MySqlConnection.ConnectionString Property

MySqlConnection.ConnectionString Property

Gets or sets the string used to connect to a MySQL Server database.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.10.9

◢Syntax

◢Remarks

The ConnectionString returned may not be exactly like what was originally set but will be indentical in terms of keyword/value pairs. Security information will not be included unless the Persist Security Info value is set to true.

You can use the ConnectionString property to connect to a database. The following example illustrates a typical connection string.

"Persist Security Info=False;database=MyDB;server=MySqlServer;user id=myUser;Password=myPass"

The ConnectionString property can be set only when the connection is closed. Many of the connection string values have corresponding read-only properties. When the connection string is set, all of these properties are updated, except when an error is detected. In this case, none of the properties are updated. MySqlConnection properties return only those settings contained in the ConnectionString.

To connect to a local machine, specify "localhost" for the server. If you do not specify a server, localhost is assumed.

Resetting the ConnectionString on a closed connection resets all connection string values (and related properties) including the password. For example, if you set a connection string that includes "Database= MyDb", and then reset the connection string to "Data Source=myserver;User Id=myUser;Password=myPass", the Database property is no longer set to MyDb.

The connection string is parsed immediately after being set. If errors in syntax are found when parsing, a runtime exception, such as ArgumentException, is generated. Other errors can be found only when an attempt is made to open the connection.

The basic format of a connection string consists of a series of keyword/value pairs separated by semicolons. The equal sign (=) connects each keyword and its value. To include values that contain a semicolon, single-quote character, or double-quote character, the value must be enclosed in double quotes. If the value contains both a semicolon and a double-quote character, the value can be enclosed in single quotes. The single quote is also useful if the value begins with a double-quote character. Conversely, the double quote can be used if the value begins with a single quote. If the value contains both single-quote and double-quote characters, the quote character used to enclose the value must be doubled each time it occurs within the value.

To include preceding or trailing spaces in the string value, the value must be enclosed in either single quotes or double quotes. Any leading or trailing spaces around integer, Boolean, or enumerated values are ignored, even if enclosed in quotes. However, spaces within a string literal keyword or value are preserved. Using .NET Framework version 1.1, single or double quotes may be used within a connection string without using delimiters (for example, Data Source= my'Server or Data Source= my"Server), unless a quote character is the first or last character in the value.

To include an equal sign (=) in a keyword or value, it must be preceded by another equal sign. For example, in the hypothetical connection string

"key==word=value"

the keyword is "key=word" and the value is "value".

If a specific keyword in a keyword= value pair occurs multiple times in a connection string, the last occurrence listed is used in the value set.

Keywords are not case sensitive.

The following table lists the valid names for keyword values within the ConnectionString.

Name	Default	Description
Connect Timeout -or- Connection Timeout	15	The length of time (in seconds) to wait for a connection to the server before terminating the attempt and generating an error.
Host -or- Server -or- Data Source -or- DataSource -or-	localhost	The name or network address of the instance of MySQL to which to connect. Multiple hosts can be specified separated by &. This can be useful where multiple MySQL servers are configured for replication and you are not concerned about the precise server you are connecting to. No attempt is made by the provider to synchronize writes to the database so care should be taken when using this option. In Unix environment with Mono, this can be a fully qualified path to MySQL socket filename. With this configuration, the Unix socket will be used instead of TCP/IP socket. Currently only a single socket name can be given so accessing MySQL in a replicated environment using Unix sockets is not currently supported.

Address -or- Addr -or- Network Address		
Port	3306	The port MySQL is using to listen for connections. This value is ignored if the connection protocol is anything but socket.
Protocol	socket	Specifies the type of connection to make to the server. Values can be: socket or tcp for a socket connection pipe for a named pipe connection unix for a Unix socket connection memory to use MySQL shared memory
CharSet -or Character Set		Specifies the character set that should be used to encode all queries sent to the server. Resultsets are still returned in the character set of the data returned.
Logging	false	When true, various pieces of information is output to any configured TraceListeners.
Allow Batch	true	When true, multiple SQL statements can be sent with one command execution. -Note- Starting with MySQL 4.1.1, batch statements should be separated by the server-defined seperator character. Commands sent to earlier versions of MySQL should be seperated with ';'.
Encrypt	false	When true, SSL/TLS encryption is used for all data sent between the client and server if the server has a certificate installed. Recognized values are true, false, yes, and no.
Initial Catalog -or- Database	mysql	The name of the database to use intially
Password -or- pwd		The password for the MySQL account being used.
Persist Security Info	false	When set to false or no (strongly recommended), security-sensitive information, such as the password, is not returned as part of the connection 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.
User Id -or- Username -or- Uid -or- User name		The MySQL login account being used.
Shared Memory Name	MYSQL	The name of the shared memory object to use for communication if the connection protocol is set to memory.
Allow Zero Datetime	false	True to have MySqlDataReader.GetValue() return a MySqlDateTime for date or datetime columns that have illegal values. False will cause a DateTime object to be returned for legal values and an exception will be thrown for illegal values.
Convert Zero Datetime	false	True to have MySqlDataReader.GetValue() and MySqlDataReader.GetDateTime() return DateTime.MinValue for date or datetime columns that have illegal values.
Pipe Name -or- Pipe	mysql	When set to the name of a named pipe, the MySqlConnection will attempt to connect to MySQL on that named pipe. This settings only applies to the Windows platform.
Use Performance Monitor	false	Posts performance data that can be tracked using perfmon

-or- UsePerformanceMonitor		
Procedure Cache Size	25	How many stored procedure definitions can be held in the cache
Ignore Prepare	true	Instructs the provider to ignore any attempts to prepare commands. This option was added to allow a user to disable prepared statements in an entire application without modifying the code. A user might want to do this if errors or bugs are encountered with MySQL prepared statements.
Use Procedure Bodies	true	Instructs the provider to attempt to call the procedure without first resolving the metadata. This is useful in situations where the calling user does not have access to the mysql.proc table. To use this mode, the parameters for the procedure must be added to the command in the same order as they appear in the procedure definition and their types must be explicitly set.
Auto Enlist	true	Indicates whether the connection should automatically enlist in the current transaction, if there is one.
Respect Binary Flags	true	Indicates whether the connection should respect all binary flags sent to the client as part of column metadata. False will cause the connector to behave like Connector/Net 5.0 and earlier.
BlobAsUTF8IncludePattern	null	Pattern that should be used to indicate which blob columns should be treated as UTF-8.
BlobAsUTF8ExcludePattern	null	Pattern that should be used to indicate which blob columns should not be treated as UTF-8.
Default Command Timeout	30	The default timeout that new MySqlCommand objects will use unless changed.
Allow User Variables	false	Should the provider expect user variables in the SQL.
Interactive -or- Interactive Session	false	Should this session be considered interactive?
Functions Return String	false	Set this option to true to force the return value of SQL functions to be string.
Use Affected Rows	false	Set this option to true to cause the affected rows reported to reflect only the rows that are actually changed. By default, the number of rows that are matched is returned.

The following table lists the valid names for connection pooling values within the ConnectionString. For more information about connection pooling, see Connection Pooling for the MySql Data Provider.

Name	Default	Description
Connection Lifetime	0	When a connection is returned to the pool, its creation time is compared with the current time, and the connection is destroyed if that time span (in seconds) exceeds the value specified by Connection Lifetime. This is useful in clustered configurations to force load balancing between a running server and a server just brought online. A value of zero (0) causes pooled connections to have the maximum connection timeout.
Max Pool Size	100	The maximum number of connections allowed in the pool.
Min Pool Size	0	The minimum number of connections allowed in the pool.
Pooling	true	When true, the MySqlConnection object is drawn from the appropriate pool, or if necessary, is created and added to the appropriate pool. Recognized values are true, false, yes, and no.
Connection Reset	false	Specifies whether the database connection should be reset when being drawn from the pool. Leaving this as **false** will yeild much faster connection opens but the user should understand the side effects of doing this such as temporary tables and user variables from the previous session not being cleared out.
Cache Server Properties	false	Specifies whether the server variables are cached between pooled connections. On systems where the variables change infrequently and there are lots of connection attempts, this can speed up things dramatically.

When setting keyword or connection pooling values that require a Boolean value, you can use 'yes' instead of 'true', and 'no' instead of 'false'.

Note The MySql Data Provider uses the native socket protocol to communicate with MySQL. Therefore, it does not support the use of an ODBC data source name (DSN) when connecting to MySQL because it does not add an ODBC layer.

CAUTION In this release, the application should use caution when constructing a connection string based on user input (for example when retrieving user ID and password information from a dialog box, and appending it to the connection string). The application should ensure that a user cannot embed extra connection string parameters in these values (for example, entering a password as "validpassword;database=somedb" in an attempt to attach to a different database).

◢Examples

The following example creates a MySqlConnection and sets some of its properties

```
public void CreateConnection()
{
MySqlConnection myConnection = new MySqlConnection();
myConnection.ConnectionString = "Persist Security Info=False;database=myDB;server=myHost;Connect Timeout=30;user id=myUser; pwd=myPass";
myConnection.Open();
}
```

◢Examples

The following example creates a MySqlConnection in Unix environment with Mono installed. MySQL socket filename used in this example is "/var/lib/mysql/mysql.sock". The actual filename depends on your MySQL configuration.

```
public void CreateConnection()
{
MySqlConnection myConnection = new MySqlConnection();
myConnection.ConnectionString = "database=myDB;server=/var/lib/mysql/mysql.sock;user id=myUser; pwd=myPass";
myConnection.Open();
}
```

◢See Also