

MySQL .NET Connection String Options

The simplest MySQL connection string for C# is:

```
new MySqlConnection("server=YOURSERVER;user=YOURUSERID;password=YOURPASSWORD")
```

For all the other options, see the tables below. MySQLConnector supports most of Oracle's [Connector/.NET connection options](#).

There are also several unique options that are supported only by MySQLConnector, a replacement for `MySql.Data` that [fixes bugs](#), adds new features, and improves database access performance. [Install it now](#).

Base Options

These are the basic options that need to be defined to connect to a MySQL database.

Name	Default	Description
Host, Server, Data Source, DataSource, Address, Addr, Network Address	localhost	The host name or network address of the MySQL Server to which to connect. Multiple hosts can be specified in a comma-delimited list. On Unix-like systems, this can be a fully qualified path to a MySQL socket file, which will cause a Unix socket to be used instead of a TCP/IP socket. Only a single socket name can be specified.
Port	3306	The TCP port on which MySQL Server is listening for connections.
User Id, UserID, Username, Uid, User name, User		The MySQL user ID.
Password, pwd		The password for the MySQL user.
Database, Initial Catalog		(Optional) The case-sensitive name of the initial database to use. This may be required if the MySQL user account only has access rights to particular databases on the server.

Socket

Name	Default	Description
Protocol, ConnectionProtocol, Connection Protocol		How to connect to the MySQL Server. This option has the following values: <ul style="list-style-type: none"> • Socket (default): Use TCP/IP sockets. • Unix: Use a Unix socket. • Pipe: Use a Windows named pipe.
Pipe, PipeName, Pipe Name	MYSQL	The name of the Windows named pipe to use to connect to the server. You must also set <code>ConnectionProtocol=pipe</code> to used named pipes.

SSL/TLS Options

These are the options that need to be used in order to configure a connection to use SSL/TLS.

Name	Default	Description
SSL Mode, SslMode	Preferred	This option has the following values: <ul style="list-style-type: none"> • Preferred - (this is the default). Use SSL if the server supports it. • None - Do not use SSL. • Required - Always use SSL. Deny connection if server does not support SSL. Does not validate CA or hostname. • VerifyCA - Always use SSL. Validates the CA but tolerates hostname mismatch. • VerifyFull - Always use SSL. Validates CA and hostname.
Certificate File, CertificateFile		Specifies the path to a certificate file in PKCS #12 (.pfx) format containing a bundled Certificate and Private Key used for Mutual Authentication. To create a PKCS #12 bundle from a PEM encoded Certificate and Key, use <code>openssl pkcs12 -in cert.pem -inkey key.pem -export -out bundle.pfx</code> . This option should not be specified if <code>SslCert</code> and <code>SslKey</code> are used.
Certificate Password, CertificatePassword		

Name	Default	Description
		Specifies the password for the certificate specified using the <code>CertificateFile</code> option. Not required if the certificate file is not password protected.
SslCert, Ssl-Cert		Specifies the path to the client's SSL certificate file in PEM format. <code>SslKey</code> must also be specified, and <code>CertificateFile</code> should not be. This option is not supported on the <code>netstandard1.3</code> or <code>netstandard2.0</code> platforms.
SslKey, Ssl-Key		Specifies the path to the client's SSL private key in PEM format. <code>SslCert</code> must also be specified, and <code>CertificateFile</code> should not be.
CA Certificate File, CACertificateFile, SslCa, Ssl-Ca		This option specifies the path to a CA certificate file in a PEM Encoded (.pem) format. This should be used with <code>SslMode=VerifyCA</code> or <code>SslMode=VerifyFull</code> to enable verification of a CA certificate that is not trusted by the Operating System's certificate store.
Certificate Store Location, CertificateStoreLocation	None	Specifies whether the connection should be encrypted with a certificate from the Certificate Store on the machine. The default value of <code>None</code> means the certificate store is not used; a value of <code>CurrentUser</code> or <code>LocalMachine</code> uses the specified store.
Certificate Thumbprint, CertificateThumbprint		Specifies which certificate should be used from the Certificate Store specified in the setting above. This option must be used to indicate which certificate in the store should be used for authentication.

Connection Pooling Options

Connection pooling is enabled by default. These options are used to configure it.

Name	Default	Description
Pooling	true	Enables connection pooling. When pooling is enabled, <code>MySQLConnection.Open</code> / <code>OpenAsync</code> retrieves an open connection from the pool if one is available, and <code>Close</code> / <code>Dispose</code> returns the open connection to the pool. If there are no available connections in the pool, and the

Name	Default	Description
		pool hasn't reached <code>MaximumPoolSize</code> connections, a new connection will be opened; otherwise, the call to <code>Open</code> / <code>OpenAsync</code> blocks until a connection becomes available or <code>ConnectionTimeout</code> is reached.
Connection Lifetime, <code>ConnectionLifeTime</code>	0	Controls the maximum length of time a connection to the server can be open. Connections that are returned to the pool are destroyed if it's been more than <code>ConnectionLifeTime</code> seconds since the connection was created. The default value of zero (0) means pooled connections will never incur a <code>ConnectionLifeTime</code> timeout.
Connection Reset, <code>ConnectionReset</code>	<code>true</code>	If <code>true</code> , the connection state is reset when it is retrieved from the pool. The default value of <code>true</code> ensures that the connection is in the same state whether it's newly created or retrieved from the pool. A value of <code>false</code> avoids making an additional server round trip when obtaining a connection, but the connection state is not reset, meaning that session variables and other session state changes from any previous use of the connection are carried over.
Connection Idle Ping Time, <code>Connection Idle Ping Time</code> <i>(Experimental)</i>	0	When a connection is retrieved from the pool, and <code>ConnectionReset</code> is <code>false</code> , the server will be pinged if the connection has been idle in the pool for longer than <code>ConnectionIdlePingTime</code> seconds. If pinging the server fails, a new connection will be opened automatically by the connection pool. This ensures that the <code>MySQLConnection</code> is in a valid, open state after the call to <code>Open</code> / <code>OpenAsync</code> , at the cost of an extra server roundtrip. For high-performance scenarios, you may wish to set <code>ConnectionIdlePingTime</code> to a non-zero value to make the connection pool assume that recently-returned connections are still open. If the connection is broken, it will throw from the first call to <code>ExecuteNonQuery</code> , <code>ExecuteReader</code> , etc.; your code should handle that failure and retry the connection. This option has no effect if <code>ConnectionReset</code> is <code>true</code> , as that will cause a connection reset packet to be sent to the server, making ping redundant.

180 The amount of time (in seconds) that a connection can remain idle in the pool. Any connection above

Name	Default	Description
Connection Idle Timeout, ConnectionIdleTimeout		<code>MinimumPoolSize</code> connections that is idle for longer than <code>ConnectionIdleTimeout</code> is subject to being closed by a background task. The background task runs every minute, or half of <code>ConnectionIdleTimeout</code> , whichever is more frequent. A value of zero (0) means pooled connections will never incur a <code>ConnectionIdleTimeout</code> , and if the pool grows to its maximum size, it will never get smaller.
Maximum Pool Size, Max Pool Size, MaximumPoolSize, maxpoolsize	100	The maximum number of connections allowed in the pool.
Minimum Pool Size, Min Pool Size, MinimumPoolSize, minpoolsize	0	The minimum number of connections to leave in the pool if <code>ConnectionIdleTimeout</code> is reached.

Connection Pooling with Multiple Servers

The `Server` option supports multiple comma-delimited host names. When this is used with connection pooling, the `LoadBalance` option controls how load is distributed across backend servers.

- `RoundRobin` (default), `Random`: A total of `MaximumPoolSize` connections will be opened, but they may be unevenly distributed across back ends.
- `LeastConnections`: A total of `MaximumPoolSize` connections will be opened, and they will be evenly distributed across back ends. The active connections will be selected from the pool in least-recently-used order, which does not ensure even load across the back ends. You should set `MaximumPoolSize` to the number of servers multiplied by the desired maximum number of open connections per backend server.
- `Failover`: All connections will initially be made to the first server in the list. You should set `MaximumPoolSize` to the maximum number of open connections you want per server.

Other Options

These are the other options that `MySQLConnector` supports. They are set to sensible defaults and typically do not need to be tweaked.

Name	Default	Description
	false	

Name	Default	Description
AllowLoadLocalInfile, Allow Load Local Infile		Allows the <code>LOAD DATA LOCAL</code> command to request files from the client. This is disabled by default as a security precaution . In order to use <code>MySqlBulkLoader</code> and set its <code>Local</code> property to <code>true</code> , you must set this option to <code>True</code> in your connection string.
AllowPublicKeyRetrieval, Allow Public Key Retrieval	false	If the user account uses <code>sha256_password</code> authentication, the password must be protected during transmission; TLS is the preferred mechanism for this, but if it is not available then RSA public key encryption will be used. To specify the server's RSA public key, use the <code>ServerRSAPublicKeyFile</code> connection string setting, or set <code>AllowPublicKeyRetrieval=True</code> to allow the client to automatically request the public key from the server. Note that <code>AllowPublicKeyRetrieval=True</code> could allow a malicious proxy to perform a MITM attack to get the plaintext password, so it is <code>False</code> by default and must be explicitly enabled.
AllowUserVariables, Allow User Variables	false	Allows user-defined variables (prefixed with <code>@</code>) to be used in SQL statements. The default value (<code>false</code>) only allows <code>@</code> -prefixed names to refer to command parameters.
AllowZeroDateTime, Allow Zero DateTime	false	If set to true all <code>DATE</code> , <code>DATETIME</code> and <code>TIMESTAMP</code> columns are returned as <code>MySqlDateTime</code> objects instead of <code>DateTime</code> . This allows the special "zero" date value <code>0000-00-00</code> to be retrieved from the database. If <code>false</code> (the default) date columns are returned as <code>DateTime</code> values, and an exception is thrown for unrepresentable dates.
ApplicationName, Application Name	null	Sets the <code>program_name</code> connection attribute passed to MySQL Server. This value may be displayed by diagnostic tools, e.g., as the "Program" column in "Client Connections" in MySQL Workbench .

utf8mb4

Name	Default	Description
CharSet, Character Set, CharSet		MySQLConnector always uses <code>utf8mb4</code> to send and receive strings from MySQL Server. This option may be specified (for backwards compatibility) but it will be ignored.
Compress, Use Compression, UseCompression	false	If true (and if the server supports compression), compresses packets sent between client and server. This option is unlikely to be useful in practice unless there is a high-latency or low-bandwidth network link between the application and the database server. You should measure performance with and without this option to determine if it's beneficial in your environment.
Connect Timeout, Connection Timeout, ConnectionTimeout	15	The length of time (in seconds) to wait for a connection to the server before terminating the attempt and generating an error.
Convert Zero Datetime, ConvertZeroDateTime	false	True to have <code>MySqlDataReader.GetValue()</code> and <code>MySqlDataReader.GetDateTime()</code> return <code>DateTime.MinValue</code> for date or datetime columns that have disallowed values.
DateTimeKind	Unspecified	The <code>DateTimeKind</code> used when <code>MySqlDataReader</code> returns a <code>DateTime</code> . If set to <code>Utc</code> or <code>Local</code> , a <code>MySqlException</code> will be thrown if a <code>DateTime</code> command parameter has a <code>Kind</code> of <code>Local</code> or <code>Utc</code> , respectively.
GuidFormat	Default	<p>Determines which column type (if any) should be read as a <code>System.Guid</code>. The options include:</p> <p>Char36 All <code>CHAR(36)</code> columns are read/written as a <code>Guid</code> using lowercase hex with hyphens, which matches <code>UUID()</code>.</p> <p>Char32 All <code>CHAR(32)</code> columns are read/written as a <code>Guid</code> using lowercase hex without hyphens.</p> <p>Binary16</p>

Name	Default	Description
		<p>All <code>BINARY(16)</code> columns are read/written as a <code>Guid</code> using big-endian byte order, which matches <code>UUID_TO_BIN(x)</code>.</p>
		<p>TimeSwapBinary16</p> <p>All <code>BINARY(16)</code> columns are read/written as a <code>Guid</code> using big-endian byte order with time parts swapped, which matches <code>UUID_TO_BIN(x,1)</code>.</p>
		<p>LittleEndianBinary16</p> <p>All <code>BINARY(16)</code> columns are read/written as a <code>Guid</code> using little-endian byte order, i.e. the byte order used by <code>Guid.ToByteArray()</code> and the <code>Guid(byte[])</code> constructor.</p>
		<p>None</p> <p>No column types are automatically read as a <code>Guid</code>.</p>
		<p>Default</p> <p>Same as <code>Char36</code> if <code>OldGuids=False</code>; same as <code>LittleEndianBinary16</code> if <code>OldGuids=True</code>.</p>
Default Command Timeout, Command Timeout, DefaultCommandTimeout	30	The length of time (in seconds) each command can execute before timing out and throwing an exception, or zero to disable timeouts. See the note in the Microsoft documentation for more explanation of how this is determined.
IgnoreCommandTransaction, Ignore Command Transaction	false	If <code>true</code> , the value of <code>MySqlCommand.Transaction</code> is ignored when commands are executed. This matches the Connector/NET behaviour and can make porting code easier. For more information, see Transaction Usage .
Interactive, Interactive Session, InteractiveSession	false	If <code>true</code> , the session <code>wait_timeout</code> variable is initialized from the global <code>interactive_timeout</code> value instead of the global <code>wait_timeout</code> value.
Keep Alive, Keepalive	0	TCP Keepalive idle time. A value of 0 indicates that the OS Default keepalive settings are used. On Windows, a value greater than 0 is the idle connection time, measured in seconds, before the first keepalive packet is sent. Due to

Name	Default	Description
Load Balance, LoadBalance	RoundRobin	<p>limitations in .NET Core, Unix-based Operating Systems will always use the OS Default keepalive settings.</p> <p>The load-balancing strategy to use when <code>Host</code> contains multiple, comma-delimited, host names. The options include:</p> <p>RoundRobin Each new connection opened for this connection pool uses the next host name (sequentially with wraparound). Requires <code>Pooling=True</code>. This is the default if <code>Pooling=True</code>.</p> <p>FailOver Each new connection tries to connect to the first host; subsequent hosts are used only if connecting to the first one fails. This is the default if <code>Pooling=False</code>.</p> <p>Random Servers are tried in a random order.</p> <p>LeastConnections Servers are tried in ascending order of number of currently-open connections in this connection pool. Requires <code>Pooling=True</code>.</p>
No Backslash Escapes, NoBackslashEscapes	false	If <code>true</code> , backslashes are not escaped in string parameters. Set this to <code>true</code> if the server's SQL mode includes <code>NO_BACKSLASH_ESCAPES</code> .
Old Guids, OldGuids	false	Obsolete; use the <code>GuidFormat</code> option instead.
Persist Security Info, PersistSecurityInfo	false	When set to <code>false</code> or no (strongly recommended), security-sensitive information, such as the password, is not returned as part of the connection string if the connection is open or has ever been in an open state. Resetting the connection string resets all connection string values, including the password. Recognized values are true, false, yes, and no.

Name	Default	Description
ServerRSAPublicKeyFile, Server RSA Public Key File		For <code>sha256_password</code> authentication. See comments under <code>AllowPublicKeyRetrieval</code> .
ServerSPN, Server SPN		For MariaDB <code>auth_gssapi_client</code> authentication. Specifies the server's Service Principal Name (to verify that authentication is occurring with the correct server).
Treat Tiny As Boolean, TreatTinyAsBoolean	true	When set to <code>true</code> , <code>TINYINT(1)</code> values are returned as booleans. Setting this to <code>false</code> causes <code>TINYINT(1)</code> to be returned as sbyte/byte.
Use Affected Rows, UseAffectedRows	false	When <code>false</code> (default), the connection reports found rows instead of changed (affected) rows. Set to <code>true</code> to report only the number of rows actually changed by <code>UPDATE</code> or <code>INSERT ... ON DUPLICATE KEY UPDATE</code> statements.
Use XA Transactions, UseXaTransactions	true	When <code>true</code> (default), using <code>TransactionScope</code> or <code>MySQLConnection.EnlistTransaction</code> will use a XA Transaction . This allows true distributed transactions, but may not be compatible with server replication; there are other limitations . When set to <code>false</code> , regular MySQL transactions are used, just like Connector/NET.

Unsupported Options

These options are used by Connector/NET but not supported by MySqlConnection. In general, they should be removed from your connection string when migrating from Connector/NET to MySqlConnection.

Name	Default	Description
AllowBatch, Allow Batch	true	MySqlConnection always allows batch statements.
CheckParameters, Check Parameters	true	MySqlConnection always checks stored procedure parameters efficiently; there's no need to disable this.
CommandInterceptors, Command Interceptors		MySqlConnection doesn't support this extensibility mechanism, which is not compatible with async operations.

Name	Default	Description
ExceptionInterceptors, Exception Interceptors		MySQLConnector doesn't support this extensibility mechanism.
FunctionsReturnString, Functions Return String	false	Not supported. BLOBs are always returned as <code>byte[]</code> .
IncludeSecurityAsserts, Include Security Asserts	false	Not supported. For partial trust environments.
IntegratedSecurity, Integrated Security	false	Windows authentication is not supported.
Logging	false	Use MySQLConnector logging (which is more flexible) instead.
OldSyntax, Old Syntax, UseOldSyntax, Use Old Syntax	false	This option is deprecated in Connector/NET and unsupported in MySQLConnector.
ProcedureCacheSize, Procedure Cache Size, ProcedureCache, Procedure Cache		MySQLConnector places no limit on the amount of stored procedure metadata that is cached. It takes a very small amount of memory.
RespectBinaryFlags, Respect Binary Flags	true	The binary type of a column is always respected.
SharedMemoryName, Shared Memory Name	true	Shared memory (on Windows) is not supported as a connection protocol.
SqlServerMode, Sql Server Mode	false	Not supported.
TreatBlobsAsUtf8, Treat BLOBs as UTF8	false	Not supported. BLOBs are always returned as <code>byte[]</code> .
UsePerformanceMonitor, Use Performance Monitor, UserPerfMon, PerfMon	false	Not supported.
UseUsageAdvisor, Use Usage Advisor, Usage Advisor	false	Not supported.

HOME

GETTING STARTED

[Installing](#)

[DbProviderFactories](#)

[Known Issues](#)

[Version History](#)

[Configuration](#)

[Logging](#)

[Use with ORMs](#)

[Using AddWithValue](#)

CONNECTION OPTIONS

TUTORIALS

[Connect to MySQL](#)

[Basic API](#)

[Best Practices](#)

[Connecting with SSH](#)

[Migrating from Connector/NET](#)

[Use with ASP.NET Core](#)

API

[MySqlBatch](#)

[MySqlBulkCopy](#)

[MySqlCommand](#)

[MySqlConnection](#)

[MySqlDataReader](#)

[MySqlTransaction](#)

TROUBLESHOOTING

[Connection Reuse](#)

[DateTime Storage](#)

[Transaction Usage](#)

[Load Data Local Infile](#)

[Retrieval of Public Key](#)