MySQL Connector/NET Release Notes

Abstract

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

For the changes in each MySQL for Visual Studio release, see the MySQL for Visual Studio Release Notes

For additional Connector/NET documentation, see MySQL Connector/NET Developer Guide. For additional MySQL for Visual Studio documentation, see MySQL for Visual Studio.

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (https://dev.mysql.com/downloads/), the version has not yet been released.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

For legal information, see the Legal Notices.

For help with using MySQL, please visit the MySQL Forums, where you can discuss your issues with other MySQL users.

Document generated on: 2021-08-06 (revision: 23103)

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 https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Changes in MySQL Connector/Net 8.0

Changes in MySQL Connector/NET 8.0.27 (Not yet released, General Availability)

Version 8.0.27 has no release notes, or they have not been published because the product version has not been released.

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

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- The `authentication_ldap_sasl_client` plugin using GSSAPI/Kerberos is disabled for .NET applications running on Windows. (Bug #32867404)
- 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.
- 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.
- 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.
- 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).
- 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 the version number with the MySQL Server 8.0.25 release.

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

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- 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)

- 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.

- 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.

- 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.

- 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 (OCI) 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.

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)

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)
- 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)
- ASP.NET applications using the MySQL provider model (`MySql.Web`) can now target .NET Framework 4.8. (Bug #31799902)
- 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)
- 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.

- 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

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)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Revisions to the `MySql.Data.EntityFrameworkCore` namespace reduced the number of public classes. (Bug #31353208)
- 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)
- 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)

- 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).
- 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.
- 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.
- 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).
- 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.

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)

Functionality Added or Changed

- 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.

- 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.

- *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.

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_inf` 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)

- 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.

- 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.

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
- Bugs Fixed

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.
- 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).
- 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).

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)

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).
- *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()`.
- 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.
- 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.
- *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.

- *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", "jsmi
};

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();
```

- 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.

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.

- *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.

- *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.

- 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.

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)

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.
- 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.
- 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

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

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).
- 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.
- 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\": 500

// 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.

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

- 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)

- 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`.

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)
```

- *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.

- *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`.

- 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.

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

- 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.
- 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`
- 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.

- *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.

- *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.

- 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.

- *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.

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

- 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)
- MySQL Connector/NET now supports MySQL servers configured to use utf8mb4 as the default character set.
- 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`

- *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.

- 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.

- *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.

- 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`

- *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.

- *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;"
);
```

- *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.

- *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()`.

- 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"
  ```

- 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.

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.
- 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.
- *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.
- *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.
- *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;"
);
```

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)

Changes in MySQL Connector/Net 7.0

Changes in MySQL Connector/NET 7.0.7 (2017-03-16, Milestone 6)

Functionality Added or Changed

- *Document Store:* Added support for Internet Protocol version 6 (IPv6) addresses. Host names can now resolve as IPv4 or IPv6 addresses.
- *Document Store:* Connection string syntax is now identical to the URI scheme, which provides a cross-product syntax for defining the connection data to establish a session. See Connecting to the Server Using URI-Like Strings or Key-Value Pairs.
- *Document Store:* Added new methods to the `Schema` class to create, alter, and drop views. Usage notes include:
  - Views created with the `Schema.CreateView().DefinedAs()` method chain are supported for use with the table `Select` method, but are not supported with the collection `Find` method.
  - A collection view created in the database by some mechanism other than the `CreateView` method is not defined as a view by `Table.IsView`.
  - Query objects assigned to a view with the `CreateView` method are static, even when the underlying query changes.

Changes in MySQL Connector/NET 7.0.6 (2016-10-28, Milestone 5)

Functionality Added or Changed

- *Document Store:* The `SessionConfig` class, `SessionConfigManager` class, and `IpersistenceHandler` interface were added for session-configuration management to represent all the information associated with a session.
- *EF Core:* Added all-platform compliance support for the .NET Framework stack (Console, WPF, WinForms, and ASP.NET) and compatibility with ASP.NET 5 when using the .NET Framework or .NET Core version.
- Added the following functionality for saving changes in EF Core:
  - Basic save operation to persist entity-instance changes to the database.

- Optimistic concurrency to protect against overwriting changes made by another user after data was fetched from the database.

Bugs Fixed

- *EF Core:* A missing function used when a model is added or changed caused the migration generation to fail. MySQLHistoryRepository now implements `get_existssql()`. (Bug #24804771)
- *EF Core:* An SQL syntax error was emitted when the `Contains` operator was used in the `Where` predicate of a LINQ query. This fix requires the use of EF Core version 1.1 or later. (Bug #24797524)
- An Entity Framework `DbContext` object created with a column of type `DateTimeOffset` returned an error indicating that `DateTimeOffset` values were incompatible with `DateTime` and `Timestamp` values. (Bug #24732001)
- *Document Store:* The `GetCollections()`, `GetTable()`, and `GetTables()` methods threw System.InvalidOperationException: name is not a valid column name in the row. (Bug #24385662)

Changes in MySQL Connector/NET 7.0.5 (2016-09-06, Milestone 4)

Functionality Added or Changed

- Added support for snapshot change-tracking by recording the original values of an entity when it is retrieved from the database, and support to access the tracked state of entities through `DbContext.Entry` and `DbContext.ChangeTracker`.
- Added concurrency tokens and shadow properties to the set of conventions used to build an Entity Framework Core model based on the shape of entity classes.
- The .NET Core version of the driver now supports connections to MySQL servers configured to use SSL (TSL for 5.7) through the `MySqlConnection` class or a `DbContext` with Entity Framework 6.x or Entity Framework Core.
- Entity Framework Core support for eager loading can be used to load related data from MySQL as part of the initial query.
- Added support for following X Protocol connection options:
  - ssl-enable: enables the use of SSL as required.

    ```
    mysqlx://user:password@192.2.0.1/?ssl-enable
    ```

  - ssl-ca: path to a local file that contains a list of trusted TLS/SSL CAs as PFX file. This option uses an implicit ssl-enable.

    ```
    mysqlx://user:password@server.example.com/?ssl-ca=(c:\cas.pfx)
    ```

  - ssl-ca-pwd: specifies the CA certificate password.

    ```
    mysqlx://user:password@server.example.com/?ssl-ca=(c:\cas.pfx)&ssl-ca-pwd=password
    ```

Changes in MySQL Connector/NET 7.0.4 (2016-08-22, Milestone 3)

MySQL Connector/Net 7.0.4 is the first development release that expands cross-platform support to Linux and macOS when using Microsoft .NET Core framework. Now, .NET developers can use X DevAPI with .NET Core and Entity Framework Core 1.0 to create server applications that run on Windows, Linux, and macOS.

Functionality Added or Changed

- Added support to X Protocol connections with flexible parameter lists that do not require string parsing.
- Added support for URI connection strings in the following formats:
  - mysqlx://[*user*[:*password*]@]*host*[:*port*]
  - mysqlx://[*user*[:*password*]@]*host*[:*port*]/*database*
  - mysqlx://[*user*[:*password*]@]*host*[:*port*]/[*database*]?*option*=*value*[&*option*=*value*]
  - mysqlx+ssh://[*user*[:*password*]@]*host*[:*port*]
- *Document Store:* Views are no longer implemented using a separate `View` class in the `MySqlX` namespace. Instead, the `IsView` property has been added to the `Table` class and views are implemented as tables.
- Added in `MySql.Data` support for .NET Core 1.0, which runs on Windows, OS X, and Linux.
- Added in `MySQL.Data.EntityFrameworkCore` support for Entity Framework (EF) Core (includes support for .NET Framework 4.5.1).

Bugs Fixed

- *Document Store:* Passing no document or a `DbDoc` object that contained an empty array to the `Add()` method of a collection would throw an exception. Now, passing in either an empty document or array of documents returns a Results object in which `RecordsAffected` is zero. (Bug #23542066)
- *Document Store:* Passing a `DbDoc` object that contained an array to the `Add()` method of a collection would throw an exception. (Bug #23542031)

Changes in MySQL Connector/NET 7.0.3 (2016-06-20, Milestone 2)

Functionality Added or Changed

- Fixed binary collations as strings instead of bytes.
- Added TLS support for TLSv1.1 and TLSv1.2 when connecting to MySQL Server 5.7.

Bugs Fixed

- *Document Store:* Added results to the `Commit()` and `Rollback()` session methods, in order to read warnings. This feature has limitations that will be addressed in a future release.
- Replaced the use of "@" for "$" in JSON path expressions for X Protocol usage. This feature has limitations that will be addressed in a future release.
- Added support for TLSv1.0 to X Protocol connections. This feature has limitations that will be addressed in a future release.

Changes in MySQL Connector/NET 7.0.2 (2016-04-11, Milestone 1)

MySQL Connector/Net 7.0.2 M1 is the first development release of MySQL Connector/Net to add support for the new X DevAPI. X DevAPI enables application developers to write code that combines the strengths of the relational and document models using a modern, NoSQL-like syntax that does not assume previous experience writing traditional SQL.

To learn more about how to write applications using X DevAPI, see X DevAPI User Guide. For more information about how the X DevAPI is implemented in Connector/Net, see http://dev.mysql.com/doc/dev/connector-net.

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

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- *Document Store:* The `getLastInsertId()` method was renamed to `getAutoIncrementValues()`.
- *Document Store:* The `getLastDocumentIds()` method was added to access the last generated ID even when using `add()` or `insert()` method chaining.
- A Column Interface was added.
- *Document Store:* Bind support was added to `NodeSession.SQL()`.
- Support was added to use a URI type string for connections. See Connecting to the Server Using URI-Like Strings or Key-Value Pairs.

Bugs Fixed

- An error was emitted when calling a stored procedure that returned a table.
- The `FetchOne()` method now returns null when there are no more rows, when before it threw an exception.
- *Document Store:* After the `Execute()` method is called and yields a successful result, parameters are now available to reuse using the same statement.
- *Document Store:* For NodeSession, the `SetCurrentSchema("[schema]")` method was added to change the schema in a session using "USE". The `GetCurrentSchema()` method was added to retrieve the current schema in a session using `"SELECT DATABASE()"`, or return null if a schema is not active.
- *Document Store:* Setting a database attribute in a connection string now automatically creates a schema object in the session object.
- *Document Store:* Creating a session using the anonymous type would throw an exception.

Changes in MySQL Connector/NET 7.0.1 (Not released, Internal)

Version 7.0.1 has no release notes, or they have not been published because the product version has not been released.

Changes in MySQL Connector/NET 7.0.0 (Not released, Internal)

Version 7.0.0 has no release notes, or they have not been published because the product version has not been released.

Changes in MySQL Connector/Net 6.10

Changes in MySQL Connector/NET 6.10.9 (2019-07-29, General Availability)

- Functionality Added or Changed

- Bugs Fixed

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 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).

Bugs Fixed

- 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)
- 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)
- The `CreateCommandBuilder` and `CreateDataAdapter` methods were added to `MySqlClientFactory` class. Thanks to Cédric Luthi for the patch. (Bug #28560189, Bug #92206)
- 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)
- Connector/NET returned the wrong time zone when the `TIMESTAMP` column was read from a MySQL table. (Bug #28156187)
- *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)
- 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)
- 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 6.10.8 (2018-08-14, General Availability)

- 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)
- All recent additions to .NET Core 2.0 now are compatible with the Connector/NET 6.10 implementation.
- With the inclusion of the `Functions.Like` extended method, scalar-function mapping, and table-splitting capabilities, Entity Framework Core 2.0 is fully supported.

Bugs Fixed

- *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)
- *EF Core:* The implementation of some methods required to scaffold an existing database were incomplete. (Bug #27898343, Bug #90368)
- 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 `Microsoft.EntityFrameworkCore` assembly (with EF Core 2.0) was not loaded and the absence generated an error when the application project was built with any version of .NET Framework. This fix ensures the following support:
  - EF Core 1.1 with .NET Framework 4.5.2 only
  - EF Core 2.0 with .NET Framework 4.6.1 or later

  (Bug #27815706, Bug #90306)
- 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 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 `TreatTinyAsBoolean` connection option was ignored when the `MySqlCommand.Prepare()` method was called. (Bug #27113566, Bug #88472)
- 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)
- 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 6.10.7 (2018-04-30, General Availability)

Functionality Added or Changed

- Connections made to MySQL 8.0 (up to and including version 8.0.3) and compatibility with the new data dictionary are now supported. For information about the data dictionary, see MySQL Data Dictionary.
- Support for the `caching_sha2_password` authentication plugin through the classic MySQL protocol was added. In addition, the `sha256_password` plugin was extended to support

authentication when RSA keys are available through non-secure connections. Caching SHA-2 pluggable authentication offers faster authentication than basic SHA-256 authentication.

- 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.

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)
- The ADO.NET Entity Data Model wizard within Visual Studio closed unexpectedly without producing the data model. Thanks to Laurents Meyer for the patch. (Bug #27420311, Bug #89338)
- An exception prevented `MySQL.Data.Entity` for Entity Framework 6 from operating as expected. Thanks to Cédric Luthi for the patch. (Bug #27360520, Bug #89134)
- Connector/NET could not be installed with NuGet packages from Microsoft Visual Studio 2015. (Bug #27251839, Bug #88838)
- With valid references to the DLLs provided, using `DbConfiguration.SetConfiguration(new MySql.Data.Entity.MySqlEFConfiguration())` to set up the DbContext threw an exception. (Bug #25185319)
- 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 6.10.6 (2018-01-25, General Availability)

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)

Bugs Fixed

- 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)
- 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 6.10.5 (2017-12-08, General Availability)

Functionality Added or Changed

- 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`
- 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.

Changes in MySQL Connector/NET 6.10.4 (2017-10-25, General Availability)

Functionality Added or Changed

- Online API reference documentation is now available for MySQL Connector/NET. This new format replaces the files compressed and deployed in a binary format with the extension .CHM (Compiled HTML) that previously shipped with each connector release.
- The deprecated `UseProcedureBodies` connection-string option was removed. Instead, the `CheckParameters` option can be used to check stored parameters against the server.
- *EF Core:* Character set and collation now are independently configurable in EF Core models. The values can be set at both the entity and entity-property levels by using either data annotation (new attributes) or new methods added to the code-first fluent API using the `MySql.Data.EntityFrameworkCore.Extensions` namespace. Specifically, the new items added for setting character set and collation are:

- Attributes: `[MySqlCharset]` and `[MySqlCollation]`
- Methods: `ForMySQLHasCharset()` and `ForMySQLHasCollation()`

Bugs Fixed

- Assemblies within NuGet packages were not fully signed. (Bug #26739307)
- *EF Core:* Inserting a date of '0000-00-00' into a column when also setting `Convert Zero Datetime=True` in the connection string of a class that derives from `DbContext` produced an exception, instead of performing the expected conversion. (Bug #26552753, Bug #87120)
- *EF Core:* Foreign key relations were not consistently created within the entity model generated by scaffolding a MySQL database. (Bug #26339430)
- 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)
- 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)

Changes in MySQL Connector/NET 6.10.3 (2017-08-18, Release Candidate)

Known limitation of this release:

The use of the SSL protocol is restricted to TCP and Unix socket connections. Connections using named pipes and shared memory do not support SSL mode.

Functionality Added or Changed

- 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`
- Support for connections using Unix domain socket files was extended to include MySQL servers deployed on the Linux hosts in .NET Core scenarios. To specify a Unix domain socket connection, set the value of the `Server` connection-string option to the path of the socket file and the `Protocol` option to `unix`. For example:

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

- 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.

- *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.
- *EF Core:* Support for scaffolding a DbContext from multiple databases was added. With multiple databases (or schemas of tables) specified, the resulting entity classes are created within a single context.

  EF Core CLI usage example:

  ```
  dotnet ef dbcontext scaffold "connection-string" MySql.Data.EntityFrameworkCore --schema world --schema s
  ```

  Package Manager Console (Visual Studio) usage example:

  ```
  Scaffold-DbContext "connection-string" MySql.Data.EntityFrameworkCore -Schemas world,sakila
  ```

- 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`

Bugs Fixed

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

Changes in MySQL Connector/NET 6.10.2 (2017-07-04, Beta)

Functionality Added or Changed

- The previously deprecated `Old Syntax` (`OldSyntax`, `Use Old Syntax`, `UseOldSyntax`) connection-string option was removed.
- *EF Core:* Tables from an existing database can be specified with command-line tools when scaffolding a DbContext for a database. The MySQL provider generates an entity type for each table in the DbContext. By default, all tables in the database are included unless a list of tables is provided.

  For Package Manager Console Tools, use the `Scaffold-DbContext` command with the `-Table <tablename, tablename, ...>` common parameter.

For .NET Core CLI Tools, use the `dotnet ef dbcontext scaffold` command with the `--table <tablename>` option for each table to add.

- *EF Core:* The MySQL provider now creates a new schema when the `entity.ToTable` method within a derived `DbContext` class specifies the name of a nonexistent schema.
- *EF Core:* The Connector/NET implementation of EF Core now includes extended maximum lengths for several string data types to enable the use of longer strings.
- Connector/NET no longer supports MySQL Fabric.

Bugs Fixed

- *EF Core:* The `--force` option when used with the `dotnet ef dbcontext scaffold` command did not overwrite the existing output files as expected. (Bug #25493508)
- *EF Core:* The Database First command used to create a `DbContext` class emitted an error when used with either the sakila or world database sample. (Bug #25493336)
- *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:* JSON data exchange format was not supported by the Database First feature. (Bug #25493143)
- *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)

Changes in MySQL Connector/NET 6.10.1 (2017-02-22, Beta)

Known limitations of this release:

- The output classes cannot exist prior to using the EF Core database first feature.
- The target database must have a table with a primary key.
- The target database cannot contain one or more views.
- The JSON data exchange format is not supported by the EF Core database first feature.
- No error message is provided when a table given as input does not exist in the database and a model context is created without any table by the EF Core database first feature.
- The EF Core database first feature does not support the following data types:
  - BINARY
  - VARBINARY
  - MEDIUMBLOB
  - LONGBLOB
  - SET
  - DATE
  - TIME
  - YEAR

- `DbContext` within an application can emit an unhandled exception. Due to this limitation, MySQL Connector/NET 6.9.9 is the preferred version to use with EF6.

This release includes the following new functionality and bug fixes:

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Added database first support, also known as Scaffold-DbContext, which enables the creation of corresponding model classes from an existing database that are compatible with Entity Framework (EF) Core 1.1.

Bugs Fixed

- EF Core: Using the `Contains` method in an expression with a variable generated an exception. (Bug #25394204, Bug #84505)
- EF Core: Using the `Skip` and `Take` methods in an assignment generated an exception. (Bug #25382036, Bug #84453)

Changes in MySQL Connector/NET 6.10.0 (2016-12-09, Alpha)

Known limitation of this release: `DbContext` within an application can emit an unhandled exception. Due to this limitation, MySQL Connector/NET 6.9.9 is the preferred version to use with EF6.

Functionality Added or Changed

- Support for compression was extended to the .NET Core version of the driver.
- Added support for Entity Framework (EF) Core 1.1.

Changes in MySQL Connector/Net 6.9

Changes in MySQL Connector/NET 6.9.12 (2018-04-30, General Availability)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Connections made to MySQL 8.0 (up to and including version 8.0.3) and compatibility with the new data dictionary are now supported. For information about the data dictionary, see MySQL Data Dictionary.
- Support for the `caching_sha2_password` authentication plugin through the classic MySQL protocol was added. In addition, the `sha256_password` plugin was extended to support authentication when RSA keys are available through non-secure connections. Caching SHA-2 pluggable authentication offers faster authentication than basic SHA-256 authentication.
- 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.

Bugs Fixed

- The `MySqlConnection.GetSchema("PROCEDURES", restrictions)` method call generated an error message, instead of returning stored procedures, when the server connection was to the MySQL 8.0 release series. (Bug #25961782)
- 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 6.9.11 (2018-01-26, General Availability)

Functionality Added or Changed

- All demos, code samples, and test-debug scripts are now optional to install, whereas before these items were installed by default. (Bug #19248623)

Bugs Fixed

- Instances of the `DataReader` class did not close connections implicitly as expected when `CommandBehavior` was set to `CloseConnection`. This fix ensures that the connection is closed properly when the `DataReader` object no longer exists. (Bug #27277013)
- 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)
- 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 6.9.10 (2017-10-23, General Availability)

Bugs Fixed

- 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)
- 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)

Changes in MySQL Connector/NET 6.9.9 (2016-07-01, General Availability)

Functionality Added or Changed

- Added TLS support for TLSv1.1 and TLSv1.2 when connecting to MySQL Server 5.7.

Bugs Fixed

- Improvements were made to how the connector handles aborted connections. (Bug #23346197, Bug #80997)
- With Entity Framework 6, building a table with a primary key would not enclose the key name in quotes, which caused a syntax error. (Bug #22696180, Bug #22696207, Bug #76292)
- The connector was not disposing the transaction returned by *DBContext.Database.BeginTransaction()* when exiting a using code block. (Bug #22514355, Bug #22514363)
- When MySQL Monitor Plugin was installed with MySQL Connector/NET from a Windows Installer (.msi) installation package or MySQL Installer, the installation failed. (Bug #21507993)
- The connector did not attempt to enumerate stored procedures via *mysql.proc()*. Instead it looked up stored procedures in INFORMATION_SCHEMA.ROUTINES. This could have led to performance degradation in certain scenarios. (Bug #20960373, Bug #23528155, Bug #74116)

Changes in MySQL Connector/NET 6.9.8 (2015-10-20, General Availability)

Functionality Added or Changed

- The GB18030 Chinese character set is now supported, a character set that was added in MySQL 5.7.4. (Bug #21098546, Bug #21803306, Bug #21803300)
- The JSON type is now supported, a type that was added in MySQL 5.7.8.
- Generated columns (GENERATION_EXPRESSION) are now supported, a feature added in MySQL 5.7.6.

Changes in MySQL Connector/NET 6.9.7 (2015-08-05, General Availability)

Bugs Fixed

- The selection of a master or slave now takes into account both the status and mode, when before it only used the mode. Ignoring the status was problematic as, for example, an unreachable server's status is marked as FAULTY while the mode does not change. (Bug #21203824)
- Using `MySqlConnection.Open()` with Connector/NET 6.9.6 would fail and raise the error "Unable to connect to Fabric server". (Bug #20983968)
- Connections to MySQL server 5.7 now default to using SSL.
- The commercial MySQL Enterprise Monitor plugin was updated to function with TLSv1.1 and TLSv1.2.

Changes in MySQL Connector/NET 6.9.6 (2015-03-04, General Availability)

Bugs Fixed

- The `MySqlSimpleRoleProvider.RoleExists` method would return true instead of false. (Bug #20325391, Bug #73880, Bug #75397)

- With the *Driver.GetTimeZoneOffset* method, the TimeZone offset was not calculated correctly, and instead it was always set to 0. In other words, this caused the time zone to always be set to UTC. (Bug #20200662, Bug #74134)
- With Entity Framework 6, passing in a string reference to the `StartWith` clause would return incorrect results. (Bug #20129927, Bug #20334228, Bug #20334274, Bug #74918)
- Repeatedly calling the *MySql.Web.Security.MySqlSimpleMembershipProvider.ValidateUser* method could leave open MySQL connections. (Bug #20109419, Bug #75662)
- The `GetTimeZoneOffset` method would return an incorrect value depending on the server Time and TimeZone, as it did not take into account the day into the context of the time difference. (Bug #20065691, Bug #20362040, Bug #20362036, Bug #74905)
- The uninstaller process would fail and not remove the connector. (Bug #19485102, Bug #73677)
- A connection string that included Chinese characters for a database or uid property would emit the following exception; "MySqlException: Unknown database '???'". (Bug #18141356, Bug #20508130, Bug #70041)
- Loading two nested collections could be reported as an invalid SQL query as MySQL Connector/NET only generated a partial query. (Bug #18049862, Bug #20409538, Bug #20409528, Bug #70941)

Changes in MySQL Connector/NET 6.9.5 (2014-11-12, General Availability)

Functionality Added or Changed

- Connector/NET now sends `EXPLAIN` report data to MySQL Enterprise Monitor as supported by the MySQL server. Before, `EXPLAIN` data was only sent for `SELECT` statements.

Bugs Fixed

- A SELECT query that had a nullable DATETIME field in a child ".Any()" clause with an ".OrderBy()" would raise an exception. (Bug #19795761)
- Generated code that matched against the start of a VARCHAR/CHAR column now correctly uses "LIKE" instead of location functions for the LINQ to Entity operators "StartsWith", "Contains", and "EndsWith". Locate functions do not make use of indexes placed on the VARCHAR/CHAR columns, which caused significant performance degradation. (Bug #19783747, Bug #19680236, Bug #19944400, Bug #72058)
- The web providers registration required the removal of the ".v20" suffix from the type in the web providers section in `machine.config`. (Bug #19715398, Bug #74080)
- Generated SQL was missing several clauses, such as OrderBy, GroupBy, and Skip), in cases that involved the "let" keyword, or in other scenarios that were translated into a DbApplyExpression. (Bug #19698010, Bug #19783760, Bug #19944549, Bug #73549)
- LINQ to Entities queries failed for the cases that contained a predicate using `IList.Contains` with an argument of "DbCastExpression, DbConstantExpression, DbParameterReferenceExpression". (Bug #19690370, Bug #19783755, Bug #19944528, Bug #73643)
- A SELECT query that had a nullable DATETIME field in a child `.Any()` clause with an `.OrderBy()` would fail and emit a `NotImplementedException` exception. (Bug #19681723, Bug #19795751, Bug #70722)
- The query optimization routine would return statements with invalid table aliases when nested queries were being optimized. This would throw an "Unknown column" exception. (Bug #19681348, Bug #19934324, Bug #72004)

- The "Feature will be installed when required" option was removed from the features list in the Connector/NET installer due to it not supporting on-demand installations because of dependencies with the assemblies. (Bug #19670596, Bug #19681113)
- A memory leak was fixed. (Bug #19467233, Bug #19474480, Bug #19474510, Bug #73122)
- The `CreateUser` function did not trim whitespace before storing values into the database, while `MembershipUser` did return trimmed values. The trimming is now performed before the data is stored. (Bug #19453313, Bug #19459461, Bug #19459436, Bug #73411)
- Queries with collations using the `_utf16le` character set were not read correctly and instead yielded a "not presented key in dictionary" error. (Bug #19446614, Bug #19355906, Bug #19446571, Bug #72737)
- MySQL Connector/NET would sometimes attempt to call stored procedures when it should not. (Bug #19446554, Bug #19446523, Bug #19325120, Bug #72736)
- "LINQ to Entities" queries that used Views with the `Take` operator were not correctly generated. This problem affected EF "Database First" scenarios. (Bug #19356006, Bug #19789288, Bug #72148)

Changes in MySQL Connector/NET 6.9.4 (2014-09-26, General Availability)

Functionality Added or Changed

- MySQL Fabric is now supported.

Bugs Fixed

- Creating a "Model First" or "Database First" model using MySQL 5.7 would set ProviderManifestToken to 5.6. (Bug #19453814, Bug #19475012)

Changes in MySQL Connector/NET 6.9.3 (2014-09-03, General Availability)

MySQL Connector/Net 6.9.3 is the first GA release for the 6.9.x series of the .NET driver for MySQL. It can be used for production environments.

Bugs Fixed

- Calling *>MySqlConnection.GetSchema("PROCEDURES WITH PARAMETERS", ...)* would generate an error about a nonexistent index. This affected Intellisense when completing stored procedures. (Bug #19289402)
- Adding a new column to an existing model as identity and PK failed when applying the migration. The generated error was: "Incorrect table definition; there can be only one auto column and it must be defined as a key". (Bug #19268382, Bug #19286383)

Changes in MySQL Connector/NET 6.9.2 (2014-07-18, Release Candidate)

Bugs Fixed

- Creating a Geometry Column with an SRID value would not save the value to the table. (Bug #19783444, Bug #19137999, Bug #19476721, Bug #71869)
- Added the SiteMap and Personalization configuration web providers to the MySql.Web Nuget package.

- Added async/await compatible methods

Changes in MySQL Connector/NET 6.9.1 (2014-05-29, Beta)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Asynchronous methods are now supported.

Bugs Fixed

- When a client refreshed a web page associated with an expired session and if the ASP.NET project was using <SessionState ... regenerateExpiredSessionId="true" ...>, a "duplicate entry" exception was generated from the `MySqlSessionProvider`. (Bug #18657550, Bug #19783515, Bug #70409)

Changes in MySQL Connector/NET 6.9.0 (2014-04-30, Alpha)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Added a `Simple Membership Web Provider`.
- Added a `SiteMap Web Provider`.
- Added a `MySQL Personalization Provider`.

Bugs Fixed

- When the connection limit was exceeded, `MySqlConnection.Open()` would leave the TCP connections in a `CLOSE_WAIT` state, but now closes them. (Bug #18665388, Bug #72025)

Changes in MySQL Connector/Net 6.8

Changes in MySQL Connector/NET 6.8.8 (2016-07-01, General Availability)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Added TLS support for TLSv1.1 and TLSv1.2 when connecting to MySQL Server 5.7.

Bugs Fixed

- Improvements were made to how the connector handles aborted connections. (Bug #23346197, Bug #80997)
- With Entity Framework 6, building a table with a primary key would not enclose the key name in quotes, which caused a syntax error. (Bug #22696180, Bug #22696207, Bug #76292)
- The connector was not disposing the transaction returned by *DBContext.Database.BeginTransaction()* when exiting a using code block. (Bug #22514355, Bug #22514363)

- The connector did not attempt to enumerate stored procedures via *mysql.proc()*. Instead it looked up stored procedures in INFORMATION_SCHEMA.ROUTINES. This could have led to performance degradation in certain scenarios. (Bug #20960373, Bug #23528155, Bug #74116)

Changes in MySQL Connector/NET 6.8.7 (2015-10-21, General Availability)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- The GB18030 Chinese character set is now supported, a character set that was added in MySQL 5.7.4. (Bug #21098546, Bug #21803306, Bug #21803300)
- The JSON type is now supported, a type that was added in MySQL 5.7.8.
- Generated columns (GENERATION_EXPRESSION) are now supported, a feature added in MySQL 5.7.6.

Bugs Fixed

- A connection string that included Chinese characters for a database or uid property would emit the following exception; "MySqlException: Unknown database '???'". To solve this, the handshake process was changed to use bytes instead of encoded strings. (Bug #18141356, Bug #70041)

Changes in MySQL Connector/NET 6.8.6 (2015-06-09, General Availability)

Bugs Fixed

- Connections to MySQL server 5.7 now default to using SSL.

Changes in MySQL Connector/NET 6.8.5 (2015-03-03, General Availability)

Bugs Fixed

- With Entity Framework 6, passing in a string reference to the `StartWith` clause would return incorrect results. (Bug #20129927, Bug #20334228, Bug #20334274, Bug #74918)
- The `GetTimeZoneOffset` method would return an incorrect value depending on the server Time and TimeZone, as it did not take into account the day into the context of the time difference. (Bug #20065691, Bug #20362040, Bug #20362036, Bug #74905)
- A memory leak was fixed. (Bug #19467233, Bug #19474480, Bug #19474510, Bug #73122)
- The `CreateUser` function did not trim whitespace before storing values into the database, while `MembershipUser` did return trimmed values. The trimming is now performed before the data is stored. (Bug #19453313, Bug #19459461, Bug #19459436, Bug #73411)
- Queries with collations using the `_utf16le` character set were not read correctly and instead yielded a "not presented key in dictionary" error. (Bug #19446614, Bug #19355906, Bug #19446571, Bug #72737)
- MySQL Connector/NET would sometimes attempt to call stored procedures when it should not. (Bug #19446554, Bug #19446523, Bug #19325120, Bug #72736)
- A connection string that included Chinese characters for a database or uid property would emit the following exception; "MySqlException: Unknown database '???'". (Bug #18141356, Bug #20508130, Bug #70041)
- Loading two nested collections could be reported as an invalid SQL query as MySQL Connector/NET only generated a partial query. (Bug #18049862, Bug #20409538, Bug #20409528, Bug #70941)

Changes in MySQL Connector/NET 6.8.4 (2014-11-11, General Availability)

Bugs Fixed

- While `MySqlDateTime.Millisecond` already allowed a value between 0 and 999 (3 digit precision), a new `MySqlDateTime.Microsecond` property was added to handle microseconds (6 digit precision) on `DateTime` values. (Bug #20019257)
- The "default table cache age" connection string property default is now 60 instead of 0. (Bug #19952133)
- Generated code that matched against the start of a VARCHAR/CHAR column now correctly uses "LIKE" instead of location functions for the LINQ to Entity operators "StartsWith", "Contains", and "EndsWith". Locate functions do not make use of indexes placed on the VARCHAR/CHAR columns, which caused significant performance degradation. (Bug #19783747, Bug #19680236, Bug #19944400, Bug #72058)
- Creating a Geometry Column with an SRID value would not save the value to the table. (Bug #19783444, Bug #19137999, Bug #19476721, Bug #71869)
- Generated SQL was missing several clauses, such as OrderBy, GroupBy, and Skip), in cases that involved the "let" keyword, or in other scenarios that were translated into a DbApplyExpression. (Bug #19698010, Bug #19783760, Bug #19944549, Bug #73549)
- LINQ to Entities queries failed for the cases that contained a predicate using `IList.Contains` with an argument of "DbCastExpression, DbConstantExpression, DbParameterReferenceExpression". (Bug #19690370, Bug #19783755, Bug #19944528, Bug #73643)
- A SELECT query that had a nullable DATETIME field in a child `.Any()` clause with an `.OrderBy()` would fail and emit a `NotImplementedException` exception. (Bug #19681723, Bug #19795751, Bug #70722)
- The query optimization routine would return statements with invalid table aliases when nested queries were being optimized. This would throw an "Unknown column" exception. (Bug #19681348, Bug #19934324, Bug #72004)
- The "Feature will be installed when required" option was removed from the features list in the Connector/NET installer due to it not supporting on-demand installations because of dependencies with the assemblies. (Bug #19670596, Bug #19681113)
- The fluent API DbModelBuilder.HasColumnType had no effect in Entity Framework 6. (Bug #19476922, Bug #19456229, Bug #19462808)
- Setting a Primary Key GUID identity in "Code First" in Entity Framework 6 did not function with MySQL server 5.7. Inserting a row with a valid value for the GUID generated an error, even when it had a trigger set to the correct value.

  As a workaround, it was necessary to redeclare the column definition to accept a dummy default, such as "default ''". (Bug #19456452, Bug #19462811, Bug #19476995)
- Non Primary Keys declared as "Identity GUID" did not have their GUID's automatically generated. (Bug #19456415, Bug #19461919, Bug #19477029)
- Creating a "Model First" or "Database First" model using MySQL 5.7 would set ProviderManifestToken to 5.6. (Bug #19453814, Bug #19475012)
- "LINQ to Entities" queries that used Views with the `Take` operator were not correctly generated. This problem affected EF "Database First" scenarios. (Bug #19356006, Bug #19789288, Bug #72148)
- Calling *>MySqlConnection.GetSchema("PROCEDURES WITH PARAMETERS", ...)* would generate an error about a nonexistent index. This affected Intellisense when completing stored procedures. (Bug #19289402)

- Adding a new column to an existing model as identity and PK failed when applying the migration. The generated error was: "Incorrect table definition; there can be only one auto column and it must be defined as a key". (Bug #19268382, Bug #19286383)
- When using Entity Framework 4.3 Code first Identifiers for Migrations and Entity Framework 6, generated foreign key identifiers could be longer than 64 characters, causing `MySQLException` errors. The fix renames any too-long identifiers using the name format `FK_guid`, where *`guid`* is a global unique identifier generated at runtime. (Bug #19211564, Bug #19483128)

  References: This issue is a regression of: Bug #67285.
- In Visual Studio, Connector/NET did not read the millisecond portion of a time value for a field of type TIME(3). (Bug #19211409, Bug #18111085)
- When the connection limit was exceeded, `MySqlConnection.Open()` would leave the TCP connections in a `CLOSE_WAIT` state, but now closes them. (Bug #18665388, Bug #72025)
- When a client refreshed a web page associated with an expired session and if the ASP.NET project was using <SessionState ... regenerateExpiredSessionId="true" ...>, a "duplicate entry" exception was generated from the `MySqlSessionProvider`. (Bug #18657550, Bug #19783515, Bug #70409)
- Entity Framework threw a `NullReferenceException` on insertion of a record into tables that had an auto-increment, unsigned, bigint primary key. (Bug #18189217, Bug #19211404, Bug #71242)
- “MaximumPoolSize” and “MinimumPoolSize” were not recognized as valid connection string options. (Bug #18182246, Bug #19484670)
- When Connector/NET's SQL generator emitted code for the LINQ `Union()` or `Concat()` operator, parentheses were not applied around the individual `SELECT` statements. That could cause a change of meaning for the query when a `Take()` operator (thus a `LIMIT` clause in the SQL code) was applied to the last `SELECT` statement. With this fix, parentheses were put around individual SELECT statements, so that the `LIMIT` clause will not be applied to the entire result of the `UNION` operation. (Bug #18049691, Bug #19211182, Bug #19483110, Bug #70828)
- During migrations with ASP.NET Identity 1.0 in Visual Studio, the code generator did not generate the indexes and foreign keys in the `Up()` class. (Bug #18049272, Bug #19483069, Bug #71287)
- The value for the `Keepalive` option in the connection string was interpreted by Connector/NET to be in milliseconds. This fix makes it to be interpreted as number of seconds, as specified in the documentation. (Bug #17981275, Bug #19211293, Bug #69484)
- When using the Code First approach in Entity Framework 5, a LINQ query that checked whether a nullable column was null resulted in a faulty SQL query being generated. (Bug #17285548, Bug #69922)

Changes in MySQL Connector/NET 6.8.3 (2013-12-20, General Availability)

MySQL Connector/Net 6.8.3 is the first GA release for the 6.8.x series of the .NET driver for MySQL. It can be used for production environments.

MySQL Connector/Net 6.8.3 supports Entity Framework 6.0. It is appropriate for use with MySQL server versions 5.0-5.6.

Bugs Fixed

- A MySQL-session-state-enabled web application threw exceptions for referencing the wrong table name `my_aspnet_Sessions` (instead of the correct name `my_aspnet_sessions`). This was due to the incorrect case handling of the SQL queries for `MySqlSessionStateStore`. (Bug #17960855, Bug #19211384, Bug #69652)
- The `RenameColumn` operation in an Entity Framework migration threw an “Unknown column 'no' in the 'field list'” error when `Update-Database` was applied. (Bug #17959787, Bug #71102)

- The Connector/NET installer failed if .NET Framework 4.0 was not installed.
- An error occurred when `MySqlProviderFactoryResolver` was used with .NET Framework 4.0 and Entity Framework 6.

Changes in MySQL Connector/NET 6.8.2 (2013-12-13, Release Candidate)

MySQL Connector/Net 6.8.2 is a new version of the .NET driver for MySQL. This is a Release Candidate (RC) release for the 6.8.x series and is not recommended for production environments.

It is appropriate for use with the MySQL server versions 5.0-5.6.

Bugs Fixed

- Connector/NET did not add the `AUTO_INCREMENT` property to a primary key column of type `BIGINT` when creating a model in Entity Framework. (Bug #17924407, Bug #17937401, Bug #70602)
- Code First automatic migration failed with foreign keys when using Entity Framework 5. (Bug #17924399, Bug #70795)
- Fractional part of a value read by `MySqlDataReadeer.GetTimeSpan()` from a `TIME(3)`-typed field was dropped. (Bug #17923814, Bug #70377)
- Connector/NET threw a `NullReferenceException` when trying to save an entity into a table with a `tinyint` or `bigint` auto-incremented primary key. (Bug #17866076, Bug #70888)
- Model First was broken for string columns with unbounded length.
- Stack overflow exceptions were thrown on spatial types.

Changes in MySQL Connector/NET 6.8.1 (2013-11-11, Beta)

MySQL Connector/Net 6.8.1 is a new version of the .NET driver for MySQL. This is a beta release for the 6.8.x series and is not recommended for production environments. It is appropriate for use with the MySQL server versions 5.0-5.6.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Added idempotent script for Entity Framework 6 migrations.
- Removed installer validation when MySQL for Visual Studio is installed.
- Changed EF migration history table to use a single column as primary key.
- Added support for Entity Framework 6.0

Bugs Fixed

- Could not open the ASP.NET Web Configuration tool in the Solution Explorer when using MySQL for Visual Studio 1.0.2 and Connector/NET 6.7.5. (Bug #17898244, Bug #69808)
- Connector/NET wrote wrong version for binding redirects.

Changes in MySQL Connector/NET 6.8.0 (Not released, Alpha)

Bugs Fixed

- The Installer did not register MySQL in the `machine.config` as a `DbProviderFactories` provider. (Bug #17601689, Bug #68760)

Changes in MySQL Connector/Net 6.7

Changes in MySQL Connector/NET 6.7.9 (2015-10-21, General Availability)

Functionality Added or Changed

- The GB18030 Chinese character set is now supported, a character set that was added in MySQL 5.7.4. (Bug #21098546, Bug #21803306, Bug #21803300)
- The JSON type is now supported, a type that was added in MySQL 5.7.8.
- Generated columns (GENERATION_EXPRESSION) are now supported, a feature added in MySQL 5.7.6.

Changes in MySQL Connector/NET 6.7.8 (2015-06-09, General Availability)

Bugs Fixed

- Connections to MySQL server 5.7 now default to using SSL.

Changes in MySQL Connector/NET 6.7.7 (2015-03-04, General Availability)

Bugs Fixed

- With Entity Framework 6, passing in a string reference to the `StartWith` clause would return incorrect results. (Bug #20129927, Bug #20334228, Bug #20334274, Bug #74918)
- The `GetTimeZoneOffset` method would return an incorrect value depending on the server Time and TimeZone, as it did not take into account the day into the context of the time difference. (Bug #20065691, Bug #20362040, Bug #20362036, Bug #74905)
- A memory leak was fixed. (Bug #19467233, Bug #19474480, Bug #19474510, Bug #73122)
- The `CreateUser` function did not trim whitespace before storing values into the database, while `MembershipUser` did return trimmed values. The trimming is now performed before the data is stored. (Bug #19453313, Bug #19459461, Bug #19459436, Bug #73411)
- Queries with collations using the `_utf16le` character set were not read correctly and instead yielded a "not presented key in dictionary" error. (Bug #19446614, Bug #19355906, Bug #19446571, Bug #72737)
- MySQL Connector/NET would sometimes attempt to call stored procedures when it should not. (Bug #19446554, Bug #19446523, Bug #19325120, Bug #72736)
- A connection string that included Chinese characters for a database or uid property would emit the following exception; "MySqlException: Unknown database '???'". (Bug #18141356, Bug #20508130, Bug #70041)
- Loading two nested collections could be reported as an invalid SQL query as MySQL Connector/NET only generated a partial query. (Bug #18049862, Bug #20409538, Bug #20409528, Bug #70941)

Changes in MySQL Connector/NET 6.7.6 (2014-11-11, General Availability)

Bugs Fixed

- While `MySqlDateTime.Millisecond` already allowed a value between 0 and 999 (3 digit precision), a new `MySqlDateTime.Microsecond` property was added to handle microseconds (6 digit precision) on `DateTime` values. (Bug #20019257)
- Generated code that matched against the start of a VARCHAR/CHAR column now correctly uses "LIKE" instead of location functions for the LINQ to Entity operators "StartsWith", "Contains",

and "EndsWith". Locate functions do not make use of indexes placed on the VARCHAR/CHAR columns, which caused significant performance degradation. (Bug #19783747, Bug #19680236, Bug #19944400, Bug #72058)

- Creating a Geometry Column with an SRID value would not save the value to the table. (Bug #19783444, Bug #19137999, Bug #19476721, Bug #71869)
- Generated SQL was missing several clauses, such as OrderBy, GroupBy, and Skip), in cases that involved the "let" keyword, or in other scenarios that were translated into a DbApplyExpression. (Bug #19698010, Bug #19783760, Bug #19944549, Bug #73549)
- LINQ to Entities queries failed for the cases that contained a predicate using `IList.Contains` with an argument of "DbCastExpression, DbConstantExpression, DbParameterReferenceExpression". (Bug #19690370, Bug #19783755, Bug #19944528, Bug #73643)
- A SELECT query that had a nullable DATETIME field in a child `.Any()` clause with an `.OrderBy()` would fail and emit a `NotImplementedException` exception. (Bug #19681723, Bug #19795751, Bug #70722)
- The query optimization routine would return statements with invalid table aliases when nested queries were being optimized. This would throw an "Unknown column" exception. (Bug #19681348, Bug #19934324, Bug #72004)
- The "Feature will be installed when required" option was removed from the features list in the Connector/NET installer due to it not supporting on-demand installations because of dependencies with the assemblies. (Bug #19670596, Bug #19681113)
- The fluent API DbModelBuilder.HasColumnType had no effect in Entity Framework 6. (Bug #19476922, Bug #19456229, Bug #19462808)
- Setting a Primary Key GUID identity in "Code First" in Entity Framework 6 did not function with MySQL server 5.7. Inserting a row with a valid value for the GUID generated an error, even when it had a trigger set to the correct value.

  As a workaround, it was necessary to redeclare the column definition to accept a dummy default, such as "default ''". (Bug #19456452, Bug #19462811, Bug #19476995)
- Non Primary Keys declared as "Identity GUID" did not have their GUID's automatically generated. (Bug #19456415, Bug #19461919, Bug #19477029)
- Creating a "Model First" or "Database First" model using MySQL 5.7 would set ProviderManifestToken to 5.6. (Bug #19453814, Bug #19475012)
- "LINQ to Entities" queries that used Views with the `Take` operator were not correctly generated. This problem affected EF "Database First" scenarios. (Bug #19356006, Bug #19789288, Bug #72148)
- Calling *>MySqlConnection.GetSchema("PROCEDURES WITH PARAMETERS", ...)* would generate an error about a nonexistent index. This affected Intellisense when completing stored procedures. (Bug #19289402)
- When using Entity Framework 4.3 Code first Identifiers for Migrations and Entity Framework 6, generated foreign key identifiers could be longer than 64 characters, causing `MySQLException` errors. The fix renames any too-long identifiers using the name format `FK_`*`guid`*, where *`guid`* is a global unique identifier generated at runtime. (Bug #19211564, Bug #19483128)

  References: This issue is a regression of: Bug #67285.
- When the connection limit was exceeded, `MySqlConnection.Open()` would leave the TCP connections in a `CLOSE_WAIT` state, but now closes them. (Bug #18665388, Bug #72025)
- When a client refreshed a web page associated with an expired session and if the ASP.NET project was using <SessionState ... regenerateExpiredSessionId="true" ...>, a "duplicate entry" exception was generated from the `MySqlSessionProvider`. (Bug #18657550, Bug #19783515, Bug #70409)

Changes in MySQL Connector/NET 6.7.5 (2014-04-04, General Availability)

MySQL Connector/Net 6.7.5 is GA release for the 6.7.x series of the .NET driver for MySQL. It can be used for production environments. It is appropriate for use with MySQL server versions 5.0-5.7.

Bugs Fixed

- Configuring replication and load balancing by passing in multiple servers as a comma-separated list to the connection string failed to function. Threading synchronization problems when using replication (specifically when getting a new server/connection) would also sometimes occur. (Bug #19484568, Bug #18112966, Bug #69832)
- In Visual Studio, Connector/NET did not read the millisecond portion of a time value for a field of type TIME(3). (Bug #19211409, Bug #18111085)
- The MySQL parser did not recognize the full string literal syntax of `[_charset_name]'string' [COLLATE collation_name]` as supported by the MySQL Server. This fix makes Connector/NET and MySQL for Visual Studio recognize the string literal syntax as specified in the specified in Character String Literal Character Set and Collation, in the MySQL Server manual. (Bug #19211249, Bug #18169145)
- Entity Framework threw a `NullReferenceException` on insertion of a record into tables that had an auto-increment, unsigned, bigint primary key. (Bug #18189217, Bug #19211404, Bug #71242)
- “MaximumPoolSize” and “MinimumPoolSize” were not recognized as valid connection string options. (Bug #18182246, Bug #19484670)
- When Connector/NET's SQL generator emitted code for the LINQ `Union()` or `Concat()` operator, parentheses were not applied around the individual `SELECT` statements. That could cause a change of meaning for the query when a `Take()` operator (thus a `LIMIT` clause in the SQL code) was applied to the last `SELECT` statement. With this fix, parentheses were put around individual SELECT statements, so that the `LIMIT` clause will not be applied to the entire result of the `UNION` operation. (Bug #18049691, Bug #19211182, Bug #19483110, Bug #70828)
- During migrations with ASP.NET Identity 1.0 in Visual Studio, the code generator did not generate the indexes and foreign keys in the `Up()` class. (Bug #18049272, Bug #19483069, Bug #71287)
- The value for the `Keepalive` option in the connection string was interpreted by Connector/NET to be in milliseconds. This fix makes it to be interpreted as number of seconds, as specified in the documentation. (Bug #17981275, Bug #19211293, Bug #69484)
- A MySQL-session-state-enabled web application threw exceptions for referencing the wrong table name `my_aspnet_Sessions` (instead of the correct name `my_aspnet_sessions`). This was due to the incorrect case handling of the SQL queries for `MySqlSessionStateStore`. (Bug #17960855, Bug #19211384, Bug #69652)
- The `RenameColumn` operation in an Entity Framework migration threw an “Unknown column 'no' in the 'field list'” error when `Update-Database` was applied. (Bug #17959787, Bug #71102)
- When using the Code First approach in Entity Framework 5, automatic migrations failed with the use of foreign keys. (Bug #17929549)
- Connector/NET did not add the `AUTO_INCREMENT` property to a primary key column of type `BIGINT` when creating a model in Entity Framework. (Bug #17924407, Bug #17937401, Bug #70602)
- Fractional part of a value read by `MySqlDataReadeer.GetTimeSpan()` from a `TIME(3)`-typed field was dropped. (Bug #17923814, Bug #70377)
- Could not open the ASP.NET Web Configuration tool in the Solution Explorer when using MySQL for Visual Studio 1.0.2 and Connector/NET 6.7.5. (Bug #17898244, Bug #69808)
- Connector/NET threw a `NullReferenceException` when trying to save an entity into a table with a `tinyint` or `bigint` auto-incremented primary key. (Bug #17866076, Bug #70888)

- The Installer did not register MySQL in the `machine.config` as a `DbProviderFactories` provider. (Bug #17601689, Bug #68760)
- When using the Code First approach in Entity Framework 5, a LINQ query that checked whether a nullable column was null resulted in a faulty SQL query being generated. (Bug #17285548, Bug #69922)
- Executing a LINQ query containing an `order by` clause and a call to the `Contains` method using the `Take` method resulted in a bad SQL query, which made reference to a no-existent alias. (Bug #17194945, Bug #69751)

Changes in MySQL Connector/NET 6.7.4 (2013-07-01, General Availability)

Bugs Fixed

- When using Entity Framework 5.0, some string patterns in a LINQ query caused MySQL to throw syntax errors when they were passed to the `Contains()`, `StartsWith()`, or `EndsWith()` methods. (Bug #16974405, Bug #69409)
- The installation of "MySQL for Visual Studio 1.0.2" was broken after installing Connector/NET 6.7.x, because the data provider was not found after uninstalling MySQL for Visual Studio. (Bug #16973456)
- The LINQ query would return the error "An error of "Unknown column 'Distinct1.nCdSite' in 'where clause'" when using `Distinct().Count()`. (Bug #16950146, Bug #68513)
- When using automatic migrations in Entity Framework 5.0, the database objects were given the 'dbo' prefix. This incorrect prefix is no longer generated. (Bug #16909439)
- When the IIS application pool reset the worker processes at a specific time, the MySQL session state store would crash the `w3wp.exe` process and the request resulted in a crash error message. There are no longer ASP.NET crash yellow pages or bad exceptions. Session expiration is now handled properly. (Bug #16909237, Bug #67665)
- Connector/NET threw a fatal error when trying to read a MySQL table that used the UTF-16 or UTF-32 character set. Mappings for UTF-16 and UTF-32 encodings have now been added. (Bug #16776818, Bug #69169)
- An "DBUpdateException saving changes" exception was thrown while inserting data that had `Identify` columns. The data is now stored in the table.

  A workaround was to `set global sql_mode = "ANSI";`. (Bug #16494585)
- Sometimes data was not returned when a socket connection was slow, interrupted, or delayed. The timeout is now properly reported as an error to the upper layers. (Bug #69039, Bug #16950212)
- Using a nested projection causes a malformed query to be created, and spurious data to be returned. (Bug #67183, Bug #16872852)
- Generated "LINQ to Entities" queries are no longer as nested. In other words, two similar queries with one nested inside the other are now flattened into a simple query. This provides better performance for large result sets. (Bug #65723, Bug #16973939)
- An exception was thrown when populating `DataTable` with query fields containing a `UNIQUE` index or constraint `NULL`. There is no longer an exception thrown, and the `DataTable.Fill` method terminates correctly (filling the data). (Bug #65065, Bug #16952323)

Changes in MySQL Connector/NET 6.7.3 (2013-05-31, Beta)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Added support for Entity Framework 5.0 when using .NET 4.0.

Bugs Fixed

- The Load Balancing and Replication functionality was unstable. (Bug #16762427)

Changes in MySQL Connector/NET 6.7.2 (2013-04-30, Beta)

Functionality Added or Changed

- WinRT support was added.

Bugs Fixed

- The Connector/NET installation wizard was installing the same version of `MySQL.Data` (v2.0.50727) to all of the assemblies. (Bug #16725274)
- The core assemblies were not listed in the **Add Reference** dialog for .NET 4.5. Only 4.0 assemblies were listed. (Bug #16704115, Bug #16463655)
- The Migration functionality failed when using Entity Framework 5.0 with Connector/NET 6.7. (Bug #16694050)
- Running the "Repair" option with the Connector/NET installation wizard would break the installation. (Bug #16630567)

Changes in MySQL Connector/NET 6.7.1 (2013-04-12, Alpha)

Bugs Fixed

- The "Bug Geometry Provider Incompatible Exception" is now handled. (Bug #16453250)

Changes in MySQL Connector/NET 6.7.0 (2013-03-22, Alpha)

Functionality Added or Changed

- Connector/NET now supports SHA-256 hashing for user account passwords. After you create an account following the steps outlined in SHA-256 Pluggable Authentication, just open a Connector/NET connection passing user and password. This feature works in both SSL and non-SSL secured connections. (Bug #15935128)
- Connector/NET now supports the MySQL 5.6 feature to store connection-specific data in the server. A Connector/NET application can supply to the server a list of key/value pairs at login time. When you specify the connection option `Connect_Attrs`, a predefined set of attribute values is sent to the server. Connector/NET automatically transmits the following attributes:
  - `_client_version`
  - `_os`
  - `_pid`
  - `_platform`

  - `_program_name`
  - `_thread`

  To examine these connection-specific attributes on the server, query the Performance Schema tables described in Performance Schema Connection Attribute Tables. (Bug #15935112)

- Connector/NET now supports the MySQL 5.6 password expiration protocol. See `ALTER USER` for the syntax to expire the password for a user. When you open a connection through Connector/NET that logs in as a user with an expired password, any statement issued through the connection produces a `SET PASSWORD` exception. When you create a connection in Visual Studio Server Explorer to a user with an expired password, a dialog prompts for a new password, after which the connection succeeds. (Bug #15935104)

- When defining an entity with a `DatabaseGeneratedOption.Identity` value with Entity Framework Code First, you can now leave this column out of the column list for an `INSERT` or `UPDATE` statement. This feature is especially useful for defining a column with a default value corresponding to the `CURRENT_TIMESTAMP()` return value:

  ```
  [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
  public DateTime DateCreated { get; set; }
  ```

  (Bug #15935094)

- The `Geometry` type is now supported, with MySQL Server 5.1 and above.

Bugs Fixed

- New geometry columns were not compatible with the `DbGeometry` class. (Bug #16446399)

- Setting `cascadeDelete` to `true` in `DbMigration.AddForeignKey()` results in `ON DELETE RESTRICT` foreign key constraints in generated DDL, rather than `ON DELETE CASCADE` as expected. (Bug #16398432, Bug #68457)

- When running a multithreaded service, you might receive the exception:

  ```
  The given key was not present in the dictionary
  ```

  The issue was fixed by enhancing the locking code within the `ConnectionStringBuilder` class. (Bug #16310698, Bug #68217)

- When using EntityFramework 4.3 and Code First, generated foreign key identifiers could be longer than 64 characters, causing `MySQLException` errors. The fix renames any too-long identifiers using the name format `FK_`*`guid`*, where *`guid`* is a global unique identifier generated at runtime. (Bug #16286397, Bug #67285)

- Connector/NET would throw a `MySqlException` exception when an invalid collection was passed to the `GetSchema` method. Now it throws an `ArgumentException` when an invalid collection name is passed to `GetSchema`, to be compliant with the API spec of `System.Data.Common.DbConnection.GetSchema`. The new behavior provides better compatibility with other data access providers such as ADO.NET. (Bug #16271425, Bug #67901)

- If a `DateTime` type in MySQL 5.6 was defined with milliseconds precision using Entity Framework Code First or EF Model First, the expected `DateTime(`*`n`*`)` column did not include the precision specification. (Bug #15972773)

- Calling the method `FirstOrDefault` “LINQ to Entities” queries caused a `System.Data.EntityCommandCompilationException` exception. (Bug #15856964, Bug #67377)

- When using Entity Frameworks: Code First in Visual Studio 2012 and using a GUID as the primary key in a class, the following error could occur:

```
Incorrect column specifier for column 'column_name'.
```

Connector/NET was applying the `AUTO_INCREMENT` attribute to the column, which requires that the column use an integer data type. (Bug #15834176, Bug #67450)

- Connection attempts using IPV6 addresses in Connector/NET would fail. (Bug #14835718, Bug #67253)
- Specifying the `Default Command Timeout` option in a connection string used with the MySQL Entity Framework provider had no effect. The command timeout was always 30 seconds. (Bug #14825670, Bug #67171)
- Visual Studio 2012 projects could sometimes give the following error dialog:

```
Failed to merge object(s).

Keyword not supported.
Parameter name: AttachDbFilename.
```

This issue affected mostly VisualStudio 2012 projects, either migrated from VisualStudio 2010 or including web sites created with VS2010, connecting to a MySQL 5.5 server. (Bug #14733472, Bug #66880)

- A `System.InvalidCastException` exception could occur when calling a stored function. A workaround was to specify the `CheckParameters=false` connection option. (Bug #13864627, Bug #64633)
- Some keywords would not work as IDs without quoting in the parser.
- The error message was improved for when attempting to authenticate with an old password.

Changes in MySQL Connector/Net 6.6

Changes in MySQL Connector/NET 6.6.7 (2014-11-11)

This is the final release of the 6.6.x branch.

Bugs Fixed

- While `MySqlDateTime.Millisecond` already allowed a value between 0 and 999 (3 digit precision), a new `MySqlDateTime.Microsecond` property was added to handle microseconds (6 digit precision) on `DateTime` values. (Bug #20019257)
- The fluent API DbModelBuilder.HasColumnType had no effect in Entity Framework 6. (Bug #19476922, Bug #19456229, Bug #19462808)
- Setting a Primary Key GUID identity in "Code First" in Entity Framework 6 did not function with MySQL server 5.7. Inserting a row with a valid value for the GUID generated an error, even when it had a trigger set to the correct value.

  As a workaround, it was necessary to redeclare the column definition to accept a dummy default, such as "default "". (Bug #19456452, Bug #19462811, Bug #19476995)
- Non Primary Keys declared as "Identity GUID" did not have their GUID's automatically generated. (Bug #19456415, Bug #19461919, Bug #19477029)
- Creating a "Model First" or "Database First" model using MySQL 5.7 would set ProviderManifestToken to 5.6. (Bug #19453814, Bug #19475012)
- When using Entity Framework 4.3 Code first Identifiers for Migrations and Entity Framework 6, generated foreign key identifiers could be longer than 64 characters, causing `MySQLException` errors. The fix renames any too-long identifiers using the name format `FK_guid`, where *guid* is a global unique identifier generated at runtime. (Bug #19211564, Bug #19483128)

References: This issue is a regression of: Bug #67285.

- When trying to alter a stored procedure or trigger that included references to session variables, an error occurred and the changes were not saved, unless "Allow User Variables=true" was set in the connection settings. An error is no longer generated, and similar issues with renaming stored procedures or triggers using Alter Routine were also fixed. (Bug #19211432)
- In Visual Studio, Connector/NET did not read the millisecond portion of a time value for a field of type TIME(3). (Bug #19211409, Bug #18111085)
- The MySQL parser did not recognize the full string literal syntax of `[_charset_name]'string' [COLLATE collation_name]` as supported by the MySQL Server. This fix makes Connector/NET and MySQL for Visual Studio recognize the string literal syntax as specified in the specified in Character String Literal Character Set and Collation, in the MySQL Server manual. (Bug #19211249, Bug #18169145)
- When the connection limit was exceeded, `MySqlConnection.Open()` would leave the TCP connections in a `CLOSE_WAIT` state, but now closes them. (Bug #18665388, Bug #72025)
- When a client refreshed a web page associated with an expired session and if the ASP.NET project was using <SessionState ... regenerateExpiredSessionId="true" ...>, a "duplicate entry" exception was generated from the `MySqlSessionProvider`. (Bug #18657550, Bug #19783515, Bug #70409)
- Entity Framework threw a `NullReferenceException` on insertion of a record into tables that had an auto-increment, unsigned, bigint primary key. (Bug #18189217, Bug #19211404, Bug #71242)
- In Visual Studio, the stored procedure debugger did not evaluate the `last_insert_id()` function in a watch expression correctly. This fix also corrects similar issues for two other information functions that query the debug data table: `row_count()` and `found_row()`. (Bug #18111085)
- When Connector/NET's SQL generator emitted code for the LINQ `Union()` or `Concat()` operator, parentheses were not applied around the individual `SELECT` statements. That could cause a change of meaning for the query when a `Take()` operator (thus a `LIMIT` clause in the SQL code) was applied to the last `SELECT` statement. With this fix, parentheses were put around individual SELECT statements, so that the `LIMIT` clause will not be applied to the entire result of the `UNION` operation. (Bug #18049691, Bug #19211182, Bug #19483110, Bug #70828)
- The MySQL parser could not parse an `if` statement when there were any spaces before the parenthesis for the arguments (for example, “if (1,1, 1)”). Besides `if`, the same issue occurred for a number of other functions like `row_count`, `ifnull`, `mod`, `repeat`, and so on, and this fix corrects the problem for all of them. (Bug #17981407, Bug #19211240)
- The value for the `Keepalive` option in the connection string was interpreted by Connector/NET to be in milliseconds. This fix makes it to be interpreted as number of seconds, as specified in the documentation. (Bug #17981275, Bug #19211293, Bug #69484)
- A MySQL-session-state-enabled web application threw exceptions for referencing the wrong table name `my_aspnet_Sessions` (instead of the correct name `my_aspnet_sessions`). This was due to the incorrect case handling of the SQL queries for `MySqlSessionStateStore`. (Bug #17960855, Bug #19211384, Bug #69652)
- The `RenameColumn` operation in an Entity Framework migration threw an “Unknown column 'no' in the 'field list'” error when `Update-Database` was applied. (Bug #17959787, Bug #71102)
- In Visual Studio, Intellisense did not treat the keywords “describe” and “desc” as synonyms of “explain”. (Bug #17956087, Bug #19211401)
- In Visual Studio, Intellisense showed views from all databases, instead of just the current one. (Bug #17954412, Bug #19211338)
- Connector/NET did not add the `AUTO_INCREMENT` property to a primary key column of type `BIGINT` when creating a model in Entity Framework. (Bug #17924407, Bug #17937401, Bug #70602)

- Fractional part of a value read by `MySqlDataReadeer.GetTimeSpan()` from a `TIME(3)`-typed field was dropped. (Bug #17923814, Bug #70377)
- When opening or creating a `.mysql` file, trying to invoke Intellisense caused an error in some cases. (Bug #17890216)
- Sometimes invoking the IntelliSense code completion (**Control** + **J**) on a MySQL file would emit an error. (Bug #17890216)
- Connector/NET threw a `NullReferenceException` when trying to save an entity into a table with a `tinyint` or `bigint` auto-incremented primary key. (Bug #17866076, Bug #70888)
- Debugger failed to debug a routine correctly when it had two functions in a single expression. That was due to an error in handling the scope of the second function, which has been fixed. (Bug #17865915)
- In Visual Studio, the debugger failed with parser errors when debugging stored procedures with a `Leave` statement. (Bug #17616344)
- When debugging a stored routine in Visual Studio with the debugger, long identifiers caused the error "data too long for column 'pvarname'" to be thrown. That was because the SQL script of the debugger did not support the same lengths for identifier names as the MySQL server does. This fix matches the debugger with the MySQL server on the maximum lengths supported for identifier names. (Bug #17568158, Bug #70159)
- When using the Code First approach in Entity Framework 5, a LINQ query that checked whether a nullable column was null resulted in a faulty SQL query being generated. (Bug #17285548, Bug #69922)

Changes in MySQL Connector/NET 6.6.6 (2013-08-20, General Availability)

Bugs Fixed

- Executing a LINQ query containing an `order by` clause and a call to the `Contains` method using the `Take` method resulted in a bad SQL query, which made reference to a no-existent alias. (Bug #17194945, Bug #69751)
- When using Entity Framework 5.0, some string patterns in a LINQ query caused MySQL to throw syntax errors when they were passed to the `Contains()`, `StartsWith()`, or `EndsWith()` methods. (Bug #16974405, Bug #69409)
- The LINQ query would return the error "An error of "Unknown column 'Distinct1.nCdSite' in 'where clause'" when using `Distinct().Count()`. (Bug #16950146, Bug #68513)
- When the IIS application pool reset the worker processes at a specific time, the MySQL session state store would crash the `w3wp.exe` process and the request resulted in a crash error message. There are no longer ASP.NET crash yellow pages or bad exceptions. Session expiration is now handled properly. (Bug #16909237, Bug #67665)
- When using Entity Framework 4.3.1 and Code First Migrations, databases were migrated more than once. This fix stops the problem by changing the `CreatedOn` column in the migration history table to use the 24-hour time format. (Bug #16869202, Bug #68889)
- Connector/NET threw a fatal error when trying to read a MySQL table that used the UTF-16 or UTF-32 character set. Mappings for UTF-16 and UTF-32 encodings have now been added. (Bug #16776818, Bug #69169)
- An "DBUpdateException saving changes" exception was thrown while inserting data that had `Identify` columns. The data is now stored in the table.

  A workaround was to `set global sql_mode = "ANSI";`. (Bug #16494585)

- After opening a stored routine in the SQL Editor in Visual Studio and then changing its name, an error occurred with the message "Unable to load the stored procedure for editing" at the attempt to save the routine. (Bug #16390757)
- When running a multithreaded service, you might receive the exception:

  ```
  The given key was not present in the dictionary
  ```

  The issue was fixed by enhancing the locking code within the `ConnectionStringBuilder` class. (Bug #16310698, Bug #68217)
- When using EntityFramework 4.3 and Code First, generated foreign key identifiers could be longer than 64 characters, causing `MySQLException` errors. The fix renames any too-long identifiers using the name format `FK_guid`, where *`guid`* is a global unique identifier generated at runtime. (Bug #16286397, Bug #67285)
- Connector/NET would throw a `MySqlException` exception when an invalid collection was passed to the `GetSchema` method. Now it throws an `ArgumentException` when an invalid collection name is passed to `GetSchema`, to be compliant with the API spec of `System.Data.Common.DbConnection.GetSchema`. The new behavior provides better compatibility with other data access providers such as ADO.NET. (Bug #16271425, Bug #67901)
- When using Entity Frameworks: Code First in Visual Studio 2012 and using a GUID as the primary key in a class, the following error could occur:

  ```
  Incorrect column specifier for column 'column_name'.
  ```

  Connector/NET was applying the `AUTO_INCREMENT` attribute to the column, which requires that the column use an integer data type. (Bug #15834176, Bug #67450)
- Specifying the `Default Command Timeout` option in a connection string used with the MySQL Entity Framework provider had no effect. The command timeout was always 30 seconds. (Bug #14825670, Bug #67171)
- A `System.InvalidCastException` exception could occur when calling a stored function. A workaround was to specify the `CheckParameters=false` connection option. (Bug #13864627, Bug #64633)
- Sometimes data was not returned when a socket connection was slow, interrupted, or delayed. The timeout is now properly reported as an error to the upper layers. (Bug #69039, Bug #16950212)
- Using a nested projection causes a malformed query to be created, and spurious data to be returned. (Bug #67183, Bug #16872852)
- Generated "LINQ to Entities" queries are no longer as nested. In other words, two similar queries with one nested inside the other are now flattened into a simple query. This provides better performance for large result sets. (Bug #65723, Bug #16973939)
- An exception was thrown when populating `DataTable` with query fields containing a `UNIQUE` index or constraint `NULL`. There is no longer an exception thrown, and the `DataTable.Fill` method terminates correctly (filling the data). (Bug #65065, Bug #16952323)

Changes in MySQL Connector/NET 6.6.5 (2013-02-05, General Availability)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Connector/NET now supports SHA-256 hashing for user account passwords. After you create an account following the steps outlined in SHA-256 Pluggable Authentication, just open a Connector/

NET connection passing user and password. This feature works in both SSL and non-SSL secured connections. (Bug #15935128)

- Connector/NET now supports the MySQL 5.6 feature to store connection-specific data in the server. A Connector/NET application can supply to the server a list of key/value pairs at login time. When you specify the connection option `Connect_Attrs`, a predefined set of attribute values is sent to the server. Connector/NET automatically transmits the following attributes:

  - `_client_version`
  - `_os`
  - `_pid`
  - `_platform`
  - `_program_name`
  - `_thread`

  To examine these connection-specific attributes on the server, query the Performance Schema tables described in Performance Schema Connection Attribute Tables. (Bug #15935112)

- Connector/NET now supports the MySQL 5.6 password expiration protocol. See `ALTER USER` for the syntax to expire the password for a user. When you open a connection through Connector/NET that logs in as a user with an expired password, any statement issued through the connection produces a `SET PASSWORD` exception. When you create a connection in Visual Studio Server Explorer to a user with an expired password, a dialog prompts for a new password, after which the connection succeeds. (Bug #15935104)

- When defining an entity with a `DatabaseGeneratedOption.Identity` value with Entity Framework Code First, you can now leave this column out of the column list for an `INSERT` or `UPDATE` statement. This feature is especially useful for defining a column with a default value corresponding to the `CURRENT_TIMESTAMP()` return value:

  ```
  [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
  public DateTime DateCreated { get; set; }
  ```

  (Bug #15935094)

Bugs Fixed

- If a `DateTime` type in MySQL 5.6 was defined with milliseconds precision using Entity Framework Code First or EF Model First, the expected `DateTime(n)` column did not include the precision specification. (Bug #15972773)

- Calling the method `FirstOrDefault` “LINQ to Entities” queries caused a `System.Data.EntityCommandCompilationException` exception. (Bug #15856964, Bug #67377)

- Connection attempts using IPV6 addresses in Connector/NET would fail. (Bug #14835718, Bug #67253)

- Visual Studio 2012 projects could sometimes give the following error dialog:

  ```
  Failed to merge object(s).

  Keyword not supported.
  Parameter name: AttachDbFilename.
  ```

  This issue affected mostly VisualStudio 2012 projects, either migrated from VisualStudio 2010 or including web sites created with VS2010, connecting to a MySQL 5.5 server. (Bug #14733472, Bug #66880)

- In Visual Studio 2005, adding a new data connection through the Server Explorer produced an error message: `Package Load Failure`. The Visual Studio plugin was dynamically linked against VS2008 version of assembly `Microsoft.VisualStudio.Data` (v9.0). The fixed plugin links with the VS2005 version (v8.0) of that library instead, which is upward compatible with later Visual Studio versions. (Bug #13491674, Bug #63073)

Changes in MySQL Connector/NET 6.6.4 (2012-10-19, Release Candidate)

Continued improvements and fixes to the 6.6 feature set. In particular, enhancements to partial trust support allow hosting services to deploy applications without installing the Connector/Net library in the GAC. This is the first release candidate for the 6.6 series.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- The medium trust support using the `MySQLClientPermissions` class is now more flexible: in addition to the original deployment method, where the library is installed in the Global Assembly Cache (GAC), you can also install the library within a `bin` or `lib` folder inside the project or solution. When the library is deployed somewhere other than the GAC, the only protocol supported is TCP/IP. Existing applications that use the library installed in the GAC must now include an extra connection option, `includesecurityasserts=true`. For details, see Working with Partial Trust / Medium Trust. (Bug #14668820, Bug #65036)

Bugs Fixed

- Since Connector/NET 6.5, `TIMESTAMP` values have been returned as `DateTime` objects with a kind property of `Local` rather than `Unspecified`. `MySqlDataReader.GetDateTime()` should have returned a date with a kind property of `UTC` when the `time_zone` connection property was `utc`. With this fix, if `time_zone` is `UTC`, `Kind` is also `UTC`; otherwise, `Kind` is `Local`.

  To work with multiple servers with different timezones, change the `time_zone` setting to UTC in all `MySqlConnection` objects. For example, if you issue the command:

  ```
  set @@GLOBAL.time_zone = '+0:00',
  ```

  then every new connection you open, or the current connection if you close and reopen it, will use the new client time zone. With this fix, you will not have to change `system_time_zone` of any of your servers. Connector/NET checks if client time zone differs from UTC by running a query like:

  ```
  select timediff( curtime(), utc_time() )
  ```

  where a return value of zero hours means UTC is being used for `time_zone`.

  With `Kind = UTC`, you can use .NET standard APIs to translate between time zones for frontend applications when required. (Bug #14740705, Bug #66964)

- When an application starts up, creates a connection, and then goes idle after a single database operation, the connections are now cleaned up more quickly: typically after an idle time of 3 minutes rather than 6 minutes. This optimization is especially useful for ASP.net applications on low-traffic sites. (Bug #14652624, Bug #66472)

Changes in MySQL Connector/NET 6.6.3 (2012-09-28, Beta)

Continued improvements and fixes to the 6.6 feature set. The support for pluggable authentication, with the ability to write you own authentication plugin, is now ready for use.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- You can now write a custom authentication plugin, taking advantage of the pluggable authentication feature available in MySQL 5.5.16 and higher. See Writing a Custom Authentication Plugin for details.

Bugs Fixed

- Under some circumstances, setting `CacheServerProperties=true` in the connection string could cause a `Packet too large` error. With connection pooling enabled and `CacheServerProperties=true`, the first connection worked as expected, but the second, third, and so on connections failed if the query exceeded 1024 bytes. (Bug #14593547, Bug #66578)

- Connector/NET did not support creating an entity with a key of type string. During database creation, a `MySqlException` was thrown saying `BLOB/TEXT column 'Name' used in key specification without a key length`. The DDL produced by the provider specified a `MEDIUMTEXT` column for the primary key without specifying a length for the key. This fix is particularly important when working with Entity Framework versions 4.3 and later, since the `__MigrationsHitory` table (which replaces the `EdmMetadata` table) uses a string property as its key. (Bug #14540202, Bug #65289, Bug #64288)

- The `ExecuteNonQuery()` could return an error `Parameter '?' must be defined`, when attempting to execute a statement such as:

  ```
  insert into table_name (Field1, Field1) VALUES(?,?)
  ```

  That is, when referencing the same field twice with two different `?` placeholders. (Bug #14499549, Bug #66060)

- When using a MySQL database set up as `UTF32` as an ASP.net membership database, web applications could give a “key too long” error, and the Website Administration Tool would not connect to providers. The cause was that the column `my_aspnet_sessions.SessionId`, when converted from `Latin1` character set to `UTF32` with 4 bytes per character, exceeded the length limit for a primary key:

  ```
  Specified key was too long; max key length is 767 bytes
  ```

  (Bug #14495292, Bug #65144)

- The MySQL Connector/NET EntityFramework provider would throw `NullReferenceException` when trying to insert a new record with an empty `VALUES` clause. Such an `INSERT` should work when the only required (`NOT NULL`) column in the table is a primary key auto-increment column. (Bug #14479715, Bug #66066)

- Using the Entity Data Model Designer `decimal` type and `CreateDatabase` function, the values were stored with 0 digits at the right of the decimal point. With this fix, the default is 2 digits to the right of the decimal point, and any precision specified through the Entity Data Model Designer is applied correctly. (Bug #14474342, Bug #65127)

- Customizing precision by calling the `HasPrecision()` method within the `OnModelCreating()` method in a Code First project would always produce precision settings (10,2) rather than the specified precision. (Bug #14469048, Bug #65001)

- When building commands through the `MySql.Data.MySqlClient.MySqlCommand()` class, memory could be leaked because some `IO.MemoryStream` instances were not being freed efficiently. The memory leak could be an issue in SQL-heavy applications, for example a logging application processing large numbers of `INSERT` statements. (Bug #14468204, Bug #65696)

- When using the ASP.net web security functionality with a MySQL database, using features that access the `my_aspnet_usersinroles` table caused an exception:

  ```
  MySql.Data.MySqlClient.MySqlException: Table 'testdb.my_aspnet_usersinrole' doesn't exist.
  ```

For example, this error could occur when trying to remove the user from a role or find users in a role. The fix corrects the spelling of the table name to `my_aspnet_usersinroles`. (Bug #14405338, Bug #65805)

- Although the member variable `MySqlCommand.LastInsertedId` was a 64-bit `long`, its value was effectively capped at the maximum value of `Int32` (2,147,483,647). If a primary key exceeded this value, the value of `LastInsertedId` was wrong. This mismatch could be an issue for tables with large numbers of rows. (Bug #14171960, Bug #65452)
- When using the Entity Framework Code First approach, the generated code could be use the `MEDIUMTEXT` data type in contexts where other types such as `VARCHAR` were more appropriate, leading to errors such as:

```
error 0064: Facet 'MaxLength' must not be specified for type 'mediumtext'.
```

(Bug #13582335, Bug #63920)

Changes in MySQL Connector/NET 6.6.2 (2012-08-25, Beta)

Continued improvements and fixes to the 6.6 feature set.

Stored Procedure Debugging

- This beta release removes several of the earlier limitations on stored procedure debugging:
  - Functions and triggers can now be debugged.
  - Intellisense is enabled in the debugger window.
  - The debugger supports the SQL grammar for all MySQL versions from 5.0 to 5.6.
  - When a debugging session is finished, stored routines that were instrumented are now restored to their original form.
  - You can now evaluate and change session variables, in addition to local variables in the procedure.
  - Conditional breakpoints are supported now.
  - When debugging a routine that has parameters, the debugger prompts for values to use for the parameters. These prompted values no longer overwrite session variables that have the same names.
  - You do not need to check the **Save password** check box when creating a new connection in Server Explorer.

  These limitations remain:

  - Some MySQL functions cannot be debugged currently (`get_lock`, `release_lock`, `begin`, `commit`, `rollback`, `set transaction` level).
  - Only one debug session may be active on a given server.

Changes in MySQL Connector/NET 6.6.1 (2012-08-08, Alpha)

Continued improvements and fixes to the 6.6 feature set.

Changes in MySQL Connector/NET 6.6.0 (2012-07-17, Alpha)

First alpha release for Connector/NET 6.6. Major features of Connector/Net 6.6:

- Stored procedure debugging in Microsoft Visual Studio.
- Entity Framework 4.3 Code First support.
- Pluggable authentication (not available in this alpha).
- Entity Framework Code First Support
- Stored Procedure Debugging
- Bugs Fixed

Entity Framework Code First Support

- To support the Entity Framework 4.3.1, Connector/NET can now use the Code First approach when developing against a model, and keep track of the changes in the Entity Model and in the Database. This new Entity Framework 4.3.1 feature is focused in allowing your database to be updated along with your Code First Model changes.

Stored Procedure Debugging

- The Connector/NET integration with Visual Studio now includes stored procedure debugging. It works in a very intuitive manner, by simply clicking **Debug Routine** from Server Explorer. The limitations in this preliminary alpha release include:
  - Can only debug stored procedures. Functions and triggers cannot be debugged yet.
  - Intellisense is currently not enabled in the debugger window.
  - Some MySQL functions cannot be debugged currently (`get_lock`, `release_lock`, `begin`, `commit`, `rollback`, `set transaction` level).
  - Only 5.1 grammar is currently supported.
  - Only one debug session may be active on a given server.
  - The debugger instruments your procedures automatically. The original procedure might not be not restored correctly.
  - Evaluating and changing session variables are not supported. Local variables in the procedure are supported.
  - Conditional breakpoints are not supported.
  - When debugging a routine that has parameters, the debugger will prompt for those values. It will create session variables out of them, so be careful to not use your own session variables that have the same name.
  - When creating a new connection in Server Explorer, please check the **Save password** check box.

Bugs Fixed

- When using Entity Framework with Connector/NET, the association property `OnDelete` was not taken into account in the `CreateDatabaseScript` function of the `ObjectContext`, leading to an error message `System.Data.UpdateException was unhandled`. The SQL generated by the `CreateDatabaseScript` function was missing `ON DELETE` and `ON UPDATE` clauses. These clauses were filled in correctly by the DDL generation wizard. (Bug #14008752, Bug #64779)
- A call to a stored procedure or function in an application using the Code First entity framework could result in an error:

```
Unhandled Exception: MySql.Data.MySqlClient.MySqlException: You have an error
```

```
in your SQL syntax; ...
```

The code change allows syntax such as the following to invoke a stored procedure, without using the `CALL` statement and without using `CommandType.StoredProcedure`.

```
int count = myContext.Database.SqlQuery<int>("GetCount").First();
```

(Bug #14008699, Bug #64999)

- When using the Entity Framework Code First approach, the generated code could be incorrect:
  - Missing length specifier for data types, such as `VARBINARY` instead of `VARBINARY(n)`.
  - `ALTER TABLE` statements referring to nonexistent tables, when private members were specified inside the main class.

  (Bug #13900091, Bug #64216)

Changes in MySQL Connector/Net 6.5

Changes in MySQL Connector/NET 6.5.7 (2013-08-26, General Availability)

Bugs Fixed

- Executing a LINQ query containing an `order by` clause and a call to the `Contains` method using the `Take` method resulted in a bad SQL query, which made reference to a no-existent alias. (Bug #17194945, Bug #69751)
- When using Entity Framework 5.0, some string patterns in a LINQ query caused MySQL to throw syntax errors when they were passed to the `Contains()`, `StartsWith()`, or `EndsWith()` methods. (Bug #16974405, Bug #69409)
- The LINQ query would return the error "An error of "Unknown column 'Distinct1.nCdSite' in 'where clause'" when using `Distinct().Count()`. (Bug #16950146, Bug #68513)
- When the IIS application pool reset the worker processes at a specific time, the MySQL session state store would crash the `w3wp.exe` process and the request resulted in a crash error message. There are no longer ASP.NET crash yellow pages or bad exceptions. Session expiration is now handled properly. (Bug #16909237, Bug #67665)
- Connector/NET threw a fatal error when trying to read a MySQL table that used the UTF-16 or UTF-32 character set. Mappings for UTF-16 and UTF-32 encodings have now been added. (Bug #16776818, Bug #69169)
- An "DBUpdateException saving changes" exception was thrown while inserting data that had `Identify` columns. The data is now stored in the table.

  A workaround was to `set global sql_mode = "ANSI";`. (Bug #16494585)
- Sometimes data was not returned when a socket connection was slow, interrupted, or delayed. The timeout is now properly reported as an error to the upper layers. (Bug #69039, Bug #16950212)
- Using a nested projection causes a malformed query to be created, and spurious data to be returned. (Bug #67183, Bug #16872852)
- Generated "LINQ to Entities" queries are no longer as nested. In other words, two similar queries with one nested inside the other are now flattened into a simple query. This provides better performance for large result sets. (Bug #65723, Bug #16973939)
- An exception was thrown when populating `DataTable` with query fields containing a `UNIQUE` index or constraint `NULL`. There is no longer an exception thrown, and the `DataTable.Fill` method terminates correctly (filling the data). (Bug #65065, Bug #16952323)

Changes in MySQL Connector/NET 6.5.6 (2013-03-23, General Availability)

Functionality Added or Changed

- Connector/NET now supports SHA-256 hashing for user account passwords. After you create an account following the steps outlined in SHA-256 Pluggable Authentication, just open a Connector/NET connection passing user and password. This feature works in both SSL and non-SSL secured connections. (Bug #15935128)

- Connector/NET now supports the MySQL 5.6 feature to store connection-specific data in the server. A Connector/NET application can supply to the server a list of key/value pairs at login time. When you specify the connection option `Connect_Attrs`, a predefined set of attribute values is sent to the server. Connector/NET automatically transmits the following attributes:

  - `_client_version`
  - `_os`
  - `_pid`
  - `_platform`
  - `_program_name`
  - `_thread`

  To examine these connection-specific attributes on the server, query the Performance Schema tables described in Performance Schema Connection Attribute Tables. (Bug #15935112)

- Connector/NET now supports the MySQL 5.6 password expiration protocol. See `ALTER USER` for the syntax to expire the password for a user. When you open a connection through Connector/NET that logs in as a user with an expired password, any statement issued through the connection produces a `SET PASSWORD` exception. When you create a connection in Visual Studio Server Explorer to a user with an expired password, a dialog prompts for a new password, after which the connection succeeds. (Bug #15935104)

- When defining an entity with a `DatabaseGeneratedOption.Identity` value with Entity Framework Code First, you can now leave this column out of the column list for an `INSERT` or `UPDATE` statement. This feature is especially useful for defining a column with a default value corresponding to the `CURRENT_TIMESTAMP()` return value:

  ```
  [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
  public DateTime DateCreated { get; set; }
  ```

  (Bug #15935094)

Bugs Fixed

- Setting `cascadeDelete` to `true` in `DbMigration.AddForeignKey()` results in `ON DELETE RESTRICT` foreign key constraints in generated DDL, rather than `ON DELETE CASCADE` as expected. (Bug #16398432, Bug #68457)

- After opening a stored routine in the SQL Editor in Visual Studio and then changing its name, an error occurred with the message "Unable to load the stored procedure for editing" at the attempt to save the routine. (Bug #16390757)

- When running a multithreaded service, you might receive the exception:

```
The given key was not present in the dictionary
```

The issue was fixed by enhancing the locking code within the `ConnectionStringBuilder` class. (Bug #16310698, Bug #68217)

- When using EntityFramework 4.3 and Code First, generated foreign key identifiers could be longer than 64 characters, causing `MySQLException` errors. The fix renames any too-long identifiers using the name format `FK_guid`, where *`guid`* is a global unique identifier generated at runtime. (Bug #16286397, Bug #67285)
- Connector/NET would throw a `MySqlException` exception when an invalid collection was passed to the `GetSchema` method. Now it throws an `ArgumentException` when an invalid collection name is passed to `GetSchema`, to be compliant with the API spec of `System.Data.Common.DbConnection.GetSchema`. The new behavior provides better compatibility with other data access providers such as ADO.NET. (Bug #16271425, Bug #67901)
- When debugging a stored routine containing a `DECIMAL` variable, you might encounter the error:

```
mismatched input ')' expecting COMMA
```

A workaround was to include a second parameter of 0 in the variable declaration:

```
DECLARE var_name DECIMAL(3, 0)
```

The fix allows the Connector/NET debugger to handle `DECIMAL` variable declarations without the optional second field. (Bug #16079735, Bug #67975)

- If a `DateTime` type in MySQL 5.6 was defined with milliseconds precision using Entity Framework Code First or EF Model First, the expected `DateTime(n)` column did not include the precision specification. (Bug #15972773)
- Specifying the `Default Command Timeout` option in a connection string used with the MySQL Entity Framework provider had no effect. The command timeout was always 30 seconds. (Bug #14825670, Bug #67171)
- A `System.InvalidCastException` exception could occur when calling a stored function. A workaround was to specify the `CheckParameters=false` connection option. (Bug #13864627, Bug #64633)

Changes in MySQL Connector/NET 6.5.5 (2012-12-03, General Availability)

Fixes issues since the 6.5.4 release.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- The medium trust support using the `MySQLClientPermissions` class is now more flexible: in addition to the original deployment method, where the library is installed in the Global Assembly Cache (GAC), you can also install the library within a `bin` or `lib` folder inside the project or solution. When the library is deployed somewhere other than the GAC, the only protocol supported is TCP/IP. Existing applications that use the library installed in the GAC must now include an extra connection option, `includesecurityasserts=true`. For details, see Working with Partial Trust / Medium Trust. (Bug #14668820, Bug #65036)

Bugs Fixed

- **Performance:** The LINQ to SQL data provider for MySQL was generating inefficient code for the `Contains()` method, producing a query with multiple `OR` clauses instead of the more efficient `IN` clause. (Bug #14016344, Bug #64934)

- **Performance:** The LINQ to SQL data provider for MySQL was generating inefficient code for the `StartsWith()` and `Contains()` methods, calling the MySQL `LOCATE()` function rather than using a `LIKE` operator with a `%` wildcard. The fix causes both methods to use the `LIKE` syntax, although only `StartsWith()` gains a substantial performance improvement. Queries involving the `StartsWith()` method can now take advantage of an index on the corresponding column. (Bug #14009363, Bug #64935)

- Calling the method `FirstOrDefault` "LINQ to Entities" queries caused a `System.Data.EntityCommandCompilationException` exception. (Bug #15856964, Bug #67377)

- When using Entity Frameworks: Code First in Visual Studio 2012 and using a GUID as the primary key in a class, the following error could occur:

  ```
  Incorrect column specifier for column 'column_name'.
  ```

  Connector/NET was applying the `AUTO_INCREMENT` attribute to the column, which requires that the column use an integer data type. (Bug #15834176, Bug #67450)

- Connection attempts using IPV6 addresses in Connector/NET would fail. (Bug #14835718, Bug #67253)

- Visual Studio 2012 projects could sometimes give the following error dialog:

  ```
  Failed to merge object(s).

  Keyword not supported.
  Parameter name: AttachDbFilename.
  ```

  This issue affected mostly VisualStudio 2012 projects, either migrated from VisualStudio 2010 or including web sites created with VS2010, connecting to a MySQL 5.5 server. (Bug #14733472, Bug #66880)

- When an application starts up, creates a connection, and then goes idle after a single database operation, the connections are now cleaned up more quickly: typically after an idle time of 3 minutes rather than 6 minutes. This optimization is especially useful for ASP.net applications on low-traffic sites. (Bug #14652624, Bug #66472)

- Under some circumstances, setting `CacheServerProperties=true` in the connection string could cause a `Packet too large` error. With connection pooling enabled and `CacheServerProperties=true`, the first connection worked as expected, but the second, third, and so on connections failed if the query exceeded 1024 bytes. (Bug #14593547, Bug #66578)

- Connector/NET did not support creating an entity with a key of type string. During database creation, a `MySqlException` was thrown saying `BLOB/TEXT column 'Name' used in key specification without a key length`. The DDL produced by the provider specified a `MEDIUMTEXT` column for the primary key without specifying a length for the key. This fix is particularly important when working with Entity Framework versions 4.3 and later, since the `__MigrationsHitory` table (which replaces the `EdmMetadata` table) uses a string property as its key. (Bug #14540202, Bug #65289, Bug #64288)

- The `ExecuteNonQuery()` could return an error `Parameter '?' must be defined`, when attempting to execute a statement such as:

  ```
  insert into table_name (Field1, Field1) VALUES(?,?)
  ```

  That is, when referencing the same field twice with two different `?` placeholders. (Bug #14499549, Bug #66060)

- When using a MySQL database set up as `UTF32` as an ASP.net membership database, web applications could give a "key too long" error, and the Website Administration Tool would not connect to providers. The cause was that the column `my_aspnet_sessions.SessionId`, when converted

from `Latin1` character set to `UTF32` with 4 bytes per character, exceeded the length limit for a primary key:

```
Specified key was too long; max key length is 767 bytes
```

(Bug #14495292, Bug #65144)

- The MySQL Connector/NET EntityFramework provider would throw `NullReferenceException` when trying to insert a new record with an empty `VALUES` clause. Such an `INSERT` should work when the only required (`NOT NULL`) column in the table is a primary key auto-increment column. (Bug #14479715, Bug #66066)

- Using the Entity Data Model Designer `decimal` type and `CreateDatabase` function, the values were stored with 0 digits at the right of the decimal point. With this fix, the default is 2 digits to the right of the decimal point, and any precision specified through the Entity Data Model Designer is applied correctly. (Bug #14474342, Bug #65127)

- Customizing precision by calling the `HasPrecision()` method within the `OnModelCreating()` method in a Code First project would always produce precision settings (10,2) rather than the specified precision. (Bug #14469048, Bug #65001)

- When building commands through the `MySql.Data.MySqlClient.MySqlCommand()` class, memory could be leaked because some `IO.MemoryStream` instances were not being freed efficiently. The memory leak could be an issue in SQL-heavy applications, for example a logging application processing large numbers of `INSERT` statements. (Bug #14468204, Bug #65696)

- When using the ASP.net web security functionality with a MySQL database, using features that access the `my_aspnet_usersinroles` table caused an exception:

  ```
  MySql.Data.MySqlClient.MySqlException: Table 'testdb.my_aspnet_usersinrole' doesn't exist.
  ```

  For example, this error could occur when trying to remove the user from a role or find users in a role. The fix corrects the spelling of the table name to `my_aspnet_usersinroles`. (Bug #14405338, Bug #65805)

- Although the member variable `MySqlCommand.LastInsertedId` was a 64-bit `long`, its value was effectively capped at the maximum value of `Int32` (2,147,483,647). If a primary key exceeded this value, the value of `LastInsertedId` was wrong. This mismatch could be an issue for tables with large numbers of rows. (Bug #14171960, Bug #65452)

- When using Entity Framework with Connector/NET, the association property `OnDelete` was not taken into account in the `CreateDatabaseScript` function of the `ObjectContext`, leading to an error message `System.Data.UpdateException was unhandled`. The SQL generated by the `CreateDatabaseScript` function was missing `ON DELETE` and `ON UPDATE` clauses. These clauses were filled in correctly by the DDL generation wizard. (Bug #14008752, Bug #64779)

- A call to a stored procedure or function in an application using the Code First entity framework could result in an error:

  ```
  Unhandled Exception: MySql.Data.MySqlClient.MySqlException: You have an error
  in your SQL syntax; ...
  ```

  The code change allows syntax such as the following to invoke a stored procedure, without using the `CALL` statement and without using `CommandType.StoredProcedure`.

  ```
  int count = myContext.Database.SqlQuery<int>("GetCount").First();
  ```

  (Bug #14008699, Bug #64999)

- When the length of a `VARCHAR` column was edited in Table Designer, the data type could be saved incorrectly as `BIT`. (Bug #13916560)

- Any sequence of `Take(n)` method calls followed by `Count` or `LongCount` would cause a `System.Data.EntityCommandCompilationException` error. (Bug #13913047, Bug #64749)
- When using the Entity Framework Code First approach, the generated code could be incorrect:
  - Missing length specifier for data types, such as `VARBINARY` instead of `VARBINARY(n)`.
  - `ALTER TABLE` statements referring to nonexistent tables, when private members were specified inside the main class.

  (Bug #13900091, Bug #64216)
- The milliseconds portion of a date/time value was not stored correctly for the datatype `DATETIME(3)`. (Bug #13881444, Bug #64686)
- When using the `MySqlProfileProvider`, calling the function `ProfileManager.DeleteProfiles` could throw an `InvalidCastException` exception. (Bug #13790123, Bug #64470)
- A timing issue with the `GetItemExclusive`, `SetAndReleaseItemExclusive`, and `GetItem` functions could cause an application to freeze for almost 2 minutes if `GetItem` was called at a particular moment when a session was already locked as read-only. (Bug #13733054, Bug #63997)
- In Visual Studio Table Designer, the name of a new index was always derived from the name of the table and could not be changed. (Bug #13613801)
- When using the Entity Framework Code First approach, the generated code could be use the `MEDIUMTEXT` data type in contexts where other types such as `VARCHAR` were more appropriate, leading to errors such as:

  ```
  error 0064: Facet 'MaxLength' must not be specified for type 'mediumtext'.
  ```

  (Bug #13582335, Bug #63920)
- In “LINQ to Entity” queries, including a child entity (1-n) and its entities (n-n) returned the wrong results. For example:

  ```
  db.Authors.Include("Books.Editions").AsEnumerable().First();
  ```

  (Bug #13491698, Bug #62801)
- In Visual Studio 2005, adding a new data connection through the Server Explorer produced an error message: `Package Load Failure`. The Visual Studio plugin was dynamically linked against VS2008 version of assembly `Microsoft.VisualStudio.Data` (v9.0). The fixed plugin links with the VS2005 version (v8.0) of that library instead, which is upward compatible with later Visual Studio versions. (Bug #13491674, Bug #63073)
- Formerly, cleanup operations for expired sessions were fully automatic, with no ability to catch the timeout event and do application-specific cleanup. This fix adds a `enableSessionExpireCallback` connection option to let developers catch the event when a session expires. When `enableSessionExpireCallback` is enabled, the `global.asax.session_end` event is raised before data is deleted from the `my_aspnet_sessions` table. When `enableSessionExpireCallback` is disabled, the data is deleted from the `my_aspnet_sessions` table without raising the event first. The timeout period for session expiry is specified in the `web.config` file, in the `timeout` option of the `sessionState` section. (Bug #13354935, Bug #62266)

Changes in MySQL Connector/NET 6.5.4 (2012-03-08, General Availability)

First GA release for Connector/NET 6.5.

Bugs Fixed

- In Visual Studio Table Designer, if you tried to save a new table using an existing table name, subsequently you would not be prompted to choose a new name, preventing you from saving the table. (Bug #13785918)
- When creating a Visual Studio Web Application Project, using the ADO.NET Entity Data Model and generating the model from a database, the Entity Framework Model was not created. This operation gave an error:

  ```
  Access denied for user 'root'@'localhost' (using password: NO)
  ```

  (Bug #13610452)
- When creating a project in VisualStudio using a .NET framework such as 3.0 or 3.5 (anything less than 4.0), the Connector/NET library (`MySql.Data.dll`) was not listed in the **Add References** dialog box. The workaround was to browse to the library and add it manually. (Bug #13491678, Bug #60462)

Changes in MySQL Connector/NET 6.5.3 (2012-02-27, Release Candidate)

Second Release Candidate (RC) release.

Bugs Fixed

- The performance when setting the `CommandText` property on the `MySqlCommand` class was improved by enhancing the efficiency of a string comparison operation. (Bug #13739383, Bug #64012)
- Fixed `MySqlTime` parsing to avoid throwing an exception when handling milliseconds (as result of a `timediff` operation). (Bug #13708884, Bug #64268)
- In Visual Studio Table Designer, when adding a second foreign key, the new name was incorrectly assigned to the first foreign key in the list. (Bug #13613824)
- In Visual Studio Table Designer, changes to a field were sometimes not detected until you switched focus away from that field. (Bug #13613755)

Changes in MySQL Connector/NET 6.5.2 (2012-02-15, Release Candidate)

First Release Candidate (RC) release.

Bugs Fixed

- When using connection pooling, the connections in the pool were not automatically closed upon application exit. With the setting `log-warnings=2`, you could encounter `Aborted connection` errors in the MySQL error log. The workaround was to explicitly call `MySql.Data.MySqlClient.MySqlConnection.ClearAllPools();` upon exiting the application. (Bug #13629471, Bug #63942)
- If `MySqlCommand.CommandText` was equal to `null`, then `MySqlCommand.ExecuteReader()` would throw the wrong exception: `NullReferenceException` instead of `InvalidOperationException`. (Bug #13624659, Bug #64092)
- In Visual Studio Table Designer, when editing a foreign key relationship, choosing a column name on the left side made that column name unavailable on the right side. (Bug #13615258)
- In Visual Studio Table Designer, it was possible to save a new foreign key relationship without filling in the fields of the **Foreign Key Relationship** dialog. (Bug #13613839)
- In Visual Studio Table Designer, modifying the **Columns** field in the **Indexes/Keys** dialog multiple times could cause an error. (Bug #13613765)

- In Visual Studio Table Designer, changing the length of a `VARCHAR` field could cause an error. (Bug #13611677)
- In Visual Studio Table Designer, deleting a foreign key relationship in the **Relationship** dialog required clicking twice. (Bug #13610283)
- In Visual Studio Table Designer, an error could occur if you added and deleted column information for foreign keys in a particular sequence. (Bug #13610235)
- The MySQL script generated by using the function `CreateDatabaseScript` used names with incorrect singular/plural forms. (Bug #13582837, Bug #62150)
- Creating a trigger on a table using the Server Explorer tree could produce an error message:

  ```
  Object reference not set to an instance of an object.
  ```

  (Bug #13511801)
- In Visual Studio Table Designer, the **Add -> Function Import...** dialog could close prematurely when you pressed the **Get Column Information** button. (Bug #13511736)
- When designating a primary key for a table in Table Designer, the key icon could fail to appear until the Table Designer was restarted. (Bug #13481246)

Changes in MySQL Connector/NET 6.5.1 (2012-01-23, Beta)

Second beta release.

Bugs Fixed

- IntelliSense would emit an error when the "`-`" (minus) character was typed. (Bug #13522344)
- After an `UPDATE` statement, Connector/NET would generate incorrect `SELECT` SQL statements if a value in the `WHERE` clause was not also present in the `SET` clause of the `UPDATE`. (Bug #13491689, Bug #62134)
- In Table Designer for Visual Studio, trying to delete foreign keys from an `InnoDB` table showed an error, and the change was not saved. (Bug #13481362)

Changes in MySQL Connector/NET 6.5.0 (2011-12-22, Beta)

First beta release.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Added “interceptor” classes for exceptions and commands. For more information, see Using the Connector/NET Interceptor Classes.
- Added fractional seconds support, as per MySQL Server 5.6 and above. For more information, see Fractional Seconds in Time Values
- Added better partial-trust support, thus allowing Connector/NET to run in a partial trust scenario. It will work correctly in a medium-trust level environment when the library is installed in the GAC. For more information, see Working with Partial Trust / Medium Trust.
- Adds a `MySqlClientPermission` class to help users define the security policies for the database connections within any application using a MySQL database.

- Added better IntelliSense support, including auto-completion when editing stored procedures or `.mysql` files.

Bugs Fixed

- The `MySqlDataReader.GetDateTime()` method was not recognizing that `TIMESTAMP` values had already been converted to the local time zone of the MySQL server, which could cause incorrect results if the value was later processed through the `ToLocalTime()` method. The fix causes the `Kind` property to be correctly set to `Local` rather than `Unspecified`. (Bug #13591554, Bug #63812)
- Visual Studio 2010 Table Designer could give an error “Object reference not set to an instance of an object” for schemas with certain combinations of column names and foreign key references. The SQL syntax was incorrect for the `ALTER TABLE` statement generated by the Table Designer. (Bug #13591545, Bug #63714)

Changes in MySQL Connector/Net 6.4

Changes in MySQL Connector/NET 6.4.6 (2012-11-26, Alpha)

This release fixes bugs since 6.4.5.

Bugs Fixed

- Calling the method `FirstOrDefault` “LINQ to Entities” queries caused a `System.Data.EntityCommandCompilationException` exception. (Bug #15856964, Bug #67377)
- When using Entity Frameworks: Code First in Visual Studio 2012 and using a GUID as the primary key in a class, the following error could occur:

  ```
  Incorrect column specifier for column 'column_name'.
  ```

  Connector/NET was applying the `AUTO_INCREMENT` attribute to the column, which requires that the column use an integer data type. (Bug #15834176, Bug #67450)
- Connection attempts using IPV6 addresses in Connector/NET would fail. (Bug #14835718, Bug #67253)
- Visual Studio 2012 projects could sometimes give the following error dialog:

  ```
  Failed to merge object(s).

  Keyword not supported.
  Parameter name: AttachDbFilename.
  ```

  This issue affected mostly VisualStudio 2012 projects, either migrated from VisualStudio 2010 or including web sites created with VS2010, connecting to a MySQL 5.5 server. (Bug #14733472, Bug #66880)
- When an application starts up, creates a connection, and then goes idle after a single database operation, the connections are now cleaned up more quickly: typically after an idle time of 3 minutes rather than 6 minutes. This optimization is especially useful for ASP.net applications on low-traffic sites. (Bug #14652624, Bug #66472)
- Under some circumstances, setting `CacheServerProperties=true` in the connection string could cause a `Packet too large` error. With connection pooling enabled and `CacheServerProperties=true`, the first connection worked as expected, but the second, third, and so on connections failed if the query exceeded 1024 bytes. (Bug #14593547, Bug #66578)
- Connector/NET did not support creating an entity with a key of type string. During database creation, a `MySqlException` was thrown saying `BLOB/TEXT column 'Name' used in`

`key specification without a key length`. The DDL produced by the provider specified a `MEDIUMTEXT` column for the primary key without specifying a length for the key. This fix is particularly important when working with Entity Framework versions 4.3 and later, since the `__MigrationsHitory` table (which replaces the `EdmMetadata` table) uses a string property as its key. (Bug #14540202, Bug #65289, Bug #64288)

- The `ExecuteNonQuery()` could return an error `Parameter '?' must be defined`, when attempting to execute a statement such as:

```
insert into table_name (Field1, Field1) VALUES(?,?)
```

  That is, when referencing the same field twice with two different `?` placeholders. (Bug #14499549, Bug #66060)

- When using a MySQL database set up as `UTF32` as an ASP.net membership database, web applications could give a “key too long” error, and the Website Administration Tool would not connect to providers. The cause was that the column `my_aspnet_sessions.SessionId`, when converted from `Latin1` character set to `UTF32` with 4 bytes per character, exceeded the length limit for a primary key:

```
Specified key was too long; max key length is 767 bytes
```

  (Bug #14495292, Bug #65144)

- The MySQL Connector/NET EntityFramework provider would throw `NullReferenceException` when trying to insert a new record with an empty `VALUES` clause. Such an `INSERT` should work when the only required (`NOT NULL`) column in the table is a primary key auto-increment column. (Bug #14479715, Bug #66066)

- Using the Entity Data Model Designer `decimal` type and `CreateDatabase` function, the values were stored with 0 digits at the right of the decimal point. With this fix, the default is 2 digits to the right of the decimal point, and any precision specified through the Entity Data Model Designer is applied correctly. (Bug #14474342, Bug #65127)

- Customizing precision by calling the `HasPrecision()` method within the `OnModelCreating()` method in a Code First project would always produce precision settings (10,2) rather than the specified precision. (Bug #14469048, Bug #65001)

- When building commands through the `MySql.Data.MySqlClient.MySqlCommand()` class, memory could be leaked because some `IO.MemoryStream` instances were not being freed efficiently. The memory leak could be an issue in SQL-heavy applications, for example a logging application processing large numbers of `INSERT` statements. (Bug #14468204, Bug #65696)

- When using the ASP.net web security functionality with a MySQL database, using features that access the `my_aspnet_usersinroles` table caused an exception:

```
MySql.Data.MySqlClient.MySqlException: Table 'testdb.my_aspnet_usersinrole' doesn't exist.
```

  For example, this error could occur when trying to remove the user from a role or find users in a role. The fix corrects the spelling of the table name to `my_aspnet_usersinroles`. (Bug #14405338, Bug #65805)

- Although the member variable `MySqlCommand.LastInsertedId` was a 64-bit `long`, its value was effectively capped at the maximum value of `Int32` (2,147,483,647). If a primary key exceeded this value, the value of `LastInsertedId` was wrong. This mismatch could be an issue for tables with large numbers of rows. (Bug #14171960, Bug #65452)

- When using Entity Framework with Connector/NET, the association property `OnDelete` was not taken into account in the `CreateDatabaseScript` function of the `ObjectContext`, leading to an error message `System.Data.UpdateException was unhandled`. The SQL generated by the `CreateDatabaseScript` function was missing `ON DELETE` and `ON UPDATE` clauses. These clauses were filled in correctly by the DDL generation wizard. (Bug #14008752, Bug #64779)

- A call to a stored procedure or function in an application using the Code First entity framework could result in an error:

  ```
  Unhandled Exception: MySql.Data.MySqlClient.MySqlException: You have an error
  in your SQL syntax; ...
  ```

  The code change allows syntax such as the following to invoke a stored procedure, without using the `CALL` statement and without using `CommandType.StoredProcedure`.

  ```
  int count = myContext.Database.SqlQuery<int>("GetCount").First();
  ```

  (Bug #14008699, Bug #64999)

- When using the Entity Framework Code First approach, the generated code could be incorrect:

  - Missing length specifier for data types, such as `VARBINARY` instead of `VARBINARY(n)`.
  - `ALTER TABLE` statements referring to nonexistent tables, when private members were specified inside the main class.

  (Bug #13900091, Bug #64216)

- A `System.InvalidCastException` exception could occur when calling a stored function. A workaround was to specify the `CheckParameters=false` connection option. (Bug #13864627, Bug #64633)

- When using the `MySqlProfileProvider`, calling the function `ProfileManager.DeleteProfiles` could throw an `InvalidCastException` exception. (Bug #13790123, Bug #64470)

- In Visual Studio Table Designer, the name of a new index was always derived from the name of the table and could not be changed. (Bug #13613801)

- When using the Entity Framework Code First approach, the generated code could be use the `MEDIUMTEXT` data type in contexts where other types such as `VARCHAR` were more appropriate, leading to errors such as:

  ```
  error 0064: Facet 'MaxLength' must not be specified for type 'mediumtext'.
  ```

  (Bug #13582335, Bug #63920)

- In Visual Studio 2005, adding a new data connection through the Server Explorer produced an error message: `Package Load Failure`. The Visual Studio plugin was dynamically linked against VS2008 version of assembly `Microsoft.VisualStudio.Data` (v9.0). The fixed plugin links with the VS2005 version (v8.0) of that library instead, which is upward compatible with later Visual Studio versions. (Bug #13491674, Bug #63073)

Changes in MySQL Connector/NET 6.4.5 (2012-05-19, Alpha)

This release fixes bugs since 6.4.4.

Bugs Fixed

- When the length of a `VARCHAR` column was edited in Table Designer, the data type could be saved incorrectly as `BIT`. (Bug #13916560)

- Any sequence of `Take(n)` method calls followed by `Count` or `LongCount` would cause a `System.Data.EntityCommandCompilationException` error. (Bug #13913047, Bug #64749)

- In Visual Studio Table Designer, if you tried to save a new table using an existing table name, subsequently you would not be prompted to choose a new name, preventing you from saving the table. (Bug #13785918)

- The performance when setting the `CommandText` property on the `MySqlCommand` class was improved by enhancing the efficiency of a string comparison operation. (Bug #13739383, Bug #64012)
- Visual Studio 2010 Table Designer could give an error “Object reference not set to an instance of an object” for schemas with certain combinations of column names and foreign key references. The SQL syntax was incorrect for the `ALTER TABLE` statement generated by the Table Designer. (Bug #13591545, Bug #63714)
- The MySQL script generated by using the function `CreateDatabaseScript` used names with incorrect singular/plural forms. (Bug #13582837, Bug #62150)
- Creating a trigger on a table using the Server Explorer tree could produce an error message:

  ```
  Object reference not set to an instance of an object.
  ```

  (Bug #13511801)
- In “LINQ to Entity” queries, including a child entity (1-n) and its entities (n-n) returned the wrong results. For example:

  ```
  db.Authors.Include("Books.Editions").AsEnumerable().First();
  ```

  (Bug #13491698, Bug #62801)
- After an `UPDATE` statement, Connector/NET would generate incorrect `SELECT` SQL statements if a value in the `WHERE` clause was not also present in the `SET` clause of the `UPDATE`. (Bug #13491689, Bug #62134)
- When creating a project in VisualStudio using a .NET framework such as 3.0 or 3.5 (anything less than 4.0), the Connector/NET library (`MySql.Data.dll`) was not listed in the **Add References** dialog box. The workaround was to browse to the library and add it manually. (Bug #13491678, Bug #60462)
- In Table Designer for Visual Studio, trying to delete foreign keys from an `InnoDB` table showed an error, and the change was not saved. (Bug #13481362)
- In Table Designer for Visual Studio, trying to create a table could fail if you saved changes immediately after entering the data type for a column. The workaround was to click somewhere else in the grid before saving changes. (Bug #13477805)
- Creating a table through the Server Explorer Window on Visual Studio 2010 could fail with a MySQL syntax error. The properties in the `CREATE TABLE` statement could be listed in incorrect order. (Bug #13475830)
- Formerly, cleanup operations for expired sessions were fully automatic, with no ability to catch the timeout event and do application-specific cleanup. This fix adds a `enableSessionExpireCallback` connection option to let developers catch the event when a session expires. When `enableSessionExpireCallback` is enabled, the `global.asax.session_end` event is raised before data is deleted from the `my_aspnet_sessions` table. When `enableSessionExpireCallback` is disabled, the data is deleted from the `my_aspnet_sessions` table without raising the event first. The timeout period for session expiry is specified in the `web.config` file, in the `timeout` option of the `sessionState` section. (Bug #13354935, Bug #62266)
- Connector/NET experienced poor performance when adding parameters to the `MySQLCommand`. (Bug #62653, Bug #13331475)
- The Unicode quotation mark character `U+0022` was not escaped by the `MySQLHelper` class. (Bug #62585, Bug #13092886)
- Using a return parameter without a name resulted in an `IndexOutOfRangeException` exception. (Bug #62416, Bug #13006969)

- Connector/NET would incorrectly map decimal values to ANSI strings. (Bug #62246, Bug #13050570)
- The `Mono` runtime did not support hashed passwords. (Bug #62203, Bug #13041618)
- Connector/NET incorrectly maps `PrimitiveTypeKind.Byte` to `tinyint`, instead of `utinyint`. And `PrimitiveTypeKind.SByte` mapping was added, to `tinyint`. (Bug #62135, Bug #13061713)

Changes in MySQL Connector/NET 6.4.4 (2011-09-26, Alpha)

This release fixes bugs since 6.4.3.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Connector/NET now enables clients to connect to the server through accounts that use Windows native authentication. For more information, see Connector/NET Authentication and Windows Pluggable Authentication.

Bugs Fixed

- The Windows Native Authentication Plugin could connect using the wrong user. (Bug #12897149)
- When creating a tableadapter through a Dataset form in Visual Studio, the `MaxLength` of the field for a `CHAR` column could be set to 3 times the length of the table column. Although this many bytes could be needed to hold a UTF-8 character value, the length value was not appropriate for restricting the length of a `TextBox`. (Bug #12860224, Bug #62094)
- An error `out of sync with server` could occur when connecting in the Visual Studio Entity Framework. The issue occurred only on some MySQL servers, all with versions earlier than MySQL 5.5. (Bug #12853286, Bug #61806)
- When adding a `MEDIUMTEXT` or `LONGTEXT` column Visual Studio, the facet `Fixed length` had to be set to `false`, even though these types allow arbitrary lengths. (Bug #12848277, Bug #54915)
- Executing a “LINQ to Entity” query could result in a `NullReferenceException` error. (Bug #12776598, Bug #61729)
- Using a combination of `ListView`, `EntityDataSource` with `TypeFilter` and `Include EntityCollection Navigation Property`, and `DataPager` caused a `NullReferenceException` error in the `System.Web.UI.WebControls.EntityDataSourceView.ExecuteSelect` function. (Bug #12776517, Bug #61714)
- `MySqlDataReader.Close` was modified to use Default behavior when clearing remaining result sets. (Bug #61690, Bug #12723423)
- `MySqlScript` was modified to enable correct processing of the `DELIMITER` command when not followed by a new line. (Bug #61680, Bug #12732279)
- The ASP.NET membership provider was modified to create and query all related tables using lowercase names. (Bug #61108, Bug #12702009)
- On Model First changed column types generated in SQL scripts to produce more suitable `MySql` types. (Bug #59244, Bug #12707104)

Changes in MySQL Connector/NET 6.4.3 (2011-07-03, Alpha)

This release fixes bugs since 6.4.2.

Bugs Fixed

- `SchemaDefinition-5.5.ssdl` was modified to treat `CHAR(36)` columns as a GUID. (Bug #61657, Bug #12708208)
- `SqlFragment.QuoteIdentifier` was modified to add MySQL quotes around identifiers. (Bug #61635, Bug #12707285)

Changes in MySQL Connector/NET 6.4.2 (2011-06-29, Alpha)

This release fixes bugs since 6.4.1.

Bugs Fixed

- Modified `MySqlConnection.BeginTransaction` to throw a `NotSupportedException` for `Snapshot` isolation level. (Bug #61589, Bug #12698020)
- Modified `ProviderManifest.xml` to map `TIMESTAMP` database columns to the `DateTime` .NET type. (Bug #55351, Bug #12652602)
- Fixed Entity Framework provider `GROUP BY` clause generation by adding all group-by keys to the `SELECT` statement. (Bug #46742, Bug #12622129)

Changes in MySQL Connector/NET 6.4.1 (2011-06-06, Alpha)

First alpha release.

Functionality Added or Changed

- Calling a stored procedure with output parameters caused a marked performance decrease. (Bug #60366, Bug #12425959)
- Changed how the procedure schema collection is retrieved. If the connection string contains "`use procedure bodies=true`" then a `SELECT` is performed on the `mysql.proc` table directly, as this is up to 50 times faster than the current Information Schema implementation. If the connection string contains "`use procedure bodies=false`", then the Information Schema collection is queried. (Bug #36694)

Changes in MySQL Connector/NET 6.4.0 (Unknown, Alpha)

Bugs Fixed

- In the ADO.NET Entity Data Model Wizard, the time to update a model scaled abnormally as the number of entities increased. (Bug #48791, Bug #12596237)

Changes in MySQL Connector/Net 6.3

Changes in MySQL Connector/NET 6.3.9 (2012-04-11)

This release fixes bugs since 6.3.8.

Bugs Fixed

- When the length of a `VARCHAR` column was edited in Table Designer, the data type could be saved incorrectly as `BIT`. (Bug #13916560)
- Any sequence of `Take(n)` method calls followed by `Count` or `LongCount` would cause a `System.Data.EntityCommandCompilationException` error. (Bug #13913047, Bug #64749)

- When adding a ADO.NET Entity Data Model and generating the model from a database containing foreign keys, the foreign keys were not included in the generated model. (Bug #13800109)
- The performance when setting the `CommandText` property on the `MySqlCommand` class was improved by enhancing the efficiency of a string comparison operation. (Bug #13739383, Bug #64012)
- Fixed `MySqlTime` parsing to avoid throwing an exception when handling milliseconds (as result of a `timediff` operation). (Bug #13708884, Bug #64268)
- When using connection pooling, the connections in the pool were not automatically closed upon application exit. With the setting `log-warnings=2`, you could encounter `Aborted connection` errors in the MySQL error log. The workaround was to explicitly call `MySql.Data.MySqlClient.MySqlConnection.ClearAllPools();` upon exiting the application. (Bug #13629471, Bug #63942)
- If `MySqlCommand.CommandText` was equal to `null`, then `MySqlCommand.ExecuteReader()` would throw the wrong exception: `NullReferenceException` instead of `InvalidOperationException`. (Bug #13624659, Bug #64092)
- In Visual Studio Table Designer, when editing a foreign key relationship, choosing a column name on the left side made that column name unavailable on the right side. (Bug #13615258)
- In Visual Studio Table Designer, it was possible to save a new foreign key relationship without filling in the fields of the **Foreign Key Relationship** dialog. (Bug #13613839)
- In Visual Studio Table Designer, when adding a second foreign key, the new name was incorrectly assigned to the first foreign key in the list. (Bug #13613824)
- In Visual Studio Table Designer, modifying the **Columns** field in the **Indexes/Keys** dialog multiple times could cause an error. (Bug #13613765)
- In Visual Studio Table Designer, changes to a field were sometimes not detected until you switched focus away from that field. (Bug #13613755)
- In Visual Studio Table Designer, changing the length of a `VARCHAR` field could cause an error. (Bug #13611677)
- When creating a Visual Studio Web Application Project, using the ADO.NET Entity Data Model and generating the model from a database, the Entity Framework Model was not created. This operation gave an error:

  ```
  Access denied for user 'root'@'localhost' (using password: NO)
  ```

  (Bug #13610452)
- In Visual Studio Table Designer, deleting a foreign key relationship in the **Relationship** dialog required clicking twice. (Bug #13610283)
- In Visual Studio Table Designer, an error could occur if you added and deleted column information for foreign keys in a particular sequence. (Bug #13610235)
- The MySQL script generated by using the function `CreateDatabaseScript` used names with incorrect singular/plural forms. (Bug #13582837, Bug #62150)
- IntelliSense would emit an error when the "`-`" (minus) character was typed. (Bug #13522344)
- Creating a trigger on a table using the Server Explorer tree could produce an error message:

  ```
  Object reference not set to an instance of an object.
  ```

  (Bug #13511801)

- In Visual Studio Table Designer, the **Add -> Function Import...** dialog could close prematurely when you pressed the **Get Column Information** button. (Bug #13511736)
- In “LINQ to Entity” queries, including a child entity (1-n) and its entities (n-n) returned the wrong results. For example:

  ```
  db.Authors.Include("Books.Editions").AsEnumerable().First();
  ```

  (Bug #13491698, Bug #62801)
- After an `UPDATE` statement, Connector/NET would generate incorrect `SELECT` SQL statements if a value in the `WHERE` clause was not also present in the `SET` clause of the `UPDATE`. (Bug #13491689, Bug #62134)
- The class `MySql.Data.Types.MySqlDateTime` was not serializable. (Bug #11750161, Bug #40555)
- Connector/NET would incorrectly map decimal values to ANSI strings. (Bug #62246, Bug #13050570)
- Connector/NET incorrectly maps `PrimitiveTypeKind.Byte` to `tinyint`, instead of `utinyint`. And `PrimitiveTypeKind.SByte` mapping was added, to `tinyint`. (Bug #62135, Bug #13061713)
- On Model First changed column types generated in SQL scripts to produce more suitable `MySql` types. (Bug #59244, Bug #12707104)

Changes in MySQL Connector/NET 6.3.8 (2011-12-16)

This release fixes bugs since 6.3.7.

Bugs Fixed

- In Visual Studio Table Designer, if you tried to save a new table using an existing table name, subsequently you would not be prompted to choose a new name, preventing you from saving the table. (Bug #13785918)
- Visual Studio 2010 Table Designer could give an error “Object reference not set to an instance of an object” for schemas with certain combinations of column names and foreign key references. The SQL syntax was incorrect for the `ALTER TABLE` statement generated by the Table Designer. (Bug #13591545, Bug #63714)
- When creating a project in VisualStudio using a .NET framework such as 3.0 or 3.5 (anything less than 4.0), the Connector/NET library (`MySql.Data.dll`) was not listed in the **Add References** dialog box. The workaround was to browse to the library and add it manually. (Bug #13491678, Bug #60462)
- The columns added in descending sort order were not included in the index, as defined within the Server Explorer. (Bug #13481709)
- In Table Designer for Visual Studio, trying to delete foreign keys from an `InnoDB` table showed an error, and the change was not saved. (Bug #13481362)
- When creating a foreign key relationship in Table Designer, changes to the `ON UPDATE` and `ON CASCADE` settings were not reflected in the actual table definition, as displayed by `SHOW CREATE TABLE`. (Bug #13481348)
- In table designer for Visual Studio, you could not create a foreign key that referenced the same table as source and destination. When adding a new relationship, the **Referenced table** list did not offer the original table as one of the choices. (Bug #13481340)
- The comment property and index type were not added in the definition of the index, as defined within the Server Explorer. (Bug #13481314)

- Removing a method was not affecting the indexes list of the table object, as defined within the Table Designer. (Bug #13481313)
- When a new column was added in Table Designer without selecting an associated data type, an error would occur trying to save the column definition. (Bug #13481298)
- Creating a table through the Server Explorer Window on Visual Studio 2010 could fail with a MySQL syntax error. The properties in the `CREATE TABLE` statement could be listed in incorrect order. (Bug #13475830)
- The Connector/NET installed for all users, and thus was not available in the `Add/Remove Programs` dialog for users other than the one who installed it. (Bug #13447941)
- The default value for `VARCHAR` and `CHAR` field types would contain single quotation marks. (Bug #13442506)
- When creating a tableadapter through a Dataset form in Visual Studio, the `MaxLength` of the field for a `CHAR` column could be set to 3 times the length of the table column. Although this many bytes could be needed to hold a UTF-8 character value, the length value was not appropriate for restricting the length of a `TextBox`. (Bug #12860224, Bug #62094)
- When adding a `MEDIUMTEXT` or `LONGTEXT` column Visual Studio, the facet `Fixed length` had to be set to `false`, even though these types allow arbitrary lengths. (Bug #12848277, Bug #54915)
- Executing a “LINQ to Entity” query could result in a `NullReferenceException` error. (Bug #12776598, Bug #61729)
- Using a combination of `ListView`, `EntityDataSource` with `TypeFilter` and `Include EntityCollection Navigation Property`, and `DataPager` caused a `NullReferenceException` error in the `System.Web.UI.WebControls.EntityDataSourceView.ExecuteSelect` function. (Bug #12776517, Bug #61714)
- Connector/NET experienced poor performance when adding parameters to the `MySQLCommand`. (Bug #62653, Bug #13331475)
- The Unicode quotation mark character `U+0022` was not escaped by the `MySQLHelper` class. (Bug #62585, Bug #13092886)
- Using a return parameter without a name resulted in an `IndexOutOfRangeException` exception. (Bug #62416, Bug #13006969)
- The `Mono` runtime did not support hashed passwords. (Bug #62203, Bug #13041618)
- `MySqlDataReader.Close` was modified to use Default behavior when clearing remaining result sets. (Bug #61690, Bug #12723423)
- `MySqlScript` was modified to enable correct processing of the `DELIMITER` command when not followed by a new line. (Bug #61680, Bug #12732279)
- `SchemaDefinition-5.5.ssdl` was modified to treat `CHAR(36)` columns as a GUID. (Bug #61657, Bug #12708208)
- `SqlFragment.QuoteIdentifier` was modified to add MySQL quotes around identifiers. (Bug #61635, Bug #12707285)
- Modified `MySqlConnection.BeginTransaction` to throw a `NotSupportedException` for `Snapshot` isolation level. (Bug #61589, Bug #12698020)
- The ASP.NET membership provider was modified to create and query all related tables using lowercase names. (Bug #61108, Bug #12702009)
- Modified `ProviderManifest.xml` to map `TIMESTAMP` database columns to the `DateTime` .NET type. (Bug #55351, Bug #12652602)

Changes in MySQL Connector/NET 6.3.7 (2011-06-22)

This release fixes bugs since 6.3.6.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Calling a stored procedure with output parameters caused a marked performance decrease. (Bug #60366, Bug #12425959)

Bugs Fixed

- `MySQLConnectionStringBuilder.ContainsKey()` incorrectly returned `false` when testing whether a keyword was part of the connection string. (Bug #11766671, Bug #59835)
- MySQL Connector/NET 6.3.6 did not work with Visual Studio 2010. (Bug #60723, Bug #12394470)
- `MysqlDataReader.GetSchemaTable` returned incorrect values and types. (Bug #59989, Bug #11776346)
- All queries other than `INSERT` were executed individually instead of as a batch even though batching was enabled for the connection. (Bug #59616, Bug #11850286)
- MySQL Connector/NET generated an exception when executing a query consisting of ';', for example:

```
mycmd(";",mycon)
mycmd.executenonquery()
```

  The exception generated was:

```
System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at MySql.Data.MySqlClient.MySqlCommand.TrimSemicolons(String sql)
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
```

  (Bug #59537, Bug #11766433)
- Setting `Membership.ApplicationName` had no effect. (Bug #59438, Bug #11770465)
- A `NullReferenceException` was thrown on disposal of a `TransactionScope` object. (Bug #59346, Bug #11766272)
- The setup wizard failed with the error “Setup Wizard ended prematurely because of an error”. This was because it assumed .NET Framework version 4.0 was located on the C: drive, when it was actually located on the E: drive. (Bug #59301)
- Fixed Entity Framework provider `GROUP BY` clause generation by adding all group-by keys to the `SELECT` statement. (Bug #46742, Bug #12622129)

Changes in MySQL Connector/NET 6.3.6 (2011-01-03)

This release fixes bugs since 6.3.5.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Changed how the procedure schema collection is retrieved. If the connection string contains "`use procedure bodies=true`" then a `SELECT` is performed on the `mysql.proc` table directly, as this is up to 50 times faster than the current Information Schema implementation. If the connection string contains "`use procedure bodies=false`", then the Information Schema collection is queried. (Bug #36694)

Bugs Fixed

- `MembershipProvider` did not generate hashes correctly if the algorithm was keyed. The Key of the algorithm should have been set if the `HashAlgorithm` was `KeyedHashAlgorithm`. (Bug #58906)
- Code introduced to fix bug #54863 proved problematic on .NET version 3.5 and above. (Bug #58853)
- The `MySqlTokenizer` contained unnecessary `Substring` and `Trim` calls:

  ```
  string token = sql.Substring(startIndex, stopIndex - startIndex).Trim();
  ```

  The variable `token` was not used anywhere in the code. (Bug #58757)
- `MySqlCommand.ExecuteReader(CommandBehavior)` threw a `NullReferenceException` when being called with `CommandBehavior.CloseConnection`, if the SQL statement contained a syntax error, or contained invalid data such as an invalid column name. (Bug #58652)
- `ReadFieldLength()` returned incorrect value for `BIGINT` autoincrement columns. (Bug #58373)
- When attempting to create an ADO.NET Entity Data Model, MySQL connections were not available. (Bug #58278)
- MySQL Connector/NET did not support the `utf8mb4` character set. When attempting to connect to `utf8mb4` tables or columns, an exception `KeyNotFoundException` was generated. (Bug #58244)
- Installation of MySQL Connector/NET 6.3.5 failed. The error reported was:

  ```
  MySQL Connector Net 6.3.5 Setup Wizard ended
  prematurely because of an error. Your system has not been modified.
  ```

  (Bug #57654)
- When the tracing driver was used and an SQL statement was longer than 300 characters, an ArgumentOutOfRangeExcpetion occurred if the statement also contained a quoted character, and the 300th character was in the middle of a quoted token. (Bug #57641)
- Calling the `Read()` method on a `DataReader` obtained from `MySqlHelper.ExecuteReader` generated the following exception:

  ```
  Unhandled Exception: MySql.Data.MySqlClient.MySqlException: Invalid attempt to R
  ead when reader is closed.
     at MySql.Data.MySqlClient.MySqlDataReader.Read()
     at MySqlTest.MainClass.Main(String[] args)
  ```

  (Bug #57501)
- When using MySQL Connector/NET on Mono 2.8 using .NET 4.0, attempting to connect to a MySQL database generated the following exception:

  ```
  Unhandled Exception: System.MissingMethodException: Method not found:
   'System.Data.Common.DbConnection.EnlistTransaction'.
    at (wrapper remoting-invoke-with-check)
   MySql.Data.MySqlClient.MySqlConnection:Open ()
  ```

  (Bug #56509)

- Default values returned for text columns were not quoted. This meant that the `COLUMN_DEFAULT` field of the `GetSchema` columns collection did not return a valid SQL expression. (Bug #56509)
- MySQL Connector/NET for .NET/Mono attempted to dynamically load the assembly `Mono.Posix.dll` when a Unix socket was used to connect to the server. This failed and the connector was not able to use a Unix socket unless the `Mono.Posix.dll` assembly was previously loaded by the program. (Bug #56410)
- The ADO.NET Entity Data Model could not add stored procedures from MySQL Server 5.0.45 but worked fine using MySQL Server 5.1. (Bug #55349)
- In the ADO.NET Entity Data Model Wizard, the time to update a model scaled abnormally as the number of entities increased. (Bug #48791, Bug #12596237)

Changes in MySQL Connector/NET 6.3.5 (2010-10-12)

This release fixes bugs since 6.3.4.

Bugs Fixed

- A typed dataset did not get the table name. (Bug #57894, Bug #11764989)
- Setting `MySqlCommand.CommandTimeout` to 0 had no effect. It should have resulted in an infinite timeout. (Bug #57265)
- When performing a row-by-row update, only the first row was updated and all other rows were ignored. (Bug #57092)
- MySQL Connector/NET experienced two problems as follows:

  1. A call to `System.Data.Objects.ObjectContext.DatabaseExists()` returned false, even if the database existed.
  2. A call to `System.Data.Objects.ObjectContext.CreateDatabase()` created a database but with a name other than the one specified in the connection string. It then failed to use it when EDM objects were processed.

  (Bug #56859)
- Setting the `Default Command Timeout` connection string option had no effect. (Bug #56806)
- When an output parameter was declared as type `MySqlDbType.Bit`, it failed to return with the correct value. (Bug #56756)
- `MySqlHelper.ExecuteReader` did not include an overload accepting `MySqlParameter` objects when using a `MySqlConnection`. However, `MySqlHelper` did include an overload for `MySqlParameter` objects when using a string object containing the connection string to the database. (Bug #56755)
- MySQL Connector/NET 6.1.3 (GA) would not install on a Windows Server 2008 (Web Edition) clean installation. There were two problems:

  - If .NET framework version 4.0 was not installed, installation failed because c:\windows \microsoft.net\v4.0.* did not exist.
  - If .NET 4.0 was subsequently installed, then the following error was generated:

```
InstallFiles: File: MySql.Data.Entity.dll,  Directory: ,  Size: 229888
MSI (s) (E0:AC) [15:20:26:196]: Assembly Error:The assembly is built by a runtime newer
than the currently loaded runtime, and cannot be loaded.
MSI (s) (E0:AC) [15:20:26:196]: Note: 1: 1935 2:  3: 0x8013101B 4: IStream 5: Commit 6:
MSI (s) (E0:A0) [15:20:26:196]: Note: 1: 1304 2: MySql.Data.Entity.dll
Error 1304. Error writing to file: MySql.Data.Entity.dll. Verify that you have access to
that directory.
```

(Bug #56580)

Changes in MySQL Connector/NET 6.3.4 (2010-09-01, General Availability)

First GA release. This release fixes bugs since 6.3.3.

Bugs Fixed

- The calculation of `lockAge` in the Session Provider sometimes generated a `System.Data.SqlTypes.SqlNullValueException`. (Bug #55701)
- Attempting to read `Double.MinValue` from a `DOUBLE` column in MySQL table generated the following exception:

```
System.OverflowException : Value was either too large or too small for a Double.

--OverflowException
at System.Number.ParseDouble(String value, NumberStyles options, NumberFormatInfo
numfmt)
at MySql.Data.Types.MySqlDouble.MySql.Data.Types.IMySqlValue.ReadValue(MySqlPacket
packet, Int64 length, Boolean nullVal)
at MySql.Data.MySqlClient.NativeDriver.ReadColumnValue(Int32 index, MySqlField field,
IMySqlValue valObject)
at MySql.Data.MySqlClient.ResultSet.ReadColumnData(Boolean outputParms)
at MySql.Data.MySqlClient.ResultSet.NextRow(CommandBehavior behavior)
at MySql.Data.MySqlClient.MySqlDataReader.Read()
```

  (Bug #55644)

- Calling `MySqlDataAdapter.Update(DataTable)` resulted in an unacceptable performance hit when updating large amounts of data. (Bug #55609)
- If using MySQL Server 5.0.x it was not possible to alter stored routines in Visual Studio. If the stored routine was clicked, and the context sensitive menu option, Alter Routine, selected, the following error was generated:

```
Unable to load object with error: Object reference not
set to an instance of an object
```

  (Bug #55170)

- When attempting to carry out an operation such as:

```
from p in db.Products where p.PostedDate>=DateTime.Now select p;
```

  Where `p.PostedDate` is a `DateTimeOffset`, and the underlying column type is a `TIMESTAMP`, the following exception was generated:

```
MySqlException occurred
Unable to serialize date/time value
```

  MySQL Connector/NET has now been changed so that all `TIMESTAMP` columns will be surfaced as `DateTime`. (Bug #52550)

- EventLog was not disposed in the SessionState provider. (Bug #52550)
- Stored procedure enumeration code generated an error if a procedure was used in a dataset that did not return any resultsets. (Bug #50671)
- The `INSERT` command was significantly slower with MySQL Connector/NET 6.x compared to 5.x, when compression was enabled. (Bug #48243)
- Opening a connection in the Visual Studio Server Explorer and choosing to alter an existing routine required another authentication at the server. (Bug #44715)

Changes in MySQL Connector/NET 6.3.3 (2010-07-27)

This release fixes bugs since 6.3.2.

Bugs Fixed

- `MySqlDataAdapter.Update()` generated concurrency violations for custom stored procedure driven update commands that used `UpdateRowSource.FirstReturnedRecord`. (Bug #54895)
- Several calls to `DataAdapter.Update()` with intervening changes to `DataTable` resulted in `ConcurrencyException` exceptions being generated. (Bug #54863)
- MySQL Connector/NET generated a null reference exception when `TransactionScope` was used by multiple threads. (Bug #54681)
- The icon for the MySQL Web Configuration Tool was not displayed in Visual Studio for Web Application Projects. (Bug #54571)
- The `MySqlHelper` object did not have an overloaded version of the `ExecuteReader` method that accepted a `MySqlConnection` object. (Bug #54570)
- If `MySqlDataAdapter` was used with an `INSERT` command where the `VALUES` clause contained an expression with parentheses in it, and set the `adapter.UpdateBatchSize` parameter to be greater than one, then the call to `adapter.Update` either generated an exception or failed to batch the commands, executing each insert individually. (Bug #54386)
- The method `MySql.Data.Common.QueryNormalizer.CollapseValueList` generated an `ArgumentOutOfRangeException`. (Bug #54152, Bug #53865)
- MySQL Connector/NET did not process `Thread.Abort()` correctly, and failed to cancel queries currently running on the server. (Bug #54012)
- MySQL Connector/NET 6.3.2 failed to install on Windows Vista. (Bug #53975)
- Garbage Collector disposal of a `MySqlConnection` object caused the following exception:

```
System.IO.EndOfStreamException: Attempted to read past the end of the stream.
MySql.Data.MySqlClient.MySqlStream.ReadFully(Stream stream, Byte[] buffer, Int32 offset,
Int32 count)
MySql.Data.MySqlClient.MySqlStream.LoadPacket()
Outer Exception Reading from the stream has failed.
...
```

  (Bug #53457)

- MySQL Connector/NET did not throw an `EndOfStreamException` exception when `net_write_timeout` was exceeded. (Bug #53439)
- After a timeout exception, if an attempt was made to reuse a connection returned to the connection pool the following exception was generated:

```
[MySqlException (0x80004005): There is already an open DataReader associated with this
Connection which must be closed first.]
   MySql.Data.MySqlClient.MySqlCommand.CheckState() +278
   MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior) +43
   MySql.Data.MySqlClient.MySqlCommand.ExecuteReader() +6
   Controls.SimpleCommand.ExecuteReader(String SQL) in ...:323
   Albums.GetImagesByAlbum(SimpleCommand Cmd, Int32 iAlbum, String Order, String Limit)
in ...:13
   Forecast.Page_Load(Object sender, EventArgs e) in ...:70
   System.Web.UI.Control.OnLoad(EventArgs e) +99
   System.Web.UI.Control.LoadRecursive() +50
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean
includeStagesAfterAsyncPoint) +627
```

(Bug #53357)

- Membership schema creation failed if the default schema collation was not Latin1. (Bug #53174)
- The MySQL Connector/NET installation failed due to `machine.config` files not being present in configuration folders.

  MySQL Connector/NET has been changed to skip over configuration folders that do not contain a `machine.config` file. (Bug #52352)
- `CHAR(36)` columns were not recognized as GUIDs when used in views with entity models. (Bug #52085)
- When an application was subjected to increased concurrent load, MySQL Connector/NET generated the following error when calling stored procedures:

  ```
  A DataTable named \'Procedure Parameters\'
  already belongs to this DataSet.
  ```

  (Bug #49118)
- When the connection string option “Connection Reset = True” was used, a connection reset used the previously used encoding for the subsequent authentication operation. This failed, for example, if UCS2 was used to read the last column before the reset. (Bug #47153)
- When batching was used in `MySqlDataAdapter`, a connection was not opened automatically in `MySqlDataAdapter.Update()`. This resulted in an `InvalidOperationException` exception being generated, with the message text “connection must be valid and open”.

  MySQL Connector/NET has been changed to behave more like SQL Server: if the connection is closed, it is opened for the duration of update operation. (Bug #38411)
- Database name was emitted into typed datasets. This prevented users using the configured default database. (Bug #33870)

Changes in MySQL Connector/NET 6.3.2 (2010-05-24, Beta)

First Beta release. This release fixes bugs since 6.3.1.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Procedure caching had a problem whereby if you created a procedure, dropped it, and recreated it with a different number of parameters an exception was generated.

  MySQL Connector/NET has been changed so that if the procedure is recreated with a different number of parameters, it will still be recognized. (Bug #52562)
- MySQL Connector/NET has been changed to include `MySqlDataReader.GetFieldType(string columnname)`. Further, `MySqlDataReader.GetOrdinal()` now includes the name of the column in the exception if the column is not found. (Bug #47467)

Bugs Fixed

- After an exception, the internal datareader, `MySqlCommand.Connection.Reader`, was not properly closed (it was not set to null). If another query was subsequently executed on that command

object an exception was generated with the message “There is already an open DataReader associated with this Connection which must be closed first.” (Bug #55558)

- MySQL Connector/NET generated an exception when used to read a `TEXT` column containing more than 32767 bytes. (Bug #54040)
- In MySQL Connector/NET, the `MySqlConnection.Abort()` method contained a `try...catch` construct, with an empty `catch` block. This meant that any exception generated at this point would not be caught. (Bug #52769)
- The procedure cache affected the MySQL Connector/NET performance, reducing it by around 65%. This was due to unnecessary calls of `String.Format()`, related to debug logging. Even though the logging was disabled the string was still being formatted, resulting in impaired performance. (Bug #52475)
- If `FunctionsReturnString=true` was used in the connection string, the decimal separator (according to locale) was not interpreted. (Bug #52187)
- In MySQL Connector/NET, the `LoadCharsetMap()` function of the `CharSetMap` class set the following incorrect mapping:

  ```
  mapping.Add("latin1", new CharacterSet("latin1", 1));
  ```

  This meant that, for example, the Euro sign was not handled correctly.

  The correct mapping should have been:

  ```
  mapping.Add("latin1", new CharacterSet("windows-1252", 1));
  ```

  This is because the MySQL `latin1` character set is the same as the `windows-cp1252` character set and it extends the official ISO 8859-1 or IANA latin1. (Bug #51927)
- A non-terminated string in SQL threw a CLR exception rather than a syntax exception. (Bug #51788)
- When calling `ExecuteNonQuery` on a command object, the following exception occurred:

  ```
  Index and length must refer to a location within the string.
  Parameter name: length
  ```

  (Bug #51610)
- MySQL Connector/NET 6.3.1 failed to install. (Bug #51407, Bug #51604)
- When using table per type inheritance and listing the contents of the parent table, the result of the query was a list of child objects, even though there was no related child record with the same parent Id. (Bug #49850)

Changes in MySQL Connector/NET 6.3.1 (2010-03-02)

This release fixes bugs since 6.3.0.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Connector/NET was not compatible with Visual Studio wizards that used square brackets to delimit symbols.

  Connector/NET has been changed to include a new connection string option `Sql Server mode` that supports use of square brackets to delimit symbols. (Bug #35852)

Bugs Fixed

- Specifying a connection string where an option had no value generated an error, rather than the value being set to the default. For example, a connection string such as the following would result in an error:

  ```
  server=localhost;user=root;compress=;database=test;port=3306;password=123456;
  ```

  (Bug #51209)

- The method `Command.TrimSemicolons` used `StringBuilder`, and therefore allocated memory for the query even if it did not need to be trimmed. This led to excessive memory consumption when executing a number of large queries. (Bug #51149)

- `MySqlCommand.Parameters.Clear()` did not work. (Bug #50444)

- Binary Columns were not displayed in the Query Builder of Visual Studio. (Bug #50171)

- When the `UpdateBatchSize` property was set to a value greater than 1, only the first row was applied to the database. (Bug #50123)

- When trying to create stored procedures from an SQL script, a `MySqlException` was thrown when attempting to redefine the `DELIMITER`:

  ```
  MySql.Data.MySqlClient.MySqlException was unhandled
  Message="You have an error in your SQL syntax; check the manual that corresponds to your
  MySQL server version for the right syntax to use near 'DELIMITER' at line 1"
  Source="MySql.Data"
  ErrorCode=-2147467259
  Number=1064
  StackTrace:
  à MySql.Data.MySqlClient.MySqlStream.ReadPacket()
  à MySql.Data.MySqlClient.NativeDriver.ReadResult(UInt64& affectedRows, Int64&
  lastInsertId)
  à MySql.Data.MySqlClient.MySqlDataReader.GetResultSet()
  à MySql.Data.MySqlClient.MySqlDataReader.NextResult()
  à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
  à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
  à MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
  à MySql.Data.MySqlClient.MySqlScript.Execute()
  ```

  Note: The `MySqlScript` class has been fixed to support the delimiter statement as it is found in SQL scripts. (Bug #46429)

- A connection string set in `web.config` could not be reused after Visual Studio 2008 Professional was shut down. It continued working for the existing controls, but did not work for new controls added. (Bug #41629)

Changes in MySQL Connector/NET 6.3.0 (2010-02-16, Alpha)

First alpha release of 6.3.

Functionality Added or Changed

- Nested transaction scopes were not supported. MySQL Connector/NET now implements nested transaction scopes. A per-thread stack of scopes is maintained, which is necessary to handle nested scopes with the `RequiresNew` or `Suppress` options. (Bug #45098)

- Support for MySQL Server 4.1 has been removed from MySQL Connector/NET starting with version 6.3.0. The connector will now throw an exception if you try to connect to a server of version less than 5.0.

Bugs Fixed

- When adding a data set in Visual Studio 2008, the following error was generated:

  ```
  Relations couldn't be added. Column 'REFERENCED_TABLE_CATALOG' does not belong to table.
  ```

  This was due to a 'REFERENCED_TABLE_CATALOG' column not being included in the foreign keys collection. (Bug #48974)

- Attempting to execute a load data local infile on a file where the user did not have write permissions, or the file was open in an editor gave an access denied error. (Bug #48944)

- The method `MySqlDataReader.GetSchemaTable()` returned 0 in the `NumericPrecision` field for decimal and newdecimal columns. (Bug #48171)

Changes in MySQL Connector/Net 6.2

Changes in MySQL Connector/NET 6.2.6 (Not released)

Bugs Fixed

- When creating a tableadapter through a Dataset form in Visual Studio, the `MaxLength` of the field for a `CHAR` column could be set to 3 times the length of the table column. Although this many bytes could be needed to hold a UTF-8 character value, the length value was not appropriate for restricting the length of a `TextBox`. (Bug #12860224, Bug #62094)

- Executing a “LINQ to Entity” query could result in a `NullReferenceException` error. (Bug #12776598, Bug #61729)

- Using a combination of `ListView`, `EntityDataSource` with `TypeFilter` and `Include EntityCollection Navigation Property`, and `DataPager` caused a `NullReferenceException` error in the `System.Web.UI.WebControls.EntityDataSourceView.ExecuteSelect` function. (Bug #12776517, Bug #61714)

- `MySqlScript` was modified to enable correct processing of the `DELIMITER` command when not followed by a new line. (Bug #61680, Bug #12732279)

- The ASP.NET membership provider was modified to create and query all related tables using lowercase names. (Bug #61108, Bug #12702009)

- On Model First changed column types generated in SQL scripts to produce more suitable `MySql` types. (Bug #59244, Bug #12707104)

Changes in MySQL Connector/NET 6.2.5 (2011-07-01)

This release fixes bugs since 6.2.4.

Bugs Fixed

- `MySQLConnectionStringBuilder.ContainsKey()` incorrectly returned `false` when testing whether a keyword was part of the connection string. (Bug #11766671, Bug #59835)

- `SchemaDefinition-5.5.ssdl` was modified to treat `CHAR(36)` columns as a GUID. (Bug #61657, Bug #12708208)

- `SqlFragment.QuoteIdentifier` was modified to add MySQL quotes around identifiers. (Bug #61635, Bug #12707285)

- Modified `MySqlConnection.BeginTransaction` to throw a `NotSupportedException` for `Snapshot` isolation level. (Bug #61589, Bug #12698020)

- `MysqlDataReader.GetSchemaTable` returned incorrect values and types. (Bug #59989, Bug #11776346)
- All queries other than `INSERT` were executed individually instead of as a batch even though batching was enabled for the connection. (Bug #59616, Bug #11850286)
- MySQL Connector/NET generated an exception when executing a query consisting of ';', for example:

```
mycmd(";",mycon)
mycmd.executenonquery()
```

  The exception generated was:

```
System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at MySql.Data.MySqlClient.MySqlCommand.TrimSemicolons(String sql)
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
```

  (Bug #59537, Bug #11766433)
- Setting `Membership.ApplicationName` had no effect. (Bug #59438, Bug #11770465)
- A `NullReferenceException` was thrown on disposal of a `TransactionScope` object. (Bug #59346, Bug #11766272)
- `MembershipProvider` did not generate hashes correctly if the algorithm was keyed. The Key of the algorithm should have been set if the `HashAlgorithm` was `KeyedHashAlgorithm`. (Bug #58906)
- Code introduced to fix bug #54863 proved problematic on .NET version 3.5 and above. (Bug #58853)
- The `MySqlTokenizer` contained unnecessary `Substring` and `Trim` calls:

```
string token = sql.Substring(startIndex, stopIndex - startIndex).Trim();
```

  The variable `token` was not used anywhere in the code. (Bug #58757)
- `MySqlCommand.ExecuteReader(CommandBehavior)` threw a `NullReferenceException` when being called with `CommandBehavior.CloseConnection`, if the SQL statement contained a syntax error, or contained invalid data such as an invalid column name. (Bug #58652)
- `ReadFieldLength()` returned incorrect value for `BIGINT` autoincrement columns. (Bug #58373)
- MySQL Connector/NET did not support the `utf8mb4` character set. When attempting to connect to `utf8mb4` tables or columns, an exception `KeyNotFoundException` was generated. (Bug #58244)
- A typed dataset did not get the table name. (Bug #57894, Bug #11764989)
- Setting `MySqlCommand.CommandTimeout` to 0 had no effect. It should have resulted in an infinite timeout. (Bug #57265)
- When performing a row-by-row update, only the first row was updated and all other rows were ignored. (Bug #57092)
- Setting the `Default Command Timeout` connection string option had no effect. (Bug #56806)
- When an output parameter was declared as type `MySqlDbType.Bit`, it failed to return with the correct value. (Bug #56756)
- `MySqlHelper.ExecuteReader` did not include an overload accepting `MySqlParameter` objects when using a `MySqlConnection`. However, `MySqlHelper` did include an overload

for `MySqlParameter` objects when using a string object containing the connection string to the database. (Bug #56755)

- Default values returned for text columns were not quoted. This meant that the `COLUMN_DEFAULT` field of the `GetSchema` columns collection did not return a valid SQL expression. (Bug #56509)
- MySQL Connector/NET for .NET/Mono attempted to dynamically load the assembly `Mono.Posix.dll` when a Unix socket was used to connect to the server. This failed and the connector was not able to use a Unix socket unless the `Mono.Posix.dll` assembly was previously loaded by the program. (Bug #56410)
- Modified `ProviderManifest.xml` to map `TIMESTAMP` database columns to the `DateTime` .NET type. (Bug #55351, Bug #12652602)
- The ADO.NET Entity Data Model could not add stored procedures from MySQL Server 5.0.45 but worked fine using MySQL Server 5.1. (Bug #55349)
- Fixed Entity Framework provider `GROUP BY` clause generation by adding all group-by keys to the `SELECT` statement. (Bug #46742, Bug #12622129)

Changes in MySQL Connector/NET 6.2.4 (2010-08-30)

This release fixes bugs since 6.2.3.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Procedure caching had a problem whereby if you created a procedure, dropped it, and recreated it with a different number of parameters an exception was generated.

  MySQL Connector/NET has been changed so that if the procedure is recreated with a different number of parameters, it will still be recognized. (Bug #52562)

Bugs Fixed

- The calculation of `lockAge` in the Session Provider sometimes generated a `System.Data.SqlTypes.SqlNullValueException`. (Bug #55701)
- Attempting to read `Double.MinValue` from a `DOUBLE` column in MySQL table generated the following exception:

```
System.OverflowException : Value was either too large or too small for a Double.

--OverflowException
at System.Number.ParseDouble(String value, NumberStyles options, NumberFormatInfo
numfmt)
at MySql.Data.Types.MySqlDouble.MySql.Data.Types.IMySqlValue.ReadValue(MySqlPacket
packet, Int64 length, Boolean nullVal)
at MySql.Data.MySqlClient.NativeDriver.ReadColumnValue(Int32 index, MySqlField field,
IMySqlValue valObject)
at MySql.Data.MySqlClient.ResultSet.ReadColumnData(Boolean outputParms)
at MySql.Data.MySqlClient.ResultSet.NextRow(CommandBehavior behavior)
at MySql.Data.MySqlClient.MySqlDataReader.Read()
```

  (Bug #55644)

- After an exception, the internal datareader, `MySqlCommand.Connection.Reader`, was not properly closed (it was not set to null). If another query was subsequently executed on that command object an exception was generated with the message “There is already an open DataReader associated with this Connection which must be closed first.” (Bug #55558)

- If using MySQL Server 5.0.x it was not possible to alter stored routines in Visual Studio. If the stored routine was clicked, and the context sensitive menu option, Alter Routine, selected, the following error was generated:

```
Unable to load object with error: Object reference not
set to an instance of an object
```

  (Bug #55170)

- `MySqlDataAdapter.Update()` generated concurrency violations for custom stored procedure driven update commands that used `UpdateRowSource.FirstReturnedRecord`. (Bug #54895)

- Several calls to `DataAdapter.Update()` with intervening changes to `DataTable` resulted in `ConcurrencyException` exceptions being generated. (Bug #54863)

- The icon for the MySQL Web Configuration Tool was not displayed in Visual Studio for Web Application Projects. (Bug #54571)

- The `MySqlHelper` object did not have an overloaded version of the `ExecuteReader` method that accepted a `MySqlConnection` object. (Bug #54570)

- If `MySqlDataAdapter` was used with an `INSERT` command where the `VALUES` clause contained an expression with parentheses in it, and set the `adapter.UpdateBatchSize` parameter to be greater than one, then the call to `adapter.Update` either generated an exception or failed to batch the commands, executing each insert individually. (Bug #54386)

- The method `MySql.Data.Common.QueryNormalizer.CollapseValueList` generated an `ArgumentOutOfRangeException`. (Bug #54152, Bug #53865)

- MySQL Connector/NET did not process `Thread.Abort()` correctly, and failed to cancel queries currently running on the server. (Bug #54012)

- Garbage Collector disposal of a `MySqlConnection` object caused the following exception:

```
System.IO.EndOfStreamException: Attempted to read past the end of the stream.
MySql.Data.MySqlClient.MySqlStream.ReadFully(Stream stream, Byte[] buffer, Int32 offset,
Int32 count)
MySql.Data.MySqlClient.MySqlStream.LoadPacket()
Outer Exception Reading from the stream has failed.
...
```

  (Bug #53457)

- MySQL Connector/NET did not throw an `EndOfStreamException` exception when `net_write_timeout` was exceeded. (Bug #53439)

- After a timeout exception, if an attempt was made to reuse a connection returned to the connection pool the following exception was generated:

```
[MySqlException (0x80004005): There is already an open DataReader associated with this
Connection which must be closed first.]
   MySql.Data.MySqlClient.MySqlCommand.CheckState() +278
   MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior) +43
   MySql.Data.MySqlClient.MySqlCommand.ExecuteReader() +6
   Controls.SimpleCommand.ExecuteReader(String SQL) in ...:323
   Albums.GetImagesByAlbum(SimpleCommand Cmd, Int32 iAlbum, String Order, String Limit)
in ...:13
   Forecast.Page_Load(Object sender, EventArgs e) in ...:70
   System.Web.UI.Control.OnLoad(EventArgs e) +99
   System.Web.UI.Control.LoadRecursive() +50
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean
includeStagesAfterAsyncPoint) +627
```

  (Bug #53357)

- Membership schema creation failed if the default schema collation was not Latin1. (Bug #53174)

- In MySQL Connector/NET, the `MySqlConnection.Abort()` method contained a `try...catch` construct, with an empty `catch` block. This meant that any exception generated at this point would not be caught. (Bug #52769)
- EventLog was not disposed in the SessionState provider. (Bug #52550)
- The procedure cache affected the MySQL Connector/NET performance, reducing it by around 65%. This was due to unnecessary calls of `String.Format()`, related to debug logging. Even though the logging was disabled the string was still being formatted, resulting in impaired performance. (Bug #52475)
- If `FunctionsReturnString=true` was used in the connection string, the decimal separator (according to locale) was not interpreted. (Bug #52187)
- Periodically the session provider threw an `SqlNullValueException` exception. When this happened, the row within the `my_aspnet_Sessions` table had `locked` always set to '1'. The locked status never changed back to '0' and the user experienced the exception on every page, until their browser was closed and reopened (recreating a new sessionID), or the `locked` value was manually changed to '0'. (Bug #52175)
- `CHAR(36)` columns were not recognized as GUIDs when used in views with entity models. (Bug #52085)
- In MySQL Connector/NET, the `LoadCharsetMap()` function of the `CharSetMap` class set the following incorrect mapping:

  ```
  mapping.Add("latin1", new CharacterSet("latin1", 1));
  ```

  This meant that, for example, the Euro sign was not handled correctly.

  The correct mapping should have been:

  ```
  mapping.Add("latin1", new CharacterSet("windows-1252", 1));
  ```

  This is because the MySQL `latin1` character set is the same as the `windows-cp1252` character set and it extends the official ISO 8859-1 or IANA latin1. (Bug #51927)
- Stored procedure enumeration code generated an error if a procedure was used in a dataset that did not return any resultsets. (Bug #50671)
- When an application was subjected to increased concurrent load, MySQL Connector/NET generated the following error when calling stored procedures:

  ```
  A DataTable named \'Procedure Parameters\'
  already belongs to this DataSet.
  ```

  (Bug #49118)
- In the ADO.NET Entity Data Model Wizard, the time to update a model scaled abnormally as the number of entities increased. (Bug #48791, Bug #12596237)
- The `INSERT` command was significantly slower with MySQL Connector/NET 6.x compared to 5.x, when compression was enabled. (Bug #48243)
- When the connection string option “Connection Reset = True” was used, a connection reset used the previously used encoding for the subsequent authentication operation. This failed, for example, if UCS2 was used to read the last column before the reset. (Bug #47153)
- Opening a connection in the Visual Studio Server Explorer and choosing to alter an existing routine required another authentication at the server. (Bug #44715)
- When batching was used in `MySqlDataAdapter`, a connection was not opened automatically in `MySqlDataAdapter.Update()`. This resulted in an `InvalidOperationException` exception being generated, with the message text “connection must be valid and open”.

MySQL Connector/NET has been changed to behave more like SQL Server: if the connection is closed, it is opened for the duration of update operation. (Bug #38411)

- Database name was emitted into typed datasets. This prevented users using the configured default database. (Bug #33870)

Changes in MySQL Connector/NET 6.2.3 (2010-04-10)

This release fixes bugs since 6.2.2.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- MySQL Connector/NET has been changed to include `MySqlDataReader.GetFieldType(string columnname)`. Further, `MySqlDataReader.GetOrdinal()` now includes the name of the column in the exception if the column is not found. (Bug #47467)

Bugs Fixed

- A non-terminated string in SQL threw a CLR exception rather than a syntax exception. (Bug #51788)
- When calling `ExecuteNonQuery` on a command object, the following exception occurred:

```
Index and length must refer to a location within the string.
Parameter name: length
```

(Bug #51610)

- Specifying a connection string where an option had no value generated an error, rather than the value being set to the default. For example, a connection string such as the following would result in an error:

```
server=localhost;user=root;compress=;database=test;port=3306;password=123456;
```

(Bug #51209)

- The method `Command.TrimSemicolons` used `StringBuilder`, and therefore allocated memory for the query even if it did not need to be trimmed. This led to excessive memory consumption when executing a number of large queries. (Bug #51149)
- `MySqlCommand.Parameters.Clear()` did not work. (Bug #50444)
- When the `MySqlScript.execute()` method was called, the following exception was generated:

```
InvalidOperationException : The CommandText property has not been properly initialized.
```

(Bug #50344)

- When using the Compact Framework the following exception occurred when attempting to connect to a MySQL Server:

```
System.InvalidOperationException was unhandled
  Message="Timeouts are not supported on this stream."
```

(Bug #50321)

- Binary Columns were not displayed in the Query Builder of Visual Studio. (Bug #50171)
- When the `UpdateBatchSize` property was set to a value greater than 1, only the first row was applied to the database. (Bug #50123)

- When using table per type inheritance and listing the contents of the parent table, the result of the query was a list of child objects, even though there was no related child record with the same parent Id. (Bug #49850)
- `MySqlDataReader.GetUInt64` returned an incorrect value when reading a `BIGINT UNSIGNED` column containing a value greater than 2147483647. (Bug #49794)
- A `FormatException` was generated when an empty string was returned from a stored function. (Bug #49642)
- When trying to create stored procedures from an SQL script, a `MySqlException` was thrown when attempting to redefine the `DELIMITER`:

```
MySql.Data.MySqlClient.MySqlException was unhandled
Message="You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'DELIMITER' at line 1"
Source="MySql.Data"
ErrorCode=-2147467259
Number=1064
StackTrace:
à MySql.Data.MySqlClient.MySqlStream.ReadPacket()
à MySql.Data.MySqlClient.NativeDriver.ReadResult(UInt64& affectedRows, Int64&
lastInsertId)
à MySql.Data.MySqlClient.MySqlDataReader.GetResultSet()
à MySql.Data.MySqlClient.MySqlDataReader.NextResult()
à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
à MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
à MySql.Data.MySqlClient.MySqlScript.Execute()
```

  Note: The `MySqlScript` class has been fixed to support the delimiter statement as it is found in SQL scripts. (Bug #46429)
- Calling a User Defined Function using Entity SQL in the Entity Framework caused a `NullReferenceException`. (Bug #45277)
- A connection string set in `web.config` could not be reused after Visual Studio 2008 Professional was shut down. It continued working for the existing controls, but did not work for new controls added. (Bug #41629)

Changes in MySQL Connector/NET 6.2.2 (2009-12-22, General Availability)

First GA release of 6.2. This release fixes bugs since 6.2.1.

Bugs Fixed

- When adding a data set in Visual Studio 2008, the following error was generated:

```
Relations couldn't be added. Column 'REFERENCED_TABLE_CATALOG' does not belong to table.
```

  This was due to a 'REFERENCED_TABLE_CATALOG' column not being included in the foreign keys collection. (Bug #48974)
- Attempting to execute a load data local infile on a file where the user did not have write permissions, or the file was open in an editor gave an access denied error. (Bug #48944)
- The method `MySqlDataReader.GetSchemaTable()` returned 0 in the `NumericPrecision` field for decimal and newdecimal columns. (Bug #48171)
- MySQL Connector/NET generated an invalid operation exception during a transaction rollback:

```
System.InvalidOperationException: Connection must be valid and open to rollback
transaction
   at MySql.Data.MySqlClient.MySqlTransaction.Rollback()
   at MySql.Data.MySqlClient.MySqlConnection.CloseFully()
```

```
   at
MySql.Data.MySqlClient.MySqlPromotableTransaction.System.Transactions.IPromotableSinglePhaseNotificat
singlePhaseEnlistment)
...
```

(Bug #35330)

- Connection objects were not garbage collected when not in use. (Bug #31996)

Changes in MySQL Connector/NET 6.2.1 (2009-11-16, Beta)

This release fixes bugs since 6.2.0.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- The `MySqlParameter` class now has a property named `PossibleValues`. This property is NULL unless the parameter is created by `MySqlCommandBuilder.DeriveParameters`. Further, it will be NULL unless the parameter is of type enum or set - in this case it will be a list of strings that are the possible values for the column. This feature is designed as an aid to the developer. (Bug #48586)

- Prior to MySQL Connector/NET 6.2, `MySqlCommand.CommandTimeout` included user processing time, that is processing time not related to direct use of the connector. Timeout was implemented through a .NET Timer, that triggered after `CommandTimeout` seconds.

  MySQL Connector/NET 6.2 introduced timeouts that are aligned with how Microsoft handles `SqlCommand.CommandTimeout`. This property is the cumulative timeout for all network reads and writes during command execution or processing of the results. A timeout can still occur in the `MySqlReader.Read` method after the first row is returned, and does not include user processing time, only IO operations.

  Further details on this can be found in the relevant Microsoft documentation.

- Starting with MySQL Connector/NET 6.2, there is a background job that runs every three minutes and removes connections from pool that have been idle (unused) for more than three minutes. The pool cleanup frees resources on both client and server side. This is because on the client side every connection uses a socket, and on the server side every connection uses a socket and a thread.

  Prior to this change, connections were never removed from the pool, and the pool always contained the peak number of open connections. For example, a web application that peaked at 1000 concurrent database connections would consume 1000 threads and 1000 open sockets at the server, without ever freeing up those resources from the connection pool.

- MySQL Connector/NET now supports the processing of certificates when connecting to an SSL-enabled MySQL Server. For more information, see Tutorial: Configuring SSL with Connector/NET.

Bugs Fixed

- Cloning of `MySqlCommand` was not typesafe. To clone a `MySqlCommand` it was necessary to do:

  ```
  MySqlCommand clone = (MySqlCommand)((ICloneable)comm).Clone();
  ```

  MySQL Connector/NET was changed so that it was possible to do:

  ```
  MySqlCommand clone = comm.Clone();
  ```

  (Bug #48460)

- When used, the `Encrypt` connection string option caused a “Keyword not supported” exception to be generated.

This option is in fact obsolete, and the option SSL Mode should be used instead. Although the `Encrypt` option has been fixed so that it does not generate an exception, it will be removed completely in version 6.4. (Bug #48290)

- When building the `MySql.Data` project with .NET Framework 3.5 installed, the following build output was displayed:

```
Project file contains ToolsVersion="4.0", which is not supported by this version of
MSBuild. Treating the project as if it had ToolsVersion="3.5".
```

  The project had been created using the .NET Framework 4.0, which was beta, instead of using the 3.5 framework. (Bug #48271)

- It was not possible to retrieve a value from a MySQL server table, if the value was larger than that supported by the .NET type `System.Decimal`.

  MySQL Connector/NET was changed to expose the `MySqlDecimal` type, along with the supporting method `GetMySqlDecimal`. (Bug #48100)

- An entity model created from a schema containing a table with a column of type `UNSIGNED BIGINT` and a view of the table did not behave correctly. When an entity was created and mapped to the view, the column that was of type `UNSIGNED BIGINT` was displayed as `BIGINT`. (Bug #47872)

- MySQL Connector/NET session support did not work with MySQL Server versions prior to 5.0, as the Session Provider used a call to `TIMESTAMPDIFF`, which was not available on servers prior to 5.0. (Bug #47219)

Changes in MySQL Connector/NET 6.2.0 (2009-10-21, Alpha)

The first alpha release of 6.2.

Bugs Fixed

- When using a `BINARY(16)` column to represent a GUID and having specified “old guids = true” in the connection string, the values were returned correctly until a null value was encountered in that field. After the null value was encountered a format exception was thrown with the following message:

```
Guid should contain 32 digits with 4 dashes (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx).
```

  (Bug #47928)

- The Session Provider created invalid “session expires” on a random basis.

  This was due to the fact that the Session Provider was incorrectly reading from the root `web.config`, rather than from the application specific `web.config`. (Bug #47815)

- When loading the `MySQLClient-mono.sln` file included with the Connector/NET source into Mono Develop, the following error occurred:

```
/home/tbedford/connector-net-src/6.1/MySQLClient-mono.sln(22):
Unsupported or unrecognized project:
'/home/tbedford/connector-net-src/6.1/Installer/Installer.wixproj'
```

  If the file was modified to remove this problem, then attempting to build the solution generated the following error:

```
/home/tbedford/connector-net-src/6.1/MySql.Data/Provider/Source/Connection.cs(280,46):
error CS0115: `MySql.Data.MySqlClient.MySqlConnection.DbProviderFactory' is marked as an
override but no suitable property found to override
```

  (Bug #47048)

Changes in MySQL Connector/Net 6.1

Changes in MySQL Connector/NET 6.1.7 (Not released)

Bugs Fixed

- `MySqlScript` was modified to enable correct processing of the `DELIMITER` command when not followed by a new line. (Bug #61680, Bug #12732279)
- `SchemaDefinition-5.5.ssdl` was modified to treat `CHAR(36)` columns as a GUID. (Bug #61657, Bug #12708208)
- `SqlFragment.QuoteIdentifier` was modified to add MySQL quotes around identifiers. (Bug #61635, Bug #12707285)
- The ASP.NET membership provider was modified to create and query all related tables using lowercase names. (Bug #61108, Bug #12702009)

Changes in MySQL Connector/NET 6.1.6 (2011-06-30)

This release fixes bugs since 6.1.5.

Bugs Fixed

- `MySQLConnectionStringBuilder.ContainsKey()` incorrectly returned `false` when testing whether a keyword was part of the connection string. (Bug #11766671, Bug #59835)
- Modified `MySqlConnection.BeginTransaction` to throw a `NotSupportedException` for `Snapshot` isolation level. (Bug #61589, Bug #12698020)
- `MysqlDataReader.GetSchemaTable` returned incorrect values and types. (Bug #59989, Bug #11776346)
- All queries other than `INSERT` were executed individually instead of as a batch even though batching was enabled for the connection. (Bug #59616, Bug #11850286)
- MySQL Connector/NET generated an exception when executing a query consisting of ';', for example:

```
mycmd(";",mycon)
mycmd.executenonquery()
```

  The exception generated was:

```
System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at MySql.Data.MySqlClient.MySqlCommand.TrimSemicolons(String sql)
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
```

  (Bug #59537, Bug #11766433)
- Setting `Membership.ApplicationName` had no effect. (Bug #59438, Bug #11770465)
- `MembershipProvider` did not generate hashes correctly if the algorithm was keyed. The Key of the algorithm should have been set if the `HashAlgorithm` was `KeyedHashAlgorithm`. (Bug #58906)
- Code introduced to fix bug #54863 proved problematic on .NET version 3.5 and above. (Bug #58853)
- The `MySqlTokenizer` contained unnecessary `Substring` and `Trim` calls:

```
string token = sql.Substring(startIndex, stopIndex - startIndex).Trim();
```

The variable `token` was not used anywhere in the code. (Bug #58757)

- `MySqlCommand.ExecuteReader(CommandBehavior)` threw a `NullReferenceException` when being called with `CommandBehavior.CloseConnection`, if the SQL statement contained a syntax error, or contained invalid data such as an invalid column name. (Bug #58652)
- `ReadFieldLength()` returned incorrect value for `BIGINT` autoincrement columns. (Bug #58373)
- MySQL Connector/NET did not support the `utf8mb4` character set. When attempting to connect to `utf8mb4` tables or columns, an exception `KeyNotFoundException` was generated. (Bug #58244)
- A typed dataset did not get the table name. (Bug #57894, Bug #11764989)
- Setting `MySqlCommand.CommandTimeout` to 0 had no effect. It should have resulted in an infinite timeout. (Bug #57265)
- When performing a row-by-row update, only the first row was updated and all other rows were ignored. (Bug #57092)
- Setting the `Default Command Timeout` connection string option had no effect. (Bug #56806)
- When an output parameter was declared as type `MySqlDbType.Bit`, it failed to return with the correct value. (Bug #56756)
- `MySqlHelper.ExecuteReader` did not include an overload accepting `MySqlParameter` objects when using a `MySqlConnection`. However, `MySqlHelper` did include an overload for `MySqlParameter` objects when using a string object containing the connection string to the database. (Bug #56755)
- Default values returned for text columns were not quoted. This meant that the `COLUMN_DEFAULT` field of the `GetSchema` columns collection did not return a valid SQL expression. (Bug #56509)
- MySQL Connector/NET for .NET/Mono attempted to dynamically load the assembly `Mono.Posix.dll` when a Unix socket was used to connect to the server. This failed and the connector was not able to use a Unix socket unless the `Mono.Posix.dll` assembly was previously loaded by the program. (Bug #56410)
- Modified `ProviderManifest.xml` to map `TIMESTAMP` database columns to the `DateTime` .NET type. (Bug #55351, Bug #12652602)
- The ADO.NET Entity Data Model could not add stored procedures from MySQL Server 5.0.45 but worked fine using MySQL Server 5.1. (Bug #55349)
- Fixed Entity Framework provider `GROUP BY` clause generation by adding all group-by keys to the `SELECT` statement. (Bug #46742, Bug #12622129)

Changes in MySQL Connector/NET 6.1.5 (2010-08-30)

This release fixes bugs since 6.1.4.

Bugs Fixed

- The calculation of `lockAge` in the Session Provider sometimes generated a `System.Data.SqlTypes.SqlNullValueException`. (Bug #55701)
- Attempting to read `Double.MinValue` from a `DOUBLE` column in MySQL table generated the following exception:

```
System.OverflowException : Value was either too large or too small for a Double.

--OverflowException
at System.Number.ParseDouble(String value, NumberStyles options, NumberFormatInfo
numfmt)
```

```
at MySql.Data.Types.MySqlDouble.MySql.Data.Types.IMySqlValue.ReadValue(MySqlPacket
packet, Int64 length, Boolean nullVal)
at MySql.Data.MySqlClient.NativeDriver.ReadColumnValue(Int32 index, MySqlField field,
IMySqlValue valObject)
at MySql.Data.MySqlClient.ResultSet.ReadColumnData(Boolean outputParms)
at MySql.Data.MySqlClient.ResultSet.NextRow(CommandBehavior behavior)
at MySql.Data.MySqlClient.MySqlDataReader.Read()
```

(Bug #55644)

- If using MySQL Server 5.0.x it was not possible to alter stored routines in Visual Studio. If the stored routine was clicked, and the context sensitive menu option, Alter Routine, selected, the following error was generated:

```
Unable to load object with error: Object reference not
set to an instance of an object
```

(Bug #55170)

- `MySqlDataAdapter.Update()` generated concurrency violations for custom stored procedure driven update commands that used `UpdateRowSource.FirstReturnedRecord`. (Bug #54895)
- Several calls to `DataAdapter.Update()` with intervening changes to `DataTable` resulted in `ConcurrencyException` exceptions being generated. (Bug #54863)
- The icon for the MySQL Web Configuration Tool was not displayed in Visual Studio for Web Application Projects. (Bug #54571)
- The `MySqlHelper` object did not have an overloaded version of the `ExecuteReader` method that accepted a `MySqlConnection` object. (Bug #54570)
- If `MySqlDataAdapter` was used with an `INSERT` command where the `VALUES` clause contained an expression with parentheses in it, and set the `adapter.UpdateBatchSize` parameter to be greater than one, then the call to `adapter.Update` either generated an exception or failed to batch the commands, executing each insert individually. (Bug #54386)
- The method `MySql.Data.Common.QueryNormalizer.CollapseValueList` generated an `ArgumentOutOfRangeException`. (Bug #54152, Bug #53865)
- Garbage Collector disposal of a `MySqlConnection` object caused the following exception:

```
System.IO.EndOfStreamException: Attempted to read past the end of the stream.
MySql.Data.MySqlClient.MySqlStream.ReadFully(Stream stream, Byte[] buffer, Int32 offset,
Int32 count)
MySql.Data.MySqlClient.MySqlStream.LoadPacket()
Outer Exception Reading from the stream has failed.
...
```

(Bug #53457)

- MySQL Connector/NET did not throw an `EndOfStreamException` exception when `net_write_timeout` was exceeded. (Bug #53439)
- After a timeout exception, if an attempt was made to reuse a connection returned to the connection pool the following exception was generated:

```
[MySqlException (0x80004005): There is already an open DataReader associated with this
Connection which must be closed first.]
   MySql.Data.MySqlClient.MySqlCommand.CheckState() +278
   MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior) +43
   MySql.Data.MySqlClient.MySqlCommand.ExecuteReader() +6
   Controls.SimpleCommand.ExecuteReader(String SQL) in ...:323
   Albums.GetImagesByAlbum(SimpleCommand Cmd, Int32 iAlbum, String Order, String Limit)
in ...:13
   Forecast.Page_Load(Object sender, EventArgs e) in ...:70
   System.Web.UI.Control.OnLoad(EventArgs e) +99
   System.Web.UI.Control.LoadRecursive() +50
```

```
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean
includeStagesAfterAsyncPoint) +627
```

(Bug #53357)

- Membership schema creation failed if the default schema collation was not Latin1. (Bug #53174)
- EventLog was not disposed in the SessionState provider. (Bug #52550)
- `CHAR(36)` columns were not recognized as GUIDs when used in views with entity models. (Bug #52085)
- Stored procedure enumeration code generated an error if a procedure was used in a dataset that did not return any resultsets. (Bug #50671)
- When an application was subjected to increased concurrent load, MySQL Connector/NET generated the following error when calling stored procedures:

```
A DataTable named \'Procedure Parameters\'
already belongs to this DataSet.
```

(Bug #49118)

- In the ADO.NET Entity Data Model Wizard, the time to update a model scaled abnormally as the number of entities increased. (Bug #48791, Bug #12596237)
- The `INSERT` command was significantly slower with MySQL Connector/NET 6.x compared to 5.x, when compression was enabled. (Bug #48243)
- When the connection string option “Connection Reset = True” was used, a connection reset used the previously used encoding for the subsequent authentication operation. This failed, for example, if UCS2 was used to read the last column before the reset. (Bug #47153)
- Opening a connection in the Visual Studio Server Explorer and choosing to alter an existing routine required another authentication at the server. (Bug #44715)
- When batching was used in `MySqlDataAdapter`, a connection was not opened automatically in `MySqlDataAdapter.Update()`. This resulted in an `InvalidOperationException` exception being generated, with the message text “connection must be valid and open”.

  MySQL Connector/NET has been changed to behave more like SQL Server: if the connection is closed, it is opened for the duration of update operation. (Bug #38411)
- Database name was emitted into typed datasets. This prevented users using the configured default database. (Bug #33870)

Changes in MySQL Connector/NET 6.1.4 (2010-04-28)

This release fixes bugs since 6.1.3.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Procedure caching had a problem whereby if you created a procedure, dropped it, and recreated it with a different number of parameters an exception was generated.

  MySQL Connector/NET has been changed so that if the procedure is recreated with a different number of parameters, it will still be recognized. (Bug #52562)
- MySQL Connector/NET has been changed to include `MySqlDataReader.GetFieldType(string columnname)`. Further,

`MySqlDataReader.GetOrdinal()` now includes the name of the column in the exception if the column is not found. (Bug #47467)

Bugs Fixed

- In MySQL Connector/NET, the `MySqlConnection.Abort()` method contained a `try...catch` construct, with an empty `catch` block. This meant that any exception generated at this point would not be caught. (Bug #52769)
- If `FunctionsReturnString=true` was used in the connection string, the decimal separator (according to locale) was not interpreted. (Bug #52187)
- In MySQL Connector/NET, the `LoadCharsetMap()` function of the `CharSetMap` class set the following incorrect mapping:

  ```
  mapping.Add("latin1", new CharacterSet("latin1", 1));
  ```

  This meant that, for example, the Euro sign was not handled correctly.

  The correct mapping should have been:

  ```
  mapping.Add("latin1", new CharacterSet("windows-1252", 1));
  ```

  This is because the MySQL `latin1` character set is the same as the `windows-cp1252` character set and it extends the official ISO 8859-1 or IANA latin1. (Bug #51927)
- A non-terminated string in SQL threw a CLR exception rather than a syntax exception. (Bug #51788)
- When calling `ExecuteNonQuery` on a command object, the following exception occurred:

  ```
  Index and length must refer to a location within the string.
  Parameter name: length
  ```

  (Bug #51610)
- The method `Command.TrimSemicolons` used `StringBuilder`, and therefore allocated memory for the query even if it did not need to be trimmed. This led to excessive memory consumption when executing a number of large queries. (Bug #51149)
- `MySqlCommand.Parameters.Clear()` did not work. (Bug #50444)
- When the `MySqlScript.execute()` method was called, the following exception was generated:

  ```
  InvalidOperationException : The CommandText property has not been properly initialized.
  ```

  (Bug #50344)
- Binary Columns were not displayed in the Query Builder of Visual Studio. (Bug #50171)
- When the `UpdateBatchSize` property was set to a value greater than 1, only the first row was applied to the database. (Bug #50123)
- When using table per type inheritance and listing the contents of the parent table, the result of the query was a list of child objects, even though there was no related child record with the same parent Id. (Bug #49850)
- `MySqlDataReader.GetUInt64` returned an incorrect value when reading a `BIGINT UNSIGNED` column containing a value greater than 2147483647. (Bug #49794)
- A `FormatException` was generated when an empty string was returned from a stored function. (Bug #49642)
- When adding a data set in Visual Studio 2008, the following error was generated:

  ```
  Relations couldn't be added. Column 'REFERENCED_TABLE_CATALOG' does not belong to table.
  ```

This was due to a 'REFERENCED_TABLE_CATALOG' column not being included in the foreign keys collection. (Bug #48974)

- Attempting to execute a load data local infile on a file where the user did not have write permissions, or the file was open in an editor gave an access denied error. (Bug #48944)
- The method `MySqlDataReader.GetSchemaTable()` returned 0 in the `NumericPrecision` field for decimal and newdecimal columns. (Bug #48171)
- When trying to create stored procedures from an SQL script, a `MySqlException` was thrown when attempting to redefine the `DELIMITER`:

```
MySql.Data.MySqlClient.MySqlException was unhandled
Message="You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'DELIMITER' at line 1"
Source="MySql.Data"
ErrorCode=-2147467259
Number=1064
StackTrace:
à MySql.Data.MySqlClient.MySqlStream.ReadPacket()
à MySql.Data.MySqlClient.NativeDriver.ReadResult(UInt64& affectedRows, Int64&
lastInsertId)
à MySql.Data.MySqlClient.MySqlDataReader.GetResultSet()
à MySql.Data.MySqlClient.MySqlDataReader.NextResult()
à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
à MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
à MySql.Data.MySqlClient.MySqlScript.Execute()
```

Note: The `MySqlScript` class has been fixed to support the delimiter statement as it is found in SQL scripts. (Bug #46429)

- Calling a User Defined Function using Entity SQL in the Entity Framework caused a `NullReferenceException`. (Bug #45277)
- A connection string set in `web.config` could not be reused after Visual Studio 2008 Professional was shut down. It continued working for the existing controls, but did not work for new controls added. (Bug #41629)

Changes in MySQL Connector/NET 6.1.3 (2009-11-16)

This release fixes bugs since 6.1.2.

Bugs Fixed

- Cloning of `MySqlCommand` was not typesafe. To clone a `MySqlCommand` it was necessary to do:

```
MySqlCommand clone = (MySqlCommand)((ICloneable)comm).Clone();
```

MySQL Connector/NET was changed so that it was possible to do:

```
MySqlCommand clone = comm.Clone();
```

(Bug #48460)

- When building the `MySql.Data` project with .NET Framework 3.5 installed, the following build output was displayed:

```
Project file contains ToolsVersion="4.0", which is not supported by this version of
MSBuild. Treating the project as if it had ToolsVersion="3.5".
```

The project had been created using the .NET Framework 4.0, which was beta, instead of using the 3.5 framework. (Bug #48271)

- If `MySqlConnection.GetSchema` was called for "Indexes" on a table named “b`a`d” as follows:

```
DataTable schemaPrimaryKeys = connection.GetSchema(
  "Indexes",
  new string[] { null, schemaName, "b`a`d"});
```

Then the following exception was generated:

```
You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near 'a`d`' at line 1
```

(Bug #48101)

- It was not possible to retrieve a value from a MySQL server table, if the value was larger than that supported by the .NET type `System.Decimal`.

  MySQL Connector/NET was changed to expose the `MySqlDecimal` type, along with the supporting method `GetMySqlDecimal`. (Bug #48100)

- For some character sets such as UTF-8, a `CHAR` column would sometimes be incorrectly interpreted as a `GUID` by MySQL Connector/NET.

  MySQL Connector/NET was changed so that a column would only be interpreted as a `GUID` if it had a character length of 36, as opposed to a byte length of 36. (Bug #47985)

- When using a `BINARY(16)` column to represent a GUID and having specified “old guids = true” in the connection string, the values were returned correctly until a null value was encountered in that field. After the null value was encountered a format exception was thrown with the following message:

  ```
  Guid should contain 32 digits with 4 dashes (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx).
  ```

  (Bug #47928)

- An entity model created from a schema containing a table with a column of type `UNSIGNED BIGINT` and a view of the table did not behave correctly. When an entity was created and mapped to the view, the column that was of type `UNSIGNED BIGINT` was displayed as `BIGINT`. (Bug #47872)

- The Session Provider created invalid “session expires” on a random basis.

  This was due to the fact that the Session Provider was incorrectly reading from the root `web.config`, rather than from the application specific `web.config`. (Bug #47815)

- Attempting to build MySQL Connector/NET 6.1 `MySQL.Data` from source code on Windows failed with the following error:

  ```
  ...\clones\6.1\MySql.Data\Provider\Source\NativeDriver.cs(519,29): error CS0122:
  'MySql.Data.MySqlClient.MySqlPacket.MySqlPacket()' is inaccessible due to its protection level
  ```

  (Bug #47354)

- When tables were auto created for the Session State Provider they were set to use the MySQL Server's default collation, rather than the default collation set for the containing database. (Bug #47332)

- When loading the `MySQLClient-mono.sln` file included with the Connector/NET source into Mono Develop, the following error occurred:

  ```
  /home/tbedford/connector-net-src/6.1/MySQLClient-mono.sln(22):
  Unsupported or unrecognized project:
  '/home/tbedford/connector-net-src/6.1/Installer/Installer.wixproj'
  ```

  If the file was modified to remove this problem, then attempting to build the solution generated the following error:

  ```
  /home/tbedford/connector-net-src/6.1/MySql.Data/Provider/Source/Connection.cs(280,46):
  ```

```
error CS0115: `MySql.Data.MySqlClient.MySqlConnection.DbProviderFactory' is marked as an
override but no suitable property found to override
```

(Bug #47048)

Changes in MySQL Connector/NET 6.1.2 (2009-09-08, General Availability)

This is the first GA release of 6.1. This release fixes bugs since 6.1.1.

Bugs Fixed

- The MySQL Connector/NET Session State Provider truncated session data to 64KB, due to its column types being set to `BLOB`. (Bug #47339)
- MySQL Connector/NET generated the following exception when using the Session State provider:

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL
server version for the right syntax to use near 'MINUTEWHERE SessionId =
'dtmgga55x35oi255nrfrxe45' AND ApplicationId = 1 AND Loc' at line 1
Description: An unhandled exception occurred during the execution of the current web
request. Please review the stack trace for more information about the error and where it
originated in the code.

Exception Details: MySql.Data.MySqlClient.MySqlException: You have an error in your SQL
syntax; check the manual that corresponds to your MySQL server version for the right
syntax to use near 'MINUTEWHERE SessionId = 'dtmgga55x35oi255nrfrxe45' AND ApplicationId =
1 AND Loc' at line 1
```

(Bug #46939)

- If an error occurred during connection to a MySQL Server, deserializing the error message from the packet buffer caused a `NullReferenceException` to be thrown. When the method `MySqlPacket::ReadString()` attempted to retrieve the error message, the following line of code threw the exception:

```
string s = encoding.GetString(bits, (int)buffer.Position, end - (int)buffer.Position);
```

This was due to the fact that the encoding field had not been initialized correctly. (Bug #46844)

- Input parameters were missing from Stored Procedures when using them with ADO.NET Data Entities. (Bug #44985)
- MySQL Connector/NET did not time out correctly. The command timeout was set to 30 secs, but MySQL Connector/NET hung for several hours. (Bug #43761)

Changes in MySQL Connector/NET 6.1.1 (2009-08-20, Beta)

This is the first Beta release of 6.1.

Bugs Fixed

- In the `MySqlDataReader` class the `GetSByte` function returned a `byte` value instead of an `sbyte` value. (Bug #46620)
- The MySQL Connector/NET Profile Provider, `MySql.Web.Profile.MySQLProfileProvider`, generated an error when running on Mono. When an attempt was made to save a string in `Profile.Name` the string was not saved to the `my_aspnet_Profiles` table. If an attempt was made to force the save with `Profile.Save()` the following error was generated:

```
Server Error in '/mono' Application

--------------------------------------------------------------------------------

The requested feature is not implemented.
Description: HTTP 500. Error processing request.
```

```
Stack Trace:

System.NotImplementedException: The requested feature is not implemented.
at MySql.Data.MySqlClient.MySqlConnection.EnlistTransaction
(System.Transactions.Transaction transaction) [0x00000]
at MySql.Data.MySqlClient.MySqlConnection.Open () [0x00000]
at MySql.Web.Profile.MySQLProfileProvider.SetPropertyValues
(System.Configuration.SettingsContext context,
System.Configuration.SettingsPropertyValueCollection collection) [0x00000]


--------------------------------------------------------------------------------
Version information: Mono Version: 2.0.50727.1433; ASP.NET Version: 2.0.50727.1433
```

(Bug #46375)

- An exception was generated when using `TIMESTAMP` columns with the Entity Framework. (Bug #46311)
- MySQL Connector/NET sometimes hung, without generating an exception. This happened if a read from a stream failed returning a 0, causing the code in `LoadPacket()` to enter an infinite loop. (Bug #46308)
- When using MySQL Connector/NET 6.0.4 and a MySQL Server 4.1 an exception was generated when trying to execute:

  ```
  connection.GetSchema("Columns", ...);
  ```

  The exception generated was:

  ```
  'connection.GetSchema("Columns")' threw an exception of type
  'System.ArgumentException'System.Data.DataTable {System.ArgumentException}
  base{"Input string was not in a correct format.Couldn't store <'Select'> in
  NUMERIC_PRECISION Column.  Expected type is UInt64."}System.Exception
  {System.ArgumentException}
  ```

  (Bug #46270)

- The MySQL Connector/NET method `StoredProcedure.GetParameters(string)` ignored the programmer's setting of the `UseProcedureBodies` option. This broke any application for which the application's parameter names did not match the parameter names in the Stored Procedure, resulting in an `ArgumentException` with the message “Parameter 'foo' not found in the collection.” and the following stack trace:

  ```
  MySql.Data.dll!MySql.Data.MySqlClient.MySqlParameterCollection.GetParameterFlexible(stri
  ng parameterName = "pStart", bool throwOnNotFound = true) Line 459C#
  MySql.Data.dll!MySql.Data.MySqlClient.StoredProcedure.Resolve() Line 157 + 0x25
  bytesC#
  MySql.Data.dll!MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(System.Data.CommandBeha
  vior behavior = SequentialAccess) Line 405 + 0xb bytesC#
  MySql.Data.dll!MySql.Data.MySqlClient.MySqlCommand.ExecuteDbDataReader(System.Data.Comma
  ndBehavior behavior = SequentialAccess) Line 884 + 0xb bytesC#
  System.Data.dll!System.Data.Common.DbCommand.System.Data.IDbCommand.ExecuteReader(System
  .Data.CommandBehavior behavior) + 0xb bytes
  System.Data.dll!System.Data.Common.DbDataAdapter.FillInternal(System.Data.DataSet
  dataset = {System.Data.DataSet}, System.Data.DataTable[] datatables = null, int
  startRecord = 0, int maxRecords = 0, string srcTable = "Table", System.Data.IDbCommand
  command = {MySql.Data.MySqlClient.MySqlCommand}, System.Data.CommandBehavior behavior) +
  0x83 bytes
  System.Data.dll!System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet dataSet, int
  startRecord, int maxRecords, string srcTable, System.Data.IDbCommand command,
  System.Data.CommandBehavior behavior) + 0x120 bytes
  System.Data.dll!System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet dataSet) +
  0x5f bytes
  ```

  (Bug #46213)

- Conversion of MySQL `TINYINT(1)` to `boolean` failed. (Bug #46205, Bug #46359, Bug #41953)

- When populating a MySQL database table in Visual Studio using the Table Editor, if a `VARCHAR(10)` column was changed to a `VARCHAR(20)` column an exception was generated:

```
SystemArgumentException: DataGridViewComboBoxCell value is not valid.
To replace this default dialog please handle the DataError Event.
```

  (Bug #46100)

- The Entity Framework provider was not calling `DBSortExpression` correctly when the `Skip` and `Take` methods were used, such as in the following statement:

```
TestModel.tblquarantine.OrderByDescending(q => q.MsgDate).Skip(100).Take(100).ToList();
```

  This resulted in the data being unsorted. (Bug #45723)

- The MySQL Connector/NET 6.0.4 installer failed with an error. The error message generated was:

```
There is a problem with this Windows Installer package. A DLL required for this
install to complete could not be run. Contact your support personnel or package vendor.
```

  When **OK** was clicked to acknowledge the error the installer exited. (Bug #45474)

- Calling the Entity Framework `SaveChanges()` method of any MySQL ORM Entity with a column type `TIME`, generated an error message:

```
Unknown PrimitiveKind Time
```

  (Bug #45457)

- Insert into two tables failed when using the Entity Framework. The exception generated was:

```
The value given is not an instance of type 'Edm.Int32'
```

  (Bug #45077)

- Errors occurred when using the Entity Framework with cultures that used a comma as the decimal separator. This was because the formatting for `SINGLE`, `DOUBLE` and `DECIMAL` values was not handled correctly. (Bug #44455)

- When attempting to connect to MySQL using the Compact Framework version of MySQL Connector/NET, an `IndexOutOfRangeException` exception was generated on trying to open the connection. (Bug #43736)

- When reading data, such as with a `MySqlDataAdapter` on a `MySqlConnection`, MySQL Connector/NET could potentially enter an infinite loop in `CompressedStream.ReadNextpacket()` if compression was enabled. (Bug #43678)

- An error occurred when building MySQL Connector/NET from source code checked out from the public SVN repository. This happened on Linux using Mono and Nant. The Mono JIT compiler version was 1.2.6.0. The Nant version was 0.85.

  When an attempt was made to build (for example) the MySQL Connector/NET 5.2 branch using the command:

```
$ nant -buildfile:Client.build
```

  The following error occurred:

```
BUILD FAILED

Error loading buildfile.
Encoding name 'Windows-1252' not supported.
Parameter name: name
```

  (Bug #42411)

- MySQL Connector/NET CHM documentation stated that MySQL Server 3.23 was supported. (Bug #42110)
- In the case of long network inactivity, especially when connection pooling was used, connections were sometimes dropped, for example, by firewalls.

  Note: The bugfix introduced a new `keepalive` parameter, which prevents disconnects by sending an empty TCP packet after a specified timeout. (Bug #40684)
- Calling a Stored Procedure with an output parameter through MySQL Connector/NET resulted in a memory leak. Calling the same Stored Procedure without an output parameter did not result in a memory leak. (Bug #36027)

Changes in MySQL Connector/NET 6.1.0 (2009-07-15, Alpha)

This is the first Alpha release of 6.1.

Functionality Added or Changed

- Changed GUID type - The back-end representation of a guid type has been changed to be CHAR(36). This is so you can use the server UUID() function to populate a GUID table. UUID generates a 36 character string. Developers of older applications can add `old guids=true` to the connection string and the old BINARY(16) type will be used instead.
- Support for native output parameters - This is supported when connected to a server that supports native output parameters. This includes servers as of 5.5.3 and 6.0.8.
- Session State Provider - This enables you to store the state of your website in a MySQL server.
- Website Configuration Dialog - This is a new wizard that is activated by clicking a button on the toolbar at the top of the Visual Studio Solution Explorer. It works in conjunction with the ASP.NET administration pages, making it easier to activate and set advanced options for the different MySQL web providers included.

Changes in MySQL Connector/Net 6.0

Changes in MySQL Connector/NET 6.0.8 (Not released)

Bugs Fixed

- `MySQLConnectionStringBuilder.ContainsKey()` incorrectly returned `false` when testing whether a keyword was part of the connection string. (Bug #11766671, Bug #59835)
- `MysqlDataReader.GetSchemaTable` returned incorrect values and types. (Bug #59989, Bug #11776346)
- All queries other than `INSERT` were executed individually instead of as a batch even though batching was enabled for the connection. (Bug #59616, Bug #11850286)
- MySQL Connector/NET generated an exception when executing a query consisting of ';', for example:

```
mycmd(";",mycon)
mycmd.executenonquery()
```

  The exception generated was:

```
System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at MySql.Data.MySqlClient.MySqlCommand.TrimSemicolons(String sql)
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
   at MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
```

(Bug #59537, Bug #11766433)

- Setting `Membership.ApplicationName` had no effect. (Bug #59438, Bug #11770465)
- `MembershipProvider` did not generate hashes correctly if the algorithm was keyed. The Key of the algorithm should have been set if the `HashAlgorithm` was `KeyedHashAlgorithm`. (Bug #58906)
- Code introduced to fix bug #54863 proved problematic on .NET version 3.5 and above. (Bug #58853)
- The `MySqlTokenizer` contained unnecessary `Substring` and `Trim` calls:

```
string token = sql.Substring(startIndex, stopIndex - startIndex).Trim();
```

  The variable `token` was not used anywhere in the code. (Bug #58757)
- `MySqlCommand.ExecuteReader(CommandBehavior)` threw a `NullReferenceException` when being called with `CommandBehavior.CloseConnection`, if the SQL statement contained a syntax error, or contained invalid data such as an invalid column name. (Bug #58652)
- `ReadFieldLength()` returned incorrect value for `BIGINT` autoincrement columns. (Bug #58373)
- MySQL Connector/NET did not support the `utf8mb4` character set. When attempting to connect to `utf8mb4` tables or columns, an exception `KeyNotFoundException` was generated. (Bug #58244)
- A typed dataset did not get the table name. (Bug #57894, Bug #11764989)
- Setting `MySqlCommand.CommandTimeout` to 0 had no effect. It should have resulted in an infinite timeout. (Bug #57265)
- When performing a row-by-row update, only the first row was updated and all other rows were ignored. (Bug #57092)
- Setting the `Default Command Timeout` connection string option had no effect. (Bug #56806)
- When an output parameter was declared as type `MySqlDbType.Bit`, it failed to return with the correct value. (Bug #56756)
- Default values returned for text columns were not quoted. This meant that the `COLUMN_DEFAULT` field of the `GetSchema` columns collection did not return a valid SQL expression. (Bug #56509)
- MySQL Connector/NET for .NET/Mono attempted to dynamically load the assembly `Mono.Posix.dll` when a Unix socket was used to connect to the server. This failed and the connector was not able to use a Unix socket unless the `Mono.Posix.dll` assembly was previously loaded by the program. (Bug #56410)
- The ADO.NET Entity Data Model could not add stored procedures from MySQL Server 5.0.45 but worked fine using MySQL Server 5.1. (Bug #55349)

Changes in MySQL Connector/NET 6.0.7 (2010-08-30)

Fixes bugs since 6.0.6.

Bugs Fixed

- Attempting to read `Double.MinValue` from a `DOUBLE` column in MySQL table generated the following exception:

```
System.OverflowException : Value was either too large or too small for a Double.

--OverflowException
at System.Number.ParseDouble(String value, NumberStyles options, NumberFormatInfo
numfmt)
at MySql.Data.Types.MySqlDouble.MySql.Data.Types.IMySqlValue.ReadValue(MySqlPacket
```

```
packet, Int64 length, Boolean nullVal)
at MySql.Data.MySqlClient.NativeDriver.ReadColumnValue(Int32 index, MySqlField field,
IMySqlValue valObject)
at MySql.Data.MySqlClient.ResultSet.ReadColumnData(Boolean outputParms)
at MySql.Data.MySqlClient.ResultSet.NextRow(CommandBehavior behavior)
at MySql.Data.MySqlClient.MySqlDataReader.Read()
```

(Bug #55644)

- `MySqlDataAdapter.Update()` generated concurrency violations for custom stored procedure driven update commands that used `UpdateRowSource.FirstReturnedRecord`. (Bug #54895)
- Several calls to `DataAdapter.Update()` with intervening changes to `DataTable` resulted in `ConcurrencyException` exceptions being generated. (Bug #54863)
- The `MySqlHelper` object did not have an overloaded version of the `ExecuteReader` method that accepted a `MySqlConnection` object. (Bug #54570)
- If `MySqlDataAdapter` was used with an `INSERT` command where the `VALUES` clause contained an expression with parentheses in it, and set the `adapter.UpdateBatchSize` parameter to be greater than one, then the call to `adapter.Update` either generated an exception or failed to batch the commands, executing each insert individually. (Bug #54386)
- The method `MySql.Data.Common.QueryNormalizer.CollapseValueList` generated an `ArgumentOutOfRangeException`. (Bug #54152, Bug #53865)
- Garbage Collector disposal of a `MySqlConnection` object caused the following exception:

```
System.IO.EndOfStreamException: Attempted to read past the end of the stream.
MySql.Data.MySqlClient.MySqlStream.ReadFully(Stream stream, Byte[] buffer, Int32 offset,
Int32 count)
MySql.Data.MySqlClient.MySqlStream.LoadPacket()
Outer Exception Reading from the stream has failed.
...
```

(Bug #53457)

- After a timeout exception, if an attempt was made to reuse a connection returned to the connection pool the following exception was generated:

```
[MySqlException (0x80004005): There is already an open DataReader associated with this
Connection which must be closed first.]
   MySql.Data.MySqlClient.MySqlCommand.CheckState() +278
   MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior) +43
   MySql.Data.MySqlClient.MySqlCommand.ExecuteReader() +6
   Controls.SimpleCommand.ExecuteReader(String SQL) in ...:323
   Albums.GetImagesByAlbum(SimpleCommand Cmd, Int32 iAlbum, String Order, String Limit)
in ...:13
   Forecast.Page_Load(Object sender, EventArgs e) in ...:70
   System.Web.UI.Control.OnLoad(EventArgs e) +99
   System.Web.UI.Control.LoadRecursive() +50
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean
includeStagesAfterAsyncPoint) +627
```

(Bug #53357)

- Membership schema creation failed if the default schema collation was not Latin1. (Bug #53174)
- EventLog was not disposed in the SessionState provider. (Bug #52550)
- Stored procedure enumeration code generated an error if a procedure was used in a dataset that did not return any resultsets. (Bug #50671)
- When an application was subjected to increased concurrent load, MySQL Connector/NET generated the following error when calling stored procedures:

```
A DataTable named \'Procedure Parameters\'
already belongs to this DataSet.
```

(Bug #49118)

- In the ADO.NET Entity Data Model Wizard, the time to update a model scaled abnormally as the number of entities increased. (Bug #48791, Bug #12596237)
- The `INSERT` command was significantly slower with MySQL Connector/NET 6.x compared to 5.x, when compression was enabled. (Bug #48243)
- When the connection string option “Connection Reset = True” was used, a connection reset used the previously used encoding for the subsequent authentication operation. This failed, for example, if UCS2 was used to read the last column before the reset. (Bug #47153)
- Opening a connection in the Visual Studio Server Explorer and choosing to alter an existing routine required another authentication at the server. (Bug #44715)
- When batching was used in `MySqlDataAdapter`, a connection was not opened automatically in `MySqlDataAdapter.Update()`. This resulted in an `InvalidOperationException` exception being generated, with the message text “connection must be valid and open”.

  MySQL Connector/NET has been changed to behave more like SQL Server: if the connection is closed, it is opened for the duration of update operation. (Bug #38411)
- Database name was emitted into typed datasets. This prevented users using the configured default database. (Bug #33870)

Changes in MySQL Connector/NET 6.0.6 (2010-04-28)

Fixes bugs since 6.0.5.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Procedure caching had a problem whereby if you created a procedure, dropped it, and recreated it with a different number of parameters an exception was generated.

  MySQL Connector/NET has been changed so that if the procedure is recreated with a different number of parameters, it will still be recognized. (Bug #52562)
- MySQL Connector/NET has been changed to include `MySqlDataReader.GetFieldType(string columnname)`. Further, `MySqlDataReader.GetOrdinal()` now includes the name of the column in the exception if the column is not found. (Bug #47467)

Bugs Fixed

- If using MySQL Server 5.0.x it was not possible to alter stored routines in Visual Studio. If the stored routine was clicked, and the context sensitive menu option, Alter Routine, selected, the following error was generated:

  ```
  Unable to load object with error: Object reference not
  set to an instance of an object
  ```

  (Bug #55170)
- In MySQL Connector/NET, the `MySqlConnection.Abort()` method contained a `try...catch` construct, with an empty `catch` block. This meant that any exception generated at this point would not be caught. (Bug #52769)
- If `FunctionsReturnString=true` was used in the connection string, the decimal separator (according to locale) was not interpreted. (Bug #52187)

- In MySQL Connector/NET, the `LoadCharsetMap()` function of the `CharSetMap` class set the following incorrect mapping:

  ```
  mapping.Add("latin1", new CharacterSet("latin1", 1));
  ```

  This meant that, for example, the Euro sign was not handled correctly.

  The correct mapping should have been:

  ```
  mapping.Add("latin1", new CharacterSet("windows-1252", 1));
  ```

  This is because the MySQL `latin1` character set is the same as the `windows-cp1252` character set and it extends the official ISO 8859-1 or IANA latin1. (Bug #51927)

- A non-terminated string in SQL threw a CLR exception rather than a syntax exception. (Bug #51788)

- When calling `ExecuteNonQuery` on a command object, the following exception occurred:

  ```
  Index and length must refer to a location within the string.
  Parameter name: length
  ```

  (Bug #51610)

- The method `Command.TrimSemicolons` used `StringBuilder`, and therefore allocated memory for the query even if it did not need to be trimmed. This led to excessive memory consumption when executing a number of large queries. (Bug #51149)

- `MySqlCommand.Parameters.Clear()` did not work. (Bug #50444)

- When the `MySqlScript.execute()` method was called, the following exception was generated:

  ```
  InvalidOperationException : The CommandText property has not been properly initialized.
  ```

  (Bug #50344)

- Binary Columns were not displayed in the Query Builder of Visual Studio. (Bug #50171)

- When the `UpdateBatchSize` property was set to a value greater than 1, only the first row was applied to the database. (Bug #50123)

- When using table per type inheritance and listing the contents of the parent table, the result of the query was a list of child objects, even though there was no related child record with the same parent Id. (Bug #49850)

- `MySqlDataReader.GetUInt64` returned an incorrect value when reading a `BIGINT UNSIGNED` column containing a value greater than 2147483647. (Bug #49794)

- A `FormatException` was generated when an empty string was returned from a stored function. (Bug #49642)

- When adding a data set in Visual Studio 2008, the following error was generated:

  ```
  Relations couldn't be added. Column 'REFERENCED_TABLE_CATALOG' does not belong to table.
  ```

  This was due to a 'REFERENCED_TABLE_CATALOG' column not being included in the foreign keys collection. (Bug #48974)

- Attempting to execute a load data local infile on a file where the user did not have write permissions, or the file was open in an editor gave an access denied error. (Bug #48944)

- The method `MySqlDataReader.GetSchemaTable()` returned 0 in the `NumericPrecision` field for decimal and newdecimal columns. (Bug #48171)

- When trying to create stored procedures from an SQL script, a `MySqlException` was thrown when attempting to redefine the `DELIMITER`:

```
MySql.Data.MySqlClient.MySqlException was unhandled
Message="You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'DELIMITER' at line 1"
Source="MySql.Data"
ErrorCode=-2147467259
Number=1064
StackTrace:
à MySql.Data.MySqlClient.MySqlStream.ReadPacket()
à MySql.Data.MySqlClient.NativeDriver.ReadResult(UInt64& affectedRows, Int64&
lastInsertId)
à MySql.Data.MySqlClient.MySqlDataReader.GetResultSet()
à MySql.Data.MySqlClient.MySqlDataReader.NextResult()
à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
à MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
à MySql.Data.MySqlClient.MySqlScript.Execute()
```

Note: The `MySqlScript` class has been fixed to support the delimiter statement as it is found in SQL scripts. (Bug #46429)

- Calling a User Defined Function using Entity SQL in the Entity Framework caused a `NullReferenceException`. (Bug #45277)
- A connection string set in `web.config` could not be reused after Visual Studio 2008 Professional was shut down. It continued working for the existing controls, but did not work for new controls added. (Bug #41629)

Changes in MySQL Connector/NET 6.0.5 (2009-11-12)

This is a new release, fixing recently discovered bugs.

Bugs Fixed

- Cloning of `MySqlCommand` was not typesafe. To clone a `MySqlCommand` it was necessary to do:

```
MySqlCommand clone = (MySqlCommand)((ICloneable)comm).Clone();
```

MySQL Connector/NET was changed so that it was possible to do:

```
MySqlCommand clone = comm.Clone();
```

(Bug #48460)

- If `MySqlConnection.GetSchema` was called for "Indexes" on a table named “b`a`d” as follows:

```
DataTable schemaPrimaryKeys = connection.GetSchema(
  "Indexes",
  new string[] { null, schemaName, "b`a`d"});
```

Then the following exception was generated:

```
You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near 'a`d`' at line 1
```

(Bug #48101)

- It was not possible to retrieve a value from a MySQL server table, if the value was larger than that supported by the .NET type `System.Decimal`.

  MySQL Connector/NET was changed to expose the `MySqlDecimal` type, along with the supporting method `GetMySqlDecimal`. (Bug #48100)

- An entity model created from a schema containing a table with a column of type `UNSIGNED BIGINT` and a view of the table did not behave correctly. When an entity was created and mapped to the view, the column that was of type `UNSIGNED BIGINT` was displayed as `BIGINT`. (Bug #47872)

- When loading the `MySQLClient-mono.sln` file included with the Connector/NET source into Mono Develop, the following error occurred:

```
/home/tbedford/connector-net-src/6.1/MySQLClient-mono.sln(22):
Unsupported or unrecognized project:
'/home/tbedford/connector-net-src/6.1/Installer/Installer.wixproj'
```

  If the file was modified to remove this problem, then attempting to build the solution generated the following error:

```
/home/tbedford/connector-net-src/6.1/MySql.Data/Provider/Source/Connection.cs(280,46):
error CS0115: `MySql.Data.MySqlClient.MySqlConnection.DbProviderFactory' is marked as an
override but no suitable property found to override
```

  (Bug #47048)

- If an error occurred during connection to a MySQL Server, deserializing the error message from the packet buffer caused a `NullReferenceException` to be thrown. When the method `MySqlPacket::ReadString()` attempted to retrieve the error message, the following line of code threw the exception:

```
string s = encoding.GetString(bits, (int)buffer.Position, end - (int)buffer.Position);
```

  This was due to the fact that the encoding field had not been initialized correctly. (Bug #46844)

- In the `MySqlDataReader` class the `GetSByte` function returned a `byte` value instead of an `sbyte` value. (Bug #46620)

- The MySQL Connector/NET Profile Provider, `MySql.Web.Profile.MySQLProfileProvider`, generated an error when running on Mono. When an attempt was made to save a string in `Profile.Name` the string was not saved to the `my_aspnet_Profiles` table. If an attempt was made to force the save with `Profile.Save()` the following error was generated:

```
Server Error in '/mono' Application

--------------------------------------------------------------------------------

The requested feature is not implemented.
Description: HTTP 500. Error processing request.

Stack Trace:

System.NotImplementedException: The requested feature is not implemented.
at MySql.Data.MySqlClient.MySqlConnection.EnlistTransaction
(System.Transactions.Transaction transaction) [0x00000]
at MySql.Data.MySqlClient.MySqlConnection.Open () [0x00000]
at MySql.Web.Profile.MySQLProfileProvider.SetPropertyValues
(System.Configuration.SettingsContext context,
System.Configuration.SettingsPropertyValueCollection collection) [0x00000]



--------------------------------------------------------------------------------
Version information: Mono Version: 2.0.50727.1433; ASP.NET Version: 2.0.50727.1433
```

  (Bug #46375)

- An exception was generated when using `TIMESTAMP` columns with the Entity Framework. (Bug #46311)

- MySQL Connector/NET sometimes hung, without generating an exception. This happened if a read from a stream failed returning a 0, causing the code in `LoadPacket()` to enter an infinite loop. (Bug #46308)

- When using MySQL Connector/NET 6.0.4 and a MySQL Server 4.1 an exception was generated when trying to execute:

```
connection.GetSchema("Columns", ...);
```

The exception generated was:

```
'connection.GetSchema("Columns")' threw an exception of type
'System.ArgumentException'System.Data.DataTable {System.ArgumentException}
base{"Input string was not in a correct format.Couldn't store <'Select'> in
NUMERIC_PRECISION Column.  Expected type is UInt64."}System.Exception
{System.ArgumentException}
```

(Bug #46270)

- The MySQL Connector/NET method `StoredProcedure.GetParameters(string)` ignored the programmer's setting of the `UseProcedureBodies` option. This broke any application for which the application's parameter names did not match the parameter names in the Stored Procedure, resulting in an `ArgumentException` with the message “Parameter 'foo' not found in the collection.” and the following stack trace:

```
MySql.Data.dll!MySql.Data.MySqlClient.MySqlParameterCollection.GetParameterFlexible(stri
ng parameterName = "pStart", bool throwOnNotFound = true) Line 459C#
MySql.Data.dll!MySql.Data.MySqlClient.StoredProcedure.Resolve() Line 157 + 0x25
bytesC#
MySql.Data.dll!MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(System.Data.CommandBeha
vior behavior = SequentialAccess) Line 405 + 0xb bytesC#
MySql.Data.dll!MySql.Data.MySqlClient.MySqlCommand.ExecuteDbDataReader(System.Data.Comma
ndBehavior behavior = SequentialAccess) Line 884 + 0xb bytesC#
System.Data.dll!System.Data.Common.DbCommand.System.Data.IDbCommand.ExecuteReader(System
.Data.CommandBehavior behavior) + 0xb bytes
System.Data.dll!System.Data.Common.DbDataAdapter.FillInternal(System.Data.DataSet
dataset = {System.Data.DataSet}, System.Data.DataTable[] datatables = null, int
startRecord = 0, int maxRecords = 0, string srcTable = "Table", System.Data.IDbCommand
command = {MySql.Data.MySqlClient.MySqlCommand}, System.Data.CommandBehavior behavior) +
0x83 bytes
System.Data.dll!System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet dataSet, int
startRecord, int maxRecords, string srcTable, System.Data.IDbCommand command,
System.Data.CommandBehavior behavior) + 0x120 bytes
System.Data.dll!System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet dataSet) +
0x5f bytes
```

(Bug #46213)

- Conversion of MySQL `TINYINT(1)` to `boolean` failed. (Bug #46205, Bug #46359, Bug #41953)
- When populating a MySQL database table in Visual Studio using the Table Editor, if a `VARCHAR(10)` column was changed to a `VARCHAR(20)` column an exception was generated:

```
SystemArgumentException: DataGridViewComboBoxCell value is not valid.
To replace this default dialog please handle the DataError Event.
```

(Bug #46100)

- In MySQL Connector/NET 6.0.4 using `GetProcData` generated an error because the `parameters` data table was only created if MySQL Server was at least version 6.0.6, or if the `UseProcedureBodies` connection string option was set to true.

  Also the `DeriveParameters` command generated a null reference exception. This was because the `parameters` data table, which was null, was used in a `for each` loop. (Bug #45952)

- The Entity Framework provider was not calling `DBSortExpression` correctly when the `Skip` and `Take` methods were used, such as in the following statement:

```
TestModel.tblquarantine.OrderByDescending(q => q.MsgDate).Skip(100).Take(100).ToList();
```

This resulted in the data being unsorted. (Bug #45723)

- The `EscapeString` code carried out escaping by calling `string.Replace` multiple times. This resulted in a performance bottleneck, as for every line a new string was allocated and another was disposed of by the garbage collector. (Bug #45699)

- Adding the `Allow Batch=False` option to the connection string caused MySQL Connector/NET to generate the error:

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL
server version for the right syntax to use near 'SET character_set_results=NULL' at line 1
```

  (Bug #45502)

- The MySQL Connector/NET 6.0.4 installer failed with an error. The error message generated was:

```
There is a problem with this Windows Installer package. A DLL required for this
install to complete could not be run. Contact your support personnel or package vendor.
```

  When **OK** was clicked to acknowledge the error the installer exited. (Bug #45474)

- A MySQL Connector/NET test program that connected to MySQL Server using the connection string option `compress=true` crashed, but only when running on Mono. The program worked as expected when running on Microsoft Windows.

  This was due to a bug in Mono. MySQL Connector/NET was modified to avoid using `WeakReferences` in the `Compressed` stream class, which was causing the crash. (Bug #45463)

- Calling the Entity Framework `SaveChanges()` method of any MySQL ORM Entity with a column type `TIME`, generated an error message:

```
Unknown PrimitiveKind Time
```

  (Bug #45457)

- Insert into two tables failed when using the Entity Framework. The exception generated was:

```
The value given is not an instance of type 'Edm.Int32'
```

  (Bug #45077)

- Input parameters were missing from Stored Procedures when using them with ADO.NET Data Entities. (Bug #44985)

- Errors occurred when using the Entity Framework with cultures that used a comma as the decimal separator. This was because the formatting for `SINGLE`, `DOUBLE` and `DECIMAL` values was not handled correctly. (Bug #44455)

- When attempting to connect to MySQL using the Compact Framework version of MySQL Connector/NET, an `IndexOutOfRangeException` exception was generated on trying to open the connection. (Bug #43736)

- When reading data, such as with a `MySqlDataAdapter` on a `MySqlConnection`, MySQL Connector/NET could potentially enter an infinite loop in `CompressedStream.ReadNextpacket()` if compression was enabled. (Bug #43678)

- An error occurred when building MySQL Connector/NET from source code checked out from the public SVN repository. This happened on Linux using Mono and Nant. The Mono JIT compiler version was 1.2.6.0. The Nant version was 0.85.

  When an attempt was made to build (for example) the MySQL Connector/NET 5.2 branch using the command:

```
$ nant -buildfile:Client.build
```

  The following error occurred:

```
BUILD FAILED

Error loading buildfile.
```

```
Encoding name 'Windows-1252' not supported.
Parameter name: name
```

(Bug #42411)

- After a Reference to "C:\Program Files\MySQL\MySQL Connector Net 5.2.4\Compact Framework \MySql.Data.CF.dll" was added to a Windows Mobile 5.0 project, the project then failed to build, generating a Microsoft Visual C# compiler error.

  The error generated was:

```
Error 2 The type 'System.Runtime.CompilerServices.CompilerGeneratedAttribute'
has no constructors defined MysqlTest
Error 3 Internal Compiler Error (0xc0000005 at address 5A7E3714):
likely culprit is 'COMPILE'.
```

  (Bug #42261)

- MySQL Connector/NET CHM documentation stated that MySQL Server 3.23 was supported. (Bug #42110)

- In the case of long network inactivity, especially when connection pooling was used, connections were sometimes dropped, for example, by firewalls.

  Note: The bugfix introduced a new `keepalive` parameter, which prevents disconnects by sending an empty TCP packet after a specified timeout. (Bug #40684)

- MySQL Connector/NET generated the following exception:

```
System.NullReferenceException: Object reference not set to an instance of an object.
      bei MySql.Data.MySqlClient.MySqlCommand.TimeoutExpired(Object commandObject)
      bei System.Threading._TimerCallback.TimerCallback_Context(Object state)
      bei System.Threading.ExecutionContext.runTryCode(Object userData)
      bei
      System.Runtime.CompilerServices.RuntimeHelpers.ExecuteCodeWithGuaranteedCleanup(TryCode
      code, CleanupCode backoutCode, Object userData)
      bei System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext,
      ContextCallback callback, Object state)
      bei System.Threading.ExecutionContext.Run(ExecutionContext executionContext,
      ContextCallback callback, Object state)
      bei System.Threading._TimerCallback.PerformTimerCallback(Object state)
```

  (Bug #40005)

- Calling a Stored Procedure with an output parameter through MySQL Connector/NET resulted in a memory leak. Calling the same Stored Procedure without an output parameter did not result in a memory leak. (Bug #36027)

- Using a `DataAdapter` with a linked `MySqlCommandBuilder` the following exception was thrown when trying to call `da.Update(DataRow[] rows)`:

```
Connection must be valid and open
```

  (Bug #34657)

Changes in MySQL Connector/NET 6.0.4 (2009-06-16)

This is the first post-GA release, fixing recently discovered bugs.

Bugs Fixed

- If a certain socket exception occurred when trying to establish a MySQL database connection, MySQL Connector/NET displayed an exception message that appeared to be unrelated to the underlying problem. This masked the problem and made diagnosing problems more difficult.

For example, if, when establishing a database connection using TCP/IP, Windows on the local machine allocated an ephemeral port that conflicted with a socket address still in use, then Windows/.NET would throw a socket exception with the following error text:

```
Only one usage of each socket address (protocol/network address/port) is normally
permitted IP ADDRESS/PORT.
```

However, MySQL Connector/NET masked this socket exception and displayed an exception with the following text:

```
Unable to connect to any of the specified MySQL hosts.
```

(Bug #45021)

- An SQL query string containing an escaped backslash caused an exception to be generated:

```
Index and length must refer to a location within the string.
Parameter name: length
at System.String.InternalSubStringWithChecks(Int32 startIndex, Int32 length, Boolean
fAlwaysCopy)
at MySql.Data.MySqlClient.MySqlTokenizer.NextParameter()
at MySql.Data.MySqlClient.Statement.InternalBindParameters(String sql,
MySqlParameterCollection parameters, MySqlPacket packet)
at MySql.Data.MySqlClient.Statement.BindParameters()
at MySql.Data.MySqlClient.PreparableStatement.Execute()
at MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
at MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
```

(Bug #44960)

- The Microsoft Visual Studio solution file `MySQL-VS2005.sln` was invalid. Several projects could not be loaded and thus it was not possible to build MySQL Connector/NET from source. (Bug #44822)
- The Data Set editor generated an error when attempts were made to modify insert, update or delete commands:

```
Error in WHERE clause near '@'.
Unable to parse query text.
```

(Bug #44512)

- The DataReader in MySQL Connector/NET 6.0.3 considered a BINARY(16) field as a GUID with a length of 16. (Bug #44507)
- When creating a new DataSet the following error was generated:

```
Failed to open a connection to database.
Cannot load type with name 'MySQL.Data.VisualStudio.StoredProcedureColumnEnumerator'
```

(Bug #44460)

- The MySQL Connector/NET MySQLRoleProvider reported that there were no roles, even when roles existed. (Bug #44414)
- MySQL Connector/NET was missing the capability to validate the server's certificate when using encryption. This made it possible to conduct a man-in-the-middle attack against the connection, which defeated the security provided by SSL. (Bug #38700)

Changes in MySQL Connector/NET 6.0.3 (2009-04-28)

First GA release.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- The `MySqlTokenizer` failed to split fieldnames from values if they were not separated by a space. This also happened if the string contained certain characters. As a result `MySqlCommand.ExecuteNonQuery` raised an index out of range exception.

  The resulting errors are illustrated by the following examples. Note, the example statements do not have delimiting spaces around the `=` operator.

  ```
  INSERT INTO anytable SET Text='test--test';
  ```

  The tokenizer incorrectly interpreted the value as containing a comment.

  ```
  UPDATE anytable SET Project='123-456',Text='Can you explain this ?',Duration=15 WHERE
        ID=4711;'
  ```

  A `MySqlException` was generated, as the `?` in the value was interpreted by the tokenizer as a parameter sign. The error message generated was:

  ```
  Fatal error encountered during command execution.
  EXCEPTION: MySqlException - Parameter '?'' must be defined.
  ```

  (Bug #44318)

Bugs Fixed

- `MySQL.Data` was not displayed as a Reference inside Microsoft Visual Studio 2008 Professional.

  When a new C# project was created in Microsoft Visual Studio 2008 Professional, `MySQL.Data` was not displayed when **References**, **Add Reference** was selected. (Bug #44141)

- Column types for `SchemaProvider` and `ISSchemaProvider` did not match.

  When the source code in `SchemaProvider.cs` and `ISSchemaProvider.cs` were compared it was apparent that they were not using the same column types. The base provider used SQL such as `SHOW CREATE TABLE`, while `ISSchemaProvider` used the schema information tables. Column types used by the base class were `INT64` and the column types used by `ISSchemaProvider` were `UNSIGNED`. (Bug #44123)

Changes in MySQL Connector/NET 6.0.2 (2009-04-07, Beta)

This is a new development release, fixing recently discovered bugs.

Bugs Fixed

- MySQL Connector/NET 6.0.1 did not load in Microsoft Visual Studio 2008 and Visual Studio 2005 Pro.

  The following error message was generated:

  ```
  .NET Framework Data Provider for MySQL: The data provider object factory service was not
  found.
  ```

  (Bug #44064)

Changes in MySQL Connector/NET 6.0.1 (2009-04-02, Beta)

This is a new Beta development release, fixing recently discovered bugs.

Bugs Fixed

- An insert and update error was generated by the decimal data type in the Entity Framework, when a German collation was used. (Bug #43574)

- Generating an Entity Data Model (EDM) schema with a table containing columns with data types `MEDIUMTEXT` and `LONGTEXT` generated a runtime error message “Max value too long or too short for Int32”. (Bug #43480)

Changes in MySQL Connector/NET 6.0.0 (2009-03-02, Alpha)

This is a new Alpha development release.

Bugs Fixed

- A null reference exception was generated when `MySqlConnection.ClearPool(connection)` was called. (Bug #42801)
- The Web Provider did not work at all on a remote host, and did not create a database when using `autogenerateschema="true"`. (Bug #39072)
- The MySQL Connector/NET installer program ended prematurely without reporting the specific error. (Bug #39019)
- When called with an incorrect password the `MembershipProvider.GetPassword()` method threw a `MySQLException` instead of a `MembershipPasswordException` . (Bug #38939)
- Possible overflow in `MySqlPacket.ReadLong()`. (Bug #36997)
- The `TokenizeSql` method was adding query overhead and causing high CPU utilization for larger queries. (Bug #36836)

Changes in MySQL Connector/Net 5.2

Changes in MySQL Connector/NET 5.2.8 (Not released)

Bugs Fixed

- If `MySqlConnection.GetSchema` was called for "Indexes" on a table named “b`a`d” as follows:

```
DataTable schemaPrimaryKeys = connection.GetSchema(
  "Indexes",
  new string[] { null, schemaName, "b`a`d"});
```

  Then the following exception was generated:

```
You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near 'a`d`' at line 1
```

  (Bug #48101)

- When the connection string option “Connection Reset = True” was used, a connection reset used the previously used encoding for the subsequent authentication operation. This failed, for example, if UCS2 was used to read the last column before the reset. (Bug #47153)
- In the `MySqlDataReader` class the `GetSByte` function returned a `byte` value instead of an `sbyte` value. (Bug #46620)
- When trying to create stored procedures from an SQL script, a `MySqlException` was thrown when attempting to redefine the `DELIMITER`:

```
MySql.Data.MySqlClient.MySqlException was unhandled
Message="You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'DELIMITER' at line 1"
Source="MySql.Data"
ErrorCode=-2147467259
Number=1064
StackTrace:
à MySql.Data.MySqlClient.MySqlStream.ReadPacket()
à MySql.Data.MySqlClient.NativeDriver.ReadResult(UInt64& affectedRows, Int64&
```

```
lastInsertId)
à MySql.Data.MySqlClient.MySqlDataReader.GetResultSet()
à MySql.Data.MySqlClient.MySqlDataReader.NextResult()
à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
à MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
à MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
à MySql.Data.MySqlClient.MySqlScript.Execute()
```

Note: The `MySqlScript` class has been fixed to support the delimiter statement as it is found in SQL scripts. (Bug #46429)

- The MySQL Connector/NET Profile Provider, `MySql.Web.Profile.MySQLProfileProvider`, generated an error when running on Mono. When an attempt was made to save a string in `Profile.Name` the string was not saved to the `my_aspnet_Profiles` table. If an attempt was made to force the save with `Profile.Save()` the following error was generated:

```
Server Error in '/mono' Application

--------------------------------------------------------------------------------

The requested feature is not implemented.
Description: HTTP 500. Error processing request.

Stack Trace:

System.NotImplementedException: The requested feature is not implemented.
at MySql.Data.MySqlClient.MySqlConnection.EnlistTransaction
(System.Transactions.Transaction transaction) [0x00000]
at MySql.Data.MySqlClient.MySqlConnection.Open () [0x00000]
at MySql.Web.Profile.MySQLProfileProvider.SetPropertyValues
(System.Configuration.SettingsContext context,
System.Configuration.SettingsPropertyValueCollection collection) [0x00000]


--------------------------------------------------------------------------------
Version information: Mono Version: 2.0.50727.1433; ASP.NET Version: 2.0.50727.1433
```

(Bug #46375)

- When using MySQL Connector/NET 6.0.4 and a MySQL Server 4.1 an exception was generated when trying to execute:

```
connection.GetSchema("Columns", ...);
```

The exception generated was:

```
'connection.GetSchema("Columns")' threw an exception of type
'System.ArgumentException'System.Data.DataTable {System.ArgumentException}
base{"Input string was not in a correct format.Couldn't store <'Select'> in
NUMERIC_PRECISION Column.  Expected type is UInt64."}System.Exception
{System.ArgumentException}
```

(Bug #46270)

- The MySQL Connector/NET method `StoredProcedure.GetParameters(string)` ignored the programmer's setting of the `UseProcedureBodies` option. This broke any application for which the application's parameter names did not match the parameter names in the Stored Procedure, resulting in an `ArgumentException` with the message “Parameter 'foo' not found in the collection.” and the following stack trace:

```
MySql.Data.dll!MySql.Data.MySqlClient.MySqlParameterCollection.GetParameterFlexible(stri
ng parameterName = "pStart", bool throwOnNotFound = true) Line 459C#
MySql.Data.dll!MySql.Data.MySqlClient.StoredProcedure.Resolve() Line 157 + 0x25
bytesC#
MySql.Data.dll!MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(System.Data.CommandBeha
vior behavior = SequentialAccess) Line 405 + 0xb bytesC#
MySql.Data.dll!MySql.Data.MySqlClient.MySqlCommand.ExecuteDbDataReader(System.Data.Comma
ndBehavior behavior = SequentialAccess) Line 884 + 0xb bytesC#
```

```
System.Data.dll!System.Data.Common.DbCommand.System.Data.IDbCommand.ExecuteReader(System
.Data.CommandBehavior behavior) + 0xb bytes
System.Data.dll!System.Data.Common.DbDataAdapter.FillInternal(System.Data.DataSet
dataset = {System.Data.DataSet}, System.Data.DataTable[] datatables = null, int
startRecord = 0, int maxRecords = 0, string srcTable = "Table", System.Data.IDbCommand
command = {MySql.Data.MySqlClient.MySqlCommand}, System.Data.CommandBehavior behavior) +
0x83 bytes
System.Data.dll!System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet dataSet, int
startRecord, int maxRecords, string srcTable, System.Data.IDbCommand command,
System.Data.CommandBehavior behavior) + 0x120 bytes
System.Data.dll!System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet dataSet) +
0x5f bytes
```

(Bug #46213)

- Conversion of MySQL `TINYINT(1)` to `boolean` failed. (Bug #46205, Bug #46359, Bug #41953)
- If the application slept for longer than the specified `net_write_timeout`, and then resumed `Read` operations on a connection, then the application failed silently. (Bug #45978)
- When reading data, such as with a `MySqlDataAdapter` on a `MySqlConnection`, MySQL Connector/NET could potentially enter an infinite loop in `CompressedStream.ReadNextpacket()` if compression was enabled. (Bug #43678)
- An error occurred when building MySQL Connector/NET from source code checked out from the public SVN repository. This happened on Linux using Mono and Nant. The Mono JIT compiler version was 1.2.6.0. The Nant version was 0.85.

  When an attempt was made to build (for example) the MySQL Connector/NET 5.2 branch using the command:

  ```
  $ nant -buildfile:Client.build
  ```

  The following error occurred:

  ```
  BUILD FAILED

  Error loading buildfile.
  Encoding name 'Windows-1252' not supported.
  Parameter name: name
  ```

  (Bug #42411)
- MySQL Connector/NET CHM documentation stated that MySQL Server 3.23 was supported. (Bug #42110)
- Using a `DataAdapter` with a linked `MySqlCommandBuilder` the following exception was thrown when trying to call `da.Update(DataRow[] rows)`:

  ```
  Connection must be valid and open
  ```

  (Bug #34657)

Changes in MySQL Connector/NET 5.2.7 (2009-07-15)

Bugs Fixed

- The `EscapeString` code carried out escaping by calling `string.Replace` multiple times. This resulted in a performance bottleneck, as for every line a new string was allocated and another was disposed of by the garbage collector. (Bug #45699)
- A MySQL Connector/NET test program that connected to MySQL Server using the connection string option `compress=true` crashed, but only when running on Mono. The program worked as expected when running on Microsoft Windows.

This was due to a bug in Mono. MySQL Connector/NET was modified to avoid using `WeakReferences` in the `Compressed` stream class, which was causing the crash. (Bug #45463)

- If a certain socket exception occurred when trying to establish a MySQL database connection, MySQL Connector/NET displayed an exception message that appeared to be unrelated to the underlying problem. This masked the problem and made diagnosing problems more difficult.

  For example, if, when establishing a database connection using TCP/IP, Windows on the local machine allocated an ephemeral port that conflicted with a socket address still in use, then Windows/.NET would throw a socket exception with the following error text:

  ```
  Only one usage of each socket address (protocol/network address/port) is normally
  permitted IP ADDRESS/PORT.
  ```

  However, MySQL Connector/NET masked this socket exception and displayed an exception with the following text:

  ```
  Unable to connect to any of the specified MySQL hosts.
  ```

  (Bug #45021)

- The Microsoft Visual Studio solution file `MySQL-VS2005.sln` was invalid. Several projects could not be loaded and thus it was not possible to build MySQL Connector/NET from source. (Bug #44822)

- The MySQL Connector/NET MySQLRoleProvider reported that there were no roles, even when roles existed. (Bug #44414)

- After a Reference to "C:\Program Files\MySQL\MySQL Connector Net 5.2.4\Compact Framework \MySql.Data.CF.dll" was added to a Windows Mobile 5.0 project, the project then failed to build, generating a Microsoft Visual C# compiler error.

  The error generated was:

  ```
  Error 2 The type 'System.Runtime.CompilerServices.CompilerGeneratedAttribute'
  has no constructors defined MysqlTest
  Error 3 Internal Compiler Error (0xc0000005 at address 5A7E3714):
  likely culprit is 'COMPILE'.
  ```

  (Bug #42261)

- MySQL Connector/NET generated the following exception:

  ```
  System.NullReferenceException: Object reference not set to an instance of an object.
        bei MySql.Data.MySqlClient.MySqlCommand.TimeoutExpired(Object commandObject)
        bei System.Threading._TimerCallback.TimerCallback_Context(Object state)
        bei System.Threading.ExecutionContext.runTryCode(Object userData)
        bei
        System.Runtime.CompilerServices.RuntimeHelpers.ExecuteCodeWithGuaranteedCleanup(TryCode
        code, CleanupCode backoutCode, Object userData)
        bei System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext,
        ContextCallback callback, Object state)
        bei System.Threading.ExecutionContext.Run(ExecutionContext executionContext,
        ContextCallback callback, Object state)
        bei System.Threading._TimerCallback.PerformTimerCallback(Object state)
  ```

  (Bug #40005)

- When a TableAdapter was created on a DataSet, it was not possible to use a stored procedure with variables. The following error was generated:

  ```
  The method or operation is not implemented
  ```

  (Bug #39409)

Changes in MySQL Connector/NET 5.2.6 (2009-04-28)

Functionality Added or Changed

- A new connection string option has been added: `use affected rows`. When `true` the connection will report changed rows instead of found rows. (Bug #44194)

Bugs Fixed

- Calling `GetSchema()` on `Indexes` or `IndexColumns` failed where index or column names were restricted.

  In `SchemaProvider.cs`, methods `GetIndexes()` and `GetIndexColumns()` passed their restrictions directly to `GetTables()`. This only worked if the restrictions were no more specific than `schemaName` and `tableName`. If `IndexName` was given, this was passed to `GetTables()` where it was treated as `TableType`. As a result no tables were returned, unless the index name happened to be `BASE TABLE` or `VIEW`. This meant that both methods failed to return any rows. (Bug #43991)

- `GetSchema("MetaDataCollections")` should have returned a table with a column named “NumberOfRestrictions” not “NumberOfRestriction”.

  This can be confirmed by referencing the Microsoft Documentation. (Bug #43990)

- Requests sent to the MySQL Connector/NET role provider to remove a user from a role failed. The query log showed the query was correctly executed within a transaction which was immediately rolled back. The rollback was caused by a missing call to the `Complete` method of the transaction. (Bug #43553)

- When using `MySqlBulkLoader.Load()`, the text file is opened by `NativeDriver.SendFileToServer`. If it encountered a problem opening the file as a stream, an exception was generated and caught. An attempt to clean up resources was then made in the `finally{}` clause by calling `fs.Close()`, but since the stream was never successfully opened, this was an attempt to execute a method of a null reference. (Bug #43332)

- A null reference exception was generated when `MySqlConnection.ClearPool(connection)` was called. (Bug #42801)

- `MySQLMembershipProvider.ValidateUser` only used the `userId` to validate. However, it should also use the `applicationId` to perform the validation correctly.

  The generated query was, for example:

  ```
  SELECT Password, PasswordKey, PasswordFormat, IsApproved, Islockedout
        FROM my_aspnet_Membership WHERE userId=13
  ```

  Note that `applicationId` is not used. (Bug #42574)

- There was an error in the `ProfileProvider` class in the `private ProfileInfoCollection GetProfiles()` function. The column of the final table was named “lastUpdatdDate” ('e' is missing) instead of the correct “lastUpdatedDate”. (Bug #41654)

- The `GetGuid()` method of `MySqlDataReader` did not treat `BINARY(16)` column data as a GUID. When operating on such a column a `FormatException` exception was generated. (Bug #41452)

- When ASP.NET membership was configured to not require password question and answer using `requiresQuestionAndAnswer="false"`, a `SqlNullValueException` was generated when using `MembershipUser.ResetPassword()` to reset the user password. (Bug #41408)

- If a `Stored Procedure` contained spaces in its parameter list, and was then called from MySQL Connector/NET, an exception was generated. However, the same `Stored Procedure` called from the MySQL Query Analyzer or the MySQL Client worked correctly.

  The exception generated was:

  ```
  Parameter '0' not found in the collection.
  ```

  (Bug #41034)

- The `DATETIME` format contained an erroneous space. (Bug #41021)

- When `MySql.Web.Profile.MySQLProfileProvider` was configured, it was not possible to assign a name other than the default name `MySQLProfileProvider`.

  If the name `SCC_MySQLProfileProvider` was assigned, an exception was generated when attempting to use `Page.Context.Profile['custom prop']`.

  The exception generated was:

  ```
  The profile default provider was not found.
  ```

  Note that the exception stated: 'the profile **default provider**...', even though a different name was explicitly requested. (Bug #40871)

- When `ExecuteNonQuery` was called with a command type of `Stored Procedure` it worked for one user but resulted in a hang for another user with the same database permissions.

  However, if `CALL` was used in the command text and `ExecuteNonQuery` was used with a command type of `Text`, the call worked for both users. (Bug #40139)

Changes in MySQL Connector/NET 5.2.5 (2008-11-19)

Bugs Fixed

- Visual Studio 2008 displayed the following error three times on start-up:

  ```
  "Package Load Failure

  Package 'MySql.Data.VisualStudio.MySqlDataProviderPackage, MySql.VisualStudio,
  Version=5.2.4, Culture=neutral, PublicKeyToken=null' has failed to load properly (GUID =
  {79A115C9-B133-4891-9E7B-242509DAD272}).  Please contact the package vendor for
  assistance.  Application restart is recommended, due to possible environment corruption.
  Would you like to disable loading the package in the future?  You may use
  'devenve/resetskippkgs' to re-enable package loading."
  ```

  (Bug #40726)

Changes in MySQL Connector/NET 5.2.4 (2008-11-13)

Bugs Fixed

- `MySqlDataReader` did not feature a `GetSByte` method. (Bug #40571)

- When working with stored procedures MySQL Connector/NET generated an exception `Unknown "table parameters" in information_schema`. (Bug #40382)

- `GetDefaultCollation` and `GetMaxLength` were not thread safe. These functions called the database to get a set of parameters and cached them in two static dictionaries in the function `InitCollections`. However, if many threads called them they would try to insert the same keys in the collections resulting in duplicate key exceptions. (Bug #40231)

- If connection pooling was not set explicitly in the connection string, MySQL Connector/NET added ";Pooling=False" to the end of the connection string when `MySqlCommand.ExecuteReader()` was called.

  If connection pooling was explicitly set in the connection string, when `MySqlConnection.Open()` was called it converted "Pooling=True" to "pooling=True".

  If `MySqlCommand.ExecuteReader()` was subsequently called, it concatenated ";Pooling=False" to the end of the connection string. The resulting connection string was thus terminated with "pooling=True;Pooling=False". This disabled connection pooling completely. (Bug #40091)

- The connection string option `Functions Return String` did not set the correct encoding for the result string. Even though the connection string option `Functions Return String=true;` is set, the result of `SELECT DES_DECRYPT()` contained "??" instead of the correct national character symbols. (Bug #40076)

- If, when using the `MySqlTransaction` transaction object, an exception was thrown, the transaction object was not disposed of and the transaction was not rolled back. (Bug #39817)

- After the `ConnectionString` property was initialized using the public setter of `DbConnectionStringBuilder`, the `GetConnectionString` method of `MySqlConnectionStringBuilder` incorrectly returned `null` when `true` was assigned to the `includePass` parameter. (Bug #39728)

- When using `ProfileProvider`, attempting to update a previously saved property failed. (Bug #39330)

- Reading a negative time value greater than -01:00:00 returned the absolute value of the original time value. (Bug #39294)

- Inserting a negative time value (negative `TimeSpan`) into a `Time` column through the use of `MySqlParameter` caused `MySqlException` to be thrown. (Bug #39275)

- When a data connection was created in the server explorer of Visual Studio 2008 Team, an error was generated when trying to expand stored procedures that had parameters.

  Also, if **TableAdapter** was right-clicked and then **Add**, **Query**, **Use Existing Stored Procedures** selected, if you then attempted to select a stored procedure, the window would close and no error message would be displayed. (Bug #39252)

- The Web Provider did not work at all on a remote host, and did not create a database when using `autogenerateschema="true"`. (Bug #39072)

- MySQL Connector/NET called hashed password methods not supported in Mono 2.0 Preview 2. (Bug #38895)

Changes in MySQL Connector/NET 5.2.3 (2008-08-19)

Functionality Added or Changed

- Error string was returned after a 28000 second `wait_timeout`. This has been changed to generate a `ConnectionState.Closed` event. (Bug #38119)

- Changed how the procedure schema collection is retrieved. If `use procedure bodies=true` then the `mysql.proc` table is selected directly as this is up to 50 times faster than the current `information_schema` implementation. If `use procedure bodies=false`, then the `information_schema` collection is queried. (Bug #36694)

- String escaping functionality has been moved from the `MySqlString` class to the `MySqlHelper` class, where it can be accessed by the `EscapeString` method. (Bug #36205)

Bugs Fixed

- The `GetOrdinal()` method failed to return the ordinal if the column name string contained an accent. (Bug #38721)
- MySQL Connector/NET uninstaller did not clean up all installed files. (Bug #38534)
- There was a short circuit evaluation error in the `MySqlCommand.CheckState()` method. When the statement `connection == null` was true a `NullReferenceException` was thrown and not the expected `InvalidOperationException`. (Bug #38276)
- The provider did not silently create the user if the user did not exist. (Bug #38243)
- Executing a command that resulted in a fatal exception did not close the connection. (Bug #37991)
- When a prepared insert query is run that contains an `UNSIGNED TINYINT` in the parameter list, the complete query and data that should be inserted is corrupted and no error is thrown. (Bug #37968)
- In a .NET application MySQL Connector/NET modifies the connection string so that it contains several occurrences of the same option with different values. This is illustrated by the example that follows.

  The original connection string:

  ```
  host=localhost;database=test;uid=*****;pwd=*****;
  connect timeout=25; auto enlist=false;pooling=false;
  ```

  The connection string after closing `MySqlDataReader`:

  ```
  host=localhost;database=test;uid=*****;pwd=*****;
  connect timeout=25;auto enlist=false;pooling=false;
  Allow User Variables=True;Allow User Variables=False;
  Allow User Variables=True;Allow User Variables=False;
  ```

  (Bug #37955)
- Unnecessary network traffic was generated for the normal case where the web provider schema was up to date. (Bug #37469)
- `MySqlReader.GetOrdinal()` performance enhancements break existing functionality. (Bug #37239)
- The `autogenerateschema` option produced tables with incorrect collations. (Bug #36444)
- `GetSchema` did not work correctly when querying for a collection, if using a non-English locale. (Bug #35459)
- When reading back a stored double or single value using the .NET provider, the value had less precision than the one stored. (Bug #33322)
- Using the MySQL Visual Studio plugin and a MySQL 4.1 server, certain field types (`ENUM`) would not be identified correctly. Also, when looking for tables, the plugin would list all tables matching a wildcard pattern of the database name supplied in the connection string, instead of only tables within the specified database. (Bug #30603)

Changes in MySQL Connector/NET 5.2.2 (2008-05-12)

Bugs Fixed

- Product documentation incorrectly stated '?' is the preferred parameter marker. (Bug #37349)

- An incorrect value for a bit field would returned in a multi-row query if a preceding value for the field returned `NULL`. (Bug #36313)
- Tables with `GEOMETRY` field types would return an unknown data type exception. (Bug #36081)
- When using the `MySQLProfileProvider`, setting profile details and then reading back saved data would result in the default values being returned instead of the updated values. (Bug #36000)
- When creating a connection, setting the `ConnectionString` property of `MySqlConnection` to `NULL` would throw an exception. (Bug #35619)
- The `DbCommandBuilder.QuoteIdentifer` method was not implemented. (Bug #35492)
- When using encrypted passwords, the `GetPassword()` function would return the wrong string. (Bug #35336)
- An error would be raised when calling `GetPassword()` with a `NULL` value. (Bug #35332)
- When retrieving data where a field has been identified as containing a GUID value, the incorrect value would be returned when a previous row contained a `NULL` value for that field. (Bug #35041)
- Using the `TableAdapter Wizard` failed when generating commands that used stored procedures due to the change in supported parameter characters. (Bug #34941)
- When creating a new stored procedures, the new parameter code which permits the use of the `@` symbol would interfere with the specification of a `DEFINER`. (Bug #34940)
- When using `SqlDataSource` to open a connection, the connection would not automatically be closed when access had completed. (Bug #34460)
- There was a high level of contention in the connection pooling code that could lead to delays when opening connections and submitting queries. The connection pooling code has been modified to try and limit the effects of the contention issue. (Bug #34001)
- Using the `TableAdapter` wizard in combination with a suitable `SELECT` statement, only the associated `INSERT` statement would also be created, rather than the required `DELETE` and `UPDATE` statements. (Bug #31338)
- Fixed problem in datagrid code related to creating a new table. This problem may have been introduced with .NET 2.0 SP1.
- Fixed profile provider that would throw an exception if you were updating a profile that already existed.

Changes in MySQL Connector/NET 5.2.1 (2008-02-27)

Bugs Fixed

- When using the provider to generate or update users and passwords, the password checking algorithm would not validate the password strength or requirements correctly. (Bug #34792)
- When executing statements that used stored procedures and functions, the new parameter code could fail to identify the correct parameter format. (Bug #34699)
- The installer failed to the DDEX provider binary if the Visual Studio 2005 component was not selected. The result would lead to MySQL Connector/NET not loading properly when using the interface to a MySQL server within Visual Studio. (Bug #34674)
- A number of issues were identified in the case, connection and schema areas of the code for `MembershipProvider`, `RoleProvider`, `ProfileProvider`. (Bug #34495)
- When using web providers, the MySQL Connector/NET would check the schema and cache the application id, even when the connection string had been set. The effect would be to break the membership provider list. (Bug #34451)

- Attempting to use an isolation level other than the default with a transaction scope would use the default isolation level. (Bug #34448)
- When altering a stored procedure within Visual Studio, the parameters to the procedure could be lost. (Bug #34359)
- A race condition could occur within the procedure cache resulting the cache contents overflowing beyond the configured cache size. (Bug #34338)
- Fixed problem with Visual Studio 2008 integration that caused pop-up menus on server explorer nodes to not function
- The provider code has been updated to fix a number of outstanding issues.

Changes in MySQL Connector/NET 5.2.0 (2008-02-11)

Functionality Added or Changed

- Performing `GetValue()` on a field `TINYINT(1)` returned a `BOOLEAN`. While not a bug, this caused problems in software that expected an `INT` to be returned. A new connection string option `Treat Tiny As Boolean` has been added with a default value of `true`. If set to `false` the provider will treat `TINYINT(1)` as `INT`. (Bug #34052)
- Added support for `DbDataAdapter UpdateBatchSize`. Batching is fully supported including collapsing inserts down into the multi-value form if possible.
- DDEX provider now works under Visual Studio 2008 beta 2.
- Added ClearPool and ClearAllPools features.

Bugs Fixed

- Some speed improvements have been implemented in the `TokenizeSql` process used to identify elements of SQL statements. (Bug #34220)
- When accessing tables from different databases within the same `TransactionScope`, the same user/password combination would be used for each database connection. MySQL Connector/NET does not handle multiple connections within the same transaction scope. An error is now returned if you attempt this process, instead of using the incorrect authorization information. (Bug #34204)
- The status of connections reported through the state change handler was not being updated correctly. (Bug #34082)
- Incorporated some connection string cache optimizations sent to us by Maxim Mass. (Bug #34000)
- In an open connection where the server had disconnected unexpectedly, the status information of the connection would not be updated properly. (Bug #33909)
- Data cached from the connection string could return invalid information because the internal routines were not using case-sensitive semantics. This lead to updated connection string options not being recognized if they were of a different case than the existing cached values. (Bug #31433)
- Column name metadata was not using the character set as defined within the connection string being used. (Bug #31185)
- Memory usage could increase and decrease significantly when updating or inserting a large number of rows. (Bug #31090)
- Commands executed from within the state change handler would fail with a `NULL` exception. (Bug #30964)

- When running a stored procedure multiple times on the same connection, the memory usage could increase indefinitely. (Bug #30116)
- Using compression in the MySQL connection with MySQL Connector/NET would be slower than using native (uncompressed) communication. (Bug #27865)
- The `MySqlDbType.Datetime` has been replaced with `MySqlDbType.DateTime`. The old format has been obsoleted. (Bug #26344)

Changes in MySQL Connector/Net 5.1

Changes in MySQL Connector/NET 5.1.8 (Not released)

Bugs Fixed

- Calling `GetSchema()` on `Indexes` or `IndexColumns` failed where index or column names were restricted.

  In `SchemaProvider.cs`, methods `GetIndexes()` and `GetIndexColumns()` passed their restrictions directly to `GetTables()`. This only worked if the restrictions were no more specific than `schemaName` and `tableName`. If `IndexName` was given, this was passed to `GetTables()` where it was treated as `TableType`. As a result no tables were returned, unless the index name happened to be `BASE TABLE` or `VIEW`. This meant that both methods failed to return any rows. (Bug #43991)

- The `DATETIME` format contained an erroneous space. (Bug #41021)
- If connection pooling was not set explicitly in the connection string, MySQL Connector/NET added ";Pooling=False" to the end of the connection string when `MySqlCommand.ExecuteReader()` was called.

  If connection pooling was explicitly set in the connection string, when `MySqlConnection.Open()` was called it converted "Pooling=True" to "pooling=True".

  If `MySqlCommand.ExecuteReader()` was subsequently called, it concatenated ";Pooling=False" to the end of the connection string. The resulting connection string was thus terminated with "pooling=True;Pooling=False". This disabled connection pooling completely. (Bug #40091)

- MySQL Connector/NET generated the following exception:

```
System.NullReferenceException: Object reference not set to an instance of an object.
      bei MySql.Data.MySqlClient.MySqlCommand.TimeoutExpired(Object commandObject)
      bei System.Threading._TimerCallback.TimerCallback_Context(Object state)
      bei System.Threading.ExecutionContext.runTryCode(Object userData)
      bei
      System.Runtime.CompilerServices.RuntimeHelpers.ExecuteCodeWithGuaranteedCleanup(TryCode
      code, CleanupCode backoutCode, Object userData)
      bei System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext,
      ContextCallback callback, Object state)
      bei System.Threading.ExecutionContext.Run(ExecutionContext executionContext,
      ContextCallback callback, Object state)
      bei System.Threading._TimerCallback.PerformTimerCallback(Object state)
```

  (Bug #40005)

- If, when using the `MySqlTransaction` transaction object, an exception was thrown, the transaction object was not disposed of and the transaction was not rolled back. (Bug #39817)
- When a prepared insert query is run that contains an `UNSIGNED TINYINT` in the parameter list, the complete query and data that should be inserted is corrupted and no error is thrown. (Bug #37968)
- Calling `MySqlDataAdapter.FillSchema` on a `SELECT` statement that referred to a table that did not exist left the connection in a bad state. After this call, all `SELECT` statements returned an

empty result set. If the `SELECT` statement referred to a table that did exist then everything worked as expected. (Bug #30518)

Changes in MySQL Connector/NET 5.1.7 (2008-08-21)

Bugs Fixed

- There was a short circuit evaluation error in the `MySqlCommand.CheckState()` method. When the statement `connection == null` was true a `NullReferenceException` was thrown and not the expected `InvalidOperationException`. (Bug #38276)
- Executing a command that resulted in a fatal exception did not close the connection. (Bug #37991)
- In a .NET application MySQL Connector/NET modifies the connection string so that it contains several occurrences of the same option with different values. This is illustrated by the example that follows.

  The original connection string:

  ```
  host=localhost;database=test;uid=*****;pwd=*****;
  connect timeout=25; auto enlist=false;pooling=false;
  ```

  The connection string after closing `MySqlDataReader`:

  ```
  host=localhost;database=test;uid=*****;pwd=*****;
  connect timeout=25;auto enlist=false;pooling=false;
  Allow User Variables=True;Allow User Variables=False;
  Allow User Variables=True;Allow User Variables=False;
  ```

  (Bug #37955)
- As `MySqlDbType.DateTime` is not available in `VB.NET` the warning `The datetime enum value is obsolete` was always shown during compilation. (Bug #37406)
- An unknown `MySqlErrorCode` was encountered when opening a connection with an incorrect password. (Bug #37398)
- Documentation incorrectly stated that “the DataColumn class in .NET 1.0 and 1.1 does not permit columns with type of UInt16, UInt32, or UInt64 to be autoincrement columns”. (Bug #37350)
- `SemaphoreFullException` is generated when application is closed. (Bug #36688)
- `GetSchema` did not work correctly when querying for a collection, if using a non-English locale. (Bug #35459)
- When reading back a stored double or single value using the .NET provider, the value had less precision than the one stored. (Bug #33322)
- Using the MySQL Visual Studio plugin and a MySQL 4.1 server, certain field types (`ENUM`) would not be identified correctly. Also, when looking for tables, the plugin would list all tables matching a wildcard pattern of the database name supplied in the connection string, instead of only tables within the specified database. (Bug #30603)

Changes in MySQL Connector/NET 5.1.6 (2008-05-12)

Bugs Fixed

- When creating a connection pool, specifying an invalid IP address will cause the entire application to crash, instead of providing an exception. (Bug #36432)
- An incorrect value for a bit field would returned in a multi-row query if a preceding value for the field returned `NULL`. (Bug #36313)

- The `MembershipProvider` will raise an exception when the connection string is configured with `enablePasswordRetrival = true` and `RequireQuestionAndAnswer = false`. (Bug #36159)
- When calling `GetNumberOfUsersOnline` an exception is raised on the submitted query due to a missing parameter. (Bug #36157)
- Tables with `GEOMETRY` field types would return an unknown data type exception. (Bug #36081)
- When creating a connection, setting the `ConnectionString` property of `MySqlConnection` to `NULL` would throw an exception. (Bug #35619)
- The `DbCommandBuilder.QuoteIdentifer` method was not implemented. (Bug #35492)
- When using `SqlDataSource` to open a connection, the connection would not automatically be closed when access had completed. (Bug #34460)
- Attempting to use an isolation level other than the default with a transaction scope would use the default isolation level. (Bug #34448)
- When altering a stored procedure within Visual Studio, the parameters to the procedure could be lost. (Bug #34359)
- A race condition could occur within the procedure cache resulting the cache contents overflowing beyond the configured cache size. (Bug #34338)
- Using the `TableAdapter` wizard in combination with a suitable `SELECT` statement, only the associated `INSERT` statement would also be created, rather than the required `DELETE` and `UPDATE` statements. (Bug #31338)
- Running queries based on a stored procedure would cause the data set designer to terminate. (Bug #26364)

Changes in MySQL Connector/NET 5.1.5 (2008-02-11)

Functionality Added or Changed

- Performing `GetValue()` on a field `TINYINT(1)` returned a `BOOLEAN`. While not a bug, this caused problems in software that expected an `INT` to be returned. A new connection string option `Treat Tiny As Boolean` has been added with a default value of `true`. If set to `false` the provider will treat `TINYINT(1)` as `INT`. (Bug #34052)

Bugs Fixed

- Some speed improvements have been implemented in the `TokenizeSql` process used to identify elements of SQL statements. (Bug #34220)
- When accessing tables from different databases within the same `TransactionScope`, the same user/password combination would be used for each database connection. MySQL Connector/NET does not handle multiple connections within the same transaction scope. An error is now returned if you attempt this process, instead of using the incorrect authorization information. (Bug #34204)
- The status of connections reported through the state change handler was not being updated correctly. (Bug #34082)
- Incorporated some connection string cache optimizations sent to us by Maxim Mass. (Bug #34000)
- In an open connection where the server had disconnected unexpectedly, the status information of the connection would not be updated properly. (Bug #33909)

- MySQL Connector/NET failed to compile properly with `nant`. (Bug #33508)
- Problem with membership provider would mean that `FindUserByEmail` failed with a `MySqlException` because it was trying to add a second parameter with the same name as the first. (Bug #33347)
- Using compression in the MySQL connection with MySQL Connector/NET would be slower than using native (uncompressed) communication. (Bug #27865)

Changes in MySQL Connector/NET 5.1.4 (2007-11-20)

Bugs Fixed

- Setting the size of a string parameter after the value could cause an exception. (Bug #32094)
- Creation of parameter objects with noninput direction using a constructor failed. This was cause by some old legacy code preventing their use. (Bug #32093)
- A date string could be returned incorrectly by `MySqlDateTime.ToString()` when the date returned by MySQL was `0000-00-00 00:00:00`. (Bug #32010)
- A syntax error in a set of batch statements could leave the data adapter in a state that appears hung. (Bug #31930)
- Installing over a failed uninstall of a previous version could result in multiple clients being registered in the `machine.config`. This would prevent certain aspects of the MySQL connection within Visual Studio to work properly. (Bug #31731)
- MySQL Connector/NET would incorrectly report success when enlisting in a distributed transaction, although distributed transactions are not supported. (Bug #31703)
- Data cached from the connection string could return invalid information because the internal routines were not using case-sensitive semantics. This lead to updated connection string options not being recognized if they were of a different case than the existing cached values. (Bug #31433)
- Trying to use a connection that was not open could return an ambiguous and misleading error message. (Bug #31262)
- Column name metadata was not using the character set as defined within the connection string being used. (Bug #31185)
- Memory usage could increase and decrease significantly when updating or inserting a large number of rows. (Bug #31090)
- Commands executed from within the state change handler would fail with a `NULL` exception. (Bug #30964)
- Extracting data through XML functions within a query returns the data as `System.Byte[]`. This was due to MySQL Connector/NET incorrectly identifying `BLOB` fields as binary, rather than text. (Bug #30233)
- When running a stored procedure multiple times on the same connection, the memory usage could increase indefinitely. (Bug #30116)
- Column types with only 1-bit (such as `BOOLEAN` and `TINYINT(1)` were not returned as boolean fields. (Bug #27959)
- When accessing certain statements, the command would timeout before the command completed. Because this cannot always be controlled through the individual command timeout options, a `default command timeout` has been added to the connection string options. (Bug #27958)
- The server error code was not updated in the `Data[]` hash, which prevented `DbProviderFactory` users from accessing the server error code. (Bug #27436)

- The `MySqlDbType.Datetime` has been replaced with `MySqlDbType.DateTime`. The old format has been obsoleted. (Bug #26344)
- Changing the connection string of a connection to one that changes the parameter marker after the connection had been assigned to a command but before the connection is opened could cause parameters to not be found. (Bug #13991)

Changes in MySQL Connector/NET 5.1.3 (2007-09-21, Beta)

This is a new Beta development release, fixing recently discovered bugs.

Bugs Fixed

- An incorrect `ConstraintException` could be raised on an `INSERT` when adding rows to a table with a multiple-column unique key index. (Bug #30204)
- A `DATE` field would be updated with a date/time value, causing a `MySqlDataAdapter.Update()` exception. (Bug #30077)
- The Saudi Hijri calendar was not supported. (Bug #29931)
- Calling `SHOW CREATE PROCEDURE` for routines with a hyphen in the catalog name produced a syntax error. (Bug #29526)
- Connecting to a MySQL server earlier than version 4.1 would raise a `NullException`. (Bug #29476)
- The availability of a MySQL server would not be reset when using pooled connections (`pooling=true`). This would lead to the server being reported as unavailable, even if the server become available while the application was still running. (Bug #29409)
- A `FormatException` error would be raised if a parameter had not been found, instead of `Resources.ParameterMustBeDefined`. (Bug #29312)
- An exception would be thrown when using the Manage Role functionality within the web administrator to assign a role to a user. (Bug #29236)
- Using the membership/role providers when `validationKey` or `decryptionKey` parameters are set to `AutoGenerate`, an exception would be raised when accessing the corresponding values. (Bug #29235)
- Certain operations would not check the `UsageAdvisor` setting, causing log messages from the Usage Advisor even when it was disabled. (Bug #29124)
- Using the same connection string multiple times would result in `Database=`*`dbname`* appearing multiple times in the resulting string. (Bug #29123)
- *Visual Studio Plugin*: Adding a new query based on a stored procedure that uses the `SELECT` statement would terminate the query/TableAdapter wizard. (Bug #29098)
- Using `TransactionScope` would cause an `InvalidOperationException`. (Bug #28709)

Changes in MySQL Connector/NET 5.1.2 (2007-06-18)

This is a new Beta development release, fixing recently discovered bugs.

Bugs Fixed

- Log messages would be truncated to 300 bytes. (Bug #28706)
- Creating a user failed due to the application name being set incorrectly. (Bug #28648)
- *Visual Studio Plugin*: Adding a new query based on a stored procedure that used a `UPDATE`, `INSERT` or `DELETE` statement would terminate the query/TableAdapter wizard. (Bug #28536)

- *Visual Studio Plugin*: Query Builder would fail to show `TINYTEXT` columns, and any columns listed after a `TINYTEXT` column correctly. (Bug #28437)
- Accessing the results from a large query when using data compression in the connection failed to return all the data. (Bug #28204)
- *Visual Studio Plugin*: Update commands would not be generated correctly when using the TableAdapter wizard. (Bug #26347)

Changes in MySQL Connector/NET 5.1.1 (2007-05-23)

Bugs Fixed

- Running the statement `SHOW PROCESSLIST` would return columns as byte arrays instead of native columns. (Bug #28448)
- Installation of the MySQL Connector/NET on Windows failed if VisualStudio had not already been installed. (Bug #28260)
- MySQL Connector/NET would look for the wrong table when executing `User.IsRole()`. (Bug #28251)
- Building a connection string within a tight loop would show slow performance. (Bug #28167)
- The `UNSIGNED` flag for parameters in a stored procedure would be ignored when using `MySqlCommandBuilder` to obtain the parameter information. (Bug #27679)
- Using `MySQLDataAdapter.FillSchema()` on a stored procedure would raise an exception: `Invalid attempt to access a field before calling Read()`. (Bug #27668)
- `DATETIME` fields from versions of MySQL before 4.1 would be incorrectly parsed, resulting in a exception. (Bug #23342)
- Fixed password property on `MySqlConnectionStringBuilder` to use `PasswordPropertyText` attribute. This causes dots to show instead of actual password text.

Changes in MySQL Connector/NET 5.1.0 (2007-05-01)

Functionality Added or Changed

- Now compiles for .NET CF 2.0.
- Rewrote stored procedure parsing code using a new SQL tokenizer. Really nasty procedures including nested comments are now supported.
- GetSchema will now report objects relative to the currently selected database. What this means is that passing in null as a database restriction will report objects on the currently selected database only.
- Added Membership and Role provider contributed by Sean Wright (thanks!).

Changes in MySQL Connector/Net 5.0

Changes in MySQL Connector/NET 5.0.10 (Not released)

Bugs Fixed

- If, when using the `MySqlTransaction` transaction object, an exception was thrown, the transaction object was not disposed of and the transaction was not rolled back. (Bug #39817)
- Executing a command that resulted in a fatal exception did not close the connection. (Bug #37991)

- When a prepared insert query is run that contains an `UNSIGNED TINYINT` in the parameter list, the complete query and data that should be inserted is corrupted and no error is thrown. (Bug #37968)
- In a .NET application MySQL Connector/NET modifies the connection string so that it contains several occurrences of the same option with different values. This is illustrated by the example that follows.

  The original connection string:

  ```
  host=localhost;database=test;uid=*****;pwd=*****;
  connect timeout=25; auto enlist=false;pooling=false;
  ```

  The connection string after closing `MySqlDataReader`:

  ```
  host=localhost;database=test;uid=*****;pwd=*****;
  connect timeout=25;auto enlist=false;pooling=false;
  Allow User Variables=True;Allow User Variables=False;
  Allow User Variables=True;Allow User Variables=False;
  ```

  (Bug #37955)
- When creating a connection pool, specifying an invalid IP address will cause the entire application to crash, instead of providing an exception. (Bug #36432)
- `GetSchema` did not work correctly when querying for a collection, if using a non-English locale. (Bug #35459)
- When reading back a stored double or single value using the .NET provider, the value had less precision than the one stored. (Bug #33322)

Changes in MySQL Connector/NET 5.0.9 (2008-04-17)

Bugs Fixed

- The `DbCommandBuilder.QuoteIdentifer` method was not implemented. (Bug #35492)
- Setting the size of a string parameter after the value could cause an exception. (Bug #32094)
- Creation of parameter objects with noninput direction using a constructor failed. This was cause by some old legacy code preventing their use. (Bug #32093)
- A date string could be returned incorrectly by `MySqlDateTime.ToString()` when the date returned by MySQL was `0000-00-00 00:00:00`. (Bug #32010)
- A syntax error in a set of batch statements could leave the data adapter in a state that appears hung. (Bug #31930)
- Installing over a failed uninstall of a previous version could result in multiple clients being registered in the `machine.config`. This would prevent certain aspects of the MySQL connection within Visual Studio to work properly. (Bug #31731)
- Data cached from the connection string could return invalid information because the internal routines were not using case-sensitive semantics. This lead to updated connection string options not being recognized if they were of a different case than the existing cached values. (Bug #31433)
- Column name metadata was not using the character set as defined within the connection string being used. (Bug #31185)
- Memory usage could increase and decrease significantly when updating or inserting a large number of rows. (Bug #31090)
- Commands executed from within the state change handler would fail with a `NULL` exception. (Bug #30964)

- When running a stored procedure multiple times on the same connection, the memory usage could increase indefinitely. (Bug #30116)
- The server error code was not updated in the `Data[]` hash, which prevented `DbProviderFactory` users from accessing the server error code. (Bug #27436)
- Changing the connection string of a connection to one that changes the parameter marker after the connection had been assigned to a command but before the connection is opened could cause parameters to not be found. (Bug #13991)

Changes in MySQL Connector/NET 5.0.8 (2007-08-21)

Note

This version introduces a new installer technology.

Bugs Fixed

- Extracting data through XML functions within a query returns the data as `System.Byte[]`. This was due to MySQL Connector/NET incorrectly identifying `BLOB` fields as binary, rather than text. (Bug #30233)
- An incorrect `ConstraintException` could be raised on an `INSERT` when adding rows to a table with a multiple-column unique key index. (Bug #30204)
- A `DATE` field would be updated with a date/time value, causing a `MySqlDataAdapter.Update()` exception. (Bug #30077)
- Fixed bug where MySQL Connector/NET was hand building some date time patterns rather than using the patterns provided under CultureInfo. This caused problems with some calendars that do not support the same ranges as Gregorian.. (Bug #29931)
- Calling `SHOW CREATE PROCEDURE` for routines with a hyphen in the catalog name produced a syntax error. (Bug #29526)
- The availability of a MySQL server would not be reset when using pooled connections (`pooling=true`). This would lead to the server being reported as unavailable, even if the server become available while the application was still running. (Bug #29409)
- A `FormatException` error would be raised if a parameter had not been found, instead of `Resources.ParameterMustBeDefined`. (Bug #29312)
- Certain operations would not check the `UsageAdvisor` setting, causing log messages from the Usage Advisor even when it was disabled. (Bug #29124)
- Using the same connection string multiple times would result in `Database=`*`dbname`* appearing multiple times in the resulting string. (Bug #29123)
- Log messages would be truncated to 300 bytes. (Bug #28706)
- Accessing the results from a large query when using data compression in the connection will fail to return all the data. (Bug #28204)
- Fixed problem where `MySqlConnection.BeginTransaction` checked the drivers status var before checking if the connection was open. The result was that the driver could report an invalid condition on a previously opened connection.
- Fixed problem where we were not closing prepared statement handles when commands are disposed. This could lead to using up all prepared statement handles on the server.
- Fixed the database schema collection so that it works on servers that are not properly respecting the `lower_case_table_names` setting.

- Fixed problem where any attempt to not read all the records returned from a select where each row of the select is greater than 1024 bytes would hang the driver.
- Fixed problem where a command timing out just after it actually finished would cause an exception to be thrown on the command timeout thread which would then be seen as an unhandled exception.
- Fixed some serious issues with command timeout and cancel that could present as exceptions about thread ownership. The issue was that not all queries cancel the same. Some produce resultsets while others don't. ExecuteReader had to be changed to check for this.

Changes in MySQL Connector/NET 5.0.7 (2007-05-18)

Bugs Fixed

- Running the statement `SHOW PROCESSLIST` would return columns as byte arrays instead of native columns. (Bug #28448)
- Building a connection string within a tight loop would show slow performance. (Bug #28167)
- Using logging (with the `logging=true` parameter to the connection string) would not generate a log file. (Bug #27765)
- The `UNSIGNED` flag for parameters in a stored procedure would be ignored when using `MySqlCommandBuilder` to obtain the parameter information. (Bug #27679)
- Using `MySQLDataAdapter.FillSchema()` on a stored procedure would raise an exception: `Invalid attempt to access a field before calling Read()`. (Bug #27668)
- If you close an open connection with an active transaction, the transaction is not automatically rolled back. (Bug #27289)
- When cloning an open `MySqlClient.MySqlConnection` with the `Persist Security Info=False` option set, the cloned connection is not usable because the security information has not been cloned. (Bug #27269)
- Enlisting a null transaction would affect the current connection object, such that further enlistment operations to the transaction are not possible. (Bug #26754)
- Attempting to change the `Connection Protocol` property within a `PropertyGrid` control would raise an exception. (Bug #26472)
- DataSet wizard would show all tables instead of only the tables available within the selected database. (Bug #26348)
- The `characterset` property would not be identified during a connection (also affected Visual Studio Plugin). (Bug #26147, Bug #27240)
- The `CreateFormat` column of the `DataTypes` collection did not contain a format specification for creating a new column type. (Bug #25947)
- `DATETIME` fields from versions of MySQL before 4.1 would be incorrectly parsed, resulting in a exception. (Bug #23342)

Changes in MySQL Connector/NET 5.0.6 (2007-03-22)

Bugs Fixed

- Publisher listed in "Add/Remove Programs" is not consistent with other MySQL products. (Bug #27253)
- `DESCRIBE ...` SQL statement returns byte arrays rather than data on MySQL versions older than 4.1.15. (Bug #27221)

- `cmd.Parameters.RemoveAt("Id")` will cause an error if the last item is requested. (Bug #27187)
- `MySqlParameterCollection` and parameters added with `Insert` method can not be retrieved later using `ParameterName`. (Bug #27135)
- Exception thrown when using large values in `UInt64` parameters. (Bug #27093)
- MySQL Visual Studio Plugin 1.1.2 does not work with MySQL Connector/NET 5.0.5. (Bug #26960)

Changes in MySQL Connector/NET 5.0.5 (2007-03-07)

Functionality Added or Changed

- Reverted behavior that required parameter names to start with the parameter marker. We apologize for this back and forth but we mistakenly changed the behavior to not match what `SqlClient` supports. We now support using either syntax for adding parameters however we also respond exactly like `SqlClient` in that if you ask for the index of a parameter using a syntax different from when you added the parameter, the result will be -1.
- Assembly now properly appears in the Visual Studio 2005 Add/Remove Reference dialog.
- Fixed problem that prevented use of `SchemaOnly` or `SingleRow` command behaviors with stored procedures or prepared statements.
- Added `MySqlParameterCollection.AddWithValue` and marked the `Add(name, value)` method as obsolete.
- Return parameters created with DeriveParameters now have the name `RETURN_VALUE`.
- Fixed problem with parameter name hashing where the hashes were not getting updated when parameters were removed from the collection.
- Fixed problem with calling stored functions when a return parameter was not given.
- Added `Use Procedure Bodies` connection string option to enable calling procedures without using procedure metadata.

Bugs Fixed

- `MySqlConnection.GetSchema` fails with `NullReferenceException` for Foreign Keys. (Bug #26660)
- MySQL Connector/NET failed to install under Windows Vista. (Bug #26430)
- Opening a connection would be slow due to host name lookup. (Bug #26152)
- Incorrect values/formats would be applied when the `OldSyntax` connection string option was used. (Bug #25950)
- Registry would be incorrectly populated with installation locations. (Bug #25928)
- Times with negative values would be returned incorrectly. (Bug #25912)
- Returned data types of a `DataTypes` collection do not contain the right correct CLR data type. (Bug #25907)
- `GetSchema` and `DataTypes` would throw an exception due to an incorrect table name. (Bug #25906)
- `MySqlConnection` throws an exception when connecting to MySQL v4.1.7. (Bug #25726)

- `SELECT` did not work correctly when using a `WHERE` clause containing a UTF-8 string. (Bug #25651)
- When closing and then re-opening a connection to a database, the character set specification is lost. (Bug #25614)
- Filling a table schema through a stored procedure triggers a runtime error. (Bug #25609)
- `BINARY` and `VARBINARY` columns would be returned as a string, not binary, data type. (Bug #25605)
- A critical `ConnectionPool` error would result in repeated `System.NullReferenceException`. (Bug #25603)
- The `UpdateRowSource.FirstReturnedRecord` method does not work. (Bug #25569)
- When connecting to a MySQL Server earlier than version 4.1, the connection would hang when reading data. (Bug #25458)
- Using `ExecuteScalar()` with more than one query, where one query fails, will hang the connection. (Bug #25443)
- When a `MySqlConversionException` is raised on a remote object, the client application would receive a `SerializationException` instead. (Bug #24957)
- When connecting to a server, the return code from the connection could be zero, even though the host name was incorrect. (Bug #24802)
- High CPU utilization would be experienced when there is no idle connection waiting when using pooled connections through `MySqlPool.GetConnection`. (Bug #24373)
- MySQL Connector/NET would not compile properly when used with Mono 1.2. (Bug #24263)
- Applications would crash when calling with `CommandType` set to `StoredProcedure`.

Changes in MySQL Connector/NET 5.0.4 (Internal)

This is an internal development release, fixing recently discovered bugs.

Changes in MySQL Connector/NET 5.0.3 (2007-01-05)

Functionality Added or Changed

- Usage Advisor has been implemented. The Usage Advisor checks your queries and will report if you are using the connection inefficiently.
- PerfMon hooks have been added to monitor the stored procedure cache hits and misses.
- The `MySqlCommand` object now supports asynchronous query methods. This is implemented useg the `BeginExecuteNonQuery` and `EndExecuteNonQuery` methods.
- Metadata from stored procedures and stored function execution are cached.
- The `CommandBuilder.DeriveParameters` function has been updated to the procedure cache.
- The `ViewColumns GetSchema` collection has been updated.
- Improved speed and performance by re-architecting certain sections of the code.
- Support for the embedded server and client library have been removed from this release. Support will be added back to a later release.

- The ShapZipLib library has been replaced with the deflate support provided within .NET 2.0.
- SSL support has been updated.

Bugs Fixed

- Additional text added to error message (Bug #25178)
- An exception would be raised, or the process would hang, if `SELECT` privileges on a database were not granted and a stored procedure was used. (Bug #25033)
- When adding parameter objects to a command object, if the parameter direction is set to `ReturnValue` before the parameter is added to the command object then when the command is executed it throws an error. (Bug #25013)
- The Add Connection dialog of the Server Explorer would freeze when accessing databases with capitalized characters in their name. (Bug #24875)
- Using `Driver.IsTooOld()` would return the wrong value. (Bug #24661)
- When using a `DbNull.Value` as the value for a parameter value, and then later setting a specific value type, the command failed with an exception because the wrong type was implied from the `DbNull.Value`. (Bug #24565)
- Stored procedure executions are not thread safe. (Bug #23905)
- Deleting a connection to a disconnected server when using the Visual Studio Plugin would cause an assertion failure. (Bug #23687)
- Nested transactions (which are unsupported)do not raise an error or warning. (Bug #22400)

Changes in MySQL Connector/NET 5.0.2 (2006-11-06)

Functionality Added or Changed

- An `Ignore Prepare` option has been added to the connection string options. If enabled, prepared statements will be disabled application-wide. The default for this option is true.
- Implemented a stored procedure cache. By default, the connector caches the metadata for the last 25 procedures that are seen. You can change the number of procedures that are cached by using the `procedure cache` connection string.
- **Important change:** Due to a number of issues with the use of server-side prepared statements, MySQL Connector/NET 5.0.2 has disabled their use by default. The disabling of server-side prepared statements does not affect the operation of the connector in any way.

  To enable server-side prepared statements you must add the following configuration property to your connector string properties:

  ```
  ignore prepare=false
  ```

  The default value of this property is true.

Bugs Fixed

- One system where IPv6 was enabled, MySQL Connector/NET would incorrectly resolve host names. (Bug #23758)

- Column names with accented characters were not parsed properly causing malformed column names in result sets. (Bug #23657)
- An exception would be thrown when calling `GetSchemaTable` and `fields` was null. (Bug #23538)
- A `System.FormatException` exception would be raised when invoking a stored procedure with an `ENUM` input parameter. (Bug #23268)
- During installation, an antivirus error message would be raised (indicating a malicious script problem). (Bug #23245)
- Creating a connection through the Server Explorer when using the Visual Studio Plugin failed. The installer for the Visual Studio Plugin has been updated to ensure that MySQL Connector/NET 5.0.2 must be installed. (Bug #23071)
- Using Windows Vista (RC2) as a nonprivileged user would raise a `Registry key 'Global' access denied`. (Bug #22882)
- Within Mono, using the `PreparedStatement` interface could result in an error due to a `BitArray` copying error. (Bug #18186)
- MySQL Connector/NET did not work as a data source for the `SqlDataSource` object used by ASP.NET 2.0. (Bug #16126)

Changes in MySQL Connector/NET 5.0.1 (2006-10-01)

Bugs Fixed

- MySQL Connector/NET on a Turkish operating system, may fail to execute certain SQL statements correctly. (Bug #22452)
- Starting a transaction on a connection created by `MySql.Data.MySqlClient.MySqlClientFactory`, using `BeginTransaction` without specifying an isolation level, causes the SQL statement to fail with a syntax error. (Bug #22042)
- The `Mysqlexception` class is now derived from the `DbException` class. (Bug #21874)
- The `#` would not be accepted within column/table names, even though it was valid. (Bug #21521)
- You can now install the MySQL Connector/NET MSI package from the command line using the `/passive`, `/quiet`, `/q` options. (Bug #19994)
- Submitting an empty string to a command object through `prepare` raises an `System.IndexOutOfRangeException`, rather than a MySQL Connector/NET exception. (Bug #18391)
- Incorrect field/data lengths could be returned for `VARCHAR` UTF8 columns. (Bug #14592)
- Using `ExecuteScalar` with a datetime field, where the value of the field is "0000-00-00 00:00:00", a `MySqlConversionException` exception would be raised. (Bug #11991)
- An `MySql.Data.Types.MySqlConversionException` would be raised when trying to update a row that contained a date field, where the date field contained a zero value (0000-00-00 00:00:00). (Bug #9619)
- Executing multiple queries as part of a transaction returns `There is already an openDataReader associated with this Connection which must be closed first`. (Bug #7248)

Changes in MySQL Connector/NET 5.0.0 (2006-08-08)

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- Replaced use of ICSharpCode with .NET 2.0 internal deflate support.
- Refactored test suite to test all protocols in a single pass.
- Added usage advisor warnings for requesting column values by the wrong type.
- Reimplemented PacketReader/PacketWriter support into `MySqlStream` class.
- Reworked connection string classes to be simpler and faster.
- Added procedure metadata caching.
- Added internal implemention of SHA1 so we don't have to distribute the OpenNetCF on mobile devices.
- Implemented `MySqlClientFactory` class.
- Added perfmon hooks for stored procedure cache hits and misses.
- Implemented classes and interfaces for ADO.NET 2.0 support.
- Added Async query methods.
- Implemented Usage Advisor.
- Completely refactored how column values are handled to avoid boxing in some cases.
- Implemented `MySqlConnectionBuilder` class.

Bugs Fixed

- CommandText: Question mark in comment line is being parsed as a parameter. (Bug #6214)

Changes in MySQL Connector/Net 1.0

Changes in MySQL Connector/NET 1.0.11 (Not released)

Bugs Fixed

- Attempting to utilize MySQL Connector/NET version 1.0.10 throws a fatal exception under Mono when pooling is enabled. (Bug #33682)
- Setting the size of a string parameter after the value could cause an exception. (Bug #32094)
- Creation of parameter objects with noninput direction using a constructor failed. This was cause by some old legacy code preventing their use. (Bug #32093)
- Memory usage could increase and decrease significantly when updating or inserting a large number of rows. (Bug #31090)
- Commands executed from within the state change handler would fail with a `NULL` exception. (Bug #30964)
- Extracting data through XML functions within a query returns the data as `System.Byte[]`. This was due to MySQL Connector/NET incorrectly identifying `BLOB` fields as binary, rather than text. (Bug #30233)
- Using compression in the MySQL connection with MySQL Connector/NET would be slower than using native (uncompressed) communication. (Bug #27865)

- Changing the connection string of a connection to one that changes the parameter marker after the connection had been assigned to a command but before the connection is opened could cause parameters to not be found. (Bug #13991)

Changes in MySQL Connector/NET 1.0.10 (2007-08-24)

Bugs Fixed

- An incorrect `ConstraintException` could be raised on an `INSERT` when adding rows to a table with a multiple-column unique key index. (Bug #30204)
- The availability of a MySQL server would not be reset when using pooled connections (`pooling=true`). This would lead to the server being reported as unavailable, even if the server become available while the application was still running. (Bug #29409)
- Publisher listed in "Add/Remove Programs" is not consistent with other MySQL products. (Bug #27253)
- `MySqlParameterCollection` and parameters added with `Insert` method can not be retrieved later using `ParameterName`. (Bug #27135)
- `BINARY` and `VARBINARY` columns would be returned as a string, not binary, data type. (Bug #25605)
- A critical `ConnectionPool` error would result in repeated `System.NullReferenceException`. (Bug #25603)
- When a `MySqlConversionException` is raised on a remote object, the client application would receive a `SerializationException` instead. (Bug #24957)
- High CPU utilization would be experienced when there is no idle connection waiting when using pooled connections through `MySqlPool.GetConnection`. (Bug #24373)

Changes in MySQL Connector/NET 1.0.9 (2007-02-02)

Functionality Added or Changed

- The ICSharpCode ZipLib is no longer used by the Connector, and is no longer distributed with it.
- **Important change:** Binaries for .NET 1.0 are no longer supplied with this release. If you need support for .NET 1.0, you must build from source.
- Improved `CommandBuilder.DeriveParameters` to first try and use the procedure cache before querying for the stored procedure metadata. Return parameters created with `DeriveParameters` now have the name `RETURN_VALUE`.
- An `Ignore Prepare` option has been added to the connection string options. If enabled, prepared statements will be disabled application-wide. The default for this option is true.
- Implemented a stored procedure cache. By default, the connector caches the metadata for the last 25 procedures that are seen. You can change the number of procedures that are cached by using the `procedure cache` connection string.
- **Important change:** Due to a number of issues with the use of server-side prepared statements, MySQL Connector/NET 5.0.2 has disabled their use by default. The disabling of server-side prepared statements does not affect the operation of the connector in any way.

  To enable server-side prepared statements you must add the following configuration property to your connector string properties:

```
ignore prepare=false
```

The default value of this property is true.

Bugs Fixed

- Times with negative values would be returned incorrectly. (Bug #25912)
- `MySqlConnection` throws a `NullReferenceException` and `ArgumentNullException` when connecting to MySQL v4.1.7. (Bug #25726)
- `SELECT` did not work correctly when using a `WHERE` clause containing a UTF-8 string. (Bug #25651)
- When closing and then re-opening a connection to a database, the character set specification is lost. (Bug #25614)
- Trying to fill a table schema through a stored procedure triggers a runtime error. (Bug #25609)
- Using `ExecuteScalar()` with more than one query, where one query fails, will hang the connection. (Bug #25443)
- Additional text added to error message. (Bug #25178)
- When adding parameter objects to a command object, if the parameter direction is set to `ReturnValue` before the parameter is added to the command object then when the command is executed it throws an error. (Bug #25013)
- When connecting to a server, the return code from the connection could be zero, even though the host name was incorrect. (Bug #24802)
- Using `Driver.IsTooOld()` would return the wrong value. (Bug #24661)
- When using a `DbNull.Value` as the value for a parameter value, and then later setting a specific value type, the command failed with an exception because the wrong type was implied from the `DbNull.Value`. (Bug #24565)
- Stored procedure executions are not thread safe. (Bug #23905)
- The `CommandBuilder` would mistakenly add insert parameters for a table column with auto incrementation enabled. (Bug #23862)
- One system where IPv6 was enabled, MySQL Connector/NET would incorrectly resolve host names. (Bug #23758)
- An `System.OverflowException` would be raised when accessing a varchar field over 255 bytes. (Bug #23749)
- Nested transactions do not raise an error or warning. (Bug #22400)
- Within Mono, using the `PreparedStatement` interface could result in an error due to a `BitArray` copying error. (Bug #18186)

Changes in MySQL Connector/NET 1.0.8 (2006-10-20)

Functionality Added or Changed

- Stored procedures are now cached.

- The method for retrieving stored procedure metadata has been changed so that users without `SELECT` privileges on the `mysql.proc` table can use a stored procedure.

Bugs Fixed

- MySQL Connector/NET on a Turkish operating system, may fail to execute certain SQL statements correctly. (Bug #22452)
- The `#` would not be accepted within column/table names, even though it was valid. (Bug #21521)
- Calling `Close` on a connection after calling a stored procedure would trigger a `NullReferenceException`. (Bug #20581)
- You can now install the MySQL Connector/NET MSI package from the command line using the `/passive`, `/quiet`, `/q` options. (Bug #19994)
- The DiscoverParameters function failed when a stored procedure used a `NUMERIC` parameter type. (Bug #19515)
- When running a query that included a date comparison, a DateReader error would be raised. (Bug #19481)
- `IDataRecord.GetString` would raise `NullPointerException` for null values in returned rows. Method now throws `SqlNullValueException`. (Bug #19294)
- Parameter substitution in queries where the order of parameters and table fields did not match would substitute incorrect values. (Bug #19261)
- Submitting an empty string to a command object through `prepare` raises an `System.IndexOutOfRangeException`, rather than a MySQL Connector/NET exception. (Bug #18391)
- An exception would be raised when using an output parameter to a `System.String` value. (Bug #17814)
- CHAR type added to MySqlDbType. (Bug #17749)
- A `SELECT` query on a table with a date with a value of `'0000-00-00'` would hang the application. (Bug #17736)
- The CommandBuilder ignored Unsigned flag at Parameter creation. (Bug #17375)
- When working with multiple threads, character set initialization would generate errors. (Bug #17106)
- When using an unsigned 64-bit integer in a stored procedure, the unsigned bit would be lost stored. (Bug #16934)
- `DataReader` would show the value of the previous row (or last row with nonnull data) if the current row contained a `datetime` field with a null value. (Bug #16884)
- Unsigned data types were not properly supported. (Bug #16788)
- The connection string parser did not permit single or double quotation marks in the password. (Bug #16659)
- The `MySqlDateTime` class did not contain constructors. (Bug #15112)
- Called `MySqlCommandBuilder.DeriveParameters` for a stored procedure that has no paramers would cause an application crash. (Bug #15077)
- Incorrect field/data lengths could be returned for `VARCHAR` UTF8 columns. (Bug #14592)
- Using `ExecuteScalar` with a datetime field, where the value of the field is "0000-00-00 00:00:00", a `MySqlConversionException` exception would be raised. (Bug #11991)

- An `MySql.Data.Types.MySqlConversionException` would be raised when trying to update a row that contained a date field, where the date field contained a zero value (0000-00-00 00:00:00). (Bug #9619)
- When using `MySqlDataAdapter`, connections to a MySQL server may remain open and active, even though the use of the connection has been completed and the data received. (Bug #8131)
- Executing multiple queries as part of a transaction returns `There is already an openDataReader associated with this Connection which must be closed first`. (Bug #7248)

Changes in MySQL Connector/NET 1.0.7 (2005-11-21)

Bugs Fixed

- Unsigned `tinyint` (NET byte) would lead to and incorrectly determined parameter type from the parameter value. (Bug #18570)
- A `#42000Query was empty` exception occurred when executing a query built with `MySqlCommandBuilder`, if the query string ended with a semicolon. (Bug #14631)
- The parameter collection object's `Add()` method added parameters to the list without first checking to see whether they already existed. Now it updates the value of the existing parameter object if it exists. (Bug #13927)
- Added support for the `cp932` character set. (Bug #13806)
- Calling a stored procedure where a parameter contained special characters (such as `'@'`) would produce an exception. Note that `ANSI_QUOTES` had to be enabled to make this possible. (Bug #13753)
- The `Ping()` method did not update the `State` property of the `Connection` object. (Bug #13658)
- Implemented the `MySqlCommandBuilder.DeriveParameters` method that is used to discover the parameters for a stored procedure. (Bug #13632)
- A statement that contained multiple references to the same parameter could not be prepared. (Bug #13541)

Changes in MySQL Connector/NET 1.0.6 (2005-10-03)

Bugs Fixed

- MySQL Connector/NET 1.0.5 could not connect on Mono. (Bug #13345)
- Serializing a parameter failed if the first value passed in was `NULL`. (Bug #13276)
- Field names that contained the following characters caused errors: `()%<>/` (Bug #13036)
- The `nant` build sequence had problems. (Bug #12978)
- The MySQL Connector/NET 1.0.5 installer would not install alongside MySQL Connector/NET 1.0.4. (Bug #12835)

Changes in MySQL Connector/NET 1.0.5 (2005-08-29)

Bugs Fixed

- MySQL Connector/NET could not connect to MySQL 4.1.14. (Bug #12771)
- With multiple hosts in the connection string, MySQL Connector/NET would not connect to the last host in the list. (Bug #12628)

- The `ConnectionString` property could not be set when a `MySqlConnection` object was added with the designer. (Bug #12551, Bug #8724)
- The `cp1250` character set was not supported. (Bug #11621)
- A call to a stored procedure caused an exception if the stored procedure had no parameters. (Bug #11542)
- Certain malformed queries would trigger a `Connection must be valid and open` error message. (Bug #11490)
- Trying to use a stored procedure when `Connection.Database` was not populated generated an exception. (Bug #11450)
- MySQL Connector/NET interpreted the new decimal data type as a byte array. (Bug #11294)
- Added support to call a stored function from MySQL Connector/NET. (Bug #10644)
- Connection could fail when .NET thread pool had no available worker threads. (Bug #10637)
- Calling `MySqlConnection.clone` when a connection string had not yet been set on the original connection would generate an error. (Bug #10281)
- Decimal parameters caused syntax errors. (Bug #10152, Bug #11550, Bug #10486)
- Parameters were not recognized when they were separated by linefeeds. (Bug #9722)
- The `MySqlCommandBuilder` class could not handle queries that referenced tables in a database other than the default database. (Bug #8382)
- Trying to read a `TIMESTAMP` column generated an exception. (Bug #7951)
- MySQL Connector/NET could not work properly with certain regional settings. (WL#8228)

Changes in MySQL Connector/NET 1.0.4 (2005-01-20)

Bugs Fixed

- `MySqlReader.GetInt32` throws exception if column is unsigned. (Bug #7755)
- Quote character \222 not quoted in `EscapeString`. (Bug #7724)
- `GetBytes` was not working. (Bug #7704)
- `MySqlDataReader.GetString(index)` returns non-Null value when field is `Null`. (Bug #7612)
- Clone method bug in `MySqlCommand`. (Bug #7478)
- Problem with Multiple resultsets. (Bug #7436)
- `MySqlAdapter.Fill` method throws error message `Non-negative number required`. (Bug #7345)
- `MySqlCommand.Connection` returns an IDbConnection. (Bug #7258)
- Calling prepare causing exception. (Bug #7243)
- Fixed problem with shared memory connections.
- Added or filled out several more topics in the API reference documentation.
- Fixed another small problem with prepared statements.

- Fixed problem that causes named pipes to not work with some blob functionality.

Changes in MySQL Connector/NET 1.0.3 (2004-10-12)

Bugs Fixed

- Invalid query string when using inout parameters (Bug #7133)
- Inserting `DateTime` causes `System.InvalidCastException` to be thrown. (Bug #7132)
- `MySqlDateTime` in Datatables sorting by Text, not Date. (Bug #7032)
- Exception stack trace lost when re-throwing exceptions. (Bug #6983)
- Errors in parsing stored procedure parameters. (Bug #6902)
- InvalidCast when using `DATE_ADD`-function. (Bug #6879)
- Int64 Support in `MySqlCommand` Parameters. (Bug #6863)
- Test suite fails with MySQL 4.0 because of case sensitivity of table names. (Bug #6831)
- `MySqlDataReader.GetChar(int i)` throws `IndexOutOfRange` exception. (Bug #6770)
- Integer "out" parameter from stored procedure returned as string. (Bug #6668)
- An Open Connection has been Closed by the Host System. (Bug #6634)
- Fixed Invalid character set index: 200. (Bug #6547)
- Connections now do not have to give a database on the connection string.
- Installer now includes options to install into GAC and create **Start Menu** items.
- Fixed major problem with detecting null values when using prepared statements.
- Fixed problem where multiple resultsets having different numbers of columns would cause a problem.
- Added `ServerThread` property to `MySqlConnection` to expose server thread id.
- Added Ping method to `MySqlConnection`.
- Changed the name of the test suite to `MySql.Data.Tests.dll`.
- Now `SHOW COLLATION` is used upon connection to retrieve the full list of charset ids.
- Made MySQL the default named pipe name.

Changes in MySQL Connector/NET 1.0.2 (2004-11-15, Gamma)

Bugs Fixed

- Fixed Objects not being disposed (Bug #6649)
- Fixed Charset-map for UCS-2 (Bug #6541)
- Fixed Zero date "0000-00-00" is returned wrong when filling Dataset (Bug #6429)
- Fixed double type handling in MySqlParameter(string parameterName, object value). (Bug #6428)
- Fixed Installation directory ignored using custom installation (Bug #6329)

- Fixed #HY000 Illegal mix of collations (latin1_swedish_ci,IMPLICIT) and (utf8_general_ (Bug #6322)
- Added the TableEditor CS and VB sample
- Added charset connection string option
- Fixed problem with MySqlBinary where string values could not be used to update extended text columns
- Provider is now using character set specified by server as default
- Updated the installer to include the new samples
- Fixed problem where setting command text leaves the command in a prepared state
- Fixed Long inserts take very long time (Bu #5453)
- Fixed problem where calling stored procedures might cause an "Illegal mix of collations" problem.

Changes in MySQL Connector/NET 1.0.1 (2004-10-27, Beta)

Bugs Fixed

- Fixed IndexOutOfBounds when reading `BLOB` with `DataReader` with `GetString(index)`. (Bug #6230)
- Fixed GetBoolean returns wrong values (Bug #6227)
- Fixed Method TokenizeSql() uses only a limited set of valid characters for parameters (Bug #6217)
- Fixed NET Connector source missing resx files (Bug #6216)
- Fixed System.OverflowException when using `YEAR` data type. (Bug #6036)
- Fixed MySqlDateTime sets IsZero property on all subseq.records after first zero found (Bug #6006)
- Fixed serializing of floating point parameters (double, numeric, single, decimal) (Bug #5900)
- Fixed missing Reference in DbType setter (Bug #5897)
- Fixed Parsing the ';' char (Bug #5876)
- Fixed DBNull Values causing problems with retrieving/updating queries. (Bug #5798)
- IsNullable error (Bug #5796)
- Fixed problem where MySqlParameterCollection.Add() would throw unclear exception when given a null value (Bug #5621)
- Fixed constructor initialize problems in MySqlCommand() (Bug #5613)
- Possible bug in MySqlParameter(string, object) constructor (Bug #5602)
- Fixed Yet Another "object reference not set to an instance of an object" (Bug #5496)
- Cannot run a stored procedure populating mysqlcommand.parameters (Bug #5474)
- Setting `DbType` threw a `NullReferenceException`. (Bug #5469)
- Calling `GetChars` on a `LONGTEXT` column threw an exception. (Bug #5458)
- `MySqlCommand` saw instances of `"?"` as parameters in string literals. (Bug #5392)
- `DataReader` reported all rows as `NULL` if one row was `NULL`. (Bug #5388)

- Fixed Can't display Chinese correctly (Bug #5288)
- Fixed MySqlDataReader and 'show tables from ...' behavior (Bug #5256)
- Fixed problem in PacketReader where it could try to allocate the wrong buffer size in EnsureCapacity
- Fixed problem where using old syntax while using the interfaces caused problems
- Added test case for resetting the command text on a prepared command
- Fixed problem where connection lifetime on the connect string was not being respected
- Field buffers being reused to decrease memory allocations and increase speed
- Added Aggregate function test (wasn't really a bug)
- Using PacketWriter instead of Packet for writing to streams
- Implemented SequentialAccess
- Fixed problem with ConnectionInternal where a key might be added more than once
- Fixed Russian character support as well
- Fixed problem where connector was not issuing a CMD_QUIT before closing the socket
- Fixed problem where Min Pool Size was not being respected
- Refactored compression code into CompressedStream to clean up NativeDriver
- CP1252 is now used for Latin1 only when the server is 4.1.2 and later
- Virtualized driver subsystem so future releases could easily support client or embedded server support

Changes in MySQL Connector/NET 1.0.0 (2004-09-01)

Bugs Fixed

- Thai encoding not correctly supported. (Bug #3889)
- Bumped version number to 1.0.0 for beta 1 release.
- Removed all of the XML comment warnings.
- Added `COPYING.rtf` file for use in installer.
- Updated many of the test cases.
- Fixed problem with using compression.
- Removed some last references to ByteFX.

Changes in MySQL Connector/Net Before 1.0

Changes in MySQL Connector/Net 0.9.0 (30 August 2004)

- Added test fixture for prepared statements.
- All type classes now implement a `SerializeBinary` method for sending their data to a `PacketWriter`.
- Added `PacketWriter` class that will enable future low-memory large object handling.

- Fixed many small bugs in running prepared statements and stored procedures.
- Changed command so that an exception will not be thrown in executing a stored procedure with parameters in old syntax mode.
- `SingleRow` behavior now working right even with limit.
- `GetBytes` now only works on binary columns.
- Logger now truncates long SQL commands so blob columns do not blow out our log.
- Host and database now have a default value of "" unless otherwise set.
- Connection Timeout seems to be ignored. (Bug #5214)
- Added test case for bug# 5051: GetSchema not working correctly.
- Fixed problem where `GetSchema` would return false for `IsUnique` when the column is key.
- `MySqlDataReader GetXXX` methods now using the field level `MySqlValue` object and not performing conversions.
- `DataReader` returning `NULL` for time column. (Bug #5097)
- Added test case for `LOAD DATA LOCAL INFILE`.
- Added replacetext custom nant task.
- Added `CommandBuilderTest` fixture.
- Added Last One Wins feature to `CommandBuilder`.
- Fixed persist security info case problem.
- Fixed `GetBool` so that 1, true, "true", and "yes" all count as true.
- Make parameter mark configurable.
- Added the "old syntax" connection string parameter to enable use of @ parameter marker.
- `MySqlCommandBuilder`. (Bug #4658)
- `ByteFX.MySqlClient` caches passwords if `Persist Security Info` is false. (Bug #4864)
- Updated license banner in all source files to include FLOSS exception.
- Added new .Types namespace and implementations for most current MySql types.
- Added `MySqlField41` as a subclass of `MySqlField`.
- Changed many classes to now use the new .Types types.
- Changed type `enum int` to `Int32`, `short` to `Int16`, and `bigint` to `Int64`.
- Added dummy types `UInt16`, `UInt32`, and `UInt64` to allow an unsigned parameter to be made.
- Connections are now reset when they are pulled from the connection pool.
- Refactored auth code in driver so it can be used for both auth and reset.
- Added `UserReset` test in `PoolingTests.cs`.
- Connections are now reset using `COM_CHANGE_USER` when pulled from the pool.
- Implemented `SingleResultSet` behavior.

- Implemented support of unicode.
- Added char set mappings for utf-8 and ucs-2.
- Time fields overflow using bytefx .net mysql driver (Bug #4520)
- Modified time test in data type test fixture to check for time spans where hours > 24.
- Wrong string with backslash escaping in `ByteFx.Data.MySqlClient.MySqlParameter`. (Bug #4505)
- Added code to Parameter test case TestQuoting to test for backslashes.
- `MySqlCommandBuilder` fails with multi-word column names. (Bug #4486)
- Fixed bug in `TokenizeSql` where underscore would terminate character capture in parameter name.
- Added test case for spaces in column names.
- `MySqlDataReader.GetBytes` do not work correctly. (Bug #4324)
- Added `GetBytes()` test case to `DataReader` test fixture.
- Now reading all server variables in `InternalConnection.Configure` into `Hashtable`.
- Now using `string[]` for index map in `CharSetMap`.
- Added CRInSQL test case for carriage returns in SQL.
- Setting maxPacketSize to default value in `Driver.ctor`.
- Setting `MySqlDbType` on a parameter doesn't set generic type. (Bug #4442)
- Removed obsolete data types `Long` and `LongLong`.
- Overflow exception thrown when using "use pipe" on connection string. (Bug #4071)
- Changed "use pipe" keyword to "pipe name" or just "pipe".
- Enable reading multiple resultsets from a single query.
- Added flags attribute to `ServerStatusFlags` enum.
- Changed name of `ServerStatus` enum to `ServerStatusFlags`.
- Inserted data row doesn't update properly.
- Error processing show create table. (Bug #4074)
- Change `Packet.ReadLenInteger` to `ReadPackedLong` and added `packet.ReadPackedInteger` that always reads integers packed with 2,3,4.
- Added `syntax.cs` test fixture to test various SQL syntax bugs.
- Improper handling of time values. Now time value of 00:00:00 is not treated as null. (Bug #4149)
- Moved all test suite files into `TestSuite` folder.
- Fixed bug where null column would move the result packet pointer backward.
- Added new nant build script.
- Clear tablename so it will be regen'ed properly during the next `GenerateSchema`. (Bug #3917)

- `GetValues` was always returning zero and was also always trying to copy all fields rather than respecting the size of the array passed in. (Bug #3915)
- Implemented shared memory access protocol.
- Implemented prepared statements for MySQL 4.1.
- Implemented stored procedures for MySQL 5.0.
- Renamed `MySqlInternalConnection` to `InternalConnection`.
- SQL is now parsed as chars, fixes problems with other languages.
- Added logging and allow batch connection string options.
- `RowUpdating` event not set when setting the `DataAdapter` property. (Bug #3888)
- Fixed bug in char set mapping.
- Implemented 4.1 authentication.
- Improved open/auth code in driver.
- Improved how connection bits are set during connection.
- Database name is now passed to server during initial handshake.
- Changed namespace for client to `MySql.Data.MySqlClient`.
- Changed assembly name of client to `MySql.Data.dll`.
- Changed license text in all source files to GPL.
- Added the `MySqlClient.build` Nant file.
- Removed the mono batch files.
- Moved some of the unused files into notused folder so nant build file can use wildcards.
- Implemented shared memory access.
- Major revamp in code structure.
- Prepared statements now working for MySql 4.1.1 and later.
- Finished implementing auth for 4.0, 4.1.0, and 4.1.1.
- Changed namespace from `MySQL.Data.MySQLClient` back to `MySql.Data.MySqlClient`.
- Fixed bug in `CharSetMapping` where it was trying to use text names as ints.
- Changed namespace to `MySQL.Data.MySQLClient`.
- Integrated auth changes from UC2004.
- Fixed bug where calling any of the GetXXX methods on a datareader before or after reading data would not throw the appropriate exception (thanks Luca Morelli).
- Added `TimeSpan` code in parameter.cs to properly serialize a timespan object to mysql time format (thanks Gianluca Colombo).
- Added `TimeStamp` to parameter serialization code. Prevented `DataAdapter` updates from working right (thanks Michael King).

- Fixed a misspelling in `MySqlHelper.cs` (thanks Patrick Kristiansen).

Changes in MySQL Connector/Net 0.76

- Driver now using charset number given in handshake to create encoding.
- Changed command editor to point to `MySqlClient.Design`.
- Fixed bug in `Version.isAtLeast`.
- Changed `DBConnectionString` to support changes done to `MySqlConnectionString`.
- Removed `SqlCommandEditor` and `DataAdapterPreviewDialog`.
- Using new long return values in many places.
- Integrated new `CompressedStream` class.
- Changed `ConnectionString` and added attributes to permit it to be used in `MySqlClient.Design`.
- Changed `packet.cs` to support newer lengths in `ReadLenInteger`.
- Changed other classes to use new properties and fields of `MySqlConnectionString`.
- `ConnectionInternal` is now using PING to see whether the server is available.
- Moved toolbox bitmaps into resource folder.
- Changed `field.cs` to permit values to come directly from row buffer.
- Changed to use the new driver.Send syntax.
- Using a new packet queueing system.
- Started work handling the "broken" compression packet handling.
- Fixed bug in `StreamCreator` where failure to connect to a host would continue to loop infinitely (thanks Kevin Casella).
- Improved connectstring handling.
- Moved designers into Pro product.
- Removed some old commented out code from `command.cs`.
- Fixed a problem with compression.
- Fixed connection object where an exception throw prior to the connection opening would not leave the connection in the connecting state (thanks Chris Cline).
- Added GUID support.
- Fixed sequence out of order bug (thanks Mark Reay).

Changes in MySQL Connector/Net 0.75

- Enum values now supported as parameter values (thanks Philipp Sumi).
- Year data type now supported.
- Fixed compression.
- Fixed bug where a parameter with a `TimeSpan` as the value would not serialize properly.

- Fixed bug where default constructor would not set default connection string values.
- Added some XML comments to some members.
- Work to fix/improve compression handling.
- Improved `ConnectionString` handling so that it better matches the standard set by `SqlClient`.
- A `MySqlException` is now thrown if a user name is not included in the connection string.
- Localhost is now used as the default if not specified on the connection string.
- An exception is now thrown if an attempt is made to set the connection string while the connection is open.
- Small changes to `ConnectionString` docs.
- Removed `MultiHostStream` and `MySqlStream`. Replaced it with `Common/StreamCreator`.
- Added support for Use Pipe connection string value.
- Added Platform class for easier access to platform utility functions.
- Fixed small pooling bug where new connection was not getting created after `IsAlive` fails.
- Added `Platform.cs` and `StreamCreator.cs`.
- Fixed `Field.cs` to properly handle 4.1 style timestamps.
- Changed `Common.Version` to `Common.DBVersion` to avoid name conflict.
- Fixed `field.cs` so that text columns return the right field type.
- Added `MySqlError` class to provide some reference for error codes (thanks Geert Veenstra).

Changes in MySQL Connector/Net 0.74

- Added Unix socket support (thanks Mohammad DAMT).
- Only calling `Thread.Sleep` when no data is available.
- Improved escaping of quote characters in parameter data.
- Removed misleading comments from `parameter.cs`.
- Fixed pooling bug.
- Fixed `ConnectionString` editor dialog (thanks marco p (pomarc)).
- `UserId` now supported in connection strings (thanks Jeff Neeley).
- Attempting to create a parameter that is not input throws an exception (thanks Ryan Gregg).
- Added much documentation.
- Checked in new `MultiHostStream` capability. Big thanks to Dan Guisinger for this. he originally submitted the code and idea of supporting multiple machines on the connect string.
- Added a lot of documentation.
- Fixed speed issue with 0.73.
- Changed to Thread.Sleep(0) in MySqlDataStream to help optimize the case where it doesn't need to wait (thanks Todd German).

- Prepopulating the idlepools to `MinPoolSize`.
- Fixed `MySqlPool` deadlock condition as well as stupid bug where CreateNewPooledConnection was not ever adding new connections to the pool. Also fixed `MySqlStream.ReadBytes` and `ReadByte` to not use `TicksPerSecond` which does not appear to always be right. (thanks Matthew J. Peddlesden)
- Fix for precision and scale (thanks Matthew J. Peddlesden).
- Added `Thread.Sleep(1)` to stream reading methods to be more cpu friendly (thanks Sean McGinnis).
- Fixed problem where `ExecuteReader` would sometime return null (thanks Lloyd Dupont).
- Fixed major bug with null field handling (thanks Naucki).
- Enclosed queries for `max_allowed_packet` and `characterset` inside try catch (and set defaults).
- Fixed problem where socket was not getting closed properly (thanks Steve!).
- Fixed problem where `ExecuteNonQuery` was not always returning the right value.
- Fixed `InternalConnection` to not use `@@session.max_allowed_packet` but use `@@max_allowed_packet`. (Thanks Miguel)
- Added many new XML doc lines.
- Fixed SQL parsing to not send empty queries (thanks Rory).
- Fixed problem where the reader was not unpeeking the packet on close.
- Fixed problem where user variables were not being handled (thanks Sami Vaaraniemi).
- Fixed loop checking in the MySqlPool (thanks Steve M. Brown)
- Fixed `ParameterCollection.Add` method to match `SqlClient` (thanks Joshua Mouch).
- Fixed `ConnectionString` parsing to handle no and yes for boolean and not lowercase values (thanks Naucki).
- Added `InternalConnection` class, changes to pooling.
- Implemented Persist Security Info.
- Added `security.cs` and `version.cs` to project
- Fixed `DateTime` handling in `Parameter.cs` (thanks Burkhard Perkens-Golomb).
- Fixed parameter serialization where some types would throw a cast exception.
- Fixed `DataReader` to convert all returned values to prevent casting errors (thanks Keith Murray).
- Added code to `Command.ExecuteReader` to return null if the initial SQL statement throws an exception (thanks Burkhard Perkens-Golomb).
- Fixed `ExecuteScalar` bug introduced with restructure.
- Restructure to permit `LOCAL DATA INFILE` and better sequencing of packets.
- Fixed several bugs related to restructure.
- Early work done to support more secure passwords in MySQL 4.1. Old passwords in 4.1 not supported yet.

- Parameters appearing after system parameters are now handled correctly (Adam M. (adammil)).
- Strings can now be assigned directly to blob fields (Adam M.).
- Fixed float parameters (thanks Pent).
- Improved Parameter constructor and `ParameterCollection.Add` methods to better match SqlClient (thanks Joshua Mouch).
- Corrected `Connection.CreateCommand` to return a `MySqlCommand` type.
- Fixed connection string designer dialog box problem (thanks Abraham Guyt).
- Fixed problem with sending commands not always reading the response packet (thanks Joshua Mouch).
- Fixed parameter serialization where some blobs types were not being handled (thanks Sean McGinnis).
- Removed spurious `MessageBox.show` from `DataReader` code (thanks Joshua Mouch).
- Fixed a nasty bug in the split SQL code (thanks everyone!).

Changes in MySQL Connector/Net 0.71

- Fixed bug in `MySqlStream` where too much data could attempt to be read (thanks Peter Belbin)
- Implemented `HasRows` (thanks Nash Pherson).
- Fixed bug where tables with more than 252 columns cause an exception (thanks Joshua Kessler).
- Fixed bug where SQL statements ending in ; would cause a problem (thanks Shane Krueger).
- Fixed bug in driver where error messages were getting truncated by 1 character (thanks Shane Krueger).
- Made `MySqlException` serializable (thanks Mathias Hasselmann).

Changes in MySQL Connector/Net 0.70

- Updated some of the character code pages to be more accurate.
- Fixed problem where readers could be opened on connections that had readers open.
- Moved test to separate assembly `MySqlClientTests`.
- Fixed stupid problem in driver with sequence out of order (Thanks Peter Belbin).
- Added some pipe tests.
- Increased default max pool size to 50.
- Compiles with Mono 0-24.
- Fixed connection and data reader dispose problems.
- Added `String` data type handling to parameter serialization.
- Fixed sequence problem in driver that occurred after thrown exception (thanks Burkhard Perkens-Golomb).
- Added support for `CommandBehavior.SingleRow` to `DataReader`.

- Fixed command SQL processing so quotation marks are better handled (thanks Theo Spears).
- Fixed parsing of double, single, and decimal values to account for non-English separators. You still have to use the right syntax if you using hard coded SQL, but if you use parameters the code will convert floating point types to use '.' appropriately internal both into the server and out.
- Added `MySqlStream` class to simplify timeouts and driver coding.
- Fixed `DataReader` so that it is closed properly when the associated connection is closed. [thanks smishra]
- Made client more SqlClient compliant so that DataReaders have to be closed before the connection can be used to run another command.
- Improved `DBNull.Value` handling in the fields.
- Added several unit tests.
- Fixed `MySqlException` base class.
- Improved driver coding
- Fixed bug where NextResult was returning false on the last resultset.
- Added more tests for MySQL.
- Improved casting problems by equating unsigned 32bit values to Int64 and unsigned 16bit values to Int32, and so forth.
- Added new constructor for `MySqlParameter` for (name, type, size, srccol)
- Fixed bug in `MySqlDataReader` where it didn't check for null fieldlist before returning field count.
- Started adding `MySqlClient` unit tests (added `MySqlClient/Tests` folder and some test cases).
- Fixed some things in Connection String handling.
- Moved `INIT_DB` to `MySqlPool`. I may move it again, this is in preparation of the conference.
- Fixed bug inside `CommandBuilder` that prevented inserts from happening properly.
- Reworked some of the internals so that all three execute methods of Command worked properly.
- Fixed many small bugs found during benchmarking.
- The first cut of `CoonectionPooling` is working. "min pool size" and "max pool size" are respected.
- Work to enable multiple resultsets to be returned.
- Character sets are handled much more intelligently now. The driver queries MySQL at startup for the default character set. That character set is then used for conversions if that code page can be loaded. If not, then the default code page for the current OS is used.
- Added code to save the inferred type in the name,value constructor of `Parameter`.
- Also, inferred type if value of null parameter is changed using `Value` property.
- Converted all files to use proper Camel case. MySQL is now MySql in all files. PgSQL is now PgSql.
- Added attribute to PgSql code to prevent designer from trying to show.
- Added `MySQLDbType` property to Parameter object and added proper conversion code to convert from `DbType` to `MySQLDbType`).

- Removed unused `ObjectToString` method from `MySQLParameter.cs`.
- Fixed `Add(..)` method in `ParameterCollection` so that it doesn't use `Add(name, value)` instead.
- Fixed `IndexOf` and `Contains` in `ParameterCollection` to be aware that parameter names are now stored without @.
- Fixed `Command.ConvertSQLToBytes` so it only permits characters that can be in MySQL variable names.
- Fixed `DataReader` and `Field` so that blob fields read their data from `Field.cs` and `GetBytes` works right.
- Added simple query builder editor to `CommandText` property of `MySQLCommand`.
- Fixed `CommandBuilder` and `Parameter` serialization to account for Parameters not storing @ in their names.
- Removed `MySQLFieldType` enum from Field.cs. Now using `MySQLDbType` enum.
- Added `Designer` attribute to several classes to prevent designer view when using VS.Net.
- Fixed Initial catalog typo in `ConnectionString` designer.
- Removed 3 parameter constructor for `MySQLParameter` that conflicted with (name, type, value).
- Changed `MySQLParameter` so `paramName` is now stored without leading @ (this fixed null inserts when using designer).
- Changed `TypeConverter` for `MySQLParameter` to use the constructor with all properties.

Changes in MySQL Connector/Net 0.68

- Fixed sequence issue in driver.
- Added `DbParametersEditor` to make parameter editing more like `SqlClient`.
- Fixed `Command` class so that parameters can be edited using the designer
- Update connection string designer to support `Use Compression` flag.
- Fixed string encoding so that European characters will work correctly.
- Creating base classes to aid in building new data providers.
- Added support for UID key in connection string.
- Field, parameter, command now using DBNull.Value instead of null.
- `CommandBuilder` using `DBNull.Value`.
- `CommandBuilder` now builds insert command correctly when an auto_insert field is not present.
- Field now uses typeof keyword to return `System.Types` (performance).

Changes in MySQL Connector/Net 0.65

- `MySQLCommandBuilder` now implemented.
- Transaction support now implemented (not all table types support this).
- `GetSchemaTable` fixed to not use xsd (for Mono).

- Driver is now Mono-compatible.
- TIME data type now supported.
- More work to improve Timestamp data type handling.
- Changed signatures of all classes to match corresponding `SqlClient` classes.

Changes in MySQL Connector/Net 0.60

- Protocol compression using SharpZipLib (www.icsharpcode.net).
- Named pipes on Windows now working properly.
- Work done to improve `Timestamp` data type handling.
- Implemented `IEnumerable` on `DataReader` so `DataGrid` would work.

Changes in MySQL Connector/Net 0.50

- Speed increased dramatically by removing bugging network sync code.
- Driver no longer buffers rows of data (more ADO.Net compliant).
- Conversion bugs related to `TIMESTAMP` and `DATETIME` fields fixed.