MySQL Connector/.NET

Connector/.NET enables developers to easily create .NET applications that require secure, high-performance data connectivity with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET aware tools. Developers can build applications using their choice of .NET languages. Connector/.NET is a fully managed ADO.NET driver written in 100% pure C#.

Connector/.NET includes full support for:

- Features provided by MySQL Server up to and including MySQL Server version 5.4.
- Large-packet support for sending and receiving rows and BLOBs up to 2 gigabytes in size.
- Protocol compression which allows for compressing the data stream between the client and server.
- Support for connecting using TCP/IP sockets, named pipes, or shared memory on Windows.
- Support for connecting using TCP/IP sockets or Unix sockets on Unix.
- Support for the Open Source Mono framework developed by Novell.
- Fully managed, does not utilize the MySQL client library.

This document is intended as a user's guide to Connector/.NET and includes a full syntax reference. Syntax information is also included within the Documentation.chm file included with the Connector/.NET distribution.

If you are using MySQL 5.0 or later, and Visual Studio as your development environment, you may want also want to use the MySQL Visual Studio Plugin. The plugin acts as a DDEX (Data Designer Extensibility) provider, enabling you to use the data design tools within Visual Studio to manipulate the schema and objects within a MySQL database. For more information, see
Visual Studio Integration.

Note:

Connector.NET 5.1.2 and later include the Visual Studio Plugin by default.

Key topics:

- For connection string properties when using the class, see Connection Options.

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Making a connection

Once the connector is installed, you can use it to create, modify, and delete connections to MySQL databases. To create a connection with a MySQL database, perform the following steps:

- Start Visual Studio, and open the Server Explorer window (View, Server Explorer option in the main Visual Studio menu, or +, hot keys).

- Right-click on the Data Connections node, and choose the Add Connection... menu item.

- Add Connection dialog opens. Press the Change button to choose MySQL Database as a data source.

Add Connection Context Menu

- Change Data Source dialog opens. Choose MySQL Database in the list of data sources (or the option, if MySQL Database is absent), and then choose .NET Framework Data Provider for MySQL in the combo box of data providers.

Choose Data Source
• Input the connection settings: the server host name (for example, localhost if the MySQL server is installed on the local machine), the user name, the password, and the default schema name. Note that you must specify the default schema name to open the connection.

Add Connection Dialog

• You can also set the port to connect with the MySQL server by pressing the Advanced button. To test connection with the MySQL server, set the server host name, the user name, and the password, and press the Test Connection button. If the test succeeds, the success confirmation dialog opens.
After you set all settings and test the connection, press OK. The newly created connection is displayed in Server Explorer. Now you can work with the MySQL server through standard Server Explorer GUI.

New Data Connection

After the connection is successfully established, all settings are saved for future use. When you start Visual Studio for the next time, just open the connection node in Server Explorer to establish a connection to the MySQL server again.

To modify and delete a connection, use the Server Explorer context menu for the corresponding node. You can modify any of the settings just by overwriting the existing values with new ones. Note that the connection may be modified or deleted only if no active editor for its objects is opened: otherwise you may loose your data.
# Editing Tables

Connector/Net contains a table editor, which enables the visual creation and modification of tables.

The Table Designer can be accessed through a mouse action on table-type node of Server Explorer. To create a new table, right-click on the node (under the connection node) and choose the Create Table command from the context menu.

To modify an existing table, double-click on the node of the table you wish to modify, or right-click on this node and choose the Design item from the context menu. Either of the commands opens the Table Designer.

The table editor is implemented in the manner of the well-known Query Browser Table Editor, but with minor differences.

Editing New Table
Table Designer consists of the following parts:

- **Columns Editor** - a data grid on top of the Table Designer. Use the Columns grid for column creation, modification, and deletion.

- **Indexes tab** - a tab on bottom of the Table Designer. Use the Indexes tab for indexes management.

- **Foreign Keys tab** - a tab on bottom of the Table Designer. Use the Foreign Keys tab for foreign keys management.

- **Column Details tab** - a tab on bottom of the Table Designer. Use the Column Details tab to set advanced column options.

- **Properties window** - a standard Visual Studio Properties window, where the properties of the edited table are displayed. Use the Properties window to set the table properties.
Each of these areas is discussed in more detail in subsequent sections.

To save changes you have made in the Table Designer, use either Save or Save All button of the Visual Studio main toolbar, or just press +. If you have not already named the table you will be prompted to do so.

Choose Table Name

Once created you can view the table in the Server Explorer.

Newly Created Table

The Table Designer main menu allows you to set a Primary Key column, edit Relationships such as Foreign Keys, and create Indexes.

Table Designer Main Menu
You can use the Column Editor to set or change the name, data type, default value, and other properties of a table column. To set the focus to a needed cell of a grid, use the mouse click. Also you can move through the grid using and + keys.

To set or change the name, data type, default value and comment of a column, activate the appropriate cell and type the desired value.

To set or unset flag-type column properties (NOT NULL, auto incremented, flags), check or uncheck the corresponding check boxes. Note that the set of column flags depends on its data type.

To reorder columns, index columns or foreign key columns in the Column Editor, select the whole column you wish to reorder by clicking on the selector column on the left of the column grid. Then move the column by using + (to move the column up) or + (to move the column down) keys.

To delete a column, select it by clicking on the selector column on the left of the column grid, then press the button on a keyboard.
Editing Indexes

Indexes management is performed via the dialog.

To add an index, select Table Designer, Indexes/Keys... from the main menu, and click Add to add a new index. You can then set the index name, index kind, index type, and a set of index columns.

Indexes Dialog

To remove an index, select it in the list box on the left, and click the Delete button.

To change index settings, select the needed index in the list box on the left. The detailed information about the index is displayed in the panel on the right hand side. Change the desired values.
Editing Foreign Keys

Foreign Keys management is performed via the dialog.

To add a foreign key, select Table Designer, Relationships... from the main menu. This displays the dialog. Click Add. You can then set the foreign key name, referenced table name, foreign key columns, and actions upon update and delete.

To remove a foreign key, select it in the list box on the left, and click the Delete button.

To change foreign key settings, select the required foreign key in the list box on the left. The detailed information about the foreign key is displayed in the right hand panel. Change the desired values.

Foreign Key Relationships Dialog
Column Properties

The tab can be used to set column options. In addition to the general column properties presented in the Column Editor, in the tab you can set additional properties such as Character Set, Collation and Precision.
Table Properties

To bring up Table Properties select the table and right click to activate the context menu. Select Properties. The dockable window will be displayed.

Table Properties Menu Item

The following table properties can be set:

- Auto Increment
- Average Row Length
- Character Set
- Collation
- Comment
- Data Directory
- Index Directory
- Maximum Rows
- Minimum Rows
- Name
- Row Format
- Schema
- Storage Engine

The property Schema is read only.

Table Properties
Editing Views

To create a new view, right click the Views node under the connection node in Server Explorer. From the node's context menu, choose the Create View command. This command opens the SQL Editor.

Editing View SQL

You can then enter the SQL for your view.

View SQL Added

To modify an existing view, double click on a node of the view you wish to modify, or right click on this node and choose the Alter View command from a context menu. Either of the commands opens the SQL Editor.

All other view properties can be set in the Properties window. These properties are:

- Catalog
- Check Option
- Definer
- Definition
- Definer
- Is Updateable
- Name
- Schema
- Security Type

Some of these properties can have arbitrary text values, others accept values from a predefined set. In the latter case you set the desired value with an embedded combobox.

The properties Is Updatable and Schema are readonly.

To save changes you have made, use either Save or Save All buttons of the Visual Studio main toolbar, or just press +.

View SQL Saved
Editing Stored Procedures and Functions

To create a new stored procedure, right-click on the node under the connection node in Server Explorer. From the node's context menu, choose the command. This command opens the SQL Editor.

Edit Stored Procedure SQL

```
CREATE PROCEDURE StoredProcedure1
/*
   
   @parameter1 INT
   OUT parameter2 datatype
   /*
BEGIN
END
```

To create a new stored function, right-click on the node under the connection node in Server Explorer. From the node's context menu, choose the command.

To modify an existing stored routine (procedure or function), double-click on the node of the routine you wish to modify, or right-click on this node and choose the command from the context menu. Either of the commands opens the SQL Editor.

To create or alter the routine definition using SQL Editor, type this definition in the SQL Editor using standard SQL. All other routine properties can be set in the Properties window. These properties are:

- Body
- Catalog
- Comment
- Creation Time
Some of these properties can have arbitrary text values, others accept values from a predefined set. In the latter case set the desired value using the embedded combo box.

You can also set all the options directly in the SQL Editor, using the standard CREATE PROCEDURE or CREATE FUNCTION statement. However, it is recommended to use the Properties window instead.

To save changes you have made, use either or buttons of the Visual Studio main
toolbar, or just press +.

Stored Procedure SQL Saved
Editing Triggers

To create a new trigger, right-click on the node of the table, for which you wish to add a trigger. From the node's context menu, choose the command. This command opens the SQL Editor.

To modify an existing trigger, double-click on the node of the trigger you wish to modify, or right-click on this node and choose the command from the context menu. Either of the commands opens the SQL Editor.

To create or alter the trigger definition using SQL Editor, type the trigger statement in the SQL Editor using standard SQL.

**Note:**

You should enter only the trigger statement, that is, the part of the CREATE TRIGGER query that is placed after the FOR EACH ROW clause.

All other trigger properties are set in the Properties window. These properties are:

- Definer
- Event Manipulation
- Name
- Timing

Some of these properties can have arbitrary text values, others accept values from a predefined set. In the latter case set the desired value using the embedded combo box.

The properties Event Table, Schema, and Server in the Properties window are read only.
To save changes you have made, use either Save or Save All buttons of the Visual Studio main toolbar, or just press +. Before changes are saved, you will be asked to confirm the execution of the corresponding SQL query in a confirmation dialog.
Editing User Defined Functions (UDF)

To create a new User Defined Function (UDF), right-click on the node under the connection node in Server Explorer. From the node's context menu, choose the Create UDF command. This command opens the UDF Editor.

To modify an existing UDF, double-click on the node of the UDF you wish to modify, or right-click on this node and choose the Alter UDF command from the context menu. Either of the commands opens the UDF Editor.

The UDF editor allows you to set the following properties:

- Name
- So-name (DLL name)
- Return type
- Is Aggregate

There are text fields for both names, a combo box for the return type, and a check box to indicate if the UDF is aggregate. All these options are also accessible via the Properties window.

The property Server in the Properties window is read only.

To save changes you have made, use either Save or Save All buttons of the Visual Studio main toolbar, or just press +. Before changes are saved, you will be asked to confirm the execution of the corresponding SQL query in a confirmation dialog.
Cloning Database Objects

Tables, views, stored procedures, and functions can be cloned using the appropriate Clone command from the context menu: Clone Table, Clone View, Clone Routine. The clone commands open the corresponding editor for a new object: the for cloning a table, and the for cloning a view or a routine.

The editor is filled with values of the original object. You can modify these values in a usual manner.

To save the cloned object, use either Save or Save All buttons of the Visual Studio main toolbar, or just press +. Before changes are saved, you will be asked to confirm the execution of the corresponding SQL query in a confirmation dialog.
Dropping Database Objects

Tables, views, stored routines, triggers, and UDFs can be dropped with the appropriate Drop command selected from its context menu: Drop Table, Drop View, Drop Routine, Drop Trigger, Drop UDF.

You will be asked to confirm the execution of the corresponding drop query in a confirmation dialog.

Dropping of multiple objects is not supported.
Using the ADO.NET Entity Framework

Connector/NET 6.0 introduced support for the ADO.NET Entity Framework. ADO.NET Entity Framework was included with .NET Framework 3.5 Service Pack 1, and Visual Studio 2008 Service Pack 1. ADO.NET Entity Framework was released on 11th August 2008.

ADO.NET Entity Framework provides an Object Relational Mapping (ORM) service, mapping the relational database schema to objects. The ADO.NET Entity Framework defines several layers, these can be summarized as:

- **Logical** - this layer defines the relational data and is defined by the Store Schema Definition Language (SSDL).

- **Conceptual** - this layer defines the .NET classes and is defined by the Conceptual Schema Definition Language (CSDL)

- **Mapping** - this layer defines the mapping from .NET classes to relational tables and associations, and is defined by Mapping Specification Language (MSL).

Connector/NET integrates with Visual Studio 2008 to provide a range of helpful tools to assist the developer.

A full treatment of ADO.NET Entity Framework is beyond the scope of this manual. You are encouraged to review the

Microsoft ADO.NET Entity Framework documentation.

MySQL Website Configuration Tool

6.1 introduced the MySQL Website Configuration Tool. This is a facility available in Visual Studio that allows you to configure the Membership, Role, Session State and Profile Provider, without having to resort to editing configuration files. You simply run the tool, set your configuration options, and the tool will modify your `web.config` file accordingly.

The MySQL Website Configuration Tool appears as a small icon on the Solution Explorer toolbar in Visual Studio, as show by the following screenshot:

MySQL Website Configuration Tool

Clicking on the Website Configuration Tool icon launches the wizard and displays the first screen:

MySQL Website Configuration Tool - Membership
This allows you to enable use of the MySQL Membership Provider. Simply click the checkbox to enable this. You can now enter the name of the application that you are creating the configuration for. You can also enter a description for the application.

You can then click the Edit... button to launch the Connection String Editor:

MySQL Website Configuration Tool - Connection String Editor
Note that if you have already defined a connection string for the providers manually in web.config, or previously using the tool, this will be automatically loaded and displayed, and can then be modified in this dialog.

You can also ensure that the necessary schema are created automatically for you by selecting the Autogenerate Schema checkbox. These schema are used to store membership information. The database used to storage is the one specified in the connection string.

You can also ensure that exceptions generated by the application will be written to the event log by selecting the checkbox.

Clicking the Advanced... button launches a dialog that allows you to set Membership Options. These options dictate such variables as password length required when a user signs up, whether the password is encrypted and whether the user can reset their password or not.
MySQL Website Configuration Tool - Advanced Options

Once information has been set up as required for configuration of the Membership Provider the Next button can be clicked to display the Roles Provider screen:

MySQL Website Configuration Tool - Roles
Again the connection string can be edited, a description added and Autogenerate Schema can be enabled before clicking Next to go to the Profiles Provider screen:

MySQL Website Configuration Tool - Profiles
This screen display similar options to the previous screens.

Click Next to proceed to the Session State configuration page:

MySQL Website Configuration Tool - Session State

![Screen shot of the MySQL Website Configuration Tool - Session State]

Once you have set up the Session State Provider as required, click Finish to exit the wizard.

At this point it is necessary to select the Authentication Type to From Internet. This can be done by launching the ASP.NET Configuration Tool, and selecting the Security tab. Click the Select authentication type link and ensure that the From the internet radio button is selected. You can now examine the database you created to store membership information. All the necessary tables will have been created for you:

MySQL Website Configuration Tool - Tables
Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
MySQL Connector/Net
Installation

Send Feedback
Connector/NET Installation

Connector/NET runs on any platform that supports the .NET framework. The .NET framework is primarily supported on recent versions of Microsoft Windows, and is supported on Linux through the Open Source Mono framework (see

http://www.mono-project.com).

Connector/NET is available for download from
http://dev.mysql.com/downloads/connector/net/5.2.html.
Installing Connector/.NET on Windows

On Windows, installation is supported either through a binary installation process or by downloading a Zip file with the Connector/.NET components.

Before installing, you should ensure that your system is up to date, including installing the latest version of the .NET Framework.
Installing Connector/.NET using the Installer

Using the installer is the most straightforward method of installing Connector/.NET on Windows and the installed components include the source code, test code and full reference documentation.

Connector/.NET is installed through the use of a Windows Installer (.msi) installation package, which can be used to install Connector/.NET on all Windows operating systems. The MSI package is contained within a ZIP archive named mysql-connector-net-version.zip, where indicates the Connector/.NET version.

To install Connector/.NET:

1. Double click on the MSI installer file extracted from the Zip you downloaded. Click Next to start the installation.

2. You must choose the type of installation that you want to perform.

   For most situations, the Typical installation will be suitable. Click the Typical button and proceed to Step 5. A Complete installation installs all the available files. To conduct a Complete installation, click the Complete button and proceed to step 5. If you want to customize your installation, including choosing the components to install and some installation options, click the Custom button and proceed to Step 3.

   The Connector/.NET installer will register the connector within the Global Assembly Cache (GAC) - this will make the Connector/.NET component available to all applications, not just those where you explicitly reference the Connector/.NET component. The installer will also create the necessary links in the Start menu to the documentation and release notes.

3. If you have chosen a custom installation, you can select the individual components that you want to install, including the core interface component, supporting documentation (a CHM file) samples and examples
and the source code. Select the items, and their installation level, and then click Next to continue the installation.

**Note:**

For Connector/NET 1.0.8 or lower and Connector 5.0.4 and lower the installer will attempt to install binaries for both 1.x and 2.x of the .NET Framework. If you only have one version of the framework installed, the connector installation may fail. If this happens, you can choose the framework version to be installed through the custom installation step.

4. You will be given a final opportunity to confirm the installation. Click Install to copy and install the files onto your machine.

5. Once the installation has been completed, click Finish to exit the installer.

Unless you choose otherwise, Connector/NET is installed in C:\Program Files\MySQL\MySQL Connector Net X.X.X, where is replaced with the version of Connector/NET you are installing. New installations do not overwrite existing versions of Connector/NET.

Depending on your installation type, the installed components will include some or all of the following components:

- **bin** - Connector/NET MySQL libraries for different versions of the .NET environment.

- **docs** - contains a CHM of the Connector/NET documentation.

- **samples** - sample code and applications that use the Connector/NET component.

- **src** - the source code for the Connector/NET component.

You may also use the /quiet or /q command-line option with the msiexec tool to install the Connector/NET package automatically (using the default options) with no notification to the user. Using this method the user cannot select options. Additionally, no prompts, messages or dialog boxes will be displayed.
C:\> msiexec /package conector-net.msi /quiet

To provide a progress bar to the user during automatic installation, use the /passive option.
Installing Connector/.NET using the Zip packages

If you are having problems running the installer, you can download a Zip file without an installer as an alternative. That file is called mysql-connector-net-version-noinstall.zip. Once downloaded, you can extract the files to a location of your choice.

The file contains the following directories:

- **bin** - Connector/.NET MySQL libraries for different versions of the .NET environment.

- **Docs** - contains a CHM of the Connector/.NET documentation.

- **Samples** - sample code and applications that use the Connector/.NET component.

Connector/.NET 6.0.x has a different directory structure:

- **Assemblies** - contains a collection of DLLs that make up the connector functionality.

- **Documentation** - contains the Connector/.NET documentation as a CHM file.

- **Samples** - sample code and applications that use the Connector/.NET component.

There is also another Zip file available for download called mysql-connector-net-version-src.zip. This file contains the source code distribution.

The file contains the following directories:

- **Documentation** - This folder contains the source files to build the documentation into the compiled HTML (CHM) format.
- **Installer** - This folder contains the source files to build the Connector/NET installer program.

- **MySql.Data** - This folder contains the source files for the core data provider.

- **MySql.VisualStudio** - This folder contains the source files for the Microsoft Visual Studio extensions.

- **MySql.Web** - This folder contains the source files for the web providers. This includes code for the membership provider, role provider and profile provider. These are used in ASP.NET web sites.

- **Samples** - This folder contains the source files for several example applications.

- **Tests** - This folder contains a spreadsheet listing test cases.

- **VisualStudio** - Contains resources used by the Visual Studio plug in.

Finally, you need to ensure that **MySql.Data.dll** is accessible to your program at build time (and run time). If using Microsoft Visual Studio you will need to add MySql.Data as a Reference to your project.
Installing Connector/NET on Unix with Mono

There is no installer available for installing the Connector/NET component on your Unix installation. Before installing, please ensure that you have a working Mono project installation. You can test whether your system has Mono installed by typing:

shell> mono --version

The version of the Mono JIT compiler will be displayed.

To compile C# source code you will also need to make sure a Mono C# compiler, is installed. Note that there are two Mono C# compilers available, mcs, which accesses the 1.0-profile libraries, and gmcs, which accesses the 2.0-profile libraries.

To install Connector/NET on Unix/Mono:

1. Download the mysql-connector-net-version-noinstall.zip and extract the contents to a directory of your choice, for example: ~/connector-net/.

2. In the directory where you unzipped the connector to, change into the bin directory. Ensure the file MySql.Data.dll is present.

3. You must register the Connector/NET component, MySql.Data, in the Global Assembly Cache (GAC). In the current directory enter the gacutil command:

   root-shell> gacutil /i MySql.Data.dll

   This will register MySql.Data into the GAC. You can check this by listing the contents of /usr/lib/mono/gac, where you will find MySql.Data if the registration has been successful.

You are now ready to compile your application. You must ensure that when you
compile your application you include the Connector/NET component using the command-line option. For example:

```
```

Note, the assemblies that need to be referenced will depend on the requirements of the application, but applications using Connector/NET will need to provide as a minimum.

You can further check your installation by running the compiled program, for example:

```
shell> mono HelloWorld.exe
```
Installing Connector/NET from the source code

Obtaining the source code

To obtain the most recent development source tree, you first need to download and install Bazaar. You can obtain Bazaar from the

Bazaar VCS Website. Bazaar is supported by any platform that supports Python, and is therefore compatible with any Linux, Unix, Windows or Mac OS X host. Instructions for downloading and installing Bazaar on the different platforms are available on the Bazaar website.

The most recent development source tree is available from our public Subversion trees at http://dev.mysql.com/tech-resources/sources.html.

To checkout out the Connector/NET sources, change to the directory where you want the copy of the Connector/NET tree to be stored, then use the following command:

shell> bzr branch lp:connectornet/trunk

To download a specific version of Connector/NET, specify the version number instead of trunk. For example, to obtain a copy of the 6.0 version of the source tree:

shell> bzr branch lp:connectornet/6.0

Source packages are also available on the downloads page.

Building the source code on Windows

The following procedure can be used to build the connector on Microsoft Windows.

- Obtain the source code, either from the Subversion server, or through one of the prepared source code packages.
• Navigate to the root of the source code tree.

• A Microsoft Visual Studio 2005 solution file is available to build the connector, this is called **MySQL-VS2005.sln**. Click on this file to load the solution into Visual Studio.

• Select Build, Build Solution from the main menu to build the solution.

**Building the source code on Unix**

Support for building Connector.NET on Mono/Unix is currently not available.

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Connector/NET Programming

Connector/NET comprises several classes that are used to connect to the database, execute queries and statements, and manage query results.

The following are the major classes of Connector/NET:

- **MySqlCommand**: Represents an SQL statement to execute against a MySQL database.
- **MySqlCommandBuilder**: Automatically generates single-table commands used to reconcile changes made to a DataSet with the associated MySQL database.
- **MySqlConnection**: Represents an open connection to a MySQL Server database.
- **MySqlDataAdapter**: Represents a set of data commands and a database connection that are used to fill a data set and update a MySQL database.
- **MySqlDataReader**: Provides a means of reading a forward-only stream of rows from a MySQL database.
- **MySqlException**: The exception that is thrown when MySQL returns an error.
- **MySqlHelper**: Helper class that makes it easier to work with the provider.
- **MySqlTransaction**: Represents an SQL transaction to be made in a MySQL database.

In the following sections you will learn about some common use cases for Connector/NET, including BLOB handling, date handling, and using Connector/NET with common tools such as Crystal Reports.
Connecting to MySQL Using Connector/.NET

Introduction

All interaction between a .NET application and the MySQL server is routed through a MySqlConnection object. Before your application can interact with the server, a MySqlConnection object must be instanced, configured, and opened.

Even when using the MySqlHelper class, a MySqlConnection object is created by the helper class.

In this section, we will describe how to connect to MySQL using the MySqlConnection object.
Creating a Connection String

The MySqlConnection object is configured using a connection string. A connection string contains sever key/value pairs, separated by semicolons. Each key/value pair is joined with an equals sign.

The following is a sample connection string:

Server=127.0.0.1;Uid=root;Pwd=12345;Database=test;

In this example, the MySqlConnection object is configured to connect to a MySQL server at 127.0.0.1, with a user name of root and a password of 12345. The default database for all statements will be the test database.

The following options are available:

⚠️ Note:

Using the '@' symbol for parameters is now the preferred approach although the old pattern of using '?' is still supported.

Please be aware however that using '@' can cause conflicts when user variables are also used. To help with this situation please see the documentation on the Allow User Variables connection string option, which can be found here:

Creating a Connection String. The Old Syntax connection string option has now been deprecated.
## Opening a Connection

Once you have created a connection string it can be used to open a connection to the MySQL server.

The following code is used to create a MySqlConnection object, assign the connection string, and open the connection.

### VB.NET

```vbnet
Dim conn As New MySql.Data.MySqlClient.MySqlConnection
Dim myConnectionString as String
myConnectionString = "server=127.0.0.1;" _
    & "uid=root;" _
    & "pwd=12345;" _
    & "database=test;"

Try
    conn.ConnectionString = myConnectionString
    conn.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    MessageBox.Show(ex.Message)
End Try
```

### C#

```csharp
MySql.Data.MySqlClient.MySqlConnection conn;
string myConnectionString;

myConnectionString = "server=127.0.0.1;uid=root;" +
    "pwd=12345;database=test;";

try
{
    conn = new MySql.Data.MySqlClient.MySqlConnection();
    conn.ConnectionString = myConnectionString;
    conn.Open();
}
catch (MySql.Data.MySqlClient.MySqlException ex)
{
    MessageBox.Show(ex.Message);
}
```
You can also pass the connection string to the constructor of the MySqlConnection class:

**VB.NET**

```vbnet
Dim myConnectionString as String

myConnectionString = "server=127.0.0.1;" & "uid=root;" & "pwd=12345;" & "database=test;"

Try
    Dim conn As New MySql.Data.MySqlClient.MySqlConnection(myConnectionString)
    conn.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    MessageBox.Show(ex.Message)
End Try
```

**C#**

```csharp
MySql.Data.MySqlClient.MySqlConnection conn;
string myConnectionString;

myConnectionString = "server=127.0.0.1;uid=root;" +
    "pwd=12345;database=test;";

try
{
    conn = new MySql.Data.MySqlClient.MySqlConnection(myConnectionString);
    conn.Open();
}
catch (MySql.Data.MySqlClient.MySqlException ex)
{
    MessageBox.Show(ex.Message);
}
```

Once the connection is open it can be used by the other Connector/.NET classes to communicate with the MySQL server.
Handling Connection Errors

Because connecting to an external server is unpredictable, it is important to add error handling to your .NET application. When there is an error connecting, the MySqlConnection class will return a MySqlException object. This object has two properties that are of interest when handling errors:

- **Message**: A message that describes the current exception.
- **Number**: The MySQL error number.

When handling errors, you can tailor your application's response based on the error number. The two most common error numbers when connecting are as follows:

- **0**: Cannot connect to server.
- **1045**: Invalid user name and/or password.

The following code shows how to adapt the application's response based on the actual error:

**VB.NET**

```vbnet
Dim myConnectionString as String

myConnectionString = "server=127.0.0.1;" &_
& "uid=root;" &_
& "pwd=12345;" &_
& "database=test;"

Try
    Dim conn As New MySql.Data.MySqlClient.MySqlConnection(myConnectionString)
    conn.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    Select Case ex.Number
        Case 0
            MessageBox.Show("Cannot connect to server. Contact admin")
        Case 1045
            MessageBox.Show("Invalid username/password, please try again")
    End Select
End Try
```
C#

```csharp
using MySql.Data.MySqlClient;

string myConnectionString;
myConnectionString = "server=127.0.0.1;uid=root;pwd=12345;database=test;";

try
{
    conn = new MySqlConnection(myConnectionString);
    conn.Open();
}

catch (MySqlException ex)
{
    switch (ex.Number)
    {
        case 0:
            MessageBox.Show("Cannot connect to server. Contact administrator");
        case 1045:
            MessageBox.Show("Invalid username/password, please try again");
    }
}
```

**Important Note:**

Note that if you are using multilanguage databases you must specify the character set in the connection string. If you do not specify the character set, the connection defaults to the latin1 charset. You can specify the character set as part of the connection string, for example:

```
C#

using MySql.Data.MySqlClient;

MySqlConnection myConnection = new MySqlConnection("server=127.0.0.1 "pwd=12345;database=test;Charset=latin1;"));
```
Using MySqlCommand

A MySqlCommand has the CommandText and CommandType properties associated with it. The CommandText will be handled differently depending on the setting of CommandType. CommandType can be one of:

1. Text - A SQL text command (default)
2. StoredProcedure - The name of a Stored Procedure
3. TableDirect - The name of a table (new in Connector/NET 6.2)

The default CommandType, Text, is used for executing queries and other SQL commands. Some example of this can be found in the following section

The MySqlCommand Object.

If CommandType is set to StoredProcedure, CommandText should be set to the name of the Stored Procedure to access.

If CommandType is set to TableDirect, all rows and columns of the named table will be returned when you call one of the Execute methods. In effect, this command performs a SELECT * on the table specified. The CommandText property is set to the name of the table you wish to query. This is illustrated by the following code snippet:

C#  

```csharp
...
MySqlCommand cmd = new MySqlCommand();
cmd.CommandText = "mytable";
cmd.Connection = someConnection;
cmd.CommandType = CommandType.TableDirect;
SqlDataReader reader = cmd.ExecuteReader();
while (reader.Read())
{
    Console.WriteLine(reader[0], reader[1]...);
}
...
```
Examples of using the CommandType of StoredProcedure can be found in the section [Calling a Stored Procedure from Connector.NET](#).

Commands can have a timeout associated with them. This is useful as you may not want a situation were a command takes up an excessive amount of time. A timeout can be set using the CommandTimeout property. The following code snippet sets a timeout of one minute:

```csharp
MySqlCommand cmd = new MySqlCommand();
cmd.CommandTimeout = 60;
```

The default value is 30 secs. A value of 0 indicates an indefinite wait and should be avoided. Note the default command timeout can be changed using the connection string option Default Command Timeout.

Prior to 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. This timer consumed a thread.

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. The 6.2 implementation uses the underlying stream timeout facility, so is more efficient in that it does not require the additional timer thread as was the case with the previous implementation.

Further details on this can be found in the relevant [Microsoft documentation](#).
Using Connector/.NET with Connection Pooling

The Connector/.NET supports connection pooling. This is enabled by default, but can be turned off via connection string options. See Creating a Connection String for further information.

Connection pooling works by keeping the native connection to the server live when the client disposes of a MySqlConnection. Subsequently, if a new MySqlConnection object is opened, it will be created from the connection pool, rather than creating a new native connection. This improves performance.

To work as designed, it is best to let the connection pooling system manage all connections. You should not create a globally accessible instance of MySqlConnection and then manually open and close it. This interferes with the way the pooling works and can lead to unpredictable results or even exceptions.

One approach that simplifies things is to avoid manually creating a MySqlConnection object. Instead use the overloaded methods that take a connection string as an argument. Using this approach, Connector/.NET will automatically create, open, close and destroy connections, using the connection pooling system for best performance.

TypedDatasets and the MembershipProvider and RoleProvider classes use this approach. Most classes that have methods that take a MySqlConnection as an argument, also have methods that take a connection string as an argument. This includes MySqlCommandAdapter.

Instead of manually creating MySqlCommand objects, you can use the static methods of the MySqlHelper class. These take a connection string as an argument, and they fully support connection pooling.

Starting with 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.
Using the Connector.NET with Prepared Statements

Introduction

As of MySQL 4.1, it is possible to use prepared statements with Connector.NET. Use of prepared statements can provide significant performance improvements on queries that are executed more than once.

Prepared execution is faster than direct execution for statements executed more than once, primarily because the query is parsed only once. In the case of direct execution, the query is parsed every time it is executed. Prepared execution also can provide a reduction of network traffic because for each execution of the prepared statement, it is necessary only to send the data for the parameters.

Another advantage of prepared statements is that it uses a binary protocol that makes data transfer between client and server more efficient.
Preparing Statements in Connector/.NET

To prepare a statement, create a command object and set the .CommandText property to your query.

After entering your statement, call the .Prepare method of the MySqlCommand object. After the statement is prepared, add parameters for each of the dynamic elements in the query.

After you enter your query and enter parameters, execute the statement using the .ExecuteNonQuery(), .ExecuteScalar(), or .ExecuteReader methods.

For subsequent executions, you need only modify the values of the parameters and call the execute method again, there is no need to set the .CommandText property or redefine the parameters.

VB.NET

```
Dim conn As New MySqlConnection
Dim cmd As New MySqlCommand

conn.ConnectionString = strConnection

Try
   conn.Open()
   cmd.Connection = conn

   cmd.CommandText = "INSERT INTO myTable VALUES(NULL, @number, @text)"
   cmd.Prepare()

   cmd.Parameters.Add("@number", 1)
   cmd.Parameters.Add("@text", "One")

   For i = 1 To 1000
      cmd.Parameters("@number").Value = i
      cmd.Parameters("@text").Value = "A string value"

      cmd.ExecuteNonQuery()
   Next
Catch ex As MySqlException
```

```csharp
    MessageBox.Show("Error " & ex.Number & " has occurred: " & ex.Message,
End Try

C#

using MySql.Data.MySqlClient;

MySqlConnection conn;
MySqlCommand cmd;

conn = new MySqlConnection();
conn.ConnectionString = strConnection;

try{
  conn.Open();
  cmd.Connection = conn;

  cmd.CommandText = "INSERT INTO myTable VALUES(NULL, @number, @text)"
  cmd.Prepare();

  cmd.Parameters.Add("@number", 1);
  cmd.Parameters.Add("@text", "One");

  for (int i=1; i <= 1000; i++)
  {
    cmd.Parameters["@number"].Value = i;
    cmd.Parameters["@text"].Value = "A string value"

    cmd.ExecuteNonQuery();
  }
}
catch (MySqlException ex)
{
  MessageBox.Show("Error " + ex.Number + " has occurred: " + ex.Message,
                 MessageBoxButtons.OK, MessageBoxIcon.Error);
}
Accessing Stored Procedures with Connector/NET

Introduction

With the release of MySQL version 5 the MySQL server now supports stored procedures with the SQL 2003 stored procedure syntax.

A stored procedure is a set of SQL statements that can be stored in the server. Once this has been done, clients do not need to keep reissuing the individual statements but can refer to the stored procedure instead.

Stored procedures can be particularly useful in situations such as the following:

- When multiple client applications are written in different languages or work on different platforms, but need to perform the same database operations.

- When security is paramount. Banks, for example, use stored procedures for all common operations. This provides a consistent and secure environment, and procedures can ensure that each operation is properly logged. In such a setup, applications and users would not get any access to the database tables directly, but can only execute specific stored procedures.

Connector/NET supports the calling of stored procedures through the MySqlCommand object. Data can be passed in and our of a MySQL stored procedure through use of the MySqlCommand.Parameters collection.

Note:

When you call a stored procedure, the command object makes an additional SELECT call to determine the parameters of the stored procedure. You must ensure that the user calling the procedure has the SELECT privilege on the mysql.proc table to enable them to verify the parameters. Failure to do this will result in an error when calling the procedure.
This section will not provide in-depth information on creating Stored Procedures. For such information, please refer to


A sample application demonstrating how to use stored procedures with Connector/NET can be found in the Samples directory of your Connector/NET installation.
Creating Stored Procedures from Connector.NET

Stored procedures in MySQL can be created using a variety of tools. First, stored procedures can be created using the command-line client. Second, stored procedures can be created using the MySQL Query Browser GUI client. Finally, stored procedures can be created using the .ExecuteNonQuery method of the MySqlCommand object:

**VB.NET**

```vbnet
Dim conn As New MySqlConnection
Dim cmd As New MySqlCommand

conn.ConnectionString = "server=127.0.0.1;uid=root;pwd=12345;database=test"

Try
    conn.Open()
    cmd.Connection = conn

    cmd.CommandText = "CREATE PROCEDURE add_emp(" +
    & "IN fname VARCHAR(20), IN lname VARCHAR(20), IN bday DATET +
    & "BEGIN INSERT INTO emp(first_name, last_name, birthdate) +
    & "VALUES(fname, lname, DATE(bday)); SET empno = LAST_INSERT +
    
    cmd.ExecuteNonQuery()
Catch ex As MySqlException
    MessageBox.Show("Error " & ex.Number & " has occurred: " & ex.Me +
End Try
```

**C#**

```csharp
using MySql.Data.MySqlClient;

MySqlConnection conn;
MySqlCommand cmd;

conn = new MySqlConnection();
cmd = new MySqlCommand();

conn.ConnectionString = "server=127.0.0.1;uid=root;" +
```
"pwd=12345;database=test;",

try
{
    conn.Open();
    cmd.Connection = conn;

    cmd.CommandText = "CREATE PROCEDURE add_emp("
        "IN fname VARCHAR(20), IN lname VARCHAR(20), IN bday DATETIME
        "BEGIN INSERT INTO emp(first_name, last_name, birthdate) " +
        "VALUES(fname, lname, DATE(bday)); SET empno = LAST_INSERT_ID());
    cmd.ExecuteNonQuery();
}

It should be noted that, unlike the command-line and GUI clients, you are not required to specify a special delimiter when creating stored procedures in Connector.NET.
Calling a Stored Procedure from Connector/NET

To call a stored procedure using Connector/NET, create a MySqlCommand object and pass the stored procedure name as the .CommandText property. Set the .CommandType property to CommandType.StoredProcedure.

After the stored procedure is named, create one MySqlCommand parameter for every parameter in the stored procedure. IN parameters are defined with the parameter name and the object containing the value, OUT parameters are defined with the parameter name and the datatype that is expected to be returned. All parameters need the parameter direction defined.

After defining parameters, call the stored procedure by using the MySqlCommand.ExecuteNonQuery() method:

**VB.NET**

```vbnet
Dim conn As New MySqlConnection
Dim cmd As New MySqlCommand

conn.ConnectionString = "server=127.0.0.1;_ 
& "uid=root;" _ 
& "pwd=12345;" _ 
& "database=test"

Try
    conn.Open()
    cmd.Connection = conn

    cmd.CommandText = "add_emp"
    cmd.CommandType = CommandType.StoredProcedure

    cmd.Parameters.Add("@lname", 'Jones')
    cmd.Parameters("@lname").Direction = ParameterDirection.Input

    cmd.Parameters.Add("@fname", 'Tom')
    cmd.Parameters("@fname").Direction = ParameterDirection.Input

    cmd.Parameters.Add("@bday", #12/13/1977 2:17:36 PM#)
    cmd.Parameters("@bday").Direction = ParameterDirection.Input

    cmd.ExecuteNonQuery()
End Try
```
```csharp
conn = new MySql.Data.MySqlClient.MySqlConnection();
conn.ConnectionString = "server=127.0.0.1;uid=root;pwd=12345;database=test;"
try {
    conn.Open();
    cmd.Connection = conn;
    cmd.CommandText = "add_emp";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add("@fname", "Tom");
    cmd.Parameters["@fname"].Direction = ParameterDirection.Input;
    cmd.Parameters["@bday"].Direction = ParameterDirection.Input;
    cmd.ExecuteNonQuery();
    MessageBox.Show(cmd.Parameters["@empno"].Value);
} catch (MySql.Data.MySqlClient.MySqlException ex) {
    MessageBox.Show("Error " + ex.Number + " has occurred: " + ex.Message,
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
```
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Once the stored procedure is called, the values of output parameters can be retrieved by using the .Value property of the MySqlConnector.Parameters collection.
Handling BLOB Data With Connector/.NET

Introduction

One common use for MySQL is the storage of binary data in BLOB columns. MySQL supports four different BLOB datatypes: TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB.

Data stored in a BLOB column can be accessed using Connector/.NET and manipulated using client-side code. There are no special requirements for using Connector/.NET with BLOB data.

Simple code examples will be presented within this section, and a full sample application can be found in the Samples directory of the Connector/.NET installation.
Preparing the MySQL Server

The first step is using MySQL with BLOB data is to configure the server. Let's start by creating a table to be accessed. In my file tables, I usually have four columns: an AUTO_INCREMENT column of appropriate size (UNSIGNED SMALLINT) to serve as a primary key to identify the file, a VARCHAR column that stores the file name, an UNSIGNED MEDIUMINT column that stores the size of the file, and a MEDIUMBLOB column that stores the file itself. For this example, I will use the following table definition:

```c#
CREATE TABLE file(
    file_id SMALLINT UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    file_name VARCHAR(64) NOT NULL,
    file_size MEDIUMINT UNSIGNED NOT NULL,
    file MEDIUMBLOB NOT NULL);
```

After creating a table, you may need to modify the max_allowed_packet system variable. This variable determines how large of a packet (that is, a single row) can be sent to the MySQL server. By default, the server will only accept a maximum size of 1MB from our client application. If you do not intend to exceed 1MB, this should be fine. If you do intend to exceed 1MB in your file transfers, this number has to be increased.

The max_allowed_packet option can be modified using MySQL Administrator's Startup Variables screen. Adjust the Maximum allowed option in the Memory section of the Networking tab to an appropriate setting. After adjusting the value, click the Apply Changes button and restart the server using the Service Control screen of MySQL Administrator. You can also adjust this value directly in the my.cnf file (add a line that reads max_allowed_packet=xxM), or use the SET max_allowed_packet=xxM; syntax from within MySQL.

Try to be conservative when setting max_allowed_packet, as transfers of BLOB data can take some time to complete. Try to set a value that will be adequate for your intended use and increase the value if necessary.
Writing a File to the Database

To write a file to a database we need to convert the file to a byte array, then use the byte array as a parameter to an INSERT query.

The following code opens a file using a FileStream object, reads it into a byte array, and inserts it into the file table:

**VB.NET**

```vbnet
Dim conn As New MySqlConnection
Dim cmd As New MySqlCommand
Dim SQL As String
Dim FileSize As UInt32
Dim rawData() As Byte
Dim fs As FileStream

conn.ConnectionString = "server=127.0.0.1;" _
& "uid=root;" _
& "pwd=12345;" _
& "database=test"

Try
FileSize = fs.Length
    rawData = New Byte(FileSize) {}
    fs.Read(rawData, 0, FileSize)
    fs.Close()
    conn.Open()

    SQL = "INSERT INTO file VALUES(NULL, @FileName, @FileSize, @File"
    cmd.Connection = conn
    cmd.CommandText = SQL
    cmd.Parameters.Add("@FileName", strFileName)
    cmd.Parameters.Add("@FileSize", FileSize)
    cmd.Parameters.Add("@File", rawData)
    cmd.ExecuteNonQuery()
```

MessageBox.Show("File Inserted into database successfully!", _
"Success!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)

conn.Close()
Catch ex As Exception
    MessageBox.Show("There was an error: " & ex.Message, "Error", _
        MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try

C#

```csharp
using MySql.Data.MySqlClient;

MySqlConnection conn;
MySqlCommand cmd;

conn = new MySqlConnection();
cmd = new MySqlCommand();

string SQL;
UInt32 FileSize;
byte[] rawData;
FileStream fs;

conn.ConnectionString = "server=127.0.0.1;uid=root;" +
    "pwd=12345;database=test;";

try{
    fs = new FileStream(@"c:\image.png", FileMode.Open, FileAccess.Read);
    FileSize = fs.Length;
    rawData = new byte[FileSize];
    fs.Read(rawData, 0, FileSize);
    fs.Close();
    conn.Open();
    SQL = "INSERT INTO file VALUES(NULL, @FileName, @FileSize, @File";
    cmd.Connection = conn;
    cmd.CommandText = SQL;
    cmd.Parameters.Add("@FileName", strFileName);
    cmd.Parameters.Add("@FileSize", FileSize);
    cmd.ExecuteNonQuery();
    MessageBox.Show("File Inserted into database successfully!",
        "Success!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
```
conn.Close();
}
catch (MySql.Data.MySqlClient.MySqlException ex)
{
    MessageBox.Show("Error " + ex.Number + " has occurred: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

The Read method of the FileStream object is used to load the file into a byte array which is sized according to the Length property of the FileStream object.

After assigning the byte array as a parameter of the MySqlCommand object, the ExecuteNonQuery method is called and the BLOB is inserted into the file table.
Reading a BLOB from the Database to a File on Disk

Once a file is loaded into the file table, we can use the MySqlDataReader class to retrieve it.

The following code retrieves a row from the file table, then loads the data into a FileStream object to be written to disk:

**VB.NET**

```vbnet
Dim conn As New MySqlConnection
Dim cmd As New MySqlCommand
Dim myData As MySqlDataReader
Dim SQL As String
Dim rawData() As Byte
Dim FileSize As UInteger
Dim fs As FileStream

conn.ConnectionString = "server=127.0.0.1;" 
& "uid=root;" 
& "pwd=12345;" 
& "database=test"

SQL = "SELECT file_name, file_size, file FROM file"

Try
        conn.Open()

        cmd.Connection = conn
        cmd.CommandText = SQL

        myData = cmd.ExecuteReader

        If Not myData.HasRows Then Throw New Exception("There are no BLO"

        myData.Read()

        FileSize = myData.GetUInteger32(myData.GetOrdinal("file_size"))
        rawData = New Byte(FileSize) {}

        myData.GetBytes(myData.GetOrdinal("file"), 0, rawData, 0, FileSi
```
fs = New FileStream("C:\newfile.png", FileMode.OpenOrCreate, FileAccess.Write)
fs.Write(rawData, 0, FileSize)
fs.Close()

MessageBox.Show("File successfully written to disk!", "Success!")

myData.Close()
conn.Close()

Catch ex As Exception
    MessageBox.Show("There was an error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try

C#

MySql.Data.MySqlClient.MySqlConnection conn;
MySql.Data.MySqlClient.MySqlDataReader myData;

conn = new MySql.Data.MySqlClient.MySqlConnection();
myData = cmd.ExecuteReader();
if (!myData.HasRows)
    throw new Exception("There are no BLOBs to save");

myData.Read();

FileSize = myData.GetUInt32(myData.GetOrdinal("file_size"));
rawData = new byte[FileSize];

conn.ConnectionString = "server=127.0.0.1;uid=root;pwd=12345;database=test;";

SQL = "SELECT file_name, file_size, file FROM file";

try
{
    conn.Open();
    cmd.Connection = conn;
    cmd.CommandText = SQL;
    myData = cmd.ExecuteReader();
    if (!myData.HasRows)
        throw new Exception("There are no BLOBs to save");
    myData.Read();
    FileSize = myData.GetUInt32(myData.GetOrdinal("file_size"));
    rawData = new byte[FileSize];
myData.GetBytes(myData.GetOrdinal("file"), 0, rawData, 0, FileSi
fs = new FileStream(@"C:\newfile.png", FileMode.OpenOrCreate, Fis.Write(rawData, 0, FileSize);
fs.Close();

MessageBox.Show("File successfully written to disk!",
    "Success!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

myData.Close();
conn.Close();
}
catch (MySql.Data.MySqlClient.MySqlException ex)
{
    MessageBox.Show("Error " + ex.Number + " has occurred: " + ex.Me
    "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

After connecting, the contents of the file table are loaded into a
MySqlDataReader object. The GetBytes method of the MySqlDataReader is
used to load the BLOB into a byte array, which is then written to disk using a
FileStream object.

The GetOrdinal method of the MySqlDataReader can be used to determine the
integer index of a named column. Use of the GetOrdinal method prevents errors
if the column order of the SELECT query is changed.
Using Connector/.NET with Crystal Reports

Introduction

Crystal Reports is a common tool used by Windows application developers to perform reporting and document generation. In this section we will show how to use Crystal Reports XI with MySQL and Connector/.NET.
Creating a Data Source

When creating a report in Crystal Reports there are two options for accessing the MySQL data while designing your report.

The first option is to use Connector/ODBC as an ADO data source when designing your report. You will be able to browse your database and choose tables and fields using drag and drop to build your report. The disadvantage of this approach is that additional work must be performed within your application to produce a data set that matches the one expected by your report.

The second option is to create a data set in VB.NET and save it as XML. This XML file can then be used to design a report. This works quite well when displaying the report in your application, but is less versatile at design time because you must choose all relevant columns when creating the data set. If you forget a column you must re-create the data set before the column can be added to the report.

The following code can be used to create a data set from a query and write it to disk:

**VB.NET**

```vbnet
Dim myData As New DataSet
Dim conn As New MySqlConnection
Dim cmd As New MySqlCommand
Dim myAdapter As New MySqlDataAdapter

conn.ConnectionString = "server=127.0.0.1;uid=root;pwd=12345;database=world"

Try
    conn.Open()
    cmd.CommandText = "SELECT city.name AS cityName, city.population & "country.name, country.population, country.continent " & "FROM country, city ORDER BY country.continent, country.name"
    cmd.Connection = conn
    myAdapter.SelectCommand = cmd

    ' Execute the command and fill the data set
    myAdapter.Fill(myData)

    ' Save the XML file
    myAdapter.ExportToXml("C:\MyData.xml")
End Try
```
myAdapter.Fill(myData)

myData.WriteXml("C:\dataset.xml", XmlWriteMode.WriteSchema)
Catch ex As Exception
    MessageBox.Show(ex.Message, "Report could not be created", MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try

C#

DataSet myData = new DataSet();
MySql.Data.MySqlClient.MySqlConnection conn;
MySql.Data.MySqlClient.MySqlDataAdapter myAdapter;

conn = new MySql.Data.MySqlClient.MySqlConnection();
conn.ConnectionString = "server=127.0.0.1;uid=root;pwd=12345;database=test;" + 
    "pwsd=12345;database=test;";

try
{
    cmd.CommandText = "SELECT city.name AS cityName, city.population AS CityPopulation, 
        "country.name, country.population, country.continent " + 
        "FROM country, city ORDER BY country.continent, country.name";
    cmd.Connection = conn;

    myAdapter.SelectCommand = cmd;
    myAdapter.Fill(myData);

    myData.WriteXml(@"C:\dataset.xml", XmlWriteMode.WriteSchema);
} catch (MySql.Data.MySqlClient.MySqlException ex)
{
    MessageBox.Show(ex.Message, "Report could not be created", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

The resulting XML file can be used as an ADO.NET XML datasource when designing your report.

If you choose to design your reports using Connector/ODBC, it can be downloaded from

dev.mysql.com.
Creating the Report

For most purposes the Standard Report wizard should help with the initial creation of a report. To start the wizard, open Crystal Reports and choose the New > Standard Report option from the File menu.

The wizard will first prompt you for a data source. If you are using Connector/ODBC as your data source, use the OLEDB provider for ODBC option from the OLE DB (ADO) tree instead of the ODBC (RDO) tree when choosing a data source. If using a saved data set, choose the ADO.NET (XML) option and browse to your saved data set.

The remainder of the report creation process is done automatically by the wizard.

After the report is created, choose the Report Options... entry of the File menu. Un-check the Save Data With Report option. This prevents saved data from interfering with the loading of data within our application.
Displaying the Report

To display a report we first populate a data set with the data needed for the report, then load the report and bind it to the data set. Finally we pass the report to the crViewer control for display to the user.

The following references are needed in a project that displays a report:

- CrystalDecisions.CrystalReports.Engine
- CrystalDecisions.ReportSource
- CrystalDecisions.Shared
- CrystalDecisions.Windows.Forms

The following code assumes that you created your report using a data set saved using the code shown in [Creating a Data Source](#), and have a crViewer control on your form named myViewer.

**VB.NET**

```vbnet
Imports CrystalDecisions.CrystalReports.Engine
Imports System.Data
Imports MySql.Data.MySqlClient

Dim myReport As New ReportDocument
Dim myData As New DataSet
Dim conn As New MySqlConnection
Dim cmd As New MySqlCommand
Dim myAdapter As New MySqlDataAdapter

conn.ConnectionString = _
    "server=127.0.0.1;" _
    & "uid=root;" _
    & "pwd=12345;" _
    & "database=test"

Try
```

conn.Open()

cmd.CommandText = "SELECT city.name AS cityName, city.population & "country.name, country.population, country.continent " _ & "FROM country, city ORDER BY country.continent, country.name"

myAdapter.SelectCommand = cmd
myAdapter.Fill(myData)

myReport.Load("\.\world_report.rpt")
myReport.SetDataSource(myData)
myViewer.ReportSource = myReport

Catch ex As Exception
    MessageBox.Show(ex.Message, "Report could not be created", Messa
End Try

C#

using CrystalDecisions.CrystalReports.Engine;
using System.Data;
using MySql.Data.MySqlClient;

DataSet myData = new DataSet();
MySql.Data.MySqlClient.MySqlConnection conn;
MySql.Data.MySqlClient.MySqlDataAdapter myAdapter;

conn = new MySql.Data.MySqlClient.MySqlConnection();
myAdapter = new MySql.Data.MySqlClient.MySqlDataAdapter();

conn.ConnectionString = "server=127.0.0.1;uid=root;" + 
"pwd=12345;database=test;";

try
{
    cmd.CommandText = "SELECT city.name AS cityName, city.population "country.name, country.population, country.continent " + 
    "FROM country, city ORDER BY country.continent, country.name"
    cmd.Connection = conn;

    myAdapter.SelectCommand = cmd;
    myAdapter.Fill(myData);

    myReport.Load("\.\world_report.rpt");
    myReport.SetDataSource(myData);
    myViewer.ReportSource = myReport;
A new data set is generated using the same query used to generate the previously saved data set. Once the data set is filled, a ReportDocument is used to load the report file and bind it to the data set. The ReportDocument is the passed as the ReportSource of the crViewer.

This same approach is taken when a report is created from a single table using Connector/ODBC. The data set replaces the table used in the report and the report is displayed properly.

When a report is created from multiple tables using Connector/ODBC, a data set with multiple tables must be created in our application. This allows each table in the report data source to be replaced with a report in the data set.

We populate a data set with multiple tables by providing multiple SELECT statements in our MySqlCommand object. These SELECT statements are based on the SQL query shown in Crystal Reports in the Database menu's Show SQL Query option. Assume the following query:

```csharp
C#
ORDER BY `country`.`Continent`, `country`.`Name`, `city`.`Name`
```

This query is converted to two SELECT queries and displayed with the following code:

```vbnet
VB.NET
Imports CrystalDecisions.CrystalReports.Engine
Imports System.Data
Imports MySql.Data.MySqlClient

Dim myReport As New ReportDocument
Dim myData As New DataSet
Dim conn As New MySqlConnection
Dim cmd As New MySqlCommand
```
Dim myAdapter As New MySqlDataAdapter

conn.ConnectionString = "server=127.0.0.1;" + 
& "uid=root;" + 
& "pwd=12345;" + 
& "database=world"

Try
    conn.Open()
    cmd.CommandText = "SELECT name, population, countrycode FROM city ORDER BY countrycode, name;" + 
            "SELECT name, population, code, continent FROM country ORDER BY continent, name"
    cmd.Connection = conn
    myAdapter.SelectCommand = cmd
    myAdapter.Fill(myData)
    myReport.Load("world_report.rpt")
    myReport.Database.Tables(0).SetDataSource(myData.Tables(0))
    myReport.Database.Tables(1).SetDataSource(myData.Tables(1))
    myViewer.ReportSource = myReport
Catch ex As Exception
    MessageBox.Show(ex.Message, "Report could not be created", MessageBoxButton.OK, MessageBoxIcon.End
End Try

C#

using CrystalDecisions.CrystalReports.Engine;
using System.Data;
using MySql.Data.MySqlClient;

DataSet myData = new DataSet();
MySql.Data.MySqlClient.MySqlConnection conn;
MySql.Data.MySqlClient.MySqlDataAdapter myAdapter;

conn = new MySql.Data.MySqlClient.MySqlConnection();
myAdapter = new MySql.Data.MySqlClient.MySqlDataAdapter();

conn.ConnectionString = "server=127.0.0.1;uid=root;" + 
"pwd=12345;database=test;";

try
{
    cmd.CommandText = "SELECT name, population, countrycode FROM city ORDER BY countrycode, name;" + 
            "SELECT name, population, code, continent FROM country ORDER BY continent, name;"
    cmd.Connection = conn;
myAdapter.SelectCommand = cmd;
myAdapter.Fill(myData);

myReport.Load("\world_report.rpt");
myReport.Database.Tables(0).SetDataSource(myData.Tables(0));
myReport.Database.Tables(1).SetDataSource(myData.Tables(1));
myViewer.ReportSource = myReport;
}
catch (MySql.Data.MySqlClient.MySqlException ex)
{
    MessageBox.Show(ex.Message, "Report could not be created",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
}

It is important to order the SELECT queries in alphabetical order, as this is the order the report will expect its source tables to be in. One SetDataSource statement is needed for each table in the report.

This approach can cause performance problems because Crystal Reports must bind the tables together on the client-side, which will be slower than using a pre-saved data set.
Handling Date and Time Information in Connector/.NET

Introduction

MySQL and the .NET languages handle date and time information differently, with MySQL allowing dates that cannot be represented by a .NET data type, such as '0000-00-00 00:00:00'. These differences can cause problems if not properly handled.

In this section we will demonstrate how to properly handle date and time information when using Connector/.NET.
Problems when Using Invalid Dates

The differences in date handling can cause problems for developers who use invalid dates. Invalid MySQL dates cannot be loaded into native .NET DateTime objects, including NULL dates.

Because of this issue, .NET DataSet objects cannot be populated by the Fill method of the MySqlDataAdapter class as invalid dates will cause a System.ArgumentOutOfRangeException exception to occur.
Restricting Invalid Dates

The best solution to the date problem is to restrict users from entering invalid dates. This can be done on either the client or the server side.

Restricting invalid dates on the client side is as simple as always using the .NET DateTime class to handle dates. The DateTime class will only allow valid dates, ensuring that the values in your database are also valid. The disadvantage of this is that it is not useful in a mixed environment where .NET and non-.NET code are used to manipulate the database, as each application must perform its own date validation.

Users of MySQL 5.0.2 and higher can use the new traditional SQL mode to restrict invalid date values. For information on using the traditional SQL mode, see [server-sql-mode].
Handling Invalid Dates

Although it is strongly recommended that you avoid the use of invalid dates within your .NET application, it is possible to use invalid dates by means of the MySqlDateTime datatype.

The MySqlDateTime datatype supports the same date values that are supported by the MySQL server. The default behavior of Connector/.NET is to return a .NET DateTime object for valid date values, and return an error for invalid dates. This default can be modified to cause Connector/.NET to return MySqlDateTime objects for invalid dates.

To instruct Connector/.NET to return a MySqlDateTime object for invalid dates, add the following line to your connection string:

Allow Zero Datetime=True

Please note that the use of the MySqlDateTime class can still be problematic. The following are some known issues:

1. Data binding for invalid dates can still cause errors (zero dates like 0000-00-00 do not seem to have this problem).

2. The ToString method return a date formatted in the standard MySQL format (for example, 2005-02-23 08:50:25). This differs from the ToString behavior of the .NET DateTime class.

3. The MySqlDateTime class supports NULL dates, while the .NET DateTime class does not. This can cause errors when trying to convert a MySqlDateTime to a DateTime if you do not check for NULL first.

Because of the known issues, the best recommendation is still to use only valid dates in your application.
Handling NULL Dates

The .NET DateTime datatype cannot handle NULL values. As such, when assigning values from a query to a DateTime variable, you must first check whether the value is in fact NULL.

When using a MySqlDataReader, use the .IsDBNull method to check whether a value is NULL before making the assignment:

**VB.NET**

```vbnet
If Not myReader.IsDBNull(myReader.GetOrdinal("mytime")) Then
    myTime = myReader.GetDateTime(myReader.GetOrdinal("mytime"))
Else
    myTime = DateTime.MinValue
End If
```

**C#**

```csharp
if (!myReader.IsDBNull(myReader.GetOrdinal("mytime")))
    myTime = myReader.GetDateTime(myReader.GetOrdinal("mytime"));
else
    myTime = DateTime.MinValue;
```

NULL values will work in a data set and can be bound to form controls without special handling.
ASP.NET Provider Model

provides support for the ASP.NET 2.0 provider model. This model allows application developers to focus on the business logic of their application instead of having to recreate such boilerplate items as membership and roles support.

supplies the following providers:

- Membership Provider
- Role Provider
- Profile Provider
- Session State Provider (6.1 and later)

The following tables show the supported providers, their default provider and the corresponding MySQL provider.

**Membership Provider**

<table>
<thead>
<tr>
<th>Default Provider</th>
<th>MySQL Provider</th>
</tr>
</thead>
</table>

**Role Provider**

<table>
<thead>
<tr>
<th>Default Provider</th>
<th>MySQL Provider</th>
</tr>
</thead>
</table>

**Profile Provider**
**Default Provider**

System.Web.Profile.SqlProfileProvider

**MySQL Provider**

MySql.Web.Profile.MySQLProfileProvider

**SessionState Provider**


**MySQL Provider**


*Note:*

The MySQL Session State provider uses slightly different capitalization on the class name compared to the other MySQL providers.

**Installing The Providers**

The installation of Connector/Net 5.1 or later will install the providers and register them in your machine's .NET configuration file, machine.config. The additional entries created will result in the system.web section appearing similar to the following code:

```csharp
<system.web>
    <processModel autoConfig="true" />
    <httpHandlers />
    <membership>
        <providers>
            <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider,	System.Web,	Version=2.0.0.0,	Culture=neutral,	PublicKeyToken=b03f5f7f11d50a3a"
        </providers>
    </membership>
    <profile>
        <providers>
            <add name="AspNetSqlProfileProvider" connectionStringName="Loc"
<add name="MySQLProfileProvider" type="MySql.Web.Profile.MySQLProfileProvider" />
</profile>

<roleManager>
<providers>
<add name="AspNetSqlRoleProvider" connectionStringName="LocalSqlServer" />
<add name="AspNetWindowsTokenRoleProvider" applicationName="/" />
</providers>
</roleManager>
</system.web>

Each provider type can have multiple provider implementations. The default provider can also be set here using the defaultProvider attribute, but usually this is set in the web.config file either manually or by using the ASP.NET configuration tool.

At time of writing the MySqlSessionStateStore is not added to machine.config at install time, and so you would need to add the following:

C#<sessionState>
<providers>
</providers>
</sessionState>

It should be pointed out that the SessionState Provider uses the customProvider attribute, rather than defaultProvider, in order to set the provider as the default. A typical web.config file might contain:

C#

<system.web>
<membership defaultProvider="MySQLMembershipProvider" />  
<roleManager defaultProvider="MySQLRoleProvider" />
<profile defaultProvider="MySQLProfileProvider" />
<sessionState customProvider="MySQLSessionStateStore" />
<compilation debug="false">
...
</compilation>
</system.web>

This sets the MySQL Providers as the defaults to be used in this web application.

The providers are implemented in the file mysql.web.dll and this file can be
found in your installation folder. There is no need to run any type of SQL script to set up the database schema as the providers create and maintain the proper schema automatically.

**Using The Providers**

The easiest way to start using the providers is to use the ASP.NET configuration tool that is available on the Solution Explorer toolbar when you have a website project loaded.

In the web pages that open you will be able to select the MySQL membership and roles providers by indicating that you want to pick a custom provider for each area.

When the provider is installed, it creates a dummy connection string named LocalMySqlServer. This has to be done so that the provider will work in the ASP.NET configuration tool. However, you will want to override this connection string in your `web.config` file. You do this by first removing the dummy connection string and then adding in the proper one, as shown in the following example:

```c#
<connectionStrings>
  <remove name="LocalMySqlServer"/>
  <add name="LocalMySqlServer" connectionString="server=xxx;uid=xxx;"/>
</connectionStrings>
```

Rather than manually editing configuration files it is recommended that you use the MySQL Website Configuration tool to config your desired provider setup. From 6.1.1 onwards all providers can be selected and configured from this wizard. The tool will modify your `website.config` file to the desired configuration. A tutorial on doing this is available in the following section

[MySQL Website Configuration Tool](#).

A tutorial demonstrating how to use the Membership and Role Providers can be found in the following section [Tutorial: ASP.NET Membership and Role Provider](#).

**Deployment**
To use the providers on a production server you will need to distribute the MySql.Data and the MySql.Web assemblies and either register them in the remote systems Global Assembly Cache or keep them in your application's *bin/* directory.
Binary/Nonbinary Issues

There are certain situations where MySQL will return incorrect metadata about one or more columns. More specifically, the server will sometimes report that a column is binary when it is not and vice versa. In these situations, it becomes practically impossible for the connector to be able to correctly identify the correct metadata.

Some examples of situations that may return incorrect metadata are:

- Execution of SHOW PROCESSLIST. Some of the columns will be returned as binary even though they only hold string data.

- When a temp table is used to process a resultset, some columns may be returned with incorrect binary flags.

- Some server functions such DATE_FORMAT will incorrectly return the column as binary.

With the availability of BINARY and VARBINARY data types it is important that we respect the metadata returned by the server. However, we are aware that some existing applications may break with this change so we are creating a connection string option to enable or disable it. By default, Connector/Net 5.1 will respect the binary flags returned by the server. This will mean that you may need to make small changes to your application to accomodate this change.

In the event that the changes required to your application would be too large, you can add 'respect binary flags=false' to your connection string. This will cause the connector to use the prior behavior. In a nutshell, that behavior was that any column that is marked as string, regardless of binary flags, will be returned as string. Only columns that are specifically marked as a BLOB will be returned as BLOB.
Character Sets

Treating Binary Blobs As UTF8

MySQL doesn't currently support 4 byte UTF8 sequences. This makes it difficult to represent some multi-byte languages such as Japanese. To try and alleviate this, Connector/Net now supports a mode where binary blobs can be treated as strings.

To do this, you set the 'Treat Blobs As UTF8' connection string keyword to yes. This is all that needs to be done to enable conversion of all binary blobs to UTF8 strings. If you wish to convert only some of your blob columns, then you can make use of the 'BlobAsUTF8IncludePattern' and 'BlobAsUTF8ExcludePattern' keywords. These should be set to the regular expression pattern that matches the column names you wish to include or exclude respectively.

One thing to note is that the regular expression patterns can both match a single column. When this happens, the include pattern is applied before the exclude pattern. The result, in this case, would be that the column would be excluded. You should also be aware that this mode does not apply to columns of type BINARY or VARBINARY and also do not apply to nonbinary BLOB columns.

Currently this mode only applies to reading strings out of MySQL. To insert 4-byte UTF8 strings into blob columns you will need to use the .NET Encoding.GetBytes function to convert your string to a series of bytes. You can then set this byte array as a parameter for a BLOB column.
Working with medium trust

.NET applications operate under a given trust level. Normal desktop applications operate under full trust while web applications that are hosted in shared environments are normally run under the medium trust level. Some hosting providers host shared applications in their own app pools and allow the application to run under full trust, but this seems to be the exception rather than the rule.

Connector/Net versions prior to 5.0.8 and 5.1.3 were not compatible with medium trust hosting. Starting with these versions, Connector/Net can be used under medium trust hosting that has been modified to allow the use of sockets for communication. By default, medium trust does not include SocketPermission. Connector/Net uses sockets to talk with the MySQL server so it is required that a new trust level be created that is an exact clone of medium trust but that has SocketPermission added.
Tracing

6.2 introduced support for .NET 2.0 compatible tracing, using TraceSource objects.

The .NET 2.0 architecture consists of four main parts:

- **Source** - This is the originator of the trace information. The source is used to send trace messages. The name of the source provided by is mysql.

- **Switch** - This defines the level of trace information to emit. Typically, this is specified in the app.config file, so that it is not necessary to recompile an application in order to change the trace level.

- **Listener** - Trace listeners define where the trace information will be written to. Supported listeners include, for example, the Visual Studio Output window, the Windows Event Log, and the console.

- **Filter** - filters can be attached to listeners. Filters determine the level of trace information that will be written. While a switch defines the level of information that will be written to all listeners, a filter can be applied on a per-listener basis, giving finer grained control of trace information.

To use tracing a TraceSource object first needs to be created. To create a TraceSource object in you would use code similar to the following:

```csharp
TraceSource ts = new TraceSource("mysql");
```

To enable trace messages you also need to configure a trace switch. There are three main switch classes, BooleanSwitch, SourceSwitch, and TraceSwitch. Trace switches also have associated with them a trace level enumeration, these are Off, Error, Warning, Info, and Verbose. The following code snippet illustrates creating a switch:

```csharp
C#

```

```csharp
ts.Switch = new SourceSwitch("MySwitch", "Verbose");
```
This creates a SourceSwitch, called MySwitch, and sets the trace level to Verbose, meaning that all trace messages will be written.

It is convenient to be able to change the trace level without having to recompile the code. This is achieved by specifying the trace level in application configuration file, **app.config**. You then simply need to specify the desired trace level in the configuration file and restart the application. The trace source is configured within the system.diagnostics section of the file. The following XML snippet illustrates this:

By default trace information is written to the Output window of Microsoft Visual Studio. However, there are a wide range of listeners than can be attached to the trace source, so that trace messages can be written out to various destinations. It is also possible to create custom listeners to allow trace messages to be written to other destinations as mobile devices and web services. A commonly used example of a listener is ConsoleTraceListener, which writes trace messages to the console.

To add a listener at run time you can use code such as the following:

```csharp
ts.Listeners.Add(new ConsoleTraceListener());
```

You can then call methods on trace source object to generate trace information. For example, the TraceInformation(), TraceEvent(), or TraceData() methods can be used.

The TraceInformation() method simply prints a string passed as a parameter. The TraceEvent() method, as well as the optional informational string, requires a TraceEventType value to be passed to indicate the trace message type, and also an application specific ID. The TraceEventType can have a value of Verbose, Information, Warning, Error, and Critical. Using the TraceData() method you can pass any object, for example an exception object, instead of a message.

To ensure than these generated trace messages gets flushed from the trace source buffers to listeners, you need to invoke the Flush() method. When you are finished using a trace source, you should call the Close() method. The Close() method first calls Flush(), to ensure any remaining data is written out. It then frees up resources, and closes the listeners associated with the trace source.
ts.TraceInformation("Informational message");
ts.TraceEvent(TraceEventType.Error, 3, "Optional error message");
ts.TraceData(TraceEventType.Error, 3, ex); // pass exception object
    ...
    ts.Close();

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
### Connector/NET Connection String Options Reference

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow Batch</td>
<td>true</td>
<td>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 separator character. Commands sent to earlier versions of MySQL should be separated with ';'.</td>
</tr>
<tr>
<td>Allow User Variables</td>
<td>false</td>
<td>Setting this to true indicates that the provider expects user variables in the SQL. This option was added in Connector/NET version 5.2.2.</td>
</tr>
<tr>
<td>Allow Zero Datetime</td>
<td>false</td>
<td>True to have MySqlDataReader.GetValue() return a MySqlDateTime for date or datetime columns that have illegal values. False will cause a System.DateTime object to be returned for legal values and an exception will</td>
</tr>
</tbody>
</table>
be thrown for illegal values.

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoEnlist</td>
<td>true</td>
</tr>
<tr>
<td>BlobAsUTF8ExcludePattern</td>
<td>null</td>
</tr>
<tr>
<td>BlobAsUTF8IncludePattern</td>
<td>null</td>
</tr>
<tr>
<td>CertificateFile</td>
<td>null</td>
</tr>
<tr>
<td>CertificatePassword</td>
<td>null</td>
</tr>
<tr>
<td>CertificateStoreLocation</td>
<td>null</td>
</tr>
<tr>
<td>CertificateThumbprint</td>
<td>null</td>
</tr>
</tbody>
</table>

This option specifies the path to a certificate file in PFX format. For an example of usage see [Tutorial: Using SSL with](#). Was introduced with 6.2.1.

This option allows you to specify a password which is used in conjunction with a certificate specified using the option CertificateFile. For an example of usage see [Tutorial: Using SSL with](#). Was introduced with 6.2.1.

This option allows you to access a certificate held in a personal store, rather than use a certificate file and password combination. For an example of usage see [Tutorial: Using SSL with](#). Was introduced with 6.2.1.

This option allows you to specify a certificate thumbprint to ensure correct
identification of a certificate contained within a personal store. For an example of usage see Tutorial: Using SSL with. Was introduced with 6.2.1.

CharSet, 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.

Connect Timeout, 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.

Connection Reset

false

Convert Zero Datetime

false

True to have MySqlDataReader.GetValue() and MySqlDataReader.GetDateTime() return DateTime.MinValue for date or datetime columns that have illegal values.

Default Command Timeout

30

Sets the default value of the command timeout to be used. This does not supercede the individual command timeout property on an individual command object. If you set the command timeout property, that will be used. This option was added in Connector/NET 5.1.4

Encrypt, UseSSL

false

For Connector/NET 5.0.3 and later, when true, SSL 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. In versions before 5.0.3, this option had no effect. From version 6.2.1 this option is deprecated and is replaced by SSL Mode. However, the option is still supported if used. If this option is set to true it is equivalent to SSL Mode = Preferred.

Functions

ReturnString

false

This will cause the connector to return binary/varbinary values as strings, if they do not have a tablename in the metadata.

Host, Server, Data Source, DataSource, Address, Addr, Network Address

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 file name. 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.

Ignore Prepare

ture

When true, instructs the provider to ignore any calls to MySqlCommand.Prepare(). This option is provided to prevent issues with corruption of the statements when use with server side prepared statements. If you want to use server-side prepare statements, set this option to false. This option was added in Connector/.NET 5.0.3 and Connector/.NET 1.0.9.

Initial Catalog, Database
mysql

The name of the database to use initially

InteractiveSession

false

Logging

false

When true, various pieces of information is output to any configured TraceListeners.

Old Guids

false

This option was introduced in Connector.NET 6.1.1. The backend representation of a GUID type was changed from BINARY(16) to CHAR(36). This was done to allow developers to use the server function UUID() to populate a GUID table - UUID() generates a 36-character string. Developers of older applications can add 'Old Guids=true' to the connection string in order to use a GUID of data type BINARY(16).

Old Syntax, OldSyntax

false

This option was deprecated in Connector.NET 5.2.2. All code should now be written using the '@' symbol as the parameter marker.

Password, 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.

Pipe Name, Pipe

mysql

When set to the name of a named pipe, the MySqlConnection will attempt to connect to MySQL on that named pipe. This setting only applies to the Windows platform.

Port

3306

The port MySQL is using to listen for connections. This value is ignored if Unix socket is used.

Procedure Cache Size

25

Sets the size of the stored procedure cache. By default, Connector.NET will store the metadata (input/output datatypes) about the last 25 stored procedures used. To disable the stored procedure cache, set the value to zero (0). This option was added in Connector.NET 5.0.2 and Connector.NET 1.0.9.

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.

Respect Binary Flags
true

Setting this option to false means that Connector.NET will ignore a column's binary flags as set by the server. This option was added in Connector.NET version 5.1.3.

Shared Memory Name

MYSQL

The name of the shared memory object to use for communication if the connection protocol is set to memory.

SSL Mode

None

This option has the following values:

- **None** - do not use SSL.
- **Preferred** - use SSL if the server supports it, but allow connection in all cases.
- **Required** - Always use SSL. Deny connection if server does not support SSL.
- **VerifyCA** - Always use SSL. Validate the CA but tolerate name mismatch.
- **VerifyFull** - Always use SSL. Fail if the host name is not correct.

This option was introduced in 6.1.1.

TreatBlobsAsUTF8

false

Treat Tiny As Boolean

true
Setting this value to false indicates that TINYINT(1) will be treated as an INT. See also [numeric-type-overview] for a further explanation of the TINYINT and BOOL data types.

Use Affected Rows

false

When true the connection will report changed rows instead of found rows. This option was added in Connector/NET version 5.2.6.

Use Procedure Bodies

true

Setting this option to false indicates that the user connecting to the database does not have the SELECT privileges for the mysql.proc (stored procedures) table. When set to false, Connector/NET will not rely on this information being available when the procedure is called. Because Connector/NET will be unable to determine this information, you should explicitly set the types of the all the parameters before the call and the parameters should be added to the command in the exact same order as they appear in the procedure definition. This option was added in Connector/NET 5.0.4 and Connector/NET 1.0.10.

User Id, Username, Uid, User name

The MySQL login account being used.

Use Compression

false

Setting this option to true enables compression of packets exchanged between the client and the server. This exchange is defined by the MySQL client-server protocol.

Compression is used if both client and server support ZLIB compression, and the client has requested compression using this option.

A compressed packet header is: packet length (3 bytes), packet number (1 byte),
and Uncompressed Packet Length (3 bytes). The Uncompressed Packet Length is the number of bytes in the original, uncompressed packet. If this is zero then the data in this packet has not been compressed. When the compression protocol is in use, either the client or the server may compress packets. However, compression will not occur if the compressed length is greater than the original length. Thus, some packets will contain compressed data while other packets will not.

Use Usage Advisor
false

Use Performance Monitor
false

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Server Configuration, CacheServerConfiguration, CacheServerConfig</td>
<td>false</td>
<td>Specifies whether server variables should be updated when a pooled connection is returned. Turning this on will yield faster opens but will also not catch any server changes made by other connections. When a connection is</td>
</tr>
</tbody>
</table>
Connection Lifetime 0

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.

The minimum number of
| Min Pool Size | 0 | 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. |
| Reset Pooled Connections, ResetConnections, ResetPooledConnections | true | Specifies whether a ping and a reset should be sent to the server before a pooled connection is returned. Not resetting will yield faster connection opens but also will not clear out session items such as temp tables. |

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
● Tutorial: An Introduction to Connector/.NET Programming

This section provides a gentle introduction to programming with Connector/.NET. The example code is written in C#, and is designed to work on both Microsoft .NET Framework and Mono.

This tutorial is designed to get you up and running with Connector/.NET as quickly as possible, it does not go into detail on any particular topic. However, the following sections of this manual describe each of the topics introduced in this tutorial in more detail. In this tutorial you are encouraged to type in and run the code, modifying it as required for your setup.

This tutorial assumes you have MySQL and Connector/.NET already installed. It also assumes that you have installed the World example database, which can be downloaded from the

MySQL Documentation page. You can also find details on how to install the database on the same page.

● Note:

Before compiling the example code make sure that you have added References to your project as required. The References required are System, System.Data and MySql.Data.
The MySqlConnection Object

For your Connector/NET application to connect to a MySQL database it needs to establish a connection. This is achieved through the use of a MySqlConnection object.

The MySqlConnection constructor takes a connection string as one of its parameters. The connection string provides necessary information to make the connection to the MySQL database. The connection string is discussed more fully in Connecting to MySQL Using Connector/NET. A reference containing a list of supported connection string options can also be found in Connection Options.

The following code shows how to create a connection object.

C#

```csharp
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;

public class Tutorial1
{
    public static void Main()
    {
        string connStr = "server=localhost;user=root;database=world;"
MySqlConnection conn = new MySqlConnection(connStr);
        try
        {
            Console.WriteLine("Connecting to MySQL...");
            conn.Open();
            // Perform database operations
            conn.Close();
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.ToString());
        }
        Console.WriteLine("Done.");
    }
}
```
When the MySqlConnection constructor is invoked it returns a connection object, which is used for subsequent database operations. The first operation in this example is to open the connection. This needs to be done before further operations take place. Before the application exits the connection to the database needs to be closed by calling Close on the connection object.

Sometimes an attempt to perform an Open on a connection object can fail, this will generate an exception that can be handled via standard exception handling code.

In this section you have learned how to create a connection to a MySQL database, and open and close the corresponding connection object.
The MySqlCommand Object

Once a connection has been established with the MySQL database, the next step is to carry out the desired database operations. This can be achieved through the use of the MySqlCommand object.

You will see how to create a MySqlCommand object. Once it has been created, there are three main methods of interest that you can call:

- **ExecuteReader** - used to query the database. Results are usually returned in a MySqlDataReader object, created by ExecuteReader.

- **ExecuteNonQuery** - used to insert and delete data.

- **ExecuteScalar** - used to return a single value.

Once a MySqlCommand object has been created, you will call one of the above methods on it to carry out a database operation, such as perform a query. The results are usually returned into a MySqlDataReader object, and then processed, for example the results might be displayed. The following code demonstrates how this could be done.

C#

```csharp
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;

public class Tutorial2
{
    public static void Main()
    {
        string connStr = "server=localhost;user=root;database=world;";
        MySqlConnection conn = new MySqlConnection(connStr);
        try
        {
            Console.WriteLine("Connecting to MySQL...");
            conn.Open();
        }
    }
}
```
string sql = "SELECT Name, HeadOfState FROM Country WHERE Continent='Oceania';"
MySqlCommand cmd = new MySqlCommand(sql, conn);
MySqlDataReader rdr = cmd.ExecuteReader();

while (rdr.Read())
{
    Console.WriteLine(rdr[0] + " -- " + rdr[1]);
}

rdr.Close();
conn.Close();

} catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
}

Console.WriteLine("Done.");
}

When a connection has been created and opened, the code then creates a
MySqlCommand object. Note that the SQL query to be executed is passed to the
MySqlCommand constructor. The ExecuteReader method is then used to
generate a MySqlReader object. The MySqlReader object contains the results
generated by the SQL executed on the command object. Once the results have
been obtained in a MySqlReader object, the results can be processed. In this case
the information is simply printed out as part of a while loop. Finally, the
MySqlReader object is disposed of by running its Close method on it.

In the next example you will see how to use the ExecuteNonQuery method.

The procedure for performing an ExecuteNonQuery method call is simpler, as
there is no need to create an object to store results. This is because
ExecuteNonQuery is only used for inserting, updating and deleting data. The
following example illustrates a simple update to the Country table:

C#
```csharp
public static void Main()
{
    string connStr = "server=localhost;user=root;database=world;
    MySqlConnection conn = new MySqlConnection(connStr);
    try
    {
        Console.WriteLine("Connecting to MySQL...");
        conn.Open();

        string sql = "INSERT INTO Country (Name, HeadOfState, Continent) VALUES ('Disneyland','Mickey Mouse', 'North America')";
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
        conn.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.ToString());
    }
    Console.WriteLine("Done.");
}
}
```

The query is constructed, the command object created and the ExecuteNonQuery method called on the command object. You can access your MySQL database with the MySQL Client program and verify that the update was carried out correctly.

Finally, you will see how the ExecuteScalar method can be used to return a single value. Again, this is straightforward, as a MySqlDataReader object is not required to store results, a simple variable will do. The following code illustrates how to use ExecuteScalar:

```csharp
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;

public class Tutorial4
{
    public static void Main()
    {
        string connStr = "server=localhost;user=root;database=world;
```
```csharp
MySqlConnection conn = new MySqlConnection(connStr);
try {
    Console.WriteLine("Connecting to MySQL...");
    conn.Open();

    string sql = "SELECT COUNT(*) FROM Country";
    MySqlCommand cmd = new MySqlCommand(sql, conn);
    object result = cmd.ExecuteScalar();
    if (result != null) {
        int r = Convert.ToInt32(result);
        Console.WriteLine("Number of countries in the World");
    }
    conn.Close();
} catch (Exception ex) {
    Console.WriteLine(ex.ToString());
} 
Console.WriteLine("Done.");
}
```

This example uses a simple query to count the rows in the Country table. The result is obtained by calling ExecuteScalar on the command object.
Working with Decoupled Data

Previously, when using MySqlDataReader, the connection to the database was continually maintained, unless explicitly closed. It is also possible to work in a manner where a connection is only established when needed. For example, in this mode, a connection could be established in order to read a chunk of data, the data could then be modified by the application as required. A connection could then be reestablished only if and when the application needs to write data back to the database. This decouples the working data set from the database.

This decouple mode of working with data is supported by Connector/NET. There are several parts involved in allowing this method to work:

- **Data Set** - The Data Set is the area in which data is loaded in order to read or modify it. A DataSet object is instantiated, which can store multiple tables of data.

- **Data Adapter** - The Data Adapter is the interface between the Data Set and the database itself. The Data Adapter is responsible for efficiently managing connections to the database, opening and closing them as required. The Data Adapter is created by instantiating an object of the MySqlDataAdapter class. The MySqlDataAdapter object has two main methods: Fill which reads data into the Data Set, and Update, which writes data from the Data Set to the database.

- **Command Builder** - The Command Builder is a support object. The Command Builder works in conjunction with the Data Adapter. When a MySqlDataAdapter object is created it is typically given an initial SELECT statement. From this SELECT statement the Command Builder can work out the corresponding INSERT, UPDATE and DELETE statements that would be required should the database need to be updated. To create the Command Builder an object of the class MySqlCommandBuilder is created.

Each of these classes will now be discussed in more detail.

**Instantiating a DataSet object**
A DataSet object can be created simply, as shown in the following example code snippet:

```csharp
DataSet dsCountry;
...
dsCountry = new DataSet();
```

Although this creates the DataSet object it has not yet filled it with data. For that a Data Adapter is required.

**Instantiating a MySqlDataAdapter object**

The MySqlDataAdapter can be created as illustrated by the following example:

```csharp
MySqlDataAdapter daCountry;
...
string sql = "SELECT Code, Name, HeadOfState FROM Country WHERE Continent='North America';
daCountry = new MySqlDataAdapter(sql, conn);
```

Note, the MySqlDataAdapter is given the SQL specifying the data you wish to work with.

**Instantiating a MySqlCommandBuilder object**

Once the MySqlDataAdapter has been created, it is necessary to generate the additional statements required for inserting, updating and deleting data. There are several ways to do this, but in this tutorial you will see how this can most easily be done with MySqlCommandBuilder. The following code snippet illustrates how this is done:

```csharp
MySqlCommandBuilder cb = new MySqlCommandBuilder(daCountry);
```

Note that the MySqlDataAdapter object is passed as a parameter to the command builder.

**Filling the Data Set**

In order to do anything useful with the data from your database, you need to load
it into a Data Set. This is one of the jobs of the MySqlDataAdapter object, and is carried out with its Fill method. The following example code illustrates this:

**C#**

```csharp
DataSet dsCountry;
...
dsCountry = new DataSet();
...
daCountry.Fill(dsCountry, "Country");
```

Note the Fill method is a MySqlDataAdapter method, the Data Adapter knows how to establish a connection with the database and retrieve the required data, and then populates the Data Set when the Fill method is called. The second parameter is the table in the Data Set to update.

**Updating the Data Set**

The data in the Data Set can now be manipulated by the application as required. At some point, changes to data will need to be written back to the database. This is achieved through a MySqlDataAdapter method, the Update method.

**C#**

```csharp
daCountry.Update(dsCountry, "Country");
```

Again, the Data Set and the table within the Data Set to update are specified.

**Working Example**

The interactions between the DataSet, MySqlDataAdapter and MySqlCommandBuilder classes can be a little confusing, so their operation can perhaps be best illustrated by working code.

In this example, data from the World database is read into a Data Grid View control. Here, the data can be viewed and changed before clicking an update button. The update button then activates code to write changes back to the database. The code uses the principles explained above. The application was built using the Microsoft Visual Studio in order to place and create the user interface controls, but the main code that uses the key classes described above is shown below, and is portable.
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace WindowsFormsApplication5
{
    public partial class Form1 : Form
    {
        MySqlDataAdapter daCountry;
        DataSet dsCountry;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            string connStr = "server=localhost;user=root;database=world;port=3306;password=******;"
            MySqlConnection conn = new MySqlConnection(connStr);
            try
            {
                label2.Text = "Connecting to MySQL...";

                string sql = "SELECT Code, Name, HeadOfState FROM Country WHERE Continent='North America';"
                daCountry = new MySqlDataAdapter(sql, conn);
                MySqlCommandBuilder cb = new MySqlCommandBuilder(daCountry);
                dsCountry = new DataSet();
                daCountry.Fill(dsCountry, "Country");
                dataGridView1.DataSource = dsCountry;
                dataGridView1.DataMember = "Country";
            }
            catch (Exception ex)
            {
                label2.Text = ex.ToString();
            }
        }
    }
}
```csharp
private void button1_Click(object sender, EventArgs e)
{
    daCountry.Update(dsCountry, "Country");
    label2.Text = "MySQL Database Updated!";
}
}
```

The application running is shown below:

World Database Application
Working with Parameters

This part of the tutorial shows you how to use parameters in your Connector/NET application.

Although it is possible to build SQL query strings directly from user input, this is not advisable as it does not prevent from erroneous or malicious information being entered. It is safer to use parameters as they will be processed as field data only. For example, imagine the following query was constructed from user input:

```csharp
string sql = "SELECT Name, HeadOfState FROM Country WHERE Continent = " + user_continent;
```

If the string user_continent came from a Text Box control, there would potentially be no control over the string entered by the user. The user could enter a string that generates a run time error, or in the worst case actually harms the system. When using parameters it is not possible to do this because a parameter is only ever treated as a field parameter, rather than an arbitrary piece of SQL code.

The same query written using a parameter for user input would be:

```csharp
string sql = "SELECT Name, HeadOfState FROM Country WHERE Continent = @Continent";
```

Note that the parameter is preceded by an '@' symbol to indicate it is to be treated as a parameter.

As well as marking the position of the parameter in the query string, it is necessary to create a parameter object that can be passed to the Command object. In Connector/NET the class MySqlConnection is used for this purpose. The use of MySqlConnection is best illustrated by a small code snippet:

```csharp
MySqlParameter param = new MySqlConnection();
param(ParameterName = "@Continent");
param.Value = "North America";
```
In this example the string "North America" is supplied as the parameter value statically, but in a more practical example it would come from a user input control. Once the parameter has its name and value set it needs to be added to the Command object using the Add method.

A further example illustrates this:

**C#**

```csharp
using System;
using System.Data;

using MySql.Data;
using MySql.Data.MySqlClient;

public class Tutorial5
{
    public static void Main()
    {
        string connStr = "server=localhost;user=root;database=world;";
        MySqlConnection conn = new MySqlConnection(connStr);
        try
        {
            Console.WriteLine("Connecting to MySQL...");
            conn.Open();

            string sql = "SELECT Name, HeadOfState FROM Country WHERE Continent=@Continent";
            MySqlCommand cmd = new MySqlCommand(sql, conn);

            Console.WriteLine("Enter a continent e.g. 'North America', 'Europe':");
            string user_input = Console.ReadLine();

            MySqlCommandParameter param = new MySqlCommandParameter();
            param.ParameterName = "@Continent";
            param.Value = user_input;
            cmd.Parameters.Add(param);

            MySqlDataReader rdr = cmd.ExecuteReader();

            while (rdr.Read())
            {
                Console.WriteLine(rdr["Name"] + " --- " + rdr["HeadOfState"]);
            }

            conn.Close();
        }
    }
}
```
In this part of the tutorial you have see how to use parameters to make your code more secure.
Working with Stored Procedures

In this section you will see how to work with Stored Procedures. This section assumes you have a basic understanding of what a Stored Procedure is, and how to create one.

For the purposes of this tutorial, you will create a simple Stored Procedure to see how it can be called from Connector/NET. In the MySQL Client program, connect to the World database and enter the following Stored Procedure:

```csharp
DELIMITER //
CREATE PROCEDURE country_hos
(IN con CHAR(20))
BEGIN
    SELECT Name, HeadOfState FROM Country
    WHERE Continent = con;
END //
DELIMITER ;
```

Test the Stored Procedure wors as expected by typing the following into the MySQL Client program:

```csharp
CALL country_hos('Europe');
```

Note that The Stored Routine takes a single parameter, which is the continent you wish to restrict your search to.

Having confirmed that the Stored Procedure is present and correct you can now move on to seeing how it can be accessed from Connector/NET.

Calling a Stored Procedure from your Connector/NET application is similar to techniques you have seen earlier in this tutorial. A MySqlCommand object is created, but rather than taking a SQL query as a parameter it takes the name of the Stored Procedure to call. The MySqlCommand object also needs to be set to the type of Stored Procedure. This is illustrated by the following code snippet:

```csharp
DELIMITER //
CREATE PROCEDURE country_hos
(IN con CHAR(20))
BEGIN
    SELECT Name, HeadOfState FROM Country
    WHERE Continent = con;
END //
DELIMITER ;
```
In this case you also need to pass a parameter to the Stored Procedure. This can be achieved using the techniques seen in the previous section on parameters, Working with Parameters. This is shown in the following code snippet:

```csharp
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;

public class Tutorial6
{
    public static void Main()
    {
        string connStr = "server=localhost;user=root;database=world;"
        MySqlConnection conn = new MySqlConnection(connStr);
        try
        {
            Console.WriteLine("Connecting to MySQL...");
            conn.Open();
```
string rtn = "country_hos";
MySqlCommand cmd = new MySqlCommand(rtn, conn);
cmd.CommandType = CommandType.StoredProcedure;
MySqlParameter param = new SqlParameter();
param.ParameterName = "@con";
param.Value = "Europe";
cmd.Parameters.Add(param);

MySqlDataReader rdr = cmd.ExecuteReader();
while (rdr.Read())
{
    Console.WriteLine(rdr[0] + " --- " + rdr[1]);
}
conn.Close();
}
catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
}
Console.WriteLine("Done.");
}

In this section you have seen how to call a Stored Procedure from Connector.NET. For the moment, this concludes our introductory tutorial on programming with Connector.NET.
**Tutorial: ASP.NET Membership and Role Provider**

Many web sites feature the facility for the user to create a user account. They can then log into the web site and enjoy a personalized experience. This requires that the developer creates database tables to store user information, along with code to gather and process this data. This represents a burden on the developer, and there is the possibility for security issues to creep into the developed code. However, ASP.NET 2.0 introduced the Membership system. This system is designed around the concept of Membership, Profile and Role Providers, which together provide all of the functionality to implement a user system, that previously would have to have been created by the developer from scratch.

Currently, provides Membership, Role, Profile and Session State Providers.

This tutorial shows you how to set up your ASP.NET web application to use the Membership and Role Providers. It assumes that you have MySQL Server installed, along with and Microsoft Visual Studio. This tutorial was tested with 6.0.4 and Microsoft Visual Studio 2008 Professional Edition.

1. Create a new database in the MySQL Server using the MySQL Command Line Client program (mysql), or other suitable tool. It does not matter what name is used for the database, but it should be noted down so that it can be specified in the connection string constructed later in this tutorial. This database will contain the tables, automatically created for you later, used to store data about users and roles.

2. Create a new ASP.NET Web Site in Visual Studio. If you are not sure how to do this, refer to the following tutorial:

   [Tutorial: Databinding in ASP.NET using LINQ on Entities](#), which demonstrates how to create a simple ASP.NET web site.

   - Add References to MySql.Data and MySql.Web to the web site project.
   - Locate the `machine.config` file on your system, which is the configuration file
for the .NET Framework.

- Search the `machine.config` file to find the membership provider `MySQLMembershipProvider`.

- Add the attribute `autogenerateschema="true"`. The appropriate section should now resemble the following (note: for the sake of brevity some information has been excluded):

```c#
<membership>
  <providers>
    <add name="AspNetSqlMembershipProvider"
         ... connectionStringName="LocalSqlServer"
         ... />
    <add name="MySQLMembershipProvider"
         autogenerateschema="true"
         type="MySql.Web.Security.MySQLMembershipProvider, MySql.Web, Version=6.0.4.0, Culture=neutral, PublicKeyToken=c5687fc88969c44d"
         connectionStringName="LocalMySqlServer"
         ... />
  </providers>
</membership>
```

Note that the name for the connection string to be used to connect to the server that contains the membership database is `LocalMySqlServer`.

The `autogenerateschema="true"` attribute will cause to silently create, or upgrade, the schema on the database server, to contain the required tables for storing membership information.

- It is now necessary to create the connection string referenced in the previous step. Load the web site's `web.config` file into Visual Studio.

- Locate the section marked `<connectionStrings>`. Add the following connection string information:

```c#
<connectionStrings>
  <remove name="LocalMySqlServer"/>
  <add name="LocalMySqlServer"
The database specified is the one created in the first step. You could alternatively have used an existing database.

- At this point build the solution to ensure no errors are present. This can be done by selecting Build, Build Solution from the main menu, or pressing .

- ASP.NET supports the concept of locally and remotely authenticated users. With local authentication the user is validated using their Windows credentials when they attempt to access the web site. This can be useful in an Intranet environment. With remote authentication a user is prompted for their login details when accessing the web site, and these credentials are checked against the membership information stored in a database server such as MySQL Server. You will now see how to choose this form of authentication.

Start the ASP.NET Web Site Administration Tool. This can be done quickly by clicking the small hammer/Earth icon in the Solution Explorer. You can also launch this tool by selecting Website, ASP.NET Configuration from the main menu.

- In the ASP.NET Web Site Administration Tool click the tab.

- Now click the link.

- Select the radio button. The web site will now need to provide a form to allow the user to enter their login details. These will be checked against membership information stored in the MySQL database.
You now need to specify the Role and Membership Provider to be used. Click the tab.

- Click the link.
- Now select the and the radio buttons.

Select Membership and Role Provider

Use this page to select a provider for each feature.

### Membership Provider

- AspNetSqlMembershipProvider
- MySQLMembershipProvider

### Role Provider

- AspNetSqlRoleProvider
- AspNetWindowsTokenRoleProvider
- MySQLRoleProvider
• In Visual Studio rebuild the solution by selecting Build, Rebuild Solution from the main menu.

• Check that the necessary schema has been created. This can be achieved using the MySQL Command Line Client program.

Membership and Role Provider Tables

• Assuming all is present and correct you can now create users and roles for your web application. The easiest way to do this is with the ASP.NET Web Site Administration Tool. However, many web applications contain their own modules for creating roles and users. For simplicity the ASP.NET Web Site Administration Tool will be used in this tutorial.

• In the ASP.NET Web Site Administration Tool click on the tab. Now that both the Membership and Role Provider are enabled you will see links for creating roles and users. Click the link.

Security Tab
You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

Use the security Setup Wizard to configure security step by step.

Click the links in the table to manage the settings for your application.

<table>
<thead>
<tr>
<th>Users</th>
<th>Roles</th>
<th>Access Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing users: 1</td>
<td>Existing roles: 2</td>
<td>Create access rules</td>
</tr>
<tr>
<td>Create user</td>
<td>Disable Roles</td>
<td>Manage access rules</td>
</tr>
<tr>
<td>Manage users</td>
<td>Create or Manage roles</td>
<td></td>
</tr>
<tr>
<td>Select authentication type</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- You can now enter the name of a new Role and click Add Role to create the new Role. Create new Roles as required.
- Click the Back button.
- Click the link. You can now fill in information about the user to be created, and also allocate that user to one or more Roles.

Create User
Using the MySQL Command Line Client program you can check that your database has been correctly populated with the Membership and Role data.

Membership and Roles Table Contents
In this tutorial you have seen how to set up the Membership and Role Providers for use in your ASP.NET web application.
Tutorial: ASP.NET Session State Provider

from version 6.1 has included a MySQL Session State Provider. This provider allows you to store session state in a MySQL database. The following tutorial shows you how to prepare to use the MySQL Session State Provider, and then store session data into the MySQL database. This tutorial uses Microsoft Visual Studio 2008 Professional Edition, 6.1.1 and MySQL Server 5.1. This tutorial also assumes you have created an empty database, for example test, where you will store session data. You could do this using the MySQL Command Line Client tool.

1. In Visual Studio create a new ASP.NET web site. If you are not sure how to do this refer to the tutorial

Tutorial: Databinding in ASP.NET using LINQ on Entities which demonstrates how to do this.

- Launch the MySQL MySQL Website Configuration tool. Due to a bug in 6.1.1 this may not appear unless you are connected to a server in the Server Explorer. If you are unfamiliar with the MySQL Website Configuration tool it is suggested that you first work through the following tutorial MySQL Website Configuration Tool.

- Navigate through the wizard to the Session State page. Make sure the checkbox is selected.

- On the same page configure the connection string to the database that will contain your session data. This database can be empty as will create the schema required to store session data.

- Ensure that the checkbox is selected so that will create the schema in your database to store the session data correctly.

- Enter the name of your application.
Click Finish. The MySQL Website Configuration tool will now update your application's web.config file with information about the connection string and default providers to be used. In this case we have selected the MySQL Session State Provider.

At this point you are ready to use the MySQL database to store session data. To test that the set up has worked you can write a simple program that uses session variables.

1. Open Default.aspx.cs. In the Page_Load method add the following code:

   ```csharp
   Session["SessionVariable1"] = "Test string";
   ```

2. Build your solution.

3. Run the solution (without debugging). When the application runs, the provider will autogenerate tables required in the database you chose when setting up the application.

4. Check that the schema was in fact created. Using the MySQL Command Line Client use the target database and then type SHOW TABLES; . You will see that has created the required schema automatically, as we selected this to happen in the MySQL Website Configuration tool.

5. Now view the contents of these tables by typing SELECT * FROM my_aspnet_sessions; in the MySQL Command Line Client. This will display the session data our application used. Note that this is stored in binary format so some data may not display as expected.

At this point you have installed the Session State Provider and carried out a preliminary test of the installation. You will now work a bit more with the Session State Provider.

In this part of the tutorial you will set and retrieve a session variable. You can work with your existing project.

1. Select the Default.aspx and switch to Design View. Add a text box and three buttons. Change the text property for the buttons to , , and . These will be Button1, Button2 and Button3 respectively. Build your solution to ensure that no errors have been introduced.
2. Still in the Design View, double click Button1. Now to the Button1_Click event handler add code some the handler resembles the following:

```csharp
protected void Button1_Click(object sender, EventArgs e)
{
    Session["SessionString"] = TextBox1.Text;
}
```

You have created a new Session variable accessed using the key. This will be set to the text that was entered into the text box when Button1 is clicked.

3. In Design View double click Button2 to add its click event handler. This button needs to clear text from the text box. The code to do this is as follows:

```csharp
protected void Button2_Click(object sender, EventArgs e)
{
    TextBox1.Text = "";
}
```

The code simply assigns an empty string to the Text property of the text box.

4. In the Desin View double click Button3 and modify the click handler as follows:

```csharp
protected void Button3_Click(object sender, EventArgs e)
{
    TextBox1.Text = (String)Session["SessionString"];  
}
```

This will retrieve the session string and display it in the text box.

5. Now modify the Page_Load method as follows:

```csharp
```
```csharp
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        TextBox1.Text = "Enter some text";
    }
}

This ensures that when the page loads the text box Text property is reset.

6. Ensure that the solution is saved and then rebuild the solution.

7. Run the solution without debugging.

8. The form will be displayed. Enter some text into the text box. Now click Store Session Variable. At this point you have stored the string in a session variable.

9. Now click Clear Text to clear the text box.

10. Now click Show Session Variable to retrieve and display the session variable.

11. Refresh the page to destroy the form and display a new form.

12. Click Show Session Variable the text box will display the stored session variable, demonstrating that the refreshing the page does not destroy the session variable.

This illustrates that the session state data is not destroyed when a page is reloaded.
Tutorial: ASP.NET Profile Provider

This tutorial shows you how to use the MySQL Profile Provider to store user profile information in a MySQL database. The tutorial uses 6.1.1, MySQL Server 5.1 and Microsoft Visual Studio 2008 Professional Edition.

Many modern web sites allow the user to create a personal profile. This requires a significant amount of code, but ASP.NET reduces this considerable by including the functionality in its Profile classes. The Profile Provider provides an abstraction between these classes and a data source. The MySQL Profile Provider allows profile data to be stored in a MySQL database. This allows the profile properties to be written to a persistent store, and be retrieved when required. The Profile Provider also allows profile data to be managed effectively, for example it allows profiles that have not been accessed since a specific date to be deleted.

The following steps show you how you can select the MySQL Profile Provider.

1. Create a new ASP.NET web project.

2. Select the MySQL Website Configuration tool. Due to a bug in 6.1.1 you may have to first connect to a server in Server Explorer before the tool's icon will display in the toolbar of the Solution Explorer.

3. In the MySQL Website Configuration tool navigate through the tool to the Profiles page.

4. Select the checkbox.

5. Select the checkbox.

6. Click the Edit... button and configure a connection string for the database that will be used to store user profile information.

7. Navigate to the last page of the tool and click Finish to save your changes and exit the tool.

At this point you are now ready to start using the MySQL Profile Provider. With
the following steps you can carry out a preliminary test of your installation.

1. Open your `web.config` file.

2. Add a simple profile such as the following:

   ```c#
   <system.web>
   <anonymousIdentification enabled="true"/>
   <profile defaultProvider="MySQLProfileProvider">
       ...
       <properties>
           <add name="Name" allowAnonymous="true"/>
           <add name="Age" allowAnonymous="true" type="System.UInt16">
               <add name="Color" allowAnonymous="true" defaultValue="Blue"/>
               <add name="Style" allowAnonymous="true" defaultValue="Plain"/>
           </group>
       </properties>
   </profile>
   ...
   
   Note that `anonymousIdentification` has been set to true. This allows users who have not been authenticated to use profiles. They are identified by a GUID in a cookie rather than by user name.

Now that the simple profile has been defined in `web.config`, the next step is to write some code to test the profile.

1. In Design View design a simple page with the following controls:

   Simple Profile Application
These will allow the user to enter some profile information. The user can also use the buttons to save their profile, clear the page, and restore their profile data.

2. In the Code View add code as follows:

```
C#

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        TextBox1.Text = Profile.Name;
        TextBox2.Text = Profile.Age.ToString();
        Label1.Text = Profile.UI.Color;
    }
}

// Store Profile
protected void Button1_Click(object sender, EventArgs e)
{
    Profile.Name = TextBox1.Text;
    Profile.Age = (UInt16)Parse(TextBox2.Text);
}
```
// Clear Form
protected void Button2_Click(object sender, EventArgs e)
{
    TextBox1.Text = "";
    TextBox2.Text = "";
    Label1.Text = "";
}

// Retrieve Profile
protected void Button3_Click(object sender, EventArgs e)
{
    TextBox1.Text = Profile.Name;
    TextBox2.Text = Profile.Age.ToString();
    Label1.Text = Profile.UI.Color;
}

protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    Profile.UI.Color = DropDownList1.SelectedValue;
}

3. Save all files and build the solution to check that no errors have been introduced.

4. Run the application.

5. Enter your name, age and select a color from the listbox. Now store this information in your profile by clicking Store Profile. Note that if you do not select a color from the listbox your profile will use the default color Blue that was specified in the web.config file.

6. Click Clear Form to clear text from the textboxes and the label that displays your chosen color.

7. Now click Retrieve Profile to restore your profile data from the MySQL database.

8. Now exit the browser to terminate the application.

9. Run the application again. Note that when the page loads your profile information is restored from the MySQL database.

In this tutorial you have seen how to using the MySQL Profile Provider with.
Tutorial: Using an Entity Framework Entity as a Windows Forms Data Source

In this tutorial you will learn how to create a Windows Forms Data Source from an Entity in an Entity Data Model. This tutorial assumes that you have installed the World example database, which can be downloaded from the MySQL Documentation page. You can also find details on how to install the database on the same page. It will also be convenient for you to create a connection to the World database after it is installed. For instructions on how to do this see Making a connection.

Creating a new Windows Forms application

The first step is to create a new Windows Forms application.

1. In Visual Studio, select File, New, Project from the main menu.

2. Choose the installed template. Click OK. The solution is created.

Adding an Entity Data Model

You will now add an Entity Data Model to your solution.

1. In the Solution Explorer, right click on your application and select Add, New Item.... From select . Click Add.

Add Entity Data Model
2. You will now see the Entity Data Model Wizard. You will use the wizard to generate the Entity Data Model from the world example database. Select the icon. Click Next.

Entity Data Model Wizard Screen 1
3. You can now select the connection you made earlier to the World database. If you have not already done so, you can create the new connection at this time by clicking on New Connection.... For further instructions on creating a connection to a database see Making a connection.

Entity Data Model Wizard Screen 2
4. Make a note of the entity connection settings to be used in App.Config, as these will be used later to write the necessary control code.

5. Click Next.

6. The Entity Data Model Wizard connects to the database. You are then presented with a tree structure of the database. From this you can select the object you would like to include in your model. If you had created Views and Stored Routines these will be displayed along with any tables. In this example you just need to select the tables. Click Finish to create the model and exit the wizard.

Entity Data Model Wizard Screen 3
7. Visual Studio will generate the model and then display it.

Entity Data Model Diagram
8. From the Visual Studio main menu select Build, Build Solution, to ensure that everything compiles correctly so far.

**Adding a new Data Source**

You will now add a new Data Source to your project and see how it can be used to read and write to the database.

1. From the Visual Studio main menu select Data, Add New Data Source.... You will be presented with the Data Source Configuration Wizard.

Entity Data Source Configuration Wizard Screen 1
2. Select the icon. Click Next.

3. You will now select the Object you wish to bind to. Expand the tree. In this tutorial you will select the city table. Once the city table has been selected click Next.

Entity Data Source Configuration Wizard Screen 2
4. The wizard will confirm that the city object is to be added. Click Finish.

Entity Data Source Configuration Wizard Screen 3
5. The city object will be displayed in the Data Sources panel. If the Data Sources panel is not displayed, select Data, Show Data Sources from the Visual Studio main menu. The docked panel will then be displayed.

Data Sources
Using the Data Source in a Windows Form

You will now learn how to use the Data Source in a Windows Form.

1. In the Data Sources panel select the Data Source you just created and drag and drop it onto the Form Designer. By default the Data Source object will be added as a Data Grid View control. Note that the Data Grid View control is bound to the cityBindingSource and the Navigator control is bound to cityBindingNavigator.

Data Form Designer
2. Save and rebuild the solution before continuing.

**Adding Code to Populate the Data Grid View**

You are now ready to add code to ensure that the Data Grid View control will be populated with data from the City database table.

1. Double click the form to access its code.

2. Add code to instatiate the Entity Data Model's EntityContainer object and retrieve data from the database to populate the control.

**Adding Code to the Form**
3. Save and rebuild the solution.

4. Run the solution. Ensure the grid is populated and you can navigate the database.

The Populated Grid Control
Adding Code to Save Changes to the Database

You will now add code to enable you to save changes to the database.

The Binding source component ensures that changes made in the Data Grid View control are also made to the Entity classes bound to it. However, that data needs to be saved back from the entities to the database itself. This can be achieved by the enabling of the Save button in the Navigator control, and the addition of some code.

1. In the Form Designer click on the Save icon in the Form toolbar and ensure that its Enabled property is set to True.

   Save Button Enabled

2. Double click the Save icon in the Form toolbar to display its code.

3. You now need to add code to ensure that data is saved to the database when the save button is clicked in the application.

   Adding Save Code to the Form
4. Once the code has been added, save the solution and rebuild it. Run the application and verify that changes made in the grid are saved.
Tutorial: Databinding in ASP.NET using LINQ on Entities

In this tutorial you create an ASP.NET web page that binds LINQ queries to entities using the Entity Framework mapping.

If you have not already done so, you should install the World example database prior to attempting this tutorial. Instructions on where to obtain the database and instructions on how to install it where given in the tutorial.

Tutorial: Using an Entity Framework Entity as a Windows Forms Data Source.

Creating an ASP.NET web site

In this part of the tutorial you will create an ASP.NET web site. The web site will use the World database. The main web page will feature a drop down list from which you can select a country, data about that country's cities will then be displayed in a grid view control.

1. From the Visual Studio main menu select File, New, Web Site....

2. From the Visual Studio installed templates select . Click OK. You will be presented with the Source view of your web page by default.

3. Click the Design view tab situated underneath the Source view panel.

   The Design Tab
4. In the Design view panel, enter some text to decorate the blank web page.

5. Click on Toolbox. From the list of controls select . Drag and drop the control to a location beneath the text on your web page.

**Drop Down List**

6. From the control's context menu, ensure that the check box is enabled. This will ensure the control's event handler is called when an item is selected. The user's choice will in turn be used to populate the control.
Enable AutoPostBack

<table>
<thead>
<tr>
<th>DropDownList Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose Data Source...</td>
</tr>
<tr>
<td>Edit Items...</td>
</tr>
<tr>
<td>✅ Enable AutoPostBack</td>
</tr>
</tbody>
</table>

7. From the Toolbox select the control.

Grid View Control

Drag and drop the Grid View control to a location just below the Drop Down List you already placed.

Placed Grid View Control
8. At this point it is recommended that you save your solution, and build the solution to ensure that there are no errors.

9. If you run the solution you will see that the text and drop down list are displayed, but the list is empty. Also, the grid view does not appear at all. Adding this functionality is described in the following sections.

At this stage you have a web site that will build, but further functionality is required. The next step will be to use the Entity Framework to create a mapping from the World database into entities that you can control programmatically.

Creating an ADO.NET Entity Data Model

In this stage of the tutorial you will add an ADO.NET Entity Data Model to your project, using the World database at the storage level. The procedure for doing this is described in the tutorial Tutorial: Using an Entity Framework Entity as a Windows Forms Data Source, and so will not be repeated here.

Populating a Drop Data List Box with using the results of a entity LINQ query

In this part of the tutorial you will write code to populate the DropDownList control. When the web page loads the data to populate the list will be achieved by using the results of a LINQ query on the model created previously.

1. In the Design view panel, double click on any blank area. This brings up the method.

2. Modify the relevant section of code according to the following listing:

   C#
   ...
   public partial class _Default : System.Web.UI.Page
   {
       worldModel.worldEntities we;
       
       protected void Page_Load(object sender, EventArgs e)
       {
           we = new worldModel.worldEntities();

           if (!IsPostBack)
```csharp
{ 
    var countryQuery = from c in we.country 
                       orderby c.Name 
                       select new { c.Code, c.Name }; 
    DropDownList1.DataValueField = "Code";
    DropDownList1.DataTextField = "Name";
    DropDownList1.DataSource = countryQuery;
   .DataBind();
} 
...
```

Note that the list control only needs to be populated when the page first loads. The conditional code ensures that if the page is subsequently reloaded, the list control is not repopulated, which would cause the user selection to be lost.

3. Save the solution, build it and run it. You should see the list control has been populated. You can select an item, but as yet the grid view control does not appear.

At this point you have a working Drop Down List control, populated by a LINQ query on your entity data model.

**Populating a Grid View control using an entity LINQ query**

In the last part of this tutorial you will populate the Grid View Control using a LINQ query on your entity data model.

1. In the Design view double click on the control. This causes its code to be displayed. This method is called when a user selects an item in the list control and thus fires an AutoPostBack event.

2. Modify the relevant section of code accordingly to the following listing:

```csharp
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e) {
    var cityQuery = from c in we.city 
                    orderby c.Name 
                    select new { c.Name, c.Population, c.Col...`
The grid view control is populated from the result of the LINQ query on the entity data model.

3. As a check compare your code to that shown in the following screenshot:

Source Code

4. Save, build and run the solution. As you select a country you will see its cities are displayed in the grid view control.

The Working Web Site
In this tutorial you have seen how to create an ASP.NET web site, you have also seen how you can access a MySQL database via LINQ queries on an entity data model.
Tutorial: Using SSL with

In this tutorial you will learn how you can use to connect to a MySQL server configured to use SSL. Support for SSL client certificates was added with 6.2.

MySQL Server uses the PEM format for certificates and private keys. This tutorial will use the test certificates from the server test suite by way of example. You can obtain the MySQL Server source code from

MySQL Downloads. The certificates can be found in the directory ./mysql-test/std_data.

To carry out the steps in this tutorial you will also need to have Open SSL installed. This can be downloaded for Microsoft Windows at no charge from Shining Light Productions.

Further details on the connection string options used in this tutorial can be found at Connection Options.

Configuring the MySQL Server to use SSL

1. In the MySQL Server configuration file, set the SSL parameters as follows:

   ```
   C#
   ssl-ca=path/to/repo/mysql-test/std_data/cacert.pem
   ssl-cert=path/to/repo/mysql-test/std_data/server-cert.pem
   ssl-key=path/to/repo/mysql-test/std_data/server-key.pem
   ```

   Adjust the directories according to the location in which you installed the MySQL source code.

2. In this step you create a test user and set the user to require SSL.

   Using the MySQL Command Line Client, connect as root and create the user sslclient.

3. To set privileges and requirements, issue the following command:
GRANT ALL PRIVILEGES ON *.* TO sslclient@'%' REQUIRE SSL;

Creating a certificate file to use with the .NET client

1. The .NET client does not use the PEM file format, as .NET does not support this format natively. You will be using test client certificates from the same server repository, for the purposes of this example. You will need to convert these to PFX format first. This format is also known as PKCS#12. An article describing this procedure can be found at the Citrix website. From the directory server-repository-root/mysql-test/std_data, issue the following command:

```
openssl pkcs12 -export -in client-cert.pem -inkey client-key.pem -certfile cacert.pem -out client.pfx
```

2. When asked for an export password, enter the password . The file `client.pfx` will be generated. This file is used in the remainder of the tutorial.

Connecting to the server using a file-based certificate

1. You will use PFX file, `client.pfx` you created in the previous step to authenticate the client. The following example demonstrates how to connect using the SSL Mode, CertificateFile and CertificatePassword connection string options:

```c#
using (MySqlConnection connection = new MySqlConnection(
    "database=test;user=sslclient;" +
    "CertificateFile=H:\bzr\mysql-trunk\mysqlest\std_data\cli" +
    "CertificatePassword=pass;" +
    "SSL Mode=Required "))
{
    connection.Open();
}
```

The path to the certificate file will need to be changed to reflect your individual installation.

Connecting to the server using a store-based certificate

1. The first step is to import the PFX file, `client.pfx`, into the Personal Store.
Double-click the file in Windows explorer. This launches the Certificate Import Wizard.

2. Follow the steps dictated by the wizard, and when prompted for the password for the PFX file, enter .

3. Click Finish to close the wizard and import the certificate into the personal store.

Examine certificates in the Personal Store

1. Start the Microsoft Management Console by entering \texttt{mmc.exe} at a command prompt.

2. Select File, Add/Remove snap-in. Click Add. Select from the list of available snap-ins in the dialog.

3. Click Add button in the dialog, and select the radio button. This is used for personal certificates.

4. Click the Finish button.

5. Click OK to close the Add/Remove Snap-in dialog.

6. You will now have displayed in the left panel of the Microsoft Management Console. Expand the Certificates - Current User tree item and select , . The right-hand panel will display a certificate issued to MySQL. This is the certificate that was previously imported. Double-click the certificate to display its details.

7. After you have imported the certificate to the Personal Store, you can use a more succinct connection string to connect to the database, as illustrated by the following code:

\begin{verbatim}
C#
using (MySqlConnection connection = new MySqlConnection(
    "database=test;user=sslclient;" +
    "Certificate Store Location=CurrentUser;" +
    "SSL Mode=Required")
{
    connection.Open();
}\end{verbatim}
Certificate Thumbprint Parameter

If you have a large number of certificates in your store, and many have the same Issuer, this can be a source of confusion and result in the wrong certificate being used. To alleviate this situation, there is an optional Certificate Thumbprint parameter that can additionally be specified as part of the connection string. As mentioned before, you can double-click on a certificate in the Microsoft Management Console to display the certificate's details. When the Certificate dialog is displayed click the tab and scroll down to see the thumbprint. The thumbprint will typically be a number such as 47 94 36 00 9a 40 f3 01 7a 14 5c f8 47 9e 76 94 d7 aa de f0. This thumbprint can be used in the connection string, as the following code illustrates:

```csharp
using (MySqlConnection connection = new MySqlConnection(
    "database=test;user=sslclient;" +
    "Certificate Store Location=CurrentUser;" +
    "Certificate Thumbprint=479436009a40f3017a145cf8479e7694d7aade" +
    "SSL Mode=Required")
{
    connection.Open();
}
```

Spaces in the thumbprint parameter are optional and the value is case-insensitive.

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Connector/NET Support

The developers of Connector/NET greatly value the input of our users in the software development process. If you find Connector/NET lacking some feature important to you, or if you discover a bug and need to file a bug report, please use the instructions in [bug-reports].
Connector/NET Community Support

- Community support for Connector/NET can be found through the forums at http://forums.mysql.com.

- Community support for Connector/NET can also be found through the mailing lists at http://lists.mysql.com.

- Paid support is available from Sun Microsystems, Inc. Additional information is available at http://www.mysql.com/support/.
How to report Connector/NET Problems or Bugs

If you encounter difficulties or problems with Connector/NET, contact the Connector/NET community

Connector/NET Community Support.

You should first try to execute the same SQL statements and commands from the client program or from admndemo. This helps you determine whether the error is in Connector/NET or MySQL.

If reporting a problem, you should ideally include the following information with the email:

- Operating system and version
- Connector/NET version
- MySQL server version
- Copies of error messages or other unexpected output
- Simple reproducible sample

Remember that the more information you can supply to us, the more likely it is that we can fix the problem.

If you believe the problem to be a bug, then you must report the bug through http://bugs.mysql.com/.
Connector/NET Change History

The Connector/NET Change History (Changelog) is located with the main Changelog for MySQL. See [connector-net-news].

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Connector/NET Versions

There are several versions of Connector/NET available:

- Connector/NET 1.0 includes support for MySQL 4.0, MySQL 4.1, MySQL 5.0 features, and full compatibility with the ADO.NET driver interface.

- Connector/NET 5.0 includes support for MySQL 4.0, MySQL 4.1, MySQL 5.0 and MySQL 5.1 features. Connector/NET 5.0 also includes full support for the ADO.Net 2.0 interfaces and subclasses, includes support for the usage advisor and performance monitor (PerfMon) hooks.

- Connector/NET 5.1 includes support for MySQL 4.0, MySQL 4.1, MySQL 5.0, MySQL 5.1 and MySQL 5.4 features. Connector/NET 5.1 also includes support for a new membership/role provider, Compact Framework 2.0, a new stored procedure parser and improvements to GetSchema. Connector/NET 5.1 also includes the Visual Studio Plugin as a standard installable component.

- Connector/NET 5.2 includes support for MySQL 4.1, MySQL 5.0, MySQL 5.1 and MySQL 5.4 features. Connector/NET 5.2 also includes support for a new membership/role provider, Compact Framework 2.0, a new stored procedure parser and improvements to GetSchema. Connector/NET 5.2 also includes the Visual Studio Plugin as a standard installable component.

- Connector/NET 6.0 includes support for MySQL 4.1, MySQL 5.0, MySQL 5.1 and MySQL 5.4.

- Connector/NET 6.1 includes support for MySQL 4.1, MySQL 5.0, MySQL 5.1, MySQL 5.4. Important new features include the MySQL Website Configuration Tool and a Session State Provider.

- Connector/NET 6.2 includes support for MySQL 4.1, MySQL 5.0, MySQL 5.1, MySQL 5.4. Important new features include a new logging system and client SSL certificates. Connector/NET 6.2 is currently available as a Beta release.
- Connector/NET 6.3 includes support for MySQL 5.0, MySQL 5.1, MySQL 5.4. Connector/NET 6.3 is not yet available.

The latest source code for Connector/NET can be downloaded from the MySQL public Subversion server. For further details see [Installing Connector/NET from the source code](#).

The following table shows the .NET Framework version required, and MySQL Server version supported by Connector/NET:

<table>
<thead>
<tr>
<th>Connector/NET version</th>
<th>ADO.NET version supported</th>
<th>.NET Framework version required</th>
<th>MySQL Server version supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.x</td>
<td>1.x</td>
<td>4.0, 4.1, 5.0</td>
</tr>
<tr>
<td>5.0</td>
<td>2.x+</td>
<td>2.x+</td>
<td>4.0, 4.1, 5.0</td>
</tr>
<tr>
<td>5.1</td>
<td>2.x+</td>
<td>2.x+</td>
<td>4.0, 4.1, 5.0, 5.1, 5.4</td>
</tr>
<tr>
<td>5.2</td>
<td>2.x+</td>
<td>2.x+</td>
<td>4.1, 5.0, 5.1, 5.4</td>
</tr>
<tr>
<td>6.0</td>
<td>2.x+</td>
<td>2.x+</td>
<td>4.1, 5.0, 5.1, 5.4</td>
</tr>
<tr>
<td>Version</td>
<td>Products</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>2.x+ 2.x+ 5.1, 5.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>2.x+ 2.x+ 4.1, 5.0, 5.1, 5.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.3</td>
<td>2.x+ 2.x+ 5.0, 5.1, 5.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

Version numbers for MySQL products are formatted as X.Y.Z, where Z=0 indicates alpha, Z=1 indicates beta, and Z>=2 indicates GA. However, Windows tools (Control Panel, properties display) may show the version numbers as XX.YY.ZZ. For example, the official MySQL formatted version number 5.0.9 may be displayed by Windows tools as 5.00.09. The two versions are the same; only the number display format is different.

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Namespaces

<table>
<thead>
<tr>
<th>Namespace</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySql.Data.MySqlClient</td>
<td></td>
</tr>
<tr>
<td>MySql.Data.Types</td>
<td></td>
</tr>
<tr>
<td>MySql.Web.Common</td>
<td></td>
</tr>
<tr>
<td>MySql.Web.Profile</td>
<td></td>
</tr>
<tr>
<td>MySql.Web.Properties</td>
<td></td>
</tr>
<tr>
<td>MySql.Web.Security</td>
<td></td>
</tr>
<tr>
<td>MySql.Web.SessionState</td>
<td></td>
</tr>
</tbody>
</table>

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
MySQL Connector/Net
MySql.Data.MySqlClient Namespace

Send Feedback
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MySqlBulkLoader</strong></td>
<td>DBProviderFactory implementation for MysqlClient.</td>
</tr>
<tr>
<td><strong>MySqlClientFactory</strong></td>
<td>Represents a SQL statement to execute against a MySQL database. This class cannot be inherited.</td>
</tr>
<tr>
<td><strong>MySqlCommand</strong></td>
<td>Represents a SQL statement to execute against a MySQL database. This class cannot be inherited.</td>
</tr>
<tr>
<td><strong>MySqlCommandBuilder</strong></td>
<td>Automatically generates single-table commands used to reconcile changes made to a DataSet with the associated MySQL database. This class cannot be inherited.</td>
</tr>
<tr>
<td><strong>MySqlConnection</strong></td>
<td>Represents an open connection to a MySQL Server database. This class cannot be inherited.</td>
</tr>
<tr>
<td><strong>MySqlConnectionStringBuilder</strong></td>
<td>Represents a set of data commands and a database connection that are used to fill a dataset and update a MySQL database. This class cannot be inherited.</td>
</tr>
<tr>
<td><strong>MySqlDataAdapter</strong></td>
<td>Provides a means of reading a forward-only stream of rows from a MySQL database. This class cannot be inherited.</td>
</tr>
<tr>
<td><strong>MySqlDataReader</strong></td>
<td>Collection of error codes that can be returned by the server</td>
</tr>
<tr>
<td><strong>MySqlError</strong></td>
<td>The exception that is thrown when MySQL returns an error. This class cannot be inherited.</td>
</tr>
<tr>
<td><strong>MySqlException</strong></td>
<td>Helper class that makes it easier to work with the provider.</td>
</tr>
<tr>
<td>Class Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MySqlInfoMessageEventArgs</td>
<td>Provides data for the InfoMessage event. This class cannot be inherited.</td>
</tr>
<tr>
<td>MySqlParameter</td>
<td>Represents a parameter to a MySqlCommand, and optionally, its mapping to</td>
</tr>
<tr>
<td></td>
<td>DataSet columns. This class cannot be inherited.</td>
</tr>
<tr>
<td>MySqlParameterCollection</td>
<td>Represents a collection of parameters relevant to a MySqlCommand as well</td>
</tr>
<tr>
<td></td>
<td>as their respective mappings to columns in a DataSet. This class cannot be</td>
</tr>
<tr>
<td></td>
<td>inherited.</td>
</tr>
<tr>
<td>MySqlRowUpdatedEventArgs</td>
<td>Provides data for the RowUpdated event. This class cannot be inherited.</td>
</tr>
<tr>
<td>MySqlRowUpdatingEventArgs</td>
<td>Provides data for the RowUpdating event. This class cannot be inherited.</td>
</tr>
<tr>
<td>MySqlScript</td>
<td>Provides a class capable of executing a SQL script containing multiple SQL</td>
</tr>
<tr>
<td></td>
<td>statements including CREATE PROCEDURE statements that require changing the</td>
</tr>
<tr>
<td></td>
<td>delimiter.</td>
</tr>
<tr>
<td>MySqlScriptErrorEventArgs</td>
<td>Represents a SQL transaction to be made in a MySQL database. This class</td>
</tr>
<tr>
<td></td>
<td>cannot be inherited.</td>
</tr>
<tr>
<td>MySqlTrace</td>
<td></td>
</tr>
<tr>
<td>MySqlTransaction</td>
<td></td>
</tr>
</tbody>
</table>
### Delegates

<table>
<thead>
<tr>
<th>Delegate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlInfoMessageEventHandler</td>
<td>Represents the method that will handle the InfoMessage event of a MySqlConnection.</td>
</tr>
<tr>
<td>MySqlRowUpdatedEventHandler</td>
<td>Represents the method that will handle the RowUpdated event of a MySqlDataAdapter.</td>
</tr>
<tr>
<td>MySqlRowUpdatingEventHandler</td>
<td>Represents the method that will handle the RowUpdating event of a MySqlDataAdapter.</td>
</tr>
<tr>
<td>MySqlScriptErrorEventHandler</td>
<td></td>
</tr>
<tr>
<td>MySqlStatementExecutedEventHandler</td>
<td></td>
</tr>
</tbody>
</table>
## Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlBulkLoaderConflictOption</td>
<td>Allows the user to specify the type of connection that should be used.</td>
</tr>
<tr>
<td>MySqlBulkLoaderPriority</td>
<td>Specifies MySQL specific data type of a field, property, for use in a</td>
</tr>
<tr>
<td></td>
<td>MySqlParameter.</td>
</tr>
<tr>
<td>MySqlCertificateStoreLocation</td>
<td>Specifies the connection types supported</td>
</tr>
<tr>
<td>MySqlConnectionProtocol</td>
<td>Provides a reference to error codes returned by MySQL.</td>
</tr>
<tr>
<td>MySqlDbType</td>
<td>SSL options for connection.</td>
</tr>
<tr>
<td>MySqlDriverType</td>
<td></td>
</tr>
<tr>
<td>MySqlErrorCode</td>
<td></td>
</tr>
<tr>
<td>MySqlSslMode</td>
<td></td>
</tr>
<tr>
<td>MySqlTraceEventType</td>
<td></td>
</tr>
<tr>
<td>UsageAdvisorWarningFlags</td>
<td></td>
</tr>
</tbody>
</table>

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySqlConnection
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public class MySqlBulkLoader

Visual Basic (Declaration)

Public Class MySqlBulkLoader

Visual C++

public ref class MySqlBulkLoader
Inheritance Hierarchy

System::Object
MySql.Data.MySqlClient::MySqlBulkLoader
The **MySqlBulkLoader** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MySqlBulkLoader</code></td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Equals**        | Determines whether the specified `Object` is equal to the current `Object`.  
(Inherited from `Object`.) |
| **Finalize**      | Allows an `Object` to attempt to free resources and perform other cleanup operations before the `Object` is reclaimed by garbage collection.  
(Inherited from `Object`.) |
| **GetHashCode**   | Serves as a hash function for a particular type.  
(Inherited from `Object`.) |
| **GetType**       | Gets the `Type` of the current instance.  
(Inherited from `Object`.) |
| **Load**          | Execute the load operation |
| **MemberwiseClone** | Creates a shallow copy of the current `Object`.  
(Inherited from `Object`.) |
| **ToString**      | Returns a `String` that represents the current `Object`.  
(Inherited from `Object`.) |
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CharacterSet</strong></td>
<td>Gets or sets the character set.</td>
</tr>
<tr>
<td><strong>Columns</strong></td>
<td>Gets the columns.</td>
</tr>
<tr>
<td><strong>ConflictOption</strong></td>
<td>Gets or sets the conflict option.</td>
</tr>
<tr>
<td><strong>Connection</strong></td>
<td>Gets or sets the connection.</td>
</tr>
<tr>
<td><strong>EscapeCharacter</strong></td>
<td>Gets or sets the escape character.</td>
</tr>
<tr>
<td><strong>Expressions</strong></td>
<td>Gets the expressions.</td>
</tr>
<tr>
<td><strong>FieldQuotationCharacter</strong></td>
<td>Gets or sets the field quotation character.</td>
</tr>
<tr>
<td><strong>FieldQuotationOptional</strong></td>
<td>Gets or sets a value indicating whether [field quotation optional].</td>
</tr>
<tr>
<td><strong>Field Terminator</strong></td>
<td>Gets or sets the field terminator.</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>Gets or sets the name of the file.</td>
</tr>
<tr>
<td><strong>LinePrefix</strong></td>
<td>Gets or sets the line prefix.</td>
</tr>
<tr>
<td><strong>Line Terminator</strong></td>
<td>Gets or sets the line terminator.</td>
</tr>
<tr>
<td><strong>Local</strong></td>
<td>Gets or sets a value indicating whether the filename that is to be loaded is local to the client or not</td>
</tr>
<tr>
<td><strong>NumberOfLinesToSkip</strong></td>
<td>Gets or sets the number of lines to skip.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>Gets or sets the priority.</td>
</tr>
<tr>
<td><strong>TableName</strong></td>
<td>Gets or sets the name of the table.</td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
<td>Gets or sets the timeout.</td>
</tr>
</tbody>
</table>
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlBulkLoader(
   MySqlConnection connection
)

Visual Basic (Declaration)

Public Sub New ( _
   connection As MySqlConnection _
)

Visual C++

public:
MySqlBulkLoader(
   MySqlConnection^ connection
)

Parameters

collection
   Type: MySql.Data.MySqlClient,:::MySqlConnection
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLBulkLoader** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Load</strong></td>
<td>Execute the load operation</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySQLBulkLoader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Execute the load operation

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Load()

Visual Basic (Declaration)

Public Function Load As Integer

Visual C++

public:
int Load()

Return Value

The number of rows inserted.
See Also

MySQLBulkLoader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlBulkLoader` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CharacterSet</strong></td>
<td>Gets or sets the character set.</td>
</tr>
<tr>
<td><strong>Columns</strong></td>
<td>Gets the columns.</td>
</tr>
<tr>
<td><strong>ConflictOption</strong></td>
<td>Gets or sets the conflict option.</td>
</tr>
<tr>
<td><strong>Connection</strong></td>
<td>Gets or sets the connection.</td>
</tr>
<tr>
<td><strong>EscapeCharacter</strong></td>
<td>Gets or sets the escape character.</td>
</tr>
<tr>
<td><strong>Expressions</strong></td>
<td>Gets the expressions.</td>
</tr>
<tr>
<td><strong>FieldQuotationCharacter</strong></td>
<td>Gets or sets the field quotation character.</td>
</tr>
<tr>
<td><strong>FieldQuotationOptional</strong></td>
<td>Gets or sets a value indicating whether [field quotation optional].</td>
</tr>
<tr>
<td><strong>FieldTerminator</strong></td>
<td>Gets or sets the field terminator.</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>Gets or sets the name of the file.</td>
</tr>
<tr>
<td><strong>LinePrefix</strong></td>
<td>Gets or sets the line prefix.</td>
</tr>
<tr>
<td><strong>LineTerminator</strong></td>
<td>Gets or sets the line terminator.</td>
</tr>
<tr>
<td><strong>Local</strong></td>
<td>Gets or sets a value indicating whether the filename that is to be loaded is local to the client or not</td>
</tr>
<tr>
<td><strong>NumberOfLinesToSkip</strong></td>
<td>Gets or sets the number of lines to skip.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>Gets or sets the priority.</td>
</tr>
<tr>
<td><strong>TableName</strong></td>
<td>Gets or sets the name of the table.</td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
<td>Gets or sets the timeout.</td>
</tr>
</tbody>
</table>
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the character set.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public string CharSet { get; set; }

Visual Basic (Declaration)

Public Property CharSet As String

Visual C++

public:
property String^ CharSet {
    String^ get ();
    void set (String^ value);
}

Field Value

The character set.
See Also

 MySqlBulkLoader Class
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the columns.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public StringCollection Columns { get; }

Visual Basic (Declaration)

Public ReadOnly Property Columns As StringCollection

Visual C++

public:
property StringCollection^ Columns {
    StringCollection^ get ();
}

Field Value

The columns.
See Also

MySQLBulkLoader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the conflict option.

**Namespace:**  MySql.Data.MySqlClient

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlBulkLoaderConflictOption ConflictOption { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property ConflictOption As MySqlBulkLoaderConflictOption
```

Visual C++

```cpp
public:
property MySqlBulkLoaderConflictOption ConflictOption {
    MySqlBulkLoaderConflictOption get ();
    void set (MySqlBulkLoaderConflictOption value);
}
```

Field Value

The conflict option.
See Also

[MySqlBulkLoader Class]
[MySql.Data.MySqlClient Namespace]
Gets or sets the connection.

**Namespace:** [MySQL.Data.MySqlClient](#)  
**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlConnection Connection { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property Connection As MySqlConnection
```

Visual C++

```cpp
public:
property MySqlConnection^ Connection {
    MySqlConnection^ get ();
    void set (MySqlConnection^ value);
}
```

Field Value

The connection.
See Also

MySQLBulkLoader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the escape character.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public char EscapeCharacter { get; set; }

Visual Basic (Declaration)

Public Property EscapeCharacter As Char

Visual C++

public:
property wchar_t EscapeCharacter {
    wchar_t get ();
    void set (wchar_t value);
}

Field Value

The escape character.
See Also

[MySQLBulkLoader Class](#)
[MySQL.Data.MySqlClient Namespace](#)

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the expressions.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public StringCollection Expressions { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Expressions As StringCollection
```

**Visual C++**

```c++
public:
property StringCollection^ Expressions {
    StringCollection^ get ();
}
```

### Field Value

The expressions.
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the field quotation character.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public char FieldQuotationCharacter { get; set; }
```

**Visual Basic (Declaration)**

```
Public Property FieldQuotationCharacter As Char
```

**Visual C++**

```cpp
public:
    property wchar_t FieldQuotationCharacter {
        wchar_t get ();
        void set (wchar_t value);
    }
```

### Field Value

The field quotation character.
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a value indicating whether [field quotation optional].

**Namespace:**  [MySQL.Data.MySqlClient]

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool FieldQuotationOptional { get; set; }

Visual Basic (Declaration)

Public Property FieldQuotationOptional As Boolean

Visual C++

public:
property bool FieldQuotationOptional {
    bool get ();
    void set (bool value);
}

Field Value

ture if [field quotation optional]; otherwise, false.
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the field terminator.

**Namespace:**  MySql.Data.MySqlClient

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string FieldTerminator { get; set; }

Visual Basic (Declaration)

Public Property FieldTerminator As String

Visual C++

public:
    property String^ FieldTerminator {
        String^ get ();
        void set (String^ value);
    }

Field Value

The field terminator.
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the name of the file.

**Namespace:**  [MySQL.Data.MySqlClient](http://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string FileName { get; set; }

Visual Basic (Declaration)

Public Property FileName As String

Visual C++

public:
property String^ FileName {
    String^ get ();
    void set (String^ value);
}

Field Value

The name of the file.
See Also

MySQLBulkLoader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the line prefix.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public string LinePrefix { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property LinePrefix As String
```

**Visual C++**

```cpp
public:
property String^ LinePrefix { 
    String^ get ();
    void set (String^ value);
}
```

**Field Value**

The line prefix.
See Also

**MySqlBulkLoader Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C# Visual Basic
Visual C++
MySQL Connector/Net

MySQLBulkLoader:::LineTerminator Property

MySQLBulkLoader Class  See Also  Send Feedback

Gets or sets the line terminator.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public string LineTerminator { get; set; }
```

Visual Basic (Declaration)

Public Property LineTerminator As String

Visual C++

```cpp
public:
property string^ LineTerminator {
    string^ get ();
    void set (string^ value);
}
```

Field Value

The line terminator.
See Also

**MySqlBulkLoader Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a value indicating whether the filename that is to be loaded is local to the client or not.

**Namespace:** MySql.Data.MySqlClient  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public bool Local { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Local As Boolean
```

**Visual C++**

```cpp
public:
property bool Local {
    bool get ();
    void set (bool value);
}
```

**Field Value**

true if local; otherwise, false.
See Also

MySQLBulkLoader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the number of lines to skip.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public int NumberOfLinesToSkip { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property NumberOfLinesToSkip As Integer
```

**Visual C++**

```cpp
public:
property int NumberOfLinesToSkip {
    int get ();
    void set (int value);
}
```

**Field Value**

The number of lines to skip.
See Also

MySQLBulkLoader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the priority.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
## Syntax

**C#**

```csharp
public MySqlBulkLoaderPriority Priority { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Priority As MySqlBulkLoaderPriority
```

**Visual C++**

```cpp
public:
property MySqlBulkLoaderPriority Priority {
    MySqlBulkLoaderPriority get ();
    void set (MySqlBulkLoaderPriority value);
}
```

## Field Value

The priority.
See Also

MySQLBulkLoader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the name of the table.

**Namespace:**  [MySQL.Data.MySqlClient](http://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public string TableName { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property TableName As String
```

**Visual C++**

```cpp
public:
property String^ TableName {
  String^ get ();
  void set (String^ value);
}
```

### Field Value

The name of the table.
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the timeout.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Timeout { get; set; }

Visual Basic (Declaration)

Public Property Timeout As Integer

Visual C++

public:
property int Timeout {
    int get ();
    void set (int value);
}

Field Value

The timeout.
See Also

MySqlBulkLoader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public enum MySqlBulkLoaderConflictOption

Visual Basic (Declaration)

Public Enumeration MySqlBulkLoaderConflictOption

Visual C++

public enum class MySqlBulkLoaderConflictOption
### Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>This is the default and indicates normal operation. In the event of a LOCAL load, this is the same as ignore. When the data file is on the server, then a key conflict will cause an error to be thrown and the rest of the data file ignored.</td>
</tr>
<tr>
<td>Replace</td>
<td>Replace column values when a key conflict occurs.</td>
</tr>
<tr>
<td>Ignore</td>
<td>Ignore any rows where the primary key conflicts.</td>
</tr>
</tbody>
</table>
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public enum MySqlBulkLoaderPriority

Visual Basic (Declaration)

Public Enumeration MySqlBulkLoaderPriority

Visual C++

public enum class MySqlBulkLoaderPriority
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>This is the default and indicates normal priority. Low priority will cause the load operation to wait until all readers of the table have finished. This only affects storage engines that use only table-level locking such as MyISAM, Memory, and Merge.</td>
</tr>
<tr>
<td>Low</td>
<td>Concurrent priority is only relevant for MyISAM tables and signals that if the table has no free blocks in the middle that other readers can retrieve data from the table while the load operation is happening.</td>
</tr>
</tbody>
</table>
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public enum MySqlCertificateStoreLocation

Visual Basic (Declaration)

Public Enumeration MySqlCertificateStoreLocation

Visual C++

public enum class MySqlCertificateStoreLocation
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Do not use certificate store</td>
</tr>
<tr>
<td>CurrentUser</td>
<td>Use certificate store for the current user</td>
</tr>
<tr>
<td>LocalMachine</td>
<td>User certificate store for the machine</td>
</tr>
</tbody>
</table>
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLClientFactory Class

DBProviderFactory implementation for MysqlClient.


**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public sealed class MySqlClientFactory : DbProviderFactory,
IServiceProvider
```

**Visual Basic (Declaration)**

```vbnet
Public NotInheritable Class MySqlClientFactory
    Inherits DbProviderFactory
    Implements IServiceProvider
```

**Visual C++**

```cpp
public ref class MySqlClientFactory sealed : public DbProviderFactory,
IServiceProvider
```
Inheritance Hierarchy

System:::Object
System.Data.Common:::DbProviderFactory
MySql.Data.MySqlClient:::MySqlClientFactory
See Also

MySqlConnection

MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlClientFactory` type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlClientFactory</td>
<td></td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateCommand</td>
<td>Returns a strongly typed DbCommand instance. (Overrides DbProviderFactory::&lt; DbCommand CreateCommand</td>
</tr>
<tr>
<td>CreateCommandBuilder</td>
<td>Returns a strongly typed DbCommandBuilder instance. (Overrides DbProviderFactory::&lt; DbCommandBuilder CreateCommandBuilder</td>
</tr>
<tr>
<td>CreateConnection</td>
<td>Returns a strongly typed DbConnection instance. (Overrides DbProviderFactory::&lt; DbConnection CreateConnection</td>
</tr>
<tr>
<td>CreateConnectionStringBuilder</td>
<td>Returns a strongly typed DbConnectionStringBuilder instance. (Overrides DbProviderFactory::&lt; DbConnectionStringBuilder CreateConnectionStringBuilder</td>
</tr>
<tr>
<td>CreateDataAdapter</td>
<td>Returns a strongly typed DbDataAdapter instance. (Overrides DbProviderFactory::&lt; DbDataAdapter CreateDataAdapter</td>
</tr>
<tr>
<td>CreateDataSourceEnumerator</td>
<td>Returns a new instance of the provider's class that implements the DbDataSourceEnumerator class. (Inherited from DbProviderFactory.</td>
</tr>
<tr>
<td>CreateParameter</td>
<td>Returns a strongly typed DbParameter instance. (Overrides DbProviderFactory::&lt; DbParameter CreateParameter</td>
</tr>
<tr>
<td>CreatePermission</td>
<td>Returns a new instance of the provider's class that implements the provider's version of the CodeAccessPermission class. (Inherited from DbProviderFactory.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources other cleanup operations before the Object is garbage collected. (Inherited from Object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Gets an instance of the <a href="#">MySqlClientFactory</a>. This can be used to retrieve strongly typed data objects.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------------------------</td>
</tr>
<tr>
<td>CanCreateDataSourceEnumerator</td>
<td>Returns true if a <code>MySqlDataSourceEnumerator</code> created; otherwise false. (Overrides <code>DbProviderFactory&lt;&gt;::CanCreateDataSourceEnumerator</code>)</td>
</tr>
</tbody>
</table>
See Also

MySQLClientFactory Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlClientFactory()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySqlClientFactory()
See Also

MySQLClientFactory Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlClientFactory` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Gets an instance of the <a href="#">MySqlClientFactory</a>. This can be used to retrieve strongly typed data objects.</td>
</tr>
</tbody>
</table>
See Also

MySQLClientFactory Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets an instance of the `MySqlConnection`. This can be used to retrieve strongly typed data objects.

**Namespace:**  `MySQL.Data.MySqlClient`  
**Assembly:**  `MySql.Data (in MySql.Data.dll) Version: 6.2.2.0`
Syntax

C#

public static MySqlClientFactory Instance

Visual Basic (Declaration)

Public Shared Instance As MySqlClientFactory

Visual C++

public:
static MySqlClientFactory^ Instance
See Also

MySQLClientFactory Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlClientFactory` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CreateCommand</strong></td>
<td>Returns a strongly typed <code>DbCommand</code> instance.</td>
</tr>
<tr>
<td>(Overrides <code>DbProviderFactory</code>:::<code>CreateCommand</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>CreateCommandBuilder</strong></td>
<td>Returns a strongly typed <code>DbCommandBuilder</code> instance.</td>
</tr>
<tr>
<td>(Overrides <code>DbProviderFactory</code>:::<code>CreateCommandBuilder</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>CreateConnection</strong></td>
<td>Returns a strongly typed <code>DbConnection</code> instance.</td>
</tr>
<tr>
<td>(Overrides <code>DbProviderFactory</code>:::<code>CreateConnection</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>CreateConnectionStringBuilder</strong></td>
<td>Returns a strongly typed <code>DbConnectionStringBuilder</code> instance.</td>
</tr>
<tr>
<td>(Overrides <code>DbProviderFactory</code>:::<code>CreateConnectionStringBuilder</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>CreateDataAdapter</strong></td>
<td>Returns a strongly typed <code>DbDataAdapter</code> instance.</td>
</tr>
<tr>
<td>(Overrides <code>DbProviderFactory</code>:::<code>CreateDataAdapter</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>CreateDataSourceEnumerator</strong></td>
<td>Returns a new instance of the provider's class that implements the <code>DbDataSourceEnumerator</code> class.</td>
</tr>
<tr>
<td>(Inherited from <code>DbProviderFactory</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>CreateParameter</strong></td>
<td>Returns a strongly typed <code>DbParameter</code> instance.</td>
</tr>
<tr>
<td>(Overrides <code>DbProviderFactory</code>:::<code>CreateParameter</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>CreatePermission</strong></td>
<td>Returns a new instance of the provider's class that implements the <code>CodeAccessPermission</code> class.</td>
</tr>
<tr>
<td>(Inherited from <code>DbProviderFactory</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is eq current <code>Object</code>.</td>
</tr>
<tr>
<td>(Inherited from <code>Object</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources other cleanup operations before the <code>Object</code> is garbage collection.</td>
</tr>
<tr>
<td>(Inherited from <code>Object</code>.)</td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>(Inherited from <code>Object</code>.)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlConnectionFactory Class**

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns a strongly typed `DbCommand` instance.

**Namespace:**  `MySql.Data.MySqlClient`

**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public override DbCommand CreateCommand()

Visual Basic (Declaration)

Public Overrides Function CreateCommand As DbCommand

Visual C++

public:
virtual DbCommand^ CreateCommand() override

Return Value

A new strongly typed instance of DbCommand.
See Also

MySQLClientFactory Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlClientFactory...::.:CreateCommandBuilder Method

MySqlClientFactory Class  See Also  Send Feedback

Returns a strongly typed DbCommandBuilder instance.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override DbCommandBuilder CreateCommandBuilder()
```

### Visual Basic (Declaration)

Public Overrides Function CreateCommandBuilder As DbCommandBuilder

### Visual C++

```cpp
public:
virtual DbCommandBuilder^ CreateCommandBuilder() override
```

## Return Value

A new strongly typed instance of `DbCommandBuilder`. 
See Also

MySqlConnectionClass
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlConnectionFactory..:: CreateConnection Method

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/connector-net/

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0

Returns a strongly typed [DbConnection](https://www.mysql.com/connector-net/) instance.
Syntax

C#

public override DbConnection CreateConnection()

Visual Basic (Declaration)

Public Overrides Function CreateConnection As DbConnection

Visual C++

public:
virtual DbConnection^ CreateConnection() override

Return Value

A new strongly typed instance of DbConnection.
See Also

MySqlConnectionClass
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnectionFactory...::CreateConnectionStringBuilder Method

MySqlConnectionFactory Class  See Also  Send Feedback

Returns a strongly typed DbConnectionStringBuilder instance.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

public override DbConnectionStringBuilder CreateConnectionStringBuilder

**Visual Basic (Declaration)**

Public Overrides Function CreateConnectionStringBuilder As DbConnect

**Visual C++**

public:
virtual DbConnectionStringBuilder^ CreateConnectionStringBuilder() 

**Return Value**

A new strongly typed instance of DbConnectionStringBuilder.
See Also

MySQLClientFactory Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySQLClientFactory..::.CreateDataAdapter Method

MySQLClientFactory Class  See Also  Send Feedback

Returns a strongly typed DbDataAdapter instance.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public override DbDataAdapter CreateDataAdapter()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function CreateDataAdapter As DbDataAdapter
```

**Visual C++**

```cpp
public:
virtual DbDataAdapter ^ CreateDataAdapter() override
```

**Return Value**

A new strongly typed instance of `DbDataAdapter`. 
See Also

MySqlConnectionFactory Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlClientFactory:::CreateParameter Method

MySqlClientFactory Class  See Also  Send Feedback

Returns a strongly typed DbParameter instance.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override DbParameter CreateParameter()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function CreateParameter As DbParameter
```

**Visual C++**

```cpp
public:
virtual DbParameter^ CreateParameter() override
```

**Return Value**

A new strongly typed instance of `DbParameter`. 
See Also

MySQLClientFactory Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLClientFactory** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanCreateDataSourceEnumerator</td>
<td>Returns true if a <code>MySqlDataSourceEnumerator</code> created; otherwise false. (Overrides <code>DbProviderFactory...CanCreateDataSourc</code></td>
</tr>
</tbody>
</table>

See Also

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlClientFactory...:CanCreateDataSourceEnumerator Property

MySqlClientFactory Class  See Also  Send Feedback

Returns true if a **MySqlDataSourceEnumerator** can be created; otherwise false.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override bool CanCreateDataSourceEnumerator { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property CanCreateDataSourceEnumerator As

Visual C++

public:
virtual property bool CanCreateDataSourceEnumerator {
    bool get () override;
}
See Also

MysqlClientFactory Class
Mysql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents a SQL statement to execute against a MySQL database. This class cannot be inherited.

**Namespace:**  [MySQL.Data.MySqlClient](https://github.com/mysql-connector-net/mysql-connector-net)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public sealed class MySqlCommand : DbCommand,
    ICloneable

Visual Basic (Declaration)

Public NotInheritable Class MySqlCommand
    Inherits DbCommand
    Implements ICloneable

Visual C++

public ref class MySqlCommand sealed : public DbCommand,
    ICloneable
Remarks

MySqlCommand features the following methods for executing commands at a MySQL database:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecuteReader</td>
<td>Executes commands that return rows.</td>
</tr>
<tr>
<td>ExecuteNonQuery</td>
<td>Executes commands such as SQL INSERT, DELETE, and UPDATE statements.</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td>Retrieves a single value (for example, an aggregate value) from a database.</td>
</tr>
</tbody>
</table>

You can reset the CommandText property and reuse the MySqlCommand object. However, you must close the MySqlDataReader before you can execute a new or previous command. If a MySqlException is generated by the method executing a MySqlCommand, the MySqlConnection remains open. It is the responsibility of the programmer to close the connection.

❖ **Note:** Using the '@' symbol for parameters is now the preferred approach although the old pattern of using '?' is still supported. Please be aware though that using '@' can cause conflicts when user variables are also used. To help with this situation please see the documentation on the 'allow user variables' connection string option. The 'old syntax' connection string option has now been deprecated.
Examples

The following example creates a MySqlCommand and a MySqlConnection. The MySqlConnection is opened and set as the Connection for the MySqlCommand. The example then calls ExecuteNonQuery, and closes the connection. To accomplish this, the ExecuteNonQuery is passed a connection string and a query string that is a SQL INSERT statement.

**VB.NET**

```vbnet
Public Sub InsertRow(myConnectionString As String)
" If the connection string is null, use a default.
If myConnectionString = "" Then
  myConnectionString = "Database=Test;Data Source=localhost;User Id=username;Password=pass"
End If
Dim myConnection As New MySqlConnection(myConnectionString)
Dim myInsertQuery As String = "INSERT INTO Orders (id, customerId, amount) Values(1001, 23, 30.66)"
Dim myCommand As New MySqlCommand(myInsertQuery)
myCommand.Connection = myConnection
myConnection.Open()
myCommand.ExecuteNonQuery()
myConnection.Close()
End Sub
```

**C#**

```csharp
public void InsertRow(string myConnectionString)
{
  // If the connection string is null, use a default.
  if (myConnectionString == "")
  {
    myConnectionString = "Database=Test;Data Source=localhost;User Id=username;Password=pass"
  }
  MySqlConnection myConnection = new MySqlConnection(myConnectionString)
  string myInsertQuery = "INSERT INTO Orders (id, customerId, amount) Values(1001, 23, 30.66)"
  MySqlCommand myCommand = new MySqlCommand(myInsertQuery); //myConnectionString
  myCommand.Connection = myConnection;
  myConnection.Open();
  myCommand.ExecuteNonQuery();
  myCommand.Connection.Close();
}
```
Inheritance Hierarchy

System..::.Object
   System..::.MarshalByRefObject
      System.ComponentModel..::.Component
         System.Data.Common..::.DbCommand
            MySql.Data.MySqlClient..::.MySqlCommand
See Also

MySQLCommand Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLCommand` type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **MySqlCommand** | Overloaded.  
<pre><code>           | Initializes a new instance of the MySqlCommand class. |
</code></pre>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeginExecuteNonQuery</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>BeginExecuteReader</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Attempts to cancel the execution of a currently active command (Overrides DbCommand:::.Cancel()()().)</td>
</tr>
<tr>
<td>Clone</td>
<td>Creates a clone of this MySqlCommand object. CommandText, Connection, and Transaction properties are included as well as the entire parameter list.</td>
</tr>
<tr>
<td>CreateDbParameter</td>
<td>Creates a new instance of a DbParameter object. (Inherited from DbCommand.)</td>
</tr>
<tr>
<td>CreateObjRef</td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>CreateParameter</td>
<td>Creates a new instance of a MySqlParameter object.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>EndExecuteNonQuery</td>
<td>Finishes asynchronous execution of a SQL statement.</td>
</tr>
<tr>
<td>EndExecuteReader</td>
<td>Finishes asynchronous execution of a SQL statement, returning the requested MySqlDataReader.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ExecuteDbDataReader</td>
<td>Executes the command text against the connection. (Inherited from DbCommand.)</td>
</tr>
<tr>
<td></td>
<td>Executes a SQL statement against the connection.</td>
</tr>
</tbody>
</table>
ExecuteNonQuery
connection and returns the number of rows affected.
(Overrides DbCommand::<code>ExecuteNonQuery()()</code>.)

ExecuteReader
Overloaded.
Executes the query, and returns the first column of the first row in the result set returned by the query. Extra columns or rows are ignored.
(Overrides DbCommand::<code>ExecuteScalar()()</code>.)

ExecuteScalar
Releases unmanaged resources and performs other cleanup operations before the
<code>Component</code> is reclaimed by garbage collection.
(Inherited from <code>Component</code>.)

Finalize
Serves as a hash function for a particular type.
(Inherited from <code>Object</code>.)
Retrieves the current lifetime service object that controls the lifetime policy for this instance.
(Inherited from <code>MarshalByRefObject</code>.)

GetHashCode
Returns an object that represents a service provided by the <code>Component</code> or by its
<code>Container</code>.
(Inherited from <code>Component</code>.)

GetLifetimeService
Gets the <code>Type</code> of the current instance.
(Inherited from <code>Object</code>.)
Obtains a lifetime service object to control the lifetime policy for this instance.
(Inherited from <code>MarshalByRefObject</code>.)

InitializeLifetimeService
Overloaded.

MemberwiseClone
Creates a prepared version of the command on an instance of MySQL Server.
(Overrides DbCommand::<code>Prepare()()</code>.)

Prepare
Returns a <code>String</code> containing the name of the <code>Component</code>, if any. This method should not be
ToString overridden.
(Inherited from Component.)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanRaiseEvents</td>
<td>Gets a value indicating whether the component can raise an event. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>CommandText</td>
<td>Gets or sets the SQL statement to execute at the data source. (Overrides <a href="#">DbCommand..::.CommandText</a>.)</td>
</tr>
<tr>
<td>CommandTimeout</td>
<td>Gets or sets the wait time before terminating the attempt to execute a command and generating an error. (Overrrides <a href="#">DbCommand..::.CommandTimeout</a>.)</td>
</tr>
<tr>
<td>CommandType</td>
<td>Gets or sets a value indicating how the <a href="#">CommandText</a> property is to be interpreted. (Overrrides <a href="#">DbCommand..::.CommandType</a>.)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets or sets the <a href="#">MySqlConnection</a> used by this instance of the <a href="#">MySqlCommand</a>.</td>
</tr>
<tr>
<td>Container</td>
<td>Gets the <a href="#">IContainer</a> that contains the <a href="#">Component</a>. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>DbConnection</td>
<td>Gets or sets the <a href="#">DbConnection</a> used by this <a href="#">DbCommand</a>. (Inherited from <a href="#">DbCommand</a>.)</td>
</tr>
<tr>
<td>DbParameterCollection</td>
<td>Gets the collection of <a href="#">DbParameter</a> objects. (Inherited from <a href="#">DbCommand</a>.)</td>
</tr>
<tr>
<td>DbTransaction</td>
<td>Gets or sets the <a href="#">DbTransaction</a> within which this <a href="#">DbCommand</a> object executes. (Inherited from <a href="#">DbCommand</a>.)</td>
</tr>
<tr>
<td>DesignMode</td>
<td>Gets a value that indicates whether the <a href="#">Component</a> is currently in design mode. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>DesignTimeVisible</strong></td>
<td>Command object should be visible in a Windows Form Designer control. (Overrides <strong>DbCommand::::DesignTimeVisible</strong>.)</td>
</tr>
<tr>
<td><strong>Events</strong></td>
<td>Gets the list of event handlers that are attached to this <strong>Component</strong>. (Inherited from <strong>Component</strong>.)</td>
</tr>
<tr>
<td><strong>IsPrepared</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LastInsertedId</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>Get the <strong>MySqlParameterCollection</strong>.</td>
</tr>
<tr>
<td><strong>Site</strong></td>
<td>Gets or sets the <strong>ISite</strong> of the <strong>Component</strong>. (Inherited from <strong>Component</strong>.)</td>
</tr>
<tr>
<td><strong>Transaction</strong></td>
<td>Gets or sets the <strong>MySQLTransaction</strong> within which the <strong>MySqlCommand</strong> executes.</td>
</tr>
<tr>
<td><strong>UpdatedRowSource</strong></td>
<td>Gets or sets how command results are applied to the <strong>DataRow</strong> when used by the <strong>Update</strong> method of the <strong>DbDataAdapter</strong>. (Overrides <strong>DbCommand::::UpdatedRowSource</strong>.)</td>
</tr>
</tbody>
</table>
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposed</td>
<td>Occurs when the component is disposed by a call to the Dispose() method. (Inherited from Component.)</td>
</tr>
</tbody>
</table>
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the MySqlCommand class.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlCommand()()</td>
<td>Initializes a new instance of the MySqlCommand class.</td>
</tr>
<tr>
<td>MySqlCommand(String)</td>
<td>Initializes a new instance of the MySqlCommand class with the text of the query.</td>
</tr>
<tr>
<td>MySqlCommand(String, MySqlConnection)</td>
<td>Initializes a new instance of the MySqlCommand class with the text of the query and a MySqlConnection.</td>
</tr>
<tr>
<td>MySqlCommand(String, MySqlConnection, MySqlTransaction)</td>
<td>Initializes a new instance of the MySqlCommand class with the text of the query, a MySqlConnection, and the MySqlTransaction.</td>
</tr>
</tbody>
</table>
Examples

The following example creates a MySqlCommand and sets some of its properties.

**Note:** This example shows how to use one of the overloaded versions of the MySqlCommand constructor. For other examples that might be available, see the individual overload topics.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand()
    Dim myConnection As New MySqlConnection("
    "Persist Security Info=False;database=test;server=myServer")
    myConnection.Open()
    Dim myTrans As MySqlTransaction = myConnection.BeginTransaction()
    Dim mySelectQuery As String = "SELECT * FROM MyTable"
    Dim myCommand As New MySqlCommand(mySelectQuery, myConnection, myTrans)
    myCommand.CommandTimeout = 20
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand()
{
    MySqlConnection myConnection = new MySqlConnection("Persist Security 
    database=test;server=myServer");
    myConnection.Open();
    MySqlTransaction myTrans = myConnection.BeginTransaction();
    string mySelectQuery = "SELECT * FROM MyTable";
    MySqlCommand myCommand = new MySqlCommand(mySelectQuery, myConnectio 
    myCommand.CommandTimeout = 20;
}
```

**C++**

```c++
public:
void CreateMySqlCommand()
{
    MySqlConnection* myConnection = new MySqlConnection(S"Persist Securi 
    database=test;server=myServer");
    myConnection->Open();
    MySqlTransaction* myTrans = myConnection->BeginTransaction();
```
String* mySelectQuery = S"SELECT * FROM myTable";
MySqlCommand* myCommand = new MySqlCommand(mySelectQuery, myConnection);
myCommand->CommandTimeout = 20;
};
See Also

MySqlCommand Class
MySqlCommand Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the MySqlCommand class.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCommand()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySqlCommand()
Remarks

The base constructor initializes all fields to their default values. The following table shows initial property values for an instance of `MySqlCommand`.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CommandText</code></td>
<td>empty string (&quot;&quot;)</td>
</tr>
<tr>
<td><code>CommandTimeout</code></td>
<td>0</td>
</tr>
<tr>
<td><code>CommandType</code></td>
<td><code>CommandType.Text</code></td>
</tr>
<tr>
<td><code>Connection</code></td>
<td><code>Null</code></td>
</tr>
</tbody>
</table>

You can change the value for any of these properties through a separate call to the property.
**Examples**

The following example creates a `MySqlCommand` and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand()
    Dim myCommand As New MySqlCommand()
    myCommand.CommandType = CommandType.Text
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand()
    {
    MySqlCommand myCommand = new MySqlCommand();
    myCommand.CommandType = CommandType.Text;
    }
```
See Also

**MySqlCommand Class**
**MySqlCommand Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlCommand Constructor (String)

Initializes a new instance of the `MySqlCommand` class with the text of the query.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public MySqlCommand(
    string cmdText
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New (_
    cmdText As String _
)
```

**Visual C++**

```cpp
public:
MySqlCommand(
    String^ cmdText
)
```

**Parameters**

cmdText

Type: `System::String`
The text of the query.
Remarks

When an instance of ` MySqlCommand` is created, the following read/write properties are set to initial values.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommandText</td>
<td><code>cmdText</code></td>
</tr>
<tr>
<td>CommandTimeout</td>
<td>0</td>
</tr>
<tr>
<td>CommandType</td>
<td>CommandType.Text</td>
</tr>
<tr>
<td>Connection</td>
<td>Null</td>
</tr>
</tbody>
</table>

You can change the value for any of these properties through a separate call to the property.
Examples

The following example creates a `MySQLCommand` and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand()
    Dim sql as String = "SELECT * FROM mytable"
    Dim myCommand As New MySqlCommand(sql)
    myCommand.CommandType = CommandType.Text
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand()
{
    string sql = "SELECT * FROM mytable";
    MySqlCommand myCommand = new MySqlCommand(sql);
    myCommand.CommandType = CommandType.Text;
}
```
See Also

MySqlCommand Class
MySqlCommand Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

`MySqlCommand Constructor (String, MySqlConnection)`

**`MySqlCommand` Class**  
**Example**  
**See Also**  
**Send Feedback**

Initializes a new instance of the **` MySqlCommand`** class with the text of the query and a **` MySqlConnection`**.

**Namespace:**  
`MySQL.Data.MySqlClient`

**Assembly:**  
`MySQL.Data` (in `MySQL.Data.dll`) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlCommand(  
    string cmdText,  
    MySqlConnection connection
)
```

Visual Basic (Declaration)

```vbnet
Public Sub New (_  
    cmdText As String, _  
    connection As MySqlConnection _
)
```

Visual C++

```cpp
public:  
    MySqlCommand(  
        String^ cmdText,  
        MySqlConnection^ connection
    )
```

Parameters

cmdText
Type: `System::String`
The text of the query.

connection
Type: `MySql.Data.MySqlClient::MySqlConnection`  
A `MySqlConnection` that represents the connection to an instance of SQL Server.
### Remarks

When an instance of `MySqlCommand` is created, the following read/write properties are set to initial values.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CommandText</code></td>
<td><code>cmdText</code></td>
</tr>
<tr>
<td><code>CommandTimeout</code></td>
<td>0</td>
</tr>
<tr>
<td><code>CommandType</code></td>
<td><code>CommandType.Text</code></td>
</tr>
<tr>
<td><code>Connection</code></td>
<td><code>connection</code></td>
</tr>
</tbody>
</table>

You can change the value for any of these properties through a separate call to the property.
**Examples**

The following example creates a `MySqlCommand` and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand()
    Dim conn as new MySqlConnection("server=myServer")
    Dim sql as String = "SELECT * FROM mytable"
    Dim myCommand As New MySqlCommand(sql, conn)
    myCommand.CommandType = CommandType.Text
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand()
{
    MySqlConnection conn = new MySqlConnection("server=myserver")
    string sql = "SELECT * FROM mytable";
    MySqlCommand myCommand = new MySqlCommand(sql, conn);
    myCommand.CommandType = CommandType.Text;
}
```
See Also

MySqlCommand Class
MySqlCommand Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlCommand` class with the text of the query, a `MySqlConnection`, and the `MySqlTransaction`.

**Namespace:**  `MySQL.Data.MySqlClient`  
**Assembly:**  `MySql.Data (in MySql.Data.dll)` Version: 6.2.2.0
Syntax

C#

public MySqlCommand(
    string cmdText,
    MySqlConnection connection,
    MySqlTransaction transaction
)

Visual Basic (Declaration)

Public Sub New ( _
    cmdText As String, _
    connection As MySqlConnection, _
    transaction As MySqlTransaction _
)

Visual C++

public:
MySqlCommand(
    String^ cmdText,
    MySqlConnection^ connection,
    MySqlTransaction^ transaction
)

Parameters

cmdText
Type: System::String
The text of the query.

connection
Type: MySql.Data.MySqlClient::MySqlConnection
A MySqlConnection that represents the connection to an instance of SQL Server.

transaction
Type: MySql.Data.MySqlClient..:::MySqlTransaction
The MySqlTransaction in which the MySqlCommand executes.
### Remarks

When an instance of `MySqlCommand` is created, the following read/write properties are set to initial values.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CommandText</code></td>
<td><code>cmdText</code></td>
</tr>
<tr>
<td><code>CommandTimeout</code></td>
<td>0</td>
</tr>
<tr>
<td><code>CommandType</code></td>
<td><code>CommandType.Text</code></td>
</tr>
<tr>
<td><code>Connection</code></td>
<td><code>connection</code></td>
</tr>
</tbody>
</table>

You can change the value for any of these properties through a separate call to the property.
Examples

The following example creates a **MySqlCommand** and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand()
    Dim conn as new MySqlConnection("server=myServer")
    conn.Open();
    Dim txn as MySqlTransaction = conn.BeginTransaction()
    Dim sql as String = "SELECT * FROM mytable"
    Dim myCommand As New MySqlCommand(sql, conn, txn)
    myCommand.CommandType = CommandType.Text
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand()
{
    MySqlConnection conn = new MySqlConnection("server=myserver")
    conn.Open();
    MySqlTransaction txn = conn.BeginTransaction();
    string sql = "SELECT * FROM mytable";
    MySqlCommand myCommand = new MySqlCommand(sql, conn, txn);
    myCommand.CommandType = CommandType.Text;
}
```
See Also

SqlConnection Class
SqlCommand Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlCommand` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BeginExecuteNonQuery</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>BeginExecuteReader</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>Cancel</code></td>
<td>Attempts to cancel the execution of a currently active command</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbCommand..::.Cancel()</code>.)</td>
</tr>
<tr>
<td><code>Clone</code></td>
<td>Creates a clone of this MySqlCommand object. CommandText, Connection, and Transaction properties are included as well as the entire parameter list.</td>
</tr>
<tr>
<td><code>CreateDbParameter</code></td>
<td>Creates a new instance of a <code>DbParameter</code> object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DbCommand</code>.)</td>
</tr>
<tr>
<td><code>CreateObjRef</code></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><code>CreateParameter</code></td>
<td>Creates a new instance of a <code>MySqlParameter</code> object.</td>
</tr>
<tr>
<td><code>Dispose</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>EndExecuteNonQuery</code></td>
<td>Finishes asynchronous execution of a SQL statement.</td>
</tr>
<tr>
<td><code>EndExecuteReader</code></td>
<td>Finishes asynchronous execution of a SQL statement, returning the requested <code>MySqlDataReader</code>.</td>
</tr>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>Executes the command text against the connection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DbCommand</code>.)</td>
</tr>
<tr>
<td><code>ExecuteDbDataReader</code></td>
<td>Executes a SQL statement against the connection.</td>
</tr>
</tbody>
</table>
**ExecuteNonQuery**
connection and returns the number of rows affected.
(Overrides
**DbCommand::<ExecuteNonQuery()>**)

**ExecuteReader**
Overloaded.
Executes the query, and returns the first column of the first row in the result set returned by the query. Extra columns or rows are ignored.
(Overrides **DbCommand::<ExecuteScalar>()**)

**ExecuteScalar**
Releases unmanaged resources and performs other cleanup operations before the

**Finalize**
**Component** is reclaimed by garbage collection.
(Inherited from **Component**)

**GetHashCode**
Serves as a hash function for a particular type.
(Inherited from **Object**)
Retrieves the current lifetime service object that controls the lifetime policy for this instance.
(Inherited from **MarshalByRefObject**)

**GetLifetimeService**
Returns an object that represents a service provided by the **Component** or by its

**GetService**
**Container**.
(Inherited from **Component**)

**GetType**
Gets the **Type** of the current instance.
(Inherited from **Object**)
Obtains a lifetime service object to control the lifetime policy for this instance.
(Inherited from **MarshalByRefObject**)

**InitializeLifetimeService**
Overloaded.

**MemberwiseClone**
Creates a prepared version of the command on an instance of MySQL Server.
(Overrides **DbCommand::<Prepare>()**)

**Prepare**
Returns a **String** containing the name of the **Component**, if any. This method should not be
ToString overridden.
(Inherited from Component.)
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Command...::BeginExecuteNonQuery Method

MySQLCommand Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BeginExecuteNonQuery()</code></td>
<td>Initiates the asynchronous execution of the SQL statement or stored procedure that is described by this <code>MySqlCommand</code>.</td>
</tr>
<tr>
<td><code>BeginExecuteNonQuery(AsyncCallback, Object)</code></td>
<td>Initiates the asynchronous execution of the SQL statement or stored procedure that is described by this <code>MySqlCommand</code>.</td>
</tr>
</tbody>
</table>
See Also

**MySqlCommand Class**
**MySqlCommand Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initiates the asynchronous execution of the SQL statement or stored procedure that is described by this `MySqlCommand`.

**Namespace:** MySql.Data.MySqlClient  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public IAsyncResult BeginExecuteNonQuery()
```

**Visual Basic (Declaration)**

```vbnet
Public Function BeginExecuteNonQuery As IAsyncResult
```

**Visual C++**

```cpp
public:
IAsyncResult^ BeginExecuteNonQuery()
```

**Return Value**

An `IAsyncResult` that can be used to poll or wait for results, or both; this value is also needed when invoking `EndExecuteNonQuery(IAsyncResult)`, which returns the number of affected rows.
See Also

**MySqlCommand Class**

*BeginExecuteNonQuery Overload*

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Command..::..BeginExecuteNonQuery Method (AsyncCallback, Object)

MySqlCommand Class  See Also  Send Feedback

Initiates the asynchronous execution of the SQL statement or stored procedure that is described by this MySqlCommand.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public IAsyncResult BeginExecuteNonQuery(
    AsyncCallback callback,
    Object stateObject
)
```

Visual Basic (Declaration)

```vbnet
Public Function BeginExecuteNonQuery ( _
    callback As AsyncCallback, _
    stateObject As Object _
) As IAsyncResult
```

Visual C++

```cpp
public: 
    IAsyncResult^ BeginExecuteNonQuery(
        AsyncCallback^ callback,
        Object^ stateObject
    )
```

Parameters

callback
Type: System::AsyncCallback
An AsyncCallback delegate that is invoked when the command's execution
has completed. Pass a null reference (Nothing in Visual Basic) to indicate
that no callback is required.

stateObject
Type: System::Object
A user-defined state object that is passed to the callback procedure. Retrieve
this object from within the callback procedure using the AsyncState
property.
Return Value

An `IAsyncResult` that can be used to poll or wait for results, or both; this value is also needed when invoking `EndExecuteNonQuery(IAsyncResult)`, which returns the number of affected rows.
See Also

MySQLCommand Class
BeginExecuteNonQuery Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BeginExecuteReader()</code></td>
<td>Initiates the asynchronous execution of the SQL statement or stored procedure that is described by this <a href="#">MySqlCommand</a>, and retrieves one or more result sets from the server.</td>
</tr>
<tr>
<td><code>BeginExecuteReader(CommandBehavior)</code></td>
<td>Initiates the asynchronous execution of the SQL statement or stored procedure that is described by this <a href="#">MySqlCommand</a> using one of the <code>CommandBehavior</code> values.</td>
</tr>
</tbody>
</table>
See Also

**MySqlCommand Class**  
**MySqlCommand Members**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
**Initiates the asynchronous execution of the SQL statement or stored procedure that is described by this `MySqlCommand`, and retrieves one or more result sets from the server.**

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data (in MySql.Data.dll) Version: 6.2.2.0`
Syntax

C#

public IAsyncResult BeginExecuteReader()

Visual Basic (Declaration)

Public Function BeginExecuteReader As IAsyncResult

Visual C++

public: IAsyncResult^ BeginExecuteReader()

Return Value

An IAsyncResult that can be used to poll, wait for results, or both; this value is also needed when invoking EndExecuteReader, which returns a MySqlDataReader instance that can be used to retrieve the returned rows.
See Also

MySqlCommand Class
BeginExecuteReader Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initiates the asynchronous execution of the SQL statement or stored procedure that is described by this `MySqlCommand` using one of the `CommandBehavior` values.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public IAsyncResult BeginExecuteReader(
    CommandBehavior behavior
)
```

Visual Basic (Declaration)

```vbnet
Public Function BeginExecuteReader (_
    behavior As CommandBehavior _
) As IAsyncResult
```

Visual C++

```cpp
public:
IAsyncResult^ BeginExecuteReader(
    CommandBehavior behavior
)
```

Parameters

behavior

Type: System.Data::CommandBehavior

One of the CommandBehavior values, indicating options for statement execution and data retrieval.

Return Value

An IAsyncResult that can be used to poll, wait for results, or both; this value is also needed when invoking EndExecuteReader, which returns a MySqlDataReader instance that can be used to retrieve the returned rows.
See Also

MySqlCommand Class
BeginExecuteReader Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Attempts to cancel the execution of a currently active command

**Namespace:**  [MySQL.Data.MySqlClient](https://docs.mysql.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

C#

```csharp
public override void Cancel()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Cancel
```

**Visual C++**

```cpp
public:
virtual void Cancel() override
```

**Implements**

```
IDbCommand::Cancel()();
```
Remarks

Cancelling a currently active query only works with MySQL versions 5.0.0 and higher.
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C# Visual Basic
Visual C++
MySQL Connector/Net

MySqlCommand..::.Clone Method

MySqlCommand Class  See Also  Send Feedback

Creates a clone of this MySqlCommand object. CommandText, Connection, and Transaction properties are included as well as the entire parameter list.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCommand Clone()

Visual Basic (Declaration)

Public Function Clone As MySqlCommand

Visual C++

public: 
MySqlCommand^ Clone()

Return Value

The cloned MySqlCommand object
See Also

MySQLCommand Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Command...:CreateParameter Method

MySQLCommand Class  See Also  Send Feedback

Creates a new instance of a **MySQLParameter** object.

**Namespace:**  [MySQL.Data.MySqlClient]

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlParameter CreateParameter()

Visual Basic (Declaration)

Public Function CreateParameter As MySqlParameter

Visual C++

public:
MySqlParameter^ CreateParameter()

Return Value

A MySqlParameter object.
Remarks

This method is a strongly-typed version of CreateParameter().
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose()</td>
<td>Releases all resources used by the Component. (Inherited from Component.)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases the unmanaged resources used by the Component and optionally releases the managed resources. (Inherited from Component.)</td>
</tr>
</tbody>
</table>
See Also

MySqlCommand Class
MySqlCommand Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Finishes asynchronous execution of a SQL statement.

**Namespace:**  [MySQL.Data.MySqlClient](https://example.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public int EndExecuteNonQuery(
    IAsyncResult asyncResult
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function EndExecuteNonQuery (_
    asyncResult As IAsyncResult _
) As Integer
```

**Visual C++**

```cpp
public:
    int EndExecuteNonQuery(
        IAsyncResult^ asyncResult
    )
```

**Parameters**

asyncResult

Type: `System::::IAsyncResult`

The `IAsyncResult` returned by the call to `BeginExecuteNonQuery()`.

**Return Value**
See Also

[Link to MySqlCommand Class]
[Link to MySql.Data.MySqlClient Namespace]

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Finishes asynchronous execution of a SQL statement, returning the requested `MySqlDataReader`.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlDataReader EndExecuteReader(IAsyncResult result)
```

Visual Basic (Declaration)

```vbnet
Public Function EndExecuteReader(_
    result As IAsyncResult _) As MySqlDataReader
```

Visual C++

```cpp
public:
    MySqlDataReader^ EndExecuteReader(IAsyncResult^ result)
```

Parameters

result

Type: `System::::IAsyncResult`

The `IAsyncResult` returned by the call to `BeginExecuteReader()`.

Return Value

A `MySqlDataReader` object that can be used to retrieve the requested rows.
See Also

`MySqlCommand Class`
`MySql.Data.MySqlClient Namespace`

Send comments on this topic to `support@mysql.com`

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
ExecuteNonQuery Method

Executes a SQL statement against the connection and returns the number of rows affected.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override int ExecuteNonQuery()
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function ExecuteNonQuery As Integer
```

Visual C++

```cpp
public:
virtual int ExecuteNonQuery() override
```

Return Value

Number of rows affected

Implements

```csharp
IDbCommand::ExecuteNonQuery()
```
**Remarks**

You can use ExecuteNonQuery to perform any type of database operation, however any resultsets returned will not be available. Any output parameters used in calling a stored procedure will be populated with data and can be retrieved after execution is complete. For UPDATE, INSERT, and DELETE statements, the return value is the number of rows affected by the command. For all other types of statements, the return value is -1.
Examples

The following example creates a MySqlCommand and then executes it using ExecuteNonQuery. The example is passed a string that is a SQL statement (such as UPDATE, INSERT, or DELETE) and a string to use to connect to the data source.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand(myExecuteQuery As String, myConnection
Dim myCommand As New MySqlCommand(myExecuteQuery, myConnection)
myCommand.Connection.Open()
myCommand.ExecuteNonQuery()
myConnection.Close()
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand(string myExecuteQuery, MySqlConnection
{
MySqlCommand myCommand = new MySqlCommand(myExecuteQuery, myConnecti
myCommand.Connection.Open();
myCommand.ExecuteNonQuery();
myConnection.Close();
}
```
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecuteReader()()()</td>
<td>Sends the CommandText to the Connection and builds a MySqlDataReader.</td>
</tr>
<tr>
<td>ExecuteReader(CommandBehavior)</td>
<td>Sends the CommandText to the Connection, and builds a MySqlDataReader using one of the CommandBehavior values.</td>
</tr>
</tbody>
</table>
See Also

MySqlCommand Class
MySqlCommand Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Sends the CommandText to the Connection and builds a MySqlDataReader.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlDataReader ExecuteReader()
```

Visual Basic (Declaration)

```vbnet
Public Function ExecuteReader As MySqlDataReader
```

Visual C++

```cpp
public:
MySqlDataReader^ ExecuteReader()
```

Return Value

A `MySqlDataReader` object.
Remarks

When the `CommandType` property is set to `StoredProcedure`, the `CommandText` property should be set to the name of the stored procedure. The command executes this stored procedure when you call `ExecuteReader`.

While the `MySqlDataReader` is in use, the associated `MySqlConnection` is busy serving the MySqlDataReader. While in this state, no other operations can be performed on the MySqlConnection other than closing it. This is the case until the `Close()` method of the MySqlDataReader is called.
Examples

The following example creates a `MySqlCommand`, then executes it by passing a string that is a SQL SELECT statement, and a string to use to connect to the data source.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommandReader(mySelectQuery As String, myConnectionString)
Dim myCommand As New MySqlCommand(mySelectQuery, myConnectionString)
myConnection.Open()
Dim myReader As MySqlDataReader
myReader = myCommand.ExecuteReader()
Try
While myReader.Read()
Console.WriteLine(myReader.GetString(0))
End While
Finally
myReader.Close()
myConnection.Close()
End Try
End Sub
```

**C#**

```csharp
public void CreateMySqlCommandReader(string mySelectQuery, MySqlConnection myConnection)
{
MySqlCommand myCommand = new MySqlCommand(mySelectQuery, myConnectionString);
myConnection.Open();
MySqlDataReader myReader;
myReader = myCommand.ExecuteReader();
try
{
while (myReader.Read())
{
Console.WriteLine(myReader.GetString(0));
}
}
finally
{
myReader.Close();
myConnection.Close();
}
```
See Also

MySqlCommand Class
ExecuteReader Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlCommand..::..ExecuteReader Method (CommandBehavior)

MySqlCommand Class  See Also  Send Feedback

Sends the CommandText to the Connection, and builds a MySqlDataReader using one of the CommandBehavior values.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlDataReader ExecuteReader(
    CommandBehavior behavior
)

Visual Basic (Declaration)

Public Function ExecuteReader ( _
    behavior As CommandBehavior _
) As MySqlDataReader

Visual C++

public:
MySqlDataReader^ ExecuteReader(
    CommandBehavior behavior
)

Parameters

behavior

Type: System.Data:::CommandBehavior
One of the CommandBehavior values.

Return Value

A MySqlDataReader object.
Remarks

When the `CommandType` property is set to `StoredProcedure`, the `CommandText` property should be set to the name of the stored procedure. The command executes this stored procedure when you call `ExecuteReader`.

The `MySqlDataReader` supports a special mode that enables large binary values to be read efficiently. For more information, see the `SequentialAccess` setting for `CommandBehavior`.

While the `MySqlDataReader` is in use, the associated `MySqlConnection` is busy serving the `MySqlDataReader`. While in this state, no other operations can be performed on the `MySqlConnection` other than closing it. This is the case until the `Close()` method of the `MySqlDataReader` is called. If the `MySqlDataReader` is created with `CommandBehavior` set to `CloseConnection`, closing the `MySqlDataReader` closes the connection automatically.

❖ **Note:** When calling `ExecuteReader` with the `SingleRow` behavior, you should be aware that using a `limit` clause in your SQL will cause all rows (up to the limit given) to be retrieved by the client. The `Read()` method will still return false after the first row but pulling all rows of data into the client will have a performance impact. If the `limit` clause is not necessary, it should be avoided.
See Also

MySqlCommand Class
ExecuteReader Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes the query, and returns the first column of the first row in the result set returned by the query. Extra columns or rows are ignored.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override Object ExecuteScalar()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function ExecuteScalar As Object
```

**Visual C++**

```c++
public:
virtual Object^ ExecuteScalar() override
```

**Return Value**

The first column of the first row in the result set, or a null reference if the result set is empty

**Implements**

`IDbCommand::ExecuteScalar()`
Remarks

Use the ExecuteScalar method to retrieve a single value (for example, an aggregate value) from a database. This requires less code than using the `ExecuteReader()` method, and then performing the operations necessary to generate the single value using the data returned by a `MySqlDataReader`
Examples

The following example creates a `MySqlCommand` and then executes it using `ExecuteScalar`. The example is passed a string that is a SQL statement that returns an aggregate result, and a string to use to connect to the data source.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand(myScalarQuery As String, myConnection
Dim myCommand As New MySqlCommand(myScalarQuery, myConnection)
myCommand.Connection.Open()
myCommand.ExecuteScalar()
myConnection.Close()
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand(string myScalarQuery, MySqlConnection
{
MySqlCommand myCommand = new MySqlCommand(myScalarQuery, myConnectio
myCommand.Connection.Open();
myCommand.ExecuteScalar();
myConnection.Close();
}
```

**C++**

```c++
public:
void CreateMySqlCommand(String* myScalarQuery, MySqlConnection* myCo
{
MySqlCommand* myCommand = new MySqlCommand(myScalarQuery, myConnecti
myCommand->Connection->Open();
myCommand->ExecuteScalar();
myConnection->Close();
}
```
See Also

MySQLCommand Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlCommand..::.MemberwiseClone Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="MemberwiseClone()" /></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td><img src="" alt="MemberwiseClone(Boolean)" /></td>
<td>Creates a shallow copy of the current <code>MarshalByRefObject</code> object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlCommand Class
MySqlCommand Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Creates a prepared version of the command on an instance of MySQL Server.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySQL.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

public override void Prepare()

Visual Basic (Declaration)

Public Overrides Sub Prepare

Visual C++

public:
virtual void Prepare() override

Implements

IDbCommand::Prepare()
Remarks

Prepared statements are only supported on MySQL version 4.1 and higher. Calling prepare while connected to earlier versions of MySQL will succeed but will execute the statement in the same way as unprepared.
The following example demonstrates the use of the **Prepare** method.

**VB.NET**

```vbnet
public sub PrepareExample()
Dim cmd as New MySqlCommand("INSERT INTO mytable VALUES (@val)", myC
cmd.Parameters.Add( "@val", 10 )
   cmd.Prepare()
   cmd.ExecuteNonQuery()
   cmd.Parameters(0).Value = 20
   cmd.ExecuteNonQuery()
end sub
```

**C#**

```csharp
private void PrepareExample()
{
   MySqlCommand cmd = new MySqlCommand("INSERT INTO mytable VALUES (@val)
cmd.Parameters.Add( "@val", 10 );
   cmd.Prepare();
   cmd.ExecuteNonQuery();
   cmd.Parameters[0].Value = 20;
   cmd.ExecuteNonQuery();
}
```
See Also

MySQLCommand Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlCommand` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanRaiseEvents</td>
<td>Gets a value indicating whether the component can raise an event. (Inherited from Component.)</td>
</tr>
<tr>
<td>CommandText</td>
<td>Gets or sets the SQL statement to execute at the data source. (Overrides DbCommand..:::CommandText.)</td>
</tr>
<tr>
<td>CommandTimeout</td>
<td>Gets or sets the wait time before terminating the attempt to execute a command and generating an error. (Overrides DbCommand..:::CommandTimeout.)</td>
</tr>
<tr>
<td>CommandType</td>
<td>Gets or sets a value indicating how the CommandType property is to be interpreted. (Overrides DbCommand..:::CommandType.)</td>
</tr>
<tr>
<td>Connection</td>
<td>Gets or sets the MySqlConnection used by this instance of the MySqlCommand. Gets the IContainer that contains the IContainer. (Inherited from Component.)</td>
</tr>
<tr>
<td>Container</td>
<td>Gets or sets the DbConnection used by this DbCommand. (Inherited from DbCommand.)</td>
</tr>
<tr>
<td>DbConnection</td>
<td>Gets or sets the DbCommand. (Inherited from DbCommand.)</td>
</tr>
<tr>
<td>DbParameterCollection</td>
<td>Gets the collection of DbParameter objects. (Inherited from DbCommand.)</td>
</tr>
<tr>
<td>DbTransaction</td>
<td>Gets or sets the DbTransaction within which this DbCommand object executes. (Inherited from DbCommand.)</td>
</tr>
<tr>
<td>DesignMode</td>
<td>Gets a value that indicates whether the Component is currently in design mode. (Inherited from Component.)</td>
</tr>
</tbody>
</table>
| DbCommand                 | Gets or sets a value indicating whether the DbCommand...
**DesignTimeVisible**
command object should be visible in a Windows Form Designer control. (Overrides **DbCommand..::.DesignTimeVisible**.)

**Events**
Gets the list of event handlers that are attached to this **Component**. (Inherited from **Component**.)

**IsPrepared**

**LastInsertedId**

**Parameters**
Get the **MySqlParameterCollection**

**Site**
Gets or sets the **ISite** of the **Component**. (Inherited from **Component**.)

**Transaction**
Gets or sets the **MySqlTransaction** within which the **MySqlCommand** executes.

**UpdatedRowSource**
Gets or sets how command results are applied to the DataRow when used by the Update method of the DbDataAdapter. (Overrides **DbCommand..::.UpdatedRowSource**.)
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the SQL statement to execute at the data source.

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysql.com/)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override string CommandText { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Overrides Property CommandText As String
```

Visual C++

```cpp
public:
virtual property String^ CommandText {
    String^ get () override;
    void set (String^ value) override;
}
```

Field Value

The SQL statement or stored procedure to execute. The default is an empty string.

Implements

`IDbCommand...::CommandText`
Remarks

When the `CommandType` property is set to `StoredProcedure`, the `CommandText` property should be set to the name of the stored procedure. The user may be required to use escape character syntax if the stored procedure name contains any special characters. The command executes this stored procedure when you call one of the `Execute` methods. Starting with Connector/Net 5.0, having both a stored function and stored procedure with the same name in the same database is not supported. It is suggested that you provide unique names for your stored routines.
Examples

The following example creates a `MySqlCommand` and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand()
    Dim myCommand As New MySqlCommand()
    myCommand.CommandText = "SELECT * FROM Mytable ORDER BY id"
    myCommand.CommandType = CommandType.Text
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand()
{
    MySqlCommand myCommand = new MySqlCommand();
    myCommand.CommandText = "SELECT * FROM mytable ORDER BY id";
    myCommand.CommandType = CommandType.Text;
}
```
See Also

MySQLCommand Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the wait time before terminating the attempt to execute a command and generating an error.

**Namespace:** [MySQL.Data.MySqlClient](#)  
**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override int CommandTimeout { get; set; }

Visual Basic (Declaration)

Public Overrides Property CommandTimeout As Integer

Visual C++

public:
virtual property int CommandTimeout {
    int get () override;
    void set (int value) override;
}

Field Value

The time (in seconds) to wait for the command to execute. The default is 30 seconds.

Implements

IDbCommand...::CommandTimeout
Remarks

CommandTimeout is dependent on the ability of MySQL to cancel an executing query. Because of this, CommandTimeout is only supported when connected to MySQL version 5.0.0 or higher.
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a value indicating how the `CommandText` property is to be interpreted.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override CommandType CommandType { get; set; }

Visual Basic (Declaration)

Public Overrides Property CommandType As CommandType

Visual C++

public:
virtual property CommandType CommandType {
    CommandType get () override;
    void set (CommandType value) override;
}

Field Value

One of the CommandType values. The default is Text.

Implements

IDbCommand, ; ; ; CommandType
Remarks

When you set the CommandType property to StoredProcedure, you should set the CommandText property to the name of the stored procedure. The command executes this stored procedure when you call one of the Execute methods.
Examples

The following example creates a **MySqlCommand** and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand()
    Dim myCommand As New MySqlCommand()
    myCommand.CommandType = CommandType.Text
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand()
{
    MySqlCommand myCommand = new MySqlCommand();
    myCommand.CommandType = CommandType.Text;
}
```
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the `SqlConnection` used by this instance of the ` MySqlCommand`.

**Namespace:**  [MySQL.Data.MySqlClient](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlConnection Connection { get; set; }

Visual Basic (Declaration)

Public Property Connection As MySqlConnection

Visual C++

public:
property MySqlConnection^ Connection {
    MySqlConnection^ get ();
    void set (MySqlConnection^ value);
}

Field Value

The connection to a data source. The default value is a null reference (Nothing in Visual Basic).
Remarks

If you set Connection while a transaction is in progress and the Transaction property is not null, an InvalidOperationException is generated. If the Transaction property is not null and the transaction has already been committed or rolled back, Transaction is set to null.
Examples

The following example creates a `MySqlCommand` and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand()
Dim mySelectQuery As String = "SELECT * FROM mytable ORDER BY id"
Dim myConnectString As String = "Persist Security Info=False;database=test;server=myServer"
Dim myCommand As New MySqlCommand(mySelectQuery)
myCommand.Connection = New MySqlConnection(myConnectString)
myCommand.CommandType = CommandType.Text
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand()
{
    string mySelectQuery = "SELECT * FROM mytable ORDER BY id";
    string myConnectString = "Persist Security Info=False;database=test;"
    MySqlCommand myCommand = new MySqlCommand(mySelectQuery);
    myCommand.Connection = new MySqlConnection(myConnectString);
    myCommand.CommandType = CommandType.Text;
}
```
See Also

MySQLCommand Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a value indicating whether the command object should be visible in a Windows Form Designer control.

**Namespace:** [MySQL.Data.MySqlClient](https://github.com/mysql-connector-dotnet/mysql-client)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override bool DesignTimeVisible { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Property DesignTimeVisible As Boolean
```

### Visual C++

```cpp
public:
virtual property bool DesignTimeVisible {
    bool get () override;
    void set (bool value) override;
}
```
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
**MySQL Command..:::IsPrepared Property**

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySQL.Data](#) (in [MySQL.Data.dll](#))  Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public bool IsPrepared { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property IsPrepared As Boolean
```

**Visual C++**

```cpp
public:
property bool IsPrepared {
    bool get ();
}
```
See Also

MySQLCommand Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlCommand..:::LastInsertedId Property

**Namespace:**  [MySQL.Data.MySqlClient](https://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public long LastInsertedId { get; }

Visual Basic (Declaration)

Public ReadOnly Property LastInsertedId As Long

Visual C++

public:
property long long LastInsertedId {
    long long get ();
}
See Also

MySQLCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlCommand..:::Parameters Property

Get the MySqlParameterCollection

**Namespace:**  MySql.Data.MySqlClient

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public MySqlParameterCollection Parameters { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Parameters As MySqlParameterCollection
```

**Visual C++**

```cpp
public:
property MySqlParameterCollection^ Parameters {
    MySqlParameterCollection^ get ();
}
```

**Field Value**

The parameters of the SQL statement or stored procedure. The default is an empty collection.
Remarks

Connector/Net does not support unnamed parameters. Every parameter added to the collection must have an associated name.
 Examples

The following example creates a `MySqlCommand` and displays its parameters. To accomplish this, the method is passed a `MySqlConnection`, a query string that is a SQL SELECT statement, and an array of `MySqlParameter` objects.

**VB.NET**

```vbnet
Public Sub CreateMySqlCommand(myConnection As MySqlConnection, _
mySelectQuery As String, myParamArray() As SqlParameter)
Dim myCommand As New MySqlCommand(mySelectQuery, myConnection)
myCommand.CommandText = "SELECT id, name FROM mytable WHERE age=@age"
myCommand.UpdatedRowSource = UpdateRowSource.Both
myCommand.Parameters.Add(myParamArray)
Dim j As Integer
For j = 0 To myCommand.Parameters.Count - 1
myCommand.Parameters.Add(myParamArray(j))
Next j
Dim myMessage As String = ""
Dim i As Integer
For i = 0 To myCommand.Parameters.Count - 1
myMessage += myCommand.Parameters(i).ToString() & ControlChars.Cr
Next i
Console.WriteLine(myMessage)
End Sub
```

**C#**

```csharp
public void CreateMySqlCommand(MySqlConnection myConnection, string mySelectQuery, SqlParameter[] myParamArray)
{
MySqlCommand myCommand = new MySqlCommand(mySelectQuery, myConnection);
myCommand.CommandText = "SELECT id, name FROM mytable WHERE age=@age"
myCommand.UpdatedRowSource = UpdateRowSource.Both;
for (int j=0; j<myParamArray.Length; j++)
{
myCommand.Parameters.Add(myParamArray[j]);
}
string myMessage = ""
for (int i = 0; i < myCommand.Parameters.Count; i++)
{
myMessage += myCommand.Parameters[i].ToString() + "\n";
}
MessageBox.Show(myMessage);
```
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the `MySqlTransaction` within which the ` MySqlCommand` executes.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public MySqlTransaction Transaction { get; set; }

Visual Basic (Declaration)

Public Property Transaction As MySqlTransaction

Visual C++

public:
property MySqlTransaction^ Transaction {
    MySqlTransaction^ get ();
    void set (MySqlTransaction^ value);
}

Field Value

The MySqlTransaction. The default value is a null reference (Nothing in Visual Basic).
Remarks

You cannot set the Transaction property if it is already set to a specific value, and the command is in the process of executing. If you set the transaction property to a `MySqlTransaction` object that is not connected to the same `MySqlConnection` as the `MySqlCommand` object, an exception will be thrown the next time you attempt to execute a statement.
See Also

MySQLCommand Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets how command results are applied to the DataRow when used by the Update method of the DbDataAdapter.

**Namespace:** [MySQL.Data.MySqlClient](#)  
**Assembly:** [MySQL.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

```csharp
public override UpdateRowSource UpdatedRowSource { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Overrides Property UpdatedRowSource As UpdateRowSource
```

Visual C++

```csharp
public:
virtual property UpdateRowSource UpdatedRowSource {
    UpdateRowSource get () override;
    void set (UpdateRowSource value) override;
}
```

Implements

```csharp
IDbCommand:::UpdatedRowSource
```
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlCommand` type exposes the following members.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposed</td>
<td>Occurs when the component is disposed by a call to the <strong>Dispose</strong> method.</td>
</tr>
</tbody>
</table>

(Inherited from **Component**.)
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLCommandBuilder Class

Automatically generates single-table commands used to reconcile changes made to a DataSet with the associated MySQL database. This class cannot be inherited.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public sealed class MySqlCommandBuilder : DbCommandBuilder

Visual Basic (Declaration)

Public NotInheritable Class MySqlCommandBuilder _
   Inherits DbCommandBuilder

Visual C++

public ref class MySqlCommandBuilder sealed : public DbCommandBuilder
Remarks

The **MySqlDataAdapter** does not automatically generate the SQL statements required to reconcile changes made to a **DataSet** with the associated instance of MySQL. However, you can create a MySqlCommandBuilder object to automatically generate SQL statements for single-table updates if you set the **SelectCommand** property of the MySqlDataAdapter. Then, any additional SQL statements that you do not set are generated by the MySqlCommandBuilder.

The MySqlCommandBuilder registers itself as a listener for RowUpdating events whenever you set the **DataAdapter** property. You can only associate one MySqlDataAdapter or MySqlCommandBuilder object with each other at one time.

To generate INSERT, UPDATE, or DELETE statements, the MySqlCommandBuilder uses the SelectCommand property to retrieve a required set of metadata automatically. If you change the SelectCommand after the metadata has is retrieved (for example, after the first update), you should call the [!:RefreshSchema] method to update the metadata.

The SelectCommand must also return at least one primary key or unique column. If none are present, an InvalidOperation exception is generated, and the commands are not generated.

The MySqlCommandBuilder also uses the **Connection**, **CommandTimeout**, and **Transaction** properties referenced by the SelectCommand. The user should call RefreshSchema if any of these properties are modified, or if the SelectCommand itself is replaced. Otherwise the **InsertCommand**, **UpdateCommand**, and **DeleteCommand** properties retain their previous values.

If you call **Dispose**, the MySqlCommandBuilder is disassociated from the MySqlDataAdapter, and the generated commands are no longer used.

**Note:** Caution must be used when using MySqlCommandBuilder on MySql 4.0 systems. With MySql 4.0, database/schema information is not provided to the connector for a query. This means that a query that pulls columns from two identically named tables in two or more different databases will not cause an
exception to be thrown but will not work correctly. Even more dangerous is the situation where your select statement references database X but is executed in database Y and both databases have tables with similar layouts. This situation can cause unwanted changes or deletes. This note does not apply to MySQL versions 4.1 and later.
Examples

The following example uses the **MySqlCommand**, along **MySqlDataAdapter** and **MySqlConnection**, to select rows from a data source. The example is passed an initialized **DataSet**, a connection string, a query string that is a SQL SELECT statement, and a string that is the name of the database table. The example then creates a MySqlCommandBuilder.

**VB.NET**

```vbnet
Public Shared Function SelectRows(myConnection As String, mySelectQuery As String) As DataSet
Dim myConn As New MySqlConnection(myConnection)
Dim myDataAdapter As New MySqlDataAdapter()
myDataAdapter.SelectCommand = New MySqlCommand(mySelectQuery, myConn)
Dim cb As SqlCommandBuilder = New MySqlCommandBuilder(myDataAdapter)
myConn.Open()

Dim ds As DataSet = New DataSet
myDataAdapter.Fill(ds, myTableName)

' Code to modify data in DataSet here

' Without the MySqlCommandBuilder this line would fail.
myDataAdapter.Update(ds, myTableName)

myConn.Close()
End Function 'SelectRows
```

**C#**

```csharp
public static DataSet SelectRows(string myConnection, string mySelectQuery)
{
    MySqlConnection myConn = new MySqlConnection(myConnection);
    MySqlDataAdapter myDataAdapter = new MySqlDataAdapter();
    myDataAdapter.SelectCommand = new MySqlCommand(mySelectQuery, myConn);
    MySqlCommandBuilder cb = new MySqlCommandBuilder(myDataAdapter);

    myConn.Open();

    DataSet ds = new DataSet();
    myDataAdapter.Fill(ds, myTableName);
```
//code to modify data in DataSet here

//Without the MySqlCommandBuilder this line would fail
myDataAdapter.Update(ds, myTableName);

myConn.Close();

return ds;
}
Inheritance Hierarchy

System..:::Object
 System..:::MarshalByRefObject
 System.ComponentModel..:::Component
 System.Data.Common..:::DbCommandBuilder
 MySql.Data.MySqlClient..:::MySqlCommandBuilder
See Also

MySqlCommandBuilder Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySqlCommandBuilder** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlCommandBuilder</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ApplyParameterInfo</strong></td>
<td>Allows the provider implementation of the <a href="#">DbCommandBuilder</a> class to handle additional parameter properties. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <a href="#">MarshalByRefObject</a>.)</td>
</tr>
<tr>
<td><strong>DeriveParameters</strong></td>
<td>Retrieves parameter information from the stored procedure specified in the MySqlCommand and populates the Parameters collection of the specified MySqlCommand object. This method is not currently supported since stored procedures are not available in MySql.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Releases unmanaged resources and performs other cleanup operations before the <a href="#">Component</a> is reclaimed by garbage collection. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td><strong>GetDeleteCommand</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetInsertCommand</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetLifetimeService</strong></td>
<td>Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a>.)</td>
</tr>
<tr>
<td><strong>GetParameterName</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Returns the placeholder for the parameter in the</td>
</tr>
</tbody>
</table>
GetParameterPlaceholder associated SQL statement.
(Inherited from DbCommandBuilder.)
Returns the schema table for the
DbCommandBuilder.
(Inherited from DbCommandBuilder.)
Returns an object that represents a service
provided by the Component or by its Container.
(Inherited from Component.)
GetService
Gets the Type of the current instance.
(Inherited from Object.)
GetType
GetUpdateCommand Overloaded.
Resets the CommandTimeout, Transaction,
CommandType, and UpdateRowSource properties
on the DbCommand.
(Inherited from DbCommandBuilder.)
Obtains a lifetime service object to control the
lifetime policy for this instance.
(Inherited from MarshalByRefObject.)
InitializeLifetimeService
MemberwiseClone
Overloaded.
(Overrides
DbCommandBuilder..::.QuoteIdentifier(String).
QuoteIdentifier
Clears the commands associated with this
DbCommandBuilder.
(Inherited from DbCommandBuilder.)
RowUpdatingHandler Adds an event handler for the RowUpdating event.
(Inherited from DbCommandBuilder.)
Registers the DbCommandBuilder to handle the
DbCommandBuilder RowUpdating event for a DbDataAdapter.
(Inherited from DbCommandBuilder.)
Returns a String containing the name of the
Component, if any. This method should not be
overridden.
(Overrides
UnquoteIdentifier
DbCommandBuilder..::.UnquoteIdentifier(String).
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanRaiseEvents</td>
<td>Gets a value indicating whether the component can raise an event.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Component.)</td>
</tr>
<tr>
<td>CatalogLocation</td>
<td>Sets or gets the CatalogLocation for an instance of the DbCommandBuilder.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DbCommandBuilder.)</td>
</tr>
<tr>
<td>CatalogSeparator</td>
<td>Sets or gets a string used as the catalog separator for an instance</td>
</tr>
<tr>
<td></td>
<td>of the DbCommandBuilder class.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DbCommandBuilder.)</td>
</tr>
<tr>
<td>ConflictOption</td>
<td>Specifies which ConflictOption is to be used by the DbCommandBuilder.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DbCommandBuilder.)</td>
</tr>
<tr>
<td>Container</td>
<td>Gets the IContainer that contains the Component.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Component.)</td>
</tr>
<tr>
<td>DataAdapter</td>
<td>Gets or sets a MySqlDataAdapter object for which SQL statements are</td>
</tr>
<tr>
<td></td>
<td>automatically generated.</td>
</tr>
<tr>
<td>DesignMode</td>
<td>Gets a value that indicates whether the Component is currently in design</td>
</tr>
<tr>
<td></td>
<td>mode.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Component.)</td>
</tr>
<tr>
<td>Events</td>
<td>Gets the list of event handlers that are attached to this Component.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Component.)</td>
</tr>
<tr>
<td>QuotePrefix</td>
<td>Gets or sets the beginning character or characters to use when specifying</td>
</tr>
<tr>
<td></td>
<td>database objects (for example, tables or columns) whose names contain</td>
</tr>
<tr>
<td></td>
<td>characters such as spaces or reserved tokens.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DbCommandBuilder.)</td>
</tr>
<tr>
<td>QuoteSuffix</td>
<td>Gets or sets the beginning character or characters to use when specifying</td>
</tr>
<tr>
<td></td>
<td>database objects (for example, tables or columns) whose names contain</td>
</tr>
<tr>
<td></td>
<td>characters such as spaces or reserved tokens.</td>
</tr>
</tbody>
</table>
### SchemaSeparator

Gets or sets the character to be used for the separator between the schema identifier and any other identifiers.

(Inherited from DbCommandBuilder.)

### SetAllValues

Specifies whether all column values in an update statement are included or only changed ones.

(Inherited from DbCommandBuilder.)

### Site

Gets or sets the ISite of the Component.

(Inherited from Component.)
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposed</td>
<td>Occurs when the component is disposed by a call to the <code>Dispose()</code> method. (Inherited from <code>Component</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlCommandBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL CommandBuilder Constructor

MySQLCommandBuilder Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlCommandBuilder()()()</td>
<td>Initializes a new instance of the MySqlCommandBuilder class.</td>
</tr>
<tr>
<td>MySqlCommandBuilder(MySqlDataAdapter)</td>
<td>Initializes a new instance of the MySqlCommandBuilder class with the associated MySqlDataAdapter object.</td>
</tr>
</tbody>
</table>
See Also

MySqlCommandBuilder Class
MySqlCommandBuilder Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlCommandBuilder` class.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/products/connector-net/

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCommandBuilder()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySqlCommandBuilder()
See Also

MySqlCommandBuilder Class
MySqlCommandBuilder Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL CommandBuilder Constructor (MySqlDataAdapter)

MySqlCommandBuilder Class  See Also  Send Feedback

Initializes a new instance of the MySqlCommandBuilder class with the associated MySqlDataAdapter object.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCommandBuilder(
    MySqlDataAdapter adapter
)

Visual Basic (Declaration)

Public Sub New ( _
    adapter As MySqlDataAdapter _
)

Visual C++

public:
MySqlCommandBuilder(
    MySqlDataAdapter^ adapter
)

Parameters

adapter
  Type: MySql.Data.MySqlClient::MySqlDataAdapter
  The MySqlDataAdapter to use.
Remarks

The `MySqlCommandBuilder` registers itself as a listener for `RowUpdating` events that are generated by the `MySqlDataAdapter` specified in this property.

When you create a new instance `MySqlCommandBuilder`, any existing `MySqlCommandBuilder` associated with this `MySqlDataAdapter` is released.
See Also

MySqlCommandBuilder Class
MySqlCommandBuilder Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlCommandBuilder` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplyParameterInfo</td>
<td>Allows the provider implementation of the DbCommandBuilder class to handle additional parameter properties. (Inherited from DbCommandBuilder.)</td>
</tr>
<tr>
<td>CreateObjRef</td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>DeriveParameters</td>
<td>Retrieves parameter information from the stored procedure specified in the MySqlCommand and populates the Parameters collection of the specified MySqlCommand object. This method is not currently supported since stored procedures are not available in MySql.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Releases unmanaged resources and performs other cleanup operations before the Component is reclaimed by garbage collection. (Inherited from Component.)</td>
</tr>
<tr>
<td>GetDeleteCommand</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetInsertCommand</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetLifetimeService</td>
<td>Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>GetParameterName</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetParameterName</td>
<td>Returns the placeholder for the parameter in the</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>GetParameterPlaceholder</td>
<td>Returns the associated SQL statement. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>GetSchemaTable</td>
<td>Returns the schema table for the <a href="#">DbCommandBuilder</a>. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>GetService</td>
<td>Returns an object that represents a service provided by the <a href="#">Component</a> or by its <a href="#">Container</a>. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetUpdateCommand</td>
<td>Overloaded. Resets the CommandTimeout, Transaction, CommandType, and UpdateRowSource properties on the <a href="#">DbCommand</a>. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>InitializeCommand</td>
<td>Resets the CommandTimeout, Transaction, CommandType, and UpdateRowSource properties on the <a href="#">DbCommand</a>. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>InitializeLifetimeService</td>
<td>Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Overloaded. (overrides <a href="#">DbCommandBuilder</a>:::MemberwiseClone.)</td>
</tr>
<tr>
<td>QuoteIdentifier</td>
<td>(Overrides <a href="#">DbCommandBuilder</a>:::QuoteIdentifier(String).)</td>
</tr>
<tr>
<td>RefreshSchema</td>
<td>Clears the commands associated with this <a href="#">DbCommandBuilder</a>. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>RowUpdatingHandler</td>
<td>Adds an event handler for the RowUpdating event. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>SetRowUpdatingHandler</td>
<td>Registers the <a href="#">DbCommandBuilder</a> to handle the RowUpdating event for a <a href="#">DbDataAdapter</a>. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String containing the name of the <a href="#">Component</a>, if any. This method should not be overridden. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>UnquoteIdentifier</td>
<td>(Overrides <a href="#">DbCommandBuilder</a>:::UnquoteIdentifier(String).)</td>
</tr>
</tbody>
</table>
See Also

**MySqlCommandBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL CommandBuilder...:::DeriveParameters Method

MySQLCommandBuilder Class  See Also  Send Feedback

Retrieves parameter information from the stored procedure specified in the MySqlCommand and populates the Parameters collection of the specified MySqlCommand object. This method is not currently supported since stored procedures are not available in MySql.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static void DeriveParameters(
    MySqlCommand command
)

Visual Basic (Declaration)

Public Shared Sub DeriveParameters ( _
    command As MySqlCommand _
)

Visual C++

public:
static void DeriveParameters(
    MySqlCommand^ command
)

Parameters

command

Type: MySql.Data.MySqlClient::MySqlCommand
The MySqlCommand referencing the stored procedure from which the parameter information is to be derived. The derived parameters are added to the Parameters collection of the MySqlCommand.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System:::InvalidOperationException</code></td>
<td>The command text is not a valid stored procedure name.</td>
</tr>
</tbody>
</table>
See Also

MySQLCommandBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL CommandBuilder...:::Dispose Method

MySQLCommandBuilder Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Disposing](Disposing) | **Dispose()**
|  | Releases all resources used by the [Component](https://example.com). (Inherited from [Component](https://example.com).) |
| ![Disposing](Disposing) | **Dispose(Boolean)**
|  | Releases the unmanaged resources used by the [DbCommandBuilder](https://example.com) and optionally releases the managed resources. (Inherited from [DbCommandBuilder](https://example.com).) |
See Also

**MySqlCommandBuilder Class**
**MySqlCommandBuilder Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net
MySqlCommandBuilder...:.GetDeleteCommand Method

MySQLCommandBuilder Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDeleteCommand()</td>
<td>Gets the delete command. Gets the automatically generated <code>DbCommand</code> object required to perform deletions at the data source, optionally using columns for parameter names. (Inherited from <code>DbCommandBuilder</code>.)</td>
</tr>
<tr>
<td>GetDeleteCommand(Boolean)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlCommandBuilder Class
MySqlCommandBuilder Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the delete command.

**Namespace:** `MySql.Data.MySqlClient`  
**Assembly:** `MySql.Data (in MySql.Data.dll) Version: 6.2.2.0`
Syntax

C#

```csharp
public MySqlCommand GetDeleteCommand()
```

Visual Basic (Declaration)

```vbnet
Public Function GetDeleteCommand As MySqlCommand
```

Visual C++

```cpp
public:
MySqlCommand^ GetDeleteCommand()
```

Return Value
See Also

SqlConnectionBuilder Class
GetDeleteCommand Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL CommandBuilder...:::GetInsertCommand Method

See Also
Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="GetInsertCommand()()" /></td>
<td>Gets the insert command. Gets the automatically generated <a href="#">DbCommand</a> object required to perform insertions at the data source, optionally using columns for parameter names. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td><img src="#" alt="GetInsertCommand(Boolean)()" /></td>
<td></td>
</tr>
</tbody>
</table>


See Also

**MySqlCommandBuilder Class**
**MySqlCommandBuilder Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlCommandBuilder...:.GetInsertCommand Method

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0

Gets the insert command.
Syntax

C#

public MySqlCommand GetInsertCommand()

Visual Basic (Declaration)

Public Function GetInsertCommand As MySqlCommand

Visual C++

public:
MySqlCommand^ GetInsertCommand()

Return Value
See Also

MySqlCommandBuilder Class
GetInsertCommand Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net
MySQLCommandBuilder...::.GetParameterName Method
MySQLCommandBuilder Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetParameterName(Int32)</code></td>
<td>Returns the name of the specified parameter in the format of @p#. Use when building a custom command builder. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td><code>GetParameterName(String)</code></td>
<td>Returns the full parameter name, given the partial parameter name. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlCommandBuilder Class
MySqlCommandBuilder Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL CommandBuilder...:::.GetUpdateCommand Method

MySQLCommandBuilder Class   See Also   Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetUpdateCommand()</code></td>
<td>Gets the update command. Gets the automatically generated <code>DbCommand</code> object required to perform updates at the data source, optionally using columns for parameter names. (Inherited from <code>DbCommandBuilder</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlCommandBuilder Class
MySqlCommandBuilder Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlCommandBuilder...:.GetUpdateCommand Method

**MySqlCommandBuilder Class**  See Also  Send Feedback

Gets the update command.

**Namespace:**  *MySQL.Data.MySqlClient*

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCommand GetUpdateCommand()

Visual Basic (Declaration)

Public Function GetUpdateCommand As MySqlCommand

Visual C++

public:
MySqlCommand^ GetUpdateCommand()

Return Value
See Also

MySqlCommandBuilder Class
GetUpdateCommand Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL CommandBuilder...:::MemberwiseClone Method

MySQLCommandBuilder Class  See Also  Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MemberwiseClone()</code>(void)</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone(Boolean)</code></td>
<td>Creates a shallow copy of the current <code>MarshalByRefObject</code> object. (Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlCommandBuilder Class
MySqlCommandBuilder Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL CommandBuilder::QuoteIdentifier Method

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override string QuoteIdentifier(
    string unquotedIdentifier
)

Visual Basic (Declaration)

Public Overrides Function QuoteIdentifier ( _
    unquotedIdentifier As String _
) As String

Visual C++

public:
virtual String^ QuoteIdentifier(
    String^ unquotedIdentifier
) override

Parameters

unquotedIdentifier
    Type: System::::String
See Also

MySqlCommandBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlCommandBuilder...:::UnquoteIdentifier Method

MySQL Command Builder Class  See Also  Send Feedback

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override string UnquoteIdentifier(
    string quotedIdentifier
)

Visual Basic (Declaration)

Public Overrides Function UnquoteIdentifier ( _
    quotedIdentifier As String _
) As String

Visual C++

public:
    virtual String^ UnquoteIdentifier(
        String^ quotedIdentifier
    ) override

Parameters

quotedIdentifier
    Type: System::String
See Also

MySqlCommandBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlCommandBuilder` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanRaiseEvents</td>
<td>Gets a value indicating whether the component can raise an event. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>CatalogLocation</td>
<td>Sets or gets the <a href="#">CatalogLocation</a> for an instance of the <a href="#">DbCommandBuilder</a> class. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>CatalogSeparator</td>
<td>Sets or gets a string used as the catalog separator for an instance of the <a href="#">DbCommandBuilder</a> class. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>ConflictOption</td>
<td>Specifies which <a href="#">ConflictOption</a> is to be used by the <a href="#">DbCommandBuilder</a>. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>Container</td>
<td>Gets the <a href="#">IContainer</a> that contains the <a href="#">Component</a>. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>DataAdapter</td>
<td>Gets or sets a <a href="#">MySqlDataAdapter</a> object for which SQL statements are automatically generated.</td>
</tr>
<tr>
<td>DesignMode</td>
<td>Gets a value that indicates whether the <a href="#">Component</a> is currently in design mode. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>Events</td>
<td>Gets the list of event handlers that are attached to this <a href="#">Component</a>. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>QuotePrefix</td>
<td>Gets or sets the beginning character or characters to use when specifying database objects (for example, tables or columns) whose names contain characters such as spaces or reserved tokens. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
<tr>
<td>QuoteSuffix</td>
<td>Gets or sets the beginning character or characters to use when specifying database objects (for example, tables or columns) whose names contain characters such as spaces or reserved tokens. (Inherited from <a href="#">DbCommandBuilder</a>.)</td>
</tr>
</tbody>
</table>
(Inherited from `DbCommandBuilder`.)

**SchemaSeparator**

Gets or sets the character to be used for the separator between the schema identifier and any other identifiers.

(Inherited from `DbCommandBuilder`.)

**SetAllValues**

Specifies whether all column values in an update statement are included or only changed ones.

(Inherited from `DbCommandBuilder`.)

**Site**

Gets or sets the `ISite` of the `Component`.

(Inherited from `Component`.)
See Also

MySQLCommandBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlCommandBuilder...:::DataAdapter Property

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/productsconnector-net/)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0

Gets or sets a [MySqlDataAdapter](https://www.mysql.com/productsconnector-net/) object for which SQL statements are automatically generated.
Syntax

C#

public MySqlDataAdapter DataAdapter { get; set; }

Visual Basic (Declaration)

Public Property DataAdapter As MySqlDataAdapter

Visual C++

public:
property MySqlDataAdapter^ DataAdapter {
    MySqlDataAdapter^ get ();
    void set (MySqlDataAdapter^ value);
}

Field Value

A MySqlDataAdapter object.
Remarks

The `MySqlCommandBuilder` registers itself as a listener for `RowUpdating` events that are generated by the `MySqlDataAdapter` specified in this property.

When you create a new instance `MySqlCommandBuilder`, any existing `MySqlCommandBuilder` associated with this `MySqlDataAdapter` is released.
See Also

**MySqlCommandBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlCommandBuilder` type exposes the following members.
### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposed</td>
<td>Occurs when the component is disposed by a call to the Dispose() method.</td>
</tr>
</tbody>
</table>

(Inherited from `Component`.)
See Also

**MySqlCommandBuilder Class**

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents an open connection to a MySQL Server database. This class cannot be inherited.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

**C#**

```
public sealed class MySqlConnection : DbConnection, ICloneable
```

**Visual Basic (Declaration)**

```
Public NotInheritable Class MySqlConnection _
    Inherits DbConnection _
    Implements ICloneable
```

**Visual C++**

```
public ref class MySqlConnection sealed : public DbConnection, ICloneable
```
Remarks

A **MySqlConnection** object represents a session to a MySQL Server data source. When you create an instance of MySqlConnection, all properties are set to their initial values. For a list of these values, see the MySqlConnection constructor.

If the MySqlConnection goes out of scope, it is not closed. Therefore, you must explicitly close the connection by calling `Close()` or `Dispose(Boolean)`.
Examples
The following example creates a MySqlCommand and a MySqlConnection. The
MySqlConnection is opened and set as the Connection for the MySqlCommand.
The example then calls ExecuteNonQuery()()(), and closes the connection. To
accomplish this, the ExecuteNonQuery is passed a connection string and a query
string that is a SQL INSERT statement.
VB.NET

Copy

<c>

Public Sub InsertRow(myConnectionString As String)
' If the connection string is null, use a default.
If myConnectionString = "" Then
myConnectionString = "Database=Test;Data Source=localhost;User Id=use
End If
Dim myConnection As New MySqlConnection(myConnectionString)
Dim myInsertQuery As String = "INSERT INTO Orders (id, customerId, am
Dim myCommand As New MySqlCommand(myInsertQuery)
myCommand.Connection = myConnection
myConnection.Open()
myCommand.ExecuteNonQuery()
myCommand.Connection.Close()
End Sub
</c>

C#

Copy

<c>

public void InsertRow(string myConnectionString)
{
// If the connection string is null, use a default.
if(myConnectionString == "")
{
myConnectionString = "Database=Test;Data Source=localhost;User Id=use
}
MySqlConnection myConnection = new MySqlConnection(myConnectionString
string myInsertQuery = "INSERT INTO Orders (id, customerId, amount) V
MySqlCommand myCommand = new MySqlCommand(myInsertQuery);
myCommand.Connection = myConnection;
myConnection.Open();
myCommand.ExecuteNonQuery();
myCommand.Connection.Close();


}
</c>
Inheritance Hierarchy

System..:::Object
System..:::MarshalByRefObject
System.ComponentModel..:::Component
System.Data.Common..:::DbConnection
MySql.Data.MySqlClient..:::MySqlConnection
See Also

 MySqlConnection Members
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySqlConnection** type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td></td>
</tr>
<tr>
<td>MySqlConnection</td>
<td>Initializes a new instance of the MySqlConnection class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginDbTransaction</strong></td>
<td>Starts a database transaction. (Inherited from <a href="#">DbConnection</a>.)</td>
</tr>
<tr>
<td><strong>BeginTransaction</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>CancelQuery</strong></td>
<td>Changes the current database for an open MySqlConnection.</td>
</tr>
<tr>
<td></td>
<td>(Overloaded <a href="#">DbConnection::ChangeDatabase(String).</a>.)</td>
</tr>
<tr>
<td><strong>ChangeDatabase</strong></td>
<td>Changes the current database for an open MySqlConnection.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <a href="#">DbConnection::ChangeDatabase(String).</a>.)</td>
</tr>
<tr>
<td><strong>ClearAllPools</strong></td>
<td>Clears all connection pools.</td>
</tr>
<tr>
<td><strong>ClearPool</strong></td>
<td>Empties the connection pool associated with the specified connection.</td>
</tr>
<tr>
<td><strong>Clone</strong></td>
<td>Creates a new MySqlConnection object with the exact same ConnectionString value</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Closes the connection to the database. This is the preferred method of closing any open connection. (Overrides <a href="#">DbConnection::Close()</a>.)</td>
</tr>
<tr>
<td><strong>CreateCommand</strong></td>
<td>Creates and returns a <a href="#">MySqlCommand</a> object associated with the <a href="#">MySqlConnection</a>.</td>
</tr>
<tr>
<td><strong>CreateDbCommand</strong></td>
<td>Creates and returns a <a href="#">DbCommand</a> object associated with the current connection. (Inherited from <a href="#">DbConnection</a>.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <a href="#">MarshalByRefObject</a>.)</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>EnlistTransaction</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Releases unmanaged resources and performs other cleanup operations before the Component is reclaimed by garbage collection.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td><strong>GetLifetimeService</strong></td>
<td>Retrieves the current lifetime service object that controls the lifetime policy for this instance.</td>
</tr>
<tr>
<td><strong>GetSchema</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetService</strong></td>
<td>Returns an object that represents a service provided by the Component or by its Container.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td><strong>InitializeLifetimeService</strong></td>
<td>Obtains a lifetime service object to control the lifetime policy for this instance.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>OnStateChange</strong></td>
<td>Raises the StateChange event.</td>
</tr>
<tr>
<td><strong>Open</strong></td>
<td>Opens a database connection with the property settings specified by the ConnectionString.</td>
</tr>
<tr>
<td><strong>Ping</strong></td>
<td>Ping</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String containing the name of the Component, if any. This method should not be overridden.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanRaiseEvents</td>
<td>Gets a value indicating whether the component can raise an event. (Inherited from <a href="#">Component</a>)</td>
</tr>
<tr>
<td>ConnectionString</td>
<td>Gets or sets the string used to connect to a MySQL Server database. (Overrides <a href="#">DbConnection..::ConnectionString</a>)</td>
</tr>
<tr>
<td>ConnectionTimeout</td>
<td>Gets the time to wait while trying to establish a connection before terminating the attempt and generating an error. (Overrides <a href="#">DbConnection..::ConnectionTimeout</a>)</td>
</tr>
<tr>
<td>Container</td>
<td>Gets the <a href="#">IContainer</a> that contains the <a href="#">Component</a>. (Inherited from <a href="#">Component</a>)</td>
</tr>
<tr>
<td>Database</td>
<td>Gets the name of the current database or the database to be used after a connection is opened. (Overrides <a href="#">DbConnection..::Database</a>)</td>
</tr>
<tr>
<td>DataSource</td>
<td>Gets the name of the MySQL server to which to connect. (Overrides <a href="#">DbConnection..::DataSource</a>)</td>
</tr>
<tr>
<td>DbProviderFactory</td>
<td>Gets the <a href="#">DbProviderFactory</a> for this <a href="#">DbConnection</a>. (Inherited from <a href="#">DbConnection</a>)</td>
</tr>
<tr>
<td>DesignMode</td>
<td>Gets a value that indicates whether the <a href="#">Component</a> is currently in design mode. (Inherited from <a href="#">Component</a>)</td>
</tr>
<tr>
<td>Events</td>
<td>Gets the list of event handlers that are attached to this <a href="#">Component</a>. (Inherited from <a href="#">Component</a>)</td>
</tr>
<tr>
<td>ServerThread</td>
<td>Returns the id of the server thread this connection is executing on</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>Gets a string containing the version of the MySQL server to which the client is connected.</td>
</tr>
</tbody>
</table>
Site

Gets or sets the ISite of the Component.
(Inherited from Component.)

State

Gets the current state of the connection.
(Overrides DbConnection...::State.)

UseCompression

Indicates if this connection should use compression when communicating with the server.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Disposed</strong></td>
<td>Occurs when the component is disposed by a call to the <code>Dispose()</code> method.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Component</code>.)</td>
</tr>
<tr>
<td><strong>InfoMessage</strong></td>
<td>Occurs when MySQL returns warnings as a result of executing a command or query.</td>
</tr>
<tr>
<td><strong>StateChange</strong></td>
<td>Occurs when the state of the event changes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DbConnection</code>.)</td>
</tr>
</tbody>
</table>
See Also

SqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `SqlConnection` class.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlConnection()()</td>
<td>Initializes a new instance of the MySqlConnection class.</td>
</tr>
<tr>
<td>MySqlConnection(String)</td>
<td>Initializes a new instance of the MySqlConnection class when given a string containing the connection string.</td>
</tr>
</tbody>
</table>
See Also

**SqlConnection Class**
**SqlConnection Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlConnection` class.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public MySqlConnection()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySqlConnection()
Remarks

When a new instance of `MySqlConnection` is created, the read/write properties are set to the following initial values unless they are specifically set using their associated keywords in the `ConnectionString` property.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConnectionString</code></td>
<td>empty string (&quot;&quot;')</td>
</tr>
<tr>
<td><code>ConnectionTimeout</code></td>
<td>15</td>
</tr>
<tr>
<td><code>Database</code></td>
<td>empty string (&quot;&quot;')</td>
</tr>
<tr>
<td><code>DataSource</code></td>
<td>empty string (&quot;&quot;')</td>
</tr>
<tr>
<td><code>ServerVersion</code></td>
<td>empty string (&quot;&quot;')</td>
</tr>
</tbody>
</table>

You can change the value for these properties only by using the `ConnectionString` property.
Examples
See Also

MySqlConnection Class
MySqlConnection Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlConnection Constructor (String)

Initializes a new instance of the `MySqlConnection` class when given a string containing the connection string.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlConnection(
    string connectionString
)
```

Visual Basic (Declaration)

```vbnet
Public Sub New ( _
    connectionString As String _
)
```

Visual C++

```cpp
public:
MySqlConnection(
    String^ connectionString
)
```

Parameters

connectionString

Type: System::String

The connection properties used to open the MySQL database.
Remarks

When a new instance of `MySqlConnection` is created, the read/write properties are set to the following initial values unless they are specifically set using their associated keywords in the `ConnectionString` property.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConnectionString</code></td>
<td>empty string (&quot;&quot;)</td>
</tr>
<tr>
<td><code>ConnectionTimeout</code></td>
<td>15</td>
</tr>
<tr>
<td><code>Database</code></td>
<td>empty string (&quot;&quot;)</td>
</tr>
<tr>
<td><code>DataSource</code></td>
<td>empty string (&quot;&quot;)</td>
</tr>
<tr>
<td><code>ServerVersion</code></td>
<td>empty string (&quot;&quot;)</td>
</tr>
</tbody>
</table>

You can change the value for these properties only by using the `ConnectionString` property.
Examples
See Also

MySqlConnection Class
MySqlConnection Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySqlConnection** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginDbTransaction</strong></td>
<td>Starts a database transaction. (Inherited from <code>DbConnection</code>.)</td>
</tr>
<tr>
<td><strong>BeginTransaction</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>CancelQuery</strong></td>
<td>Changes the current database for an open <code>MySqlConnection</code>. (Overrides <code>DbConnection::ChangeDatabase(String)</code>.)</td>
</tr>
<tr>
<td><strong>ChangeDatabase</strong></td>
<td>Changes the current database for an open <code>MySqlConnection</code>. (Overrides <code>DbConnection::ChangeDatabase(String)</code>.)</td>
</tr>
<tr>
<td><strong>ClearAllPools</strong></td>
<td>Clears all connection pools.</td>
</tr>
<tr>
<td><strong>ClearPool</strong></td>
<td>Empties the connection pool associated with the specified connection.</td>
</tr>
<tr>
<td><strong>Clone</strong></td>
<td>Creates a new <code>MySqlConnection</code> object with the exact same <code>ConnectionString</code> value</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Closes the connection to the database. This is the preferred method of closing any open connection. (Overrides <code>DbConnection::Close()</code>.)</td>
</tr>
<tr>
<td><strong>CreateCommand</strong></td>
<td>Creates and returns a <code>MySqlCommand</code> object associated with the <code>MySqlConnection</code>.</td>
</tr>
<tr>
<td><strong>CreateDbCommand</strong></td>
<td>Creates and returns a <code>DbCommand</code> object associated with the current connection. (Inherited from <code>DbConnection</code>.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>EnlistTransaction</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
Finalize

Releases unmanaged resources and performs other cleanup operations before the
Component is reclaimed by garbage collection.
(Inherited from Component.)

GetHashCode

Serves as a hash function for a particular type.
(Inherited from Object.)

GetLifetimeService

Retrieves the current lifetime service object
that controls the lifetime policy for this
instance.
(Inherited from MarshalByRefObject.)

GetSchema

Overloaded.

GetService

Returns an object that represents a service
provided by the Component or by its
Container.
(Inherited from Component.)

GetType

Gets the Type of the current instance.
(Inherited from Object.)

InitializeLifetimeService

Obtains a lifetime service object to control the
lifetime policy for this instance.
(Inherited from MarshalByRefObject.)

MemberwiseClone

Overloaded.

OnStateChange

Raises the StateChange event.
(Inherited from DbConnection.)

Open

Opens a database connection with the
property settings specified by the
ConnectionString.
(Overrides DbConnection..::.Open()Q().)

Ping

Ping

Returns a String containing the name of the
Component, if any. This method should not be
overridden.
(Inherited from Component.)
See Also

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BeginTransaction()</code></td>
<td>Begins a database transaction.</td>
</tr>
<tr>
<td><code>BeginTransaction(IsolationLevel)</code></td>
<td>Begins a database transaction with the specified isolation level.</td>
</tr>
</tbody>
</table>
See Also

MySqlConnection Class
MySqlConnection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Begins a database transaction.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlConnection BeginTransaction()

Visual Basic (Declaration)

Public Function BeginTransaction As MySqlTransaction

Visual C++

public:
MySqlConnection^ BeginTransaction()

Return Value

An object representing the new transaction.
# Remarks

This command is equivalent to the MySQL BEGIN TRANSACTION command.

You must explicitly commit or roll back the transaction using the `Commit()` or `Rollback()` method.

**Note:** If you do not specify an isolation level, the default isolation level is used. To specify an isolation level with the `BeginTransaction()` method, use the overload that takes the `iso` parameter. Also note that any attempt to begin a transaction while a transaction is in progress will throw an exception on MySQL 4.1 and higher. On MySQL 4.0, an exception will not be thrown because servers 4.0 and earlier did not report their transaction status.
Examples

The following example creates a `MySqlConnection` and a `MySqlTransaction`. It also demonstrates how to use the `BeginTransaction`, a `Commit()()`, and `Rollback()()` methods.

**VB.NET**

```vbnet
Public Sub RunTransaction(myConnString As String)
    Dim myConnection As New MySqlConnection(myConnString)
    myConnection.Open()

    Dim myCommand As MySqlCommand = myConnection.CreateCommand()
    Dim myTrans As MySqlTransaction

    ' Start a local transaction
    myTrans = myConnection.BeginTransaction()
    ' Must assign both transaction object and connection
    ' to Command object for a pending local transaction
    myCommand.Connection = myConnection
    myCommand.Transaction = myTrans

    Try
        myCommand.CommandText = "Insert into Test (id, desc) VALUES (100, 'D"
        myCommand.ExecuteNonQuery()
    myCommand.CommandText = "Insert into Test (id, desc) VALUES (101, 'D"
        myCommand.ExecuteNonQuery()
        myTrans.Commit()
        Console.WriteLine("Both records are written to database.")
    Catch e As Exception
        Try
            myTrans.Rollback()
        Catch ex As MySqlException
            If Not myTrans.Connection Is Nothing Then
                Console.WriteLine("An exception of type " + ex.GetType().ToString() + " was encountered while attempting to roll back the transaction.")
            End If
        End Try
        Console.WriteLine("An exception of type " + e.GetType().ToString() + " was encountered while inserting the data.")
        Finally
            myConnection.Close()
    End Try
```
public void RunTransaction(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();

    MySqlCommand myCommand = myConnection.CreateCommand();
    MySqlTransaction myTrans;

    // Start a local transaction
    myTrans = myConnection.BeginTransaction();
    // Must assign both transaction object and connection
    // to Command object for a pending local transaction
    myCommand.Connection = myConnection;
    myCommand.Transaction = myTrans;

    try
    {
        myCommand.CommandText = "insert into Test (id, desc) VALUES (100, 'D"
        myCommand.ExecuteNonQuery();
        myCommand.CommandText = "insert into Test (id, desc) VALUES (101, 'D"
        myCommand.ExecuteNonQuery();
        myTrans.Commit();
        Console.WriteLine("Both records are written to database.");
    }
    catch(Exception e)
    {
        try
        {
            myTrans.Rollback();
        }
        catch (SqlException ex)
        {
            if (myTrans.Connection != null)
            {
                Console.WriteLine("An exception of type " + ex.GetType() + " was encountered while attempting to roll back the transaction.");
            }
        }
        Console.WriteLine("An exception of type " + e.GetType() + " was encountered while inserting the data.");
        Console.WriteLine("Neither record was written to database.");
    }
    finally
    {

myConnection.Close();
}
}
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System...InvalidOperationException</code></td>
<td>Parallel transactions are not supported.</td>
</tr>
</tbody>
</table>
See Also

SqlConnection Class
BeginTransaction Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Begins a database transaction with the specified isolation level.

**Namespace:** MySQL.Data.MySqlClient  
**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlTransaction BeginTransaction(
    IsolationLevel iso
)
```

Visual Basic (Declaration)

```vbnet
Public Function BeginTransaction ( _
    iso As IsolationLevel _
) As MySqlTransaction
```

Visual C++

```cpp
public:
MySqlTransaction^ BeginTransaction(
    IsolationLevel iso
)
```

Parameters

iso

Type: `System.Data::IsolationLevel`
The isolation level under which the transaction should run.

Return Value

An object representing the new transaction.
Remarks

This command is equivalent to the MySQL BEGIN TRANSACTION command.

You must explicitly commit or roll back the transaction using the Commit() or Rollback() method.

☑️ Note: If you do not specify an isolation level, the default isolation level is used. To specify an isolation level with the BeginTransaction() method, use the overload that takes the iso parameter. Also note that any attempt to begin a transaction while a transaction is in progress will throw an exception on MySQL 4.1 and higher. On MySQL 4.0, an exception will not be thrown because servers 4.0 and earlier did not report their transaction status.
Examples

The following example creates a **MySqlConnection** and a **MySqlTransaction**. It also demonstrates how to use the BeginTransaction, a **Commit()**), and **Rollback()** methods.

**VB.NET**

```vbnet
Public Sub RunTransaction(myConnString As String)
    Dim myConnection As New MySqlConnection(myConnString)
    myConnection.Open()

    Dim myCommand As MySqlCommand = myConnection.CreateCommand()
    Dim myTrans As MySqlTransaction

    ' Start a local transaction
    myTrans = myConnection.BeginTransaction()
    ' Must assign both transaction object and connection
    ' to Command object for a pending local transaction
    myCommand.Connection = myConnection
    myCommand.Transaction = myTrans

    Try
        myCommand.CommandText = "Insert into Test (id, desc) VALUES (100, 'D"
        myCommand.ExecuteNonQuery()
        myCommand.CommandText = "Insert into Test (id, desc) VALUES (101, 'D"
        myCommand.ExecuteNonQuery()
        myTrans.Commit()
        Console.WriteLine("Both records are written to database.")
    Catch e As Exception
        Try
            myTrans.Rollback()
        Catch ex As MySqlException
            If Not myTrans.Connection Is Nothing Then
                Console.WriteLine("An exception of type " + ex.GetType().ToString() + " was encountered while attempting to roll back the transaction.")
            End If
        End Try
        Console.WriteLine("An exception of type " + e.GetType().ToString() + " was encountered while inserting the data.")
    End Try

    Finally
        myConnection.Close()
    End Try
```
C#

```csharp
public void RunTransaction(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();

    MySqlCommand myCommand = myConnection.CreateCommand();
    MySqlTransaction myTrans;

    // Start a local transaction
    myTrans = myConnection.BeginTransaction();
    // Must assign both transaction object and connection
    // to Command object for a pending local transaction
    myCommand.Connection = myConnection;
    myCommand.Transaction = myTrans;

    try
    {
        myCommand.CommandText = "insert into Test (id, desc) VALUES (100, 'D"
        myCommand.ExecuteNonQuery();
        myCommand.CommandText = "insert into Test (id, desc) VALUES (101, 'D"
        myCommand.ExecuteNonQuery();
        myTrans.Commit();
        Console.WriteLine("Both records are written to database.");
    }
    catch (Exception e)
    {
        try
        {
            myTrans.Rollback();
        }
        catch (SqlException ex)
        {
            if (myTrans.Connection != null)
            {
                Console.WriteLine("An exception of type " + ex.GetType() +
" was encountered while attempting to roll back the transaction.");
            }
        }
    }

    Console.WriteLine("An exception of type " + e.GetType() +
" was encountered while inserting the data.");
    Console.WriteLine("Neither record was written to database.");
}
finally
{
}
```
myConnection.Close();
}
}
# Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System...::InvalidOperationException</code></td>
<td>Parallel exceptions are not supported.</td>
</tr>
</tbody>
</table>
See Also

SqlConnection Class
beginTransaction Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLConnection...:::CancelQuery Method

MySQLConnection Class  See Also  Send Feedback

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public void CancelQuery(
  int timeout
)

Visual Basic (Declaration)

Public Sub CancelQuery ( _
  timeout As Integer _
)

Visual C++

public:
void CancelQuery(
  int timeout
)

Parameters

timeout
  Type: System::Int32
See Also

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Changes the current database for an open MySqlConnection.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySql.Data](#) (in [MySql.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

```csharp
public override void ChangeDatabase(
    string databaseName
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub ChangeDatabase (
        databaseName As String
    )
```

Visual C++

```cpp
public:
    virtual void ChangeDatabase(
        String^ databaseName
    ) override
```

Parameters

databaseName
    Type: System::String
    The name of the database to use.

Implements

IDbConnection::ChangeDatabase(String)
Remarks

The value supplied in the database parameter must be a valid database name. The database parameter cannot contain a null value, an empty string, or a string with only blank characters.

When you are using connection pooling against MySQL, and you close the connection, it is returned to the connection pool. The next time the connection is retrieved from the pool, the reset connection request executes before the user performs any operations.
Examples

The following example creates a `MySqlConnection` and displays some of its read-only properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlConnection()
    Dim myConnString As String = _
    "Persist Security Info=False;database=test;server=localhost;user id=
    Dim myConnection As New MySqlConnection( myConnString )
    myConnection.Open()
    MessageBox.Show( "Server Version: " + myConnection.ServerVersion _
    + ControlChars.NewLine + "Database: " + myConnection.Database )
    myConnection.ChangeDatabase( "test2" )
    MessageBox.Show( "ServerVersion: " + myConnection.ServerVersion _
    + ControlChars.NewLine + "Database: " + myConnection.Database )
    myConnection.Close()
End Sub
```

**C#**

```csharp
public void CreateMySqlConnection()
{
    string myConnString =
    "Persist Security Info=False;database=test;server=localhost;user id=
    MySqlConnection myConnection = new MySqlConnection( myConnString );
    myConnection.Open();
    MessageBox.Show( "Server Version: " + myConnection.ServerVersion
    + \nDatabase: " + myConnection.Database );
    myConnection.ChangeDatabase( "test2" );
    MessageBox.Show( "ServerVersion: " + myConnection.ServerVersion
    + \nDatabase: " + myConnection.Database );
    myConnection.Close();
}
```
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System...:: ArgumentException</td>
<td>The database name is not valid.</td>
</tr>
<tr>
<td>System...:: InvalidOperationException</td>
<td>The connection is not open.</td>
</tr>
<tr>
<td>MySql.Data.MySqlClient...:: MySqlException</td>
<td>Cannot change the database.</td>
</tr>
</tbody>
</table>
See Also

SqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Clears all connection pools.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

class public static void ClearAllPools()

Visual Basic (Declaration)

Public Shared Sub ClearAllPools

Visual C++

public:
static void ClearAllPools()
Remarks

ClearAllPools essentially performs a `ClearPool(MySqlConnection)` on all current connection pools.
See Also

MySQLConnection Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Empties the connection pool associated with the specified connection.

**Namespace:**  [MySql.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public static void ClearPool(
    MySqlConnection connection
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Sub ClearPool ( _
    connection As MySqlConnection _
)
```

### Visual C++

```cpp
public:
    static void ClearPool(
        MySqlConnection^ connection
    )
```

## Parameters

**connection**

Type: `System.Data.MySqlClient::MySqlConnection`

The `MySqlConnection` associated with the pool to be cleared.
Remarks

ClearPool clears the connection pool that is associated with the connection. If additional connections associated with connection are in use at the time of the call, they are marked appropriately and are discarded (instead of being returned to the pool) when Close is called on them.
See Also

**SqlConnection Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Create a new MySqlConnection object with the exact same ConnectionString value

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

    public MySqlConnection Clone()

Visual Basic (Declaration)

Public Function Clone As MySqlConnection

Visual C++

    public: 
    MySqlConnection^ Clone()

Return Value

A cloned MySqlConnection object
See Also

** MySqlConnection Class**  
** MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Closes the connection to the database. This is the preferred method of closing any open connection.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override void Close()

Visual Basic (Declaration)

Public Overrides Sub Close

Visual C++

public:
virtual void Close() override

Implements

IDbConnection::Close()()
Remarks

The Close method rolls back any pending transactions. It then releases the connection to the connection pool, or closes the connection if connection pooling is disabled.

An application can call Close more than one time. No exception is generated.
Examples

The following example creates a `MySqlConnection`, opens it, displays some of its properties, then closes the connection.

**VB.NET**

```vbnet
Public Sub CreateMySqlConnection(myConnString As String)
    Dim myConnection As New MySqlConnection(myConnString)
    myConnection.Open()
    myConnection.Close()
End Sub
```

**C#**

```csharp
public void CreateMySqlConnection(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();
    myConnection.Close();
}
```
See Also

MySQLConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic  Visual C++
MySQL Connector/Net
MySQLConnection...::CreateCommand Method
MySQLConnection Class  See Also  Send Feedback

Creates and returns a **MySqlCommand** object associated with the **MySQLConnection**.

**Namespace:**  MySql.Data.MySqlClient
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCommand CreateCommand()

Visual Basic (Declaration)

Public Function CreateCommand As MySqlCommand

Visual C++

public:
MySQLCommand^ CreateCommand()

Return Value

A MySqlCommand object.
See Also

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C# Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net
MySqlConnection...::Dispose Method

MySQLConnection Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose()()</td>
<td>Releases all resources used by the Component. (Inherited from Component.)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases the unmanaged resources used by the Component and optionally releases the managed resources. (Inherited from Component.)</td>
</tr>
</tbody>
</table>
See Also

MySQLConnection Class
MySQLConnection Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Include Inherited Members

MySQL Connector/Net

MySqlConnection...::EnlistTransaction Method

See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnlistTransaction(Transaction)</td>
<td>Enlists in the specified transaction. (Inherited from DbConnection.)</td>
</tr>
<tr>
<td>EnlistTransaction(Transaction)</td>
<td>Enlists in the specified transaction.</td>
</tr>
</tbody>
</table>
See Also

MySqlConnection Class
MySqlConnection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlConnection...::EnlistTransaction Method (Transaction)

Enlists in the specified transaction.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public virtual void EnlistTransaction(
    Transaction transaction
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overridable Sub EnlistTransaction ( _
    transaction As Transaction _
)
```

**Visual C++**

```cpp
public:
    virtual void EnlistTransaction(
        Transaction^ transaction
    )
```

### Parameters

`transaction`

Type: `System.Transactions::Transaction`

A reference to an existing `Transaction` in which to enlist.
See Also

 MySqlConnection Class
 EnlistTransaction Overload
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlConnection...::GetSchema Method

MySQLConnection Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSchema()()</td>
<td>Returns schema information for the data source of this DbConnection. (Overrides DbConnection::GetSchema()().)</td>
</tr>
<tr>
<td>GetSchema(String)</td>
<td>Returns schema information for the data source of this DbConnection using the specified string for the schema name. (Overrides DbConnection::GetSchema(String).)</td>
</tr>
<tr>
<td>GetSchema(String, array&lt;String&gt;[])</td>
<td>Returns schema information for the data source of this DbConnection using the specified string for the schema name and the specified string array for the restriction values. (Overrides DbConnection::GetSchema(String, array&lt;String&gt;[]).)</td>
</tr>
</tbody>
</table>
See Also

MySQLConnection Class
MySQLConnection Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

`MySqlConnection...::GetSchema Method`

`MySqlConnection` See Also Send Feedback

Returns schema information for the data source of this `DbConnection`.

**Namespace:**  [MySql.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#
public override DataTable GetSchema()

Visual Basic (Declaration)
Public Overrides Function GetSchema As DataTable

Visual C++
public:
virtual DataTable^ GetSchema() override

Return Value

A DataTable that contains schema information.
See Also

**SqlConnection Class**
**GetSchema Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns schema information for the data source of this `DbConnection` using the specified string for the schema name.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override DataTable GetSchema(
    string collectionName
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function GetSchema ( _
    collectionName As String _
) As DataTable
```

### Visual C++

```cpp
public:
    virtual DataTable^ GetSchema(
        String^ collectionName
    ) override
```

### Parameters

collectionName
- **Type:** `System::String`
  - Specifies the name of the schema to return.

### Return Value

A `DataTable` that contains schema information.
See Also

MySqlConnection Class
GetSchema Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySQLConnection...::GetSchema Method (String, array<String>[][])

Returns schema information for the data source of this DbConnection using the specified string for the schema name and the specified string array for the restriction values.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override DataTable GetSchema(
    string collectionName,
    string[] restrictionValues
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function GetSchema ( _
    collectionName As String, _
    restrictionValues As String() _
) As DataTable
```

Visual C++

```cpp
public:
virtual DataTable^ GetSchema(
    String^ collectionName, 
    array<String^>^ restrictionValues
) override
```

Parameters

collectionName
Type: System::::String
Specifies the name of the schema to return.

restrictionValues
Type: array< System::::String >[]
Specifies a set of restriction values for the requested schema.

Return Value

A DataTable that contains schema information.
See Also

SqlConnection Class
GetSchema Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connection...:::MemberwiseClone Method

MySQLConnection Class  See Also  Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberwiseClone()()</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>MemberwiseClone(Boolean)</td>
<td>Creates a shallow copy of the current MarshalByRefObject object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">MarshalByRefObject</a>.)</td>
</tr>
</tbody>
</table>
See Also

SqlConnection Class
SqlConnection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Opens a database connection with the property settings specified by the ConnectionString.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override void Open()

Visual Basic (Declaration)

Public Overrides Sub Open

Visual C++

public:
virtual void Open() override

Implements

IDbConnection...::Open()()
Remarks

The MySqlConnection draws an open connection from the connection pool if one is available. Otherwise, it establishes a new connection to an instance of MySQL.
Examples

The following example creates a MySqlConnection, opens it, displays some of its properties, then closes the connection.

**VB.NET**

```vbnet
Public Sub CreateMySqlConnection(myConnString As String)
    Dim myConnection As New MySqlConnection(myConnString)
    myConnection.Open()
    myConnection.Close()
End Sub
```

**C#**

```csharp
public void CreateMySqlConnection(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();
    MessageBox.Show("ServerVersion: 	" + myConnection.ServerVersion + 
                    "\nState: 	" + myConnection.State.ToString());
    myConnection.Close();
}
```
# Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System::::InvalidOperationException</code></td>
<td>Cannot open a connection without specifying a data source or server.</td>
</tr>
<tr>
<td><code>MySql.Data.MySqlClient::::MySqlException</code></td>
<td>A connection-level error occurred while opening the connection.</td>
</tr>
</tbody>
</table>
See Also

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Ping

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool Ping()

Visual Basic (Declaration)

Public Function Ping As Boolean

Visual C++

public:
bool Ping()

Return Value
See Also

SqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlConnection` type exposes the following members.

MySqlConnection Class  See Also  Send Feedback
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanRaiseEvents</td>
<td>Gets a value indicating whether the component can raise an event. <em>(Inherited from <a href="#">Component</a>)</em></td>
</tr>
<tr>
<td>ConnectionString</td>
<td>Gets or sets the string used to connect to a MySQL Server database. <em>( Overrides <a href="#">DbConnection..::.ConnectionString</a> )</em></td>
</tr>
<tr>
<td>ConnectionTimeout</td>
<td>Gets the time to wait while trying to establish a connection before terminating the attempt and generating an error. <em>( Overrides <a href="#">DbConnection..::.ConnectionTimeout</a> )</em></td>
</tr>
<tr>
<td>Container</td>
<td>Gets the <a href="#">.IContainer</a> that contains the <a href="#">Component</a>. <em>(Inherited from <a href="#">Component</a>)</em></td>
</tr>
<tr>
<td>Database</td>
<td>Gets the name of the current database or the database to be used after a connection is opened. <em>( Overrides <a href="#">DbConnection..::.Database</a> )</em></td>
</tr>
<tr>
<td>DataSource</td>
<td>Gets the name of the MySQL server to which to connect. <em>( Overrides <a href="#">DbConnection..::.DataSource</a> )</em></td>
</tr>
<tr>
<td>DbProviderFactory</td>
<td>Gets the <a href="#">DbProviderFactory</a> for this <a href="#">DbConnection</a>. <em>(Inherited from <a href="#">DbConnection</a>)</em></td>
</tr>
<tr>
<td>DesignMode</td>
<td>Gets a value that indicates whether the <a href="#">Component</a> is currently in design mode. <em>(Inherited from <a href="#">Component</a>)</em></td>
</tr>
<tr>
<td>Events</td>
<td>Gets the list of event handlers that are attached to this <a href="#">Component</a>. <em>(Inherited from <a href="#">Component</a>)</em></td>
</tr>
<tr>
<td>ServerThread</td>
<td>Returns the id of the server thread this connection is executing on.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>Gets a string containing the version of the MySQL server to which the client is connected.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Site</td>
<td>Gets or sets the ISite of the Component. (Inherited from Component.)</td>
</tr>
<tr>
<td>State</td>
<td>Gets the current state of the connection. (Overrides DbConnection:::State.)</td>
</tr>
<tr>
<td>UseCompression</td>
<td>Indicates if this connection should use compression when communicating with the server.</td>
</tr>
</tbody>
</table>
See Also

SqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
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: 6.2.2.0
Syntax

C#

public override string ConnectionString { get; set; }

Visual Basic (Declaration)

Public Overrides Property ConnectionString As String

Visual C++

public:
virtual property String^ ConnectionString {
     String^ get () override;
     void set (String^ value) override;
}

Implements

IDbConnection::*:ConnectionString
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;UserId=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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Connect Timeout              | -or-    | The length of time (in seconds) to wait for a connection to the server before
                              | 15                              | terminating the attempt and generating an error. |
| Host                         |         | The name or network address of the instance of MySQL to which                 |


-or-
Server
-or-
Data Source
-or-
DataSource
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.
-or-
Address
-or-
Addr
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.
-or-
Network Address
-or-
Port
The port MySQL is using to listen for connections. This value is ignored if the connection protocol is anything but socket. Specifies the type of connection to make to the server.
-or-
Protocol
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
<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CharSet</td>
<td>-or</td>
<td>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.</td>
</tr>
<tr>
<td>Character Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logging</td>
<td>false</td>
<td>When true, various pieces of information is output to any configured TraceListeners. When true, multiple SQL statements can be sent with one command execution.</td>
</tr>
<tr>
<td>Allow Batch</td>
<td>true</td>
<td>Starting with MySQL 4.1.1, batch statements should be separated by the server-defined separator character.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Commands sent to earlier versions of MySQL should be seperated with ';'.</td>
</tr>
<tr>
<td>Encrypt</td>
<td>false</td>
<td>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.</td>
</tr>
<tr>
<td>Initial Catalog</td>
<td>mysql</td>
<td>The name of the database to use intially</td>
</tr>
<tr>
<td>Database Password</td>
<td>-or</td>
<td>The password for the MySQL account being used.</td>
</tr>
<tr>
<td></td>
<td>pwd</td>
<td>When set to false or no (strongly recommended), security-sensitive</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Persist Security Info</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>User Id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-or-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-or-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-or-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shared Memory Name</td>
<td>MYSQL</td>
<td></td>
</tr>
<tr>
<td>Allow Zero Datetime</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Convert Zero Datetime</td>
<td>false</td>
<td></td>
</tr>
</tbody>
</table>

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.

The MySQL login account being used.

The name of the shared memory object to use for communication if the connection protocol is set to memory.

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.

True to have `MySqlDataReader.GetValue()` and `MySqlDataReader.GetDateTime()` return `DateTime.MinValue` for
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipe Name</td>
<td>mysql</td>
</tr>
<tr>
<td>Pipe</td>
<td>mysql</td>
</tr>
<tr>
<td>Use Performance Monitor</td>
<td>false</td>
</tr>
<tr>
<td>Procedure Cache Size</td>
<td>25</td>
</tr>
<tr>
<td>Ignore Prepare</td>
<td>true</td>
</tr>
<tr>
<td>Use Procedure Bodies</td>
<td>true</td>
</tr>
</tbody>
</table>

- **Pipe Name**: date or datetime columns that have illegal values. When set to the name of a named pipe, the MySqlConnection will attempt to connect to MySQL on that named pipe.

- **Pipe**: This setting only applies to the Windows platform.

- **Use Performance Monitor**: Posts performance data that can be tracked using perfmon.

- **Procedure Cache Size**: How many stored procedure definitions can be held in the cache. 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.

- **Ignore Prepare**: 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.
<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Enlist</td>
<td>true</td>
<td>Indicates whether the connection should automatically enlist in the current transaction, if there is one.</td>
</tr>
<tr>
<td>Respect Binary Flags</td>
<td>true</td>
<td>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.</td>
</tr>
<tr>
<td>BlobAsUTF8IncludePattern</td>
<td>null</td>
<td>Pattern that should be used to indicate which blob columns should be treated as UTF-8.</td>
</tr>
<tr>
<td>BlobAsUTF8ExcludePattern</td>
<td>null</td>
<td>Pattern that should be used to indicate which blob columns should not be treated as UTF-8.</td>
</tr>
<tr>
<td>Default Command Timeout</td>
<td>30</td>
<td>The default timeout that new MySqlCommand objects will use unless changed.</td>
</tr>
<tr>
<td>Allow User Variables</td>
<td>false</td>
<td>Should the provider expect user variables in the SQL.</td>
</tr>
<tr>
<td>Interactive -or- Interactive Session</td>
<td>false</td>
<td>Should this session be considered interactive?</td>
</tr>
<tr>
<td>Functions Return String</td>
<td>false</td>
<td>Set this option to true to force the return value of SQL functions to be string.</td>
</tr>
<tr>
<td>Use Affected Rows</td>
<td>false</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Lifetime</td>
<td>0</td>
<td>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.</td>
</tr>
<tr>
<td>Max Pool Size</td>
<td>100</td>
<td>The maximum number of connections allowed in the pool.</td>
</tr>
<tr>
<td>Min Pool Size</td>
<td>0</td>
<td>The minimum number of connections allowed in the pool. 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. Specifies whether the database connection should be reset when</td>
</tr>
<tr>
<td>Pooling</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>
Connection Reset  false

being drawn from the pool. Leaving this as false will yield 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.

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

**VB.NET**

```vbnet
Public Sub CreateConnection()
  Dim myConnection As New MySqlConnection()
  myConnection.ConnectionString = "Persist Security Info=False;database=myDB;server=myHost;Connect Timeout=30;user id=myUser;pwd=myPass"
  myConnection.Open()
End Sub 'CreateConnection
```

**C#**

```csharp
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.

**VB.NET**

```vbnet
Public Sub CreateConnection()
    Dim myConnection As New MySqlConnection()
    myConnection.ConnectionString = "database=myDB;server=/var/lib/mysql/mysql.sock;user=myUser;pwd=myPass"
    myConnection.Open()
End Sub 'CreateConnection
```

**C#**

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

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the time to wait while trying to establish a connection before terminating the attempt and generating an error.

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysqlconnector.net)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override int ConnectionTimeout { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property ConnectionTimeout As Integer

Visual C++

public:
virtual property int ConnectionTimeout {
    int get () override;
}

Implements

IDbConnection:::ConnectionTimeout
Remarks

A value of 0 indicates no limit, and should be avoided in a **ConnectionString** because an attempt to connect will wait indefinitely.
Examples

The following example creates a MySqlConnection and sets some of its properties in the connection string.

**VB.NET**

```vbnet
Public Sub CreateSqlConnection()
    Dim myConnection As New MySqlConnection()
    myConnection.ConnectionString = "Persist Security Info=False;Username=user;Password=pass;database=test1;server=localhost;Connect Timeout=30"
    myConnection.Open()
End Sub
```

**C#**

```csharp
public void CreateSqlConnection()
{
    MySqlConnection myConnection = new MySqlConnection();
    myConnection.ConnectionString = "Persist Security Info=False;Username=user;Password=pass;database=test1;server=localhost;Connect Timeout=30"
    myConnection.Open();
}
```
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System...(ArgumentException)</code></td>
<td>The value set is less than 0.</td>
</tr>
</tbody>
</table>
See Also

**MySqlConnection Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic  Visual C++
MySQL Connector/Net
MySQLConnection..:::Database Property
MySQLConnection Class  Example  See Also  Send Feedback

Gets the name of the current database or the database to be used after a connection is opened.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override string Database { get; }
```

### Visual Basic (Declaration)

```vbnet
Public Overrides ReadOnly Property Database As String
```

### Visual C++

```cpp
public:
virtual property String^ Database {
    String^ get () override;
}
```

## Return Value

The name of the current database or the name of the database to be used after a connection is opened. The default value is an empty string.

## Implements

`IDbConnection::Database`
Remarks

The Database property does not update dynamically. If you change the current database using a SQL statement, then this property may reflect the wrong value. If you change the current database using the ChangeDatabase(String) method, this property is updated to reflect the new database.
**Examples**

The following example creates a **MySqlConnection** and displays some of its read-only properties.

**VB.NET**

```vbnet
Public Sub CreateMySqlConnection()
    Dim myConnString As String = _
        "Persist Security Info=False;database=test;server=localhost;user id=
    Dim myConnection As New MySqlConnection( myConnString )
    myConnection.Open()
    MessageBox.Show( "Server Version: " + myConnection.ServerVersion _
        + ControlChars.NewLine + "Database: " + myConnection.Database )
    myConnection.ChangeDatabase( "test2" )
    MessageBox.Show( "ServerVersion: " + myConnection.ServerVersion _
        + ControlChars.NewLine + "Database: " + myConnection.Database )
    myConnection.Close()
End Sub
```

**C#**

```csharp
public void CreateMySqlConnection()
{
    string myConnString = _
        "Persist Security Info=False;database=test;server=localhost;user id=
    MySqlConnection myConnection = new MySqlConnection( myConnString );
    myConnection.Open();
    MessageBox.Show( "Server Version: " + myConnection.ServerVersion _
        + ControlChars.NewLine + "Database: " + myConnection.Database );
    myConnection.ChangeDatabase( "test2" );
    MessageBox.Show( "ServerVersion: " + myConnection.ServerVersion _
        + ControlChars.NewLine + "Database: " + myConnection.Database );
    myConnection.Close();
}
```
See Also

**MySqlConnection Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the name of the MySQL server to which to connect.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/products/connector-dotnet/)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override string DataSource { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property DataSource As String

Visual C++

public:
virtual property String^ DataSource {
    String^ get () override;
}
See Also

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ServerThread Property

Returns the id of the server thread this connection is executing on

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public int ServerThread { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property ServerThread As Integer
```

**Visual C++**

```cpp
public:
property int ServerThread {
    int get () ;
}
```
See Also

MySQLConnection Class
Mysql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a string containing the version of the MySQL server to which the client is connected.

**Namespace:** [MySql.Data.MySqlClient](https://www.mysql.connector.net/

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override string ServerVersion { get; }
```

Visual Basic (Declaration)

```
Public Overrides ReadOnly Property ServerVersion As String
```

Visual C++

```c++
public:
virtual property String^ ServerVersion {
    String^ get () override;
}
```

Return Value

The version of the instance of MySQL.
Examples

The following example creates a **MySqlConnection**, opens it, displays some of its properties, then closes the connection.

**VB.NET**

```vbnet
Public Sub CreateMySqlConnection(myConnString As String)
Dim myConnection As New MySqlConnection(myConnString)
myConnection.Open()
myConnection.Close()
End Sub
```

**C#**

```csharp
public void CreateMySqlConnection(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();
    myConnection.Close();
}
```
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System...::InvalidOperationException</code></td>
<td>The connection is closed.</td>
</tr>
</tbody>
</table>
See Also

**MySqlConnection Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the current state of the connection.

**Namespace:**  [MySQL.Data.MySqlClient](https://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override ConnectionState State { get; }
```

Visual Basic (Declaration)

```vbnet
Public Overrides ReadOnly Property State As ConnectionState
```

Visual C++

```cpp
public:
virtual property ConnectionState State {
    ConnectionState get () override;
}
```

Return Value

A bitwise combination of the ConnectionState values. The default is Closed.

Implements

IDbConnection::State
Remarks

The allowed state changes are:

- From Closed to Open, using the Open method of the connection object.
- From Open to Closed, using either the Close method or the Dispose method of the connection object.
Examples

The following example creates a **MySqlConnection**, opens it, displays some of its properties, then closes the connection.

**VB.NET**

```vbnet
Public Sub CreateMySqlConnection(myConnString As String)
    Dim myConnection As New MySqlConnection(myConnString)
    myConnection.Open()
    myConnection.Close()
End Sub
```

**C#**

```csharp
public void CreateMySqlConnection(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();
    myConnection.Close();
}
```
See Also

MySQLConnection Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
UseCompression Property

Indicates if this connection should use compression when communicating with the server.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool UseCompression { get; }

Visual Basic (Declaration)

Public ReadOnly Property UseCompression As Boolean

Visual C++

public:
property bool UseCompression {
    bool get ();
}
}
See Also

SqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The MySqlConnection type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposed</td>
<td>Occurs when the component is disposed by a call to the Dispose() method.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Component.)</td>
</tr>
<tr>
<td>InfoMessage</td>
<td>Occurs when MySQL returns warnings as a result of executing a command or query.</td>
</tr>
<tr>
<td>StateChange</td>
<td>Occurs when the state of the event changes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DbConnection.)</td>
</tr>
</tbody>
</table>
See Also

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Occurs when MySQL returns warnings as a result of executing a command or query.

**Namespace:** MySql.Data.MySqlClient  **Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public event MySqlInfoMessageEventHandler InfoMessage

Visual Basic (Declaration)

Public Event InfoMessage As MySqlInfoMessageEventHandler

Visual C++

public:
    event MySqlInfoMessageEventHandler^ InfoMessage {
        void add (MySqlInfoMessageEventHandler^ value);
        void remove (MySqlInfoMessageEventHandler^ value);
    }
Remarks
See Also

SqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Allows the user to specify the type of connection that should be used.

**Namespace:** [MySQL.Data.MySqlClient](#)  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public enum MySqlConnectionProtocol
```

**Visual Basic (Declaration)**

```vbnet
Public Enumeration MySqlConnectionProtocol
```

**Visual C++**

```cpp
public enum class MySqlConnectionProtocol
```
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sockets</td>
<td>TCP/IP style connection. Works everywhere.</td>
</tr>
<tr>
<td>Socket</td>
<td>TCP/IP style connection. Works everywhere.</td>
</tr>
<tr>
<td>Tcp</td>
<td>TCP/IP style connection. Works everywhere.</td>
</tr>
<tr>
<td>Pipe</td>
<td>Named pipe connection. Works only on Windows systems.</td>
</tr>
<tr>
<td>NamedPipe</td>
<td>Named pipe connection. Works only on Windows systems.</td>
</tr>
<tr>
<td>UnixSocket</td>
<td>Unix domain socket connection. Works only with Unix systems.</td>
</tr>
<tr>
<td>Unix</td>
<td>Unix domain socket connection. Works only with Unix systems.</td>
</tr>
<tr>
<td>SharedMemory</td>
<td>Shared memory connection. Currently works only with Windows systems.</td>
</tr>
<tr>
<td>Memory</td>
<td>Shared memory connection. Currently works only with Windows systems.</td>
</tr>
</tbody>
</table>
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlConnectionStringBuilder Class

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public class MySqlConnectionStringBuilder : DbConnectionStringBuilder
```

**Visual Basic (Declaration)**

```vbnet
Public Class MySqlConnectionStringBuilder
    Inherits DbConnectionStringBuilder
```

**Visual C++**

```c++
public ref class MySqlConnectionConnectionStringBuilder : public DbConnectionString
```
Inheritance Hierarchy

System..:::Object
   System.Data.Common..:::DbConnectionStringBuilder
      MySql.Data.MySqlClient..:::MySqlConnectionStringBuilder
See Also

MySqlConnectionStringBuilder Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLConnectionStringBuilder` type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlConnectionStringBuilder</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Adds an entry with the specified key and value into the <code>DbConnectionStringBuilder</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DbConnectionStringBuilder</code>.)</td>
</tr>
<tr>
<td><strong>Clear</strong></td>
<td>Clears the collection of <code>PropertyDescriptor</code> objects on the associated <code>DbConnectionStringBuilder</code>.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbConnectionStringBuilder..::.Clear()</code>.)</td>
</tr>
<tr>
<td><strong>ClearPropertyDescriptors</strong></td>
<td>Determines whether the <code>DbConnectionStringBuilder</code> contains a specific key.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DbConnectionStringBuilder</code>.)</td>
</tr>
<tr>
<td><strong>ContainsKey</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Compares the connection information in this <code>DbConnectionStringBuilder</code> object with the connection information in the supplied object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DbConnectionStringBuilder</code>.)</td>
</tr>
<tr>
<td><strong>EquivalentTo</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetConnectionString</strong></td>
<td>Fills a supplied <code>Hashtable</code> with information about all the properties of this <code>DbConnectionStringBuilder</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DbConnectionStringBuilder</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><strong>GetProperties</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Remove</strong></td>
<td>Indicates whether the specified key exists in this <strong>DbConnectionStringBuilder</strong> instance.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <strong>DbConnectionStringBuilder::Remove(String)</strong>.)</td>
</tr>
<tr>
<td><strong>ShouldSerialize</strong></td>
<td>Returns the connection string associated with this <strong>DbConnectionStringBuilder</strong> instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>DbConnectionStringBuilder</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from <strong>DbConnectionStringBuilder</strong>.)</td>
</tr>
<tr>
<td></td>
<td>(Overrides)</td>
</tr>
<tr>
<td><strong>TryGetValue</strong></td>
<td>(Overrides <strong>DbConnectionStringBuilder::TryGetValue(String, Object%)</strong>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowBatch</td>
<td>Gets or sets a boolean value that indicates whether this connection will allow commands to send multiple SQL statements in one execution.</td>
</tr>
<tr>
<td>AllowUserVariables</td>
<td>Gets or sets a boolean value that indicates if zero date time values are supported.</td>
</tr>
<tr>
<td>AllowZeroDateTime</td>
<td></td>
</tr>
<tr>
<td>AutoEnlist</td>
<td>Gets or sets the pattern that matches the columns that should not be treated as UTF8.</td>
</tr>
<tr>
<td>BlobAsUTF8ExcludePattern</td>
<td>Gets or sets the pattern that matches the columns that should be treated as UTF8.</td>
</tr>
<tr>
<td>BlobAsUTF8IncludePattern</td>
<td>Gets or sets a value that indicates whether the ConnectionString property is visible in Visual Studio designers. (Inherited from DbConnectionStringBuilder.)</td>
</tr>
<tr>
<td>BrowsableConnectionString</td>
<td>Gets or sets the character set that should be used for sending queries to the server.</td>
</tr>
<tr>
<td>CertificateFile</td>
<td>Gets or sets the lifetime of a pooled connection.</td>
</tr>
<tr>
<td>CertificatePassword</td>
<td>Gets or sets the protocol that should be used for communicating with MySQL.</td>
</tr>
<tr>
<td>CertificateStoreLocation</td>
<td>Gets or sets a boolean value indicating if the connection should be reset when retrieved from the pool.</td>
</tr>
<tr>
<td>CertificateThumbprint</td>
<td>Gets or sets the connection string associated with the DbConnectionStringBuilder. (Inherited from DbConnectionStringBuilder.)</td>
</tr>
<tr>
<td>CharacterSet</td>
<td></td>
</tr>
<tr>
<td>ConnectionLifeTime</td>
<td></td>
</tr>
<tr>
<td>ConnectionProtocol</td>
<td></td>
</tr>
<tr>
<td>ConnectionReset</td>
<td></td>
</tr>
<tr>
<td>ConnectionString</td>
<td></td>
</tr>
</tbody>
</table>
**ConnectionTimeout**
- Gets or sets the connection timeout.

**ConvertZeroDateTime**
- Gets or sets a boolean value indicating if zero datetime values should be converted to DateTime.MinValue.

**Count**
- Gets the current number of keys that are contained within the **ConnectionString** property. 
  (Inherited from **DbConnectionStringBuilder**.)

**Database**
- Gets or sets the name of the database the connection should initially connect to.

**DefaultCommandTimeout**
- Gets or sets the default command timeout.

**FunctionsReturnString**
- Gets or sets a boolean value indicating if calls to Prepare() should be ignored.

**IgnorePrepare**
- Gets or sets a boolean value indicating if the password should be persisted in the connection

**InteractiveSession**
- Gets a value that indicates whether the **DbConnectionStringBuilder** has a fixed size. 
  (Inherited from **DbConnectionStringBuilder**.)

**IsFixedSize**
- Gets a value that indicates whether the **DbConnectionStringBuilder** is read-only. 
  (Inherited from **DbConnectionStringBuilder**.)

**IsReadOnly**
- Gets a value that indicates whether logging is enabled. 
  (Overrides **DbConnectionStringBuilder..::..Item[[((String)]])**)

**Keepalive**
- Gets an ICollection that contains the keys in the **DbConnectionStringBuilder**. 
  (Inherited from **DbConnectionStringBuilder**.)

**Keys**
- Gets or sets a boolean value that indicates whether logging is enabled. 
  Gets or sets the maximum connection pool setting.

**Logging**
- Gets or sets the minimum connection pool size.

**MaximumPoolSize**
- Gets the minimum connection pool size.

**MinimumPoolSize**
- Gets the password that should be used to connect with.

**OldGuids**
- Gets or sets a boolean value that indicates if the password should be persisted in the connection
PipeName

Gets or sets the name of the named pipe that should be used for communicating with MySQL.

Pooling

Gets or sets a boolean value indicating if connection pooling is enabled.

Port

Gets or sets the port number that is used when the socket protocol is being used.

ProcedureCacheSize

Gets or sets the size of the stored procedure cache.

RespectBinaryFlags

Gets or sets the name of the server.

Server

Gets or sets the base name of the shared memory objects used to communicate with MySQL when the shared memory protocol is being used.

SharedMemoryName

Indicates whether to use SSL connections and how to handle server certificate errors.

SslMode

Indicates whether the driver should treat binary blobs as UTF8.

TreatBlobsAsUTF8

Gets or sets a boolean value that indicates whether this connection should use compression.

TreatTinyAsBoolean

Obsolete.

UseAffectedRows

Gets or sets a boolean value that indicates whether this connection uses the old style (@) parameter markers or the new (?) style.

UseCompression

Gets or sets the user id that should be used to connect with.

UseOldSyntax

Gets or sets a boolean value indicating if the Usage Advisor should be enabled.

UsePerformanceMonitor

UseProcedureBodies

UseUsageAdvisor

Gets an ICollection that contains the values in the DbConnectionStringBuilder.

(UserId)

(Inherited from DbConnectionStringBuilder.)
See Also

**MySqlConnectionStringBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlConnectionStringBuilder()()()</td>
<td></td>
</tr>
<tr>
<td>MySqlConnectionStringBuilder(String)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

[link to MySqlConnectionStringBuilder Class]
[link to MySqlConnectionStringBuilder Members]
[link to MySql.Data.MySqlClient Namespace]

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

**MySqlConnectionStringBuilder Constructor**

**Namespace:** [MySql.Data.MySqlClient](https://example.com)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlConnectionStringBuilder()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySqlConnectionStringBuilder()
See Also

MySqlConnectionStringBuilder Class
MySqlConnectionStringBuilder Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlParameter Constructor (String)

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public MySqlConnectionStringBuilder(
    string connStr
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub New ( _
    connStr As String _
)
```

### Visual C++

```cpp
public:
MySqlConnectionStringBuilder(
    String^ connStr
)
```

## Parameters

connStr
Type: System::String
See Also

MySqlConnectionStringBuilder Class
MySqlConnectionStringBuilder Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlConnectionStringBuilder` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Add**              | Adds an entry with the specified key and value into the `DbConnectionStringBuilder`.
|                      | (Inherited from `DbConnectionStringBuilder`.)                                |
| **Clear**            | Clears the collection of `PropertyDescriptor` objects on the associated `DbConnectionStringBuilder`.
<p>|                      | (Overrides <code>DbConnectionStringBuilder::Clear()</code>)                            |
| <strong>ClearPropertyDescriptors</strong> | Determines whether the <code>DbConnectionStringBuilder</code> contains a specific key.  |
|                      | (Inherited from <code>DbConnectionStringBuilder</code>.)                                |
| <strong>ContainsKey</strong>      | Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.  |
|                      | (Inherited from <code>Object</code>.)                                                  |
| <strong>Equals</strong>           | Compares the connection information in this <code>DbConnectionStringBuilder</code> object with the connection information in the supplied object. |
|                      | (Inherited from <code>DbConnectionStringBuilder</code>.)                                |
| <strong>EquivalentTo</strong>     | Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. |
|                      | (Inherited from <code>Object</code>.)                                                  |
| <strong>Finalize</strong>         | Serves as a hash function for a particular type.                           |
|                      | (Inherited from <code>Object</code>.)                                                  |
| <strong>GetConnectionString</strong> | Fills a supplied <code>Hashtable</code> with information about all the properties of this <code>DbConnectionStringBuilder</code>. |
|                      | (Inherited from <code>DbConnectionStringBuilder</code>.)                                |
| <strong>GetHashCode</strong>      | Gets the <code>Type</code> of the current instance.                                    |
| <strong>GetProperties</strong>    |                                                                             |
|                      |                                                                             |
| <strong>GetType</strong>          |                                                                             |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Remove</strong></td>
<td>DbConnectionStringBuilder..:::Remove(String). (Overrides)</td>
</tr>
<tr>
<td><strong>ShouldSerialize</strong></td>
<td>Indicates whether the specified key exists in this DbConnectionStringBuilder instance. (Inherited from DbConnectionStringBuilder.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>DbConnectionStringBuilder. (Inherited from DbConnectionStringBuilder.) (Overrides)</td>
</tr>
<tr>
<td><strong>TryGetValue</strong></td>
<td>DbConnectionStringBuilder..:::TryGetValue(String, Object%). (Overrides)</td>
</tr>
</tbody>
</table>
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

`MySqlConnectionStringBuilder::Clear` Method

*Namespace:*  `MySql.Data.MySqlClient`

*Assembly:*  `MySql.Data` (in `MySql.Data.dll`)  Version: 6.2.2.0
Syntax

C#

public override void Clear()

Visual Basic (Declaration)

Public Overrides Sub Clear

Visual C++

public:
    virtual void Clear() override

Implements

IDictionary...::Clear()()
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string GetConnectionString(
    bool includePass
)

Visual Basic (Declaration)

Public Function GetConnectionString ( _
    includePass As Boolean _
) As String

Visual C++

public:
String^ GetConnectionString(
    bool includePass
)

Parameters

includePass
    Type: System::Boolean
See Also

`MySqlConnectionStringBuilder` Class
`MySql.Data.MySqlClient` Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ConnectionStringBuilder...::Remove Method

`MySqlConnectionStringBuilder` Class

**Namespace:**  `MySQL.Data.MySqlClient`

**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public override bool Remove(
    string keyword
)

Visual Basic (Declaration)

Public Overrides Function Remove ( _
    keyword As String _
) As Boolean

Visual C++

public:
    virtual bool Remove(
        String^ keyword
    ) override

Parameters

keyword
    Type: System:::String
See Also

**MySqlConnectionStringBuilder Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

`MySqlConnectionStringBuilder...:TryGetValue Method`  
`MySqlConnectionStringBuilder Class   See Also   Send Feedback`

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
# Syntax

**C#**

```csharp
public override bool TryGetValue(  
    string keyword,  
    out Object value  
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function TryGetValue ( _
    keyword As String, _
    _
    <OutAttribute> ByRef value As Object _
) As Boolean
```

**Visual C++**

```cpp
public: bool TryGetValue(  
    String^ keyword,  
    [OutAttribute] Object^% value  
) override
```

## Parameters

**keyword**
- Type: `System::String`

**value**
- Type: `System::Object`
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlConnectionStringBuilder` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowBatch</td>
<td>Gets or sets a boolean value that indicates whether this connection will allow commands to send multiple SQL statements in one execution.</td>
</tr>
<tr>
<td>AllowUserVariables</td>
<td>Gets or sets a boolean value that indicates if zero date time values are supported.</td>
</tr>
<tr>
<td>AllowZeroDateTime</td>
<td></td>
</tr>
<tr>
<td>AutoEnlist</td>
<td>Gets or sets the pattern that matches the columns that should not be treated as UTF8</td>
</tr>
<tr>
<td>BlobAsUTF8ExcludePattern</td>
<td>Gets or sets the pattern that matches the columns that should be treated as UTF8</td>
</tr>
<tr>
<td>BlobAsUTF8IncludePattern</td>
<td>Gets or sets a value that indicates whether the ConnectionString property is visible in Visual Studio designers.</td>
</tr>
<tr>
<td>(Inherited from DbConnectionStringBuilder.)</td>
<td></td>
</tr>
<tr>
<td>CertificateFile</td>
<td>Gets or sets the character set that should be used for sending queries to the server.</td>
</tr>
<tr>
<td>CertificatePassword</td>
<td>Gets or sets the lifetime of a pooled connection.</td>
</tr>
<tr>
<td>CertificateStoreLocation</td>
<td>Gets or sets the protocol that should be used for communicating with MySQL.</td>
</tr>
<tr>
<td>CertificateThumbprint</td>
<td>Gets or sets a boolean value indicating if the connection should be reset when retrieved from the pool.</td>
</tr>
<tr>
<td>(Inherited from DbConnectionStringBuilder.)</td>
<td></td>
</tr>
<tr>
<td>connection</td>
<td>Gets or sets the connection string associated with the DbConnectionStringBuilder.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ConnectionTimeout</td>
<td>Gets or sets the connection timeout.</td>
</tr>
<tr>
<td>ConvertZeroDateTime</td>
<td>Gets or sets a boolean value indicating if zero datetime values should be converted to DateTime.MinValue.</td>
</tr>
<tr>
<td>Count</td>
<td>Gets the current number of keys that are contained within theConnectionString property. (Inherited from DbConnectionStringBuilder.)</td>
</tr>
<tr>
<td>Database</td>
<td>Gets or sets the name of the database the connection should initially connect to.</td>
</tr>
<tr>
<td>DefaultCommandTimeout</td>
<td>Gets or sets the default command timeout.</td>
</tr>
<tr>
<td>FunctionsReturnValue</td>
<td>Gets or sets a boolean value indicating if calls to Prepare() should be ignored.</td>
</tr>
<tr>
<td>IgnorePrepare</td>
<td>Gets a value that indicates whether the DbConnectionStringBuilder has a fixed size. (Inherited from DbConnectionStringBuilder.)</td>
</tr>
<tr>
<td>InteractiveSession</td>
<td>Gets a value that indicates whether the DbConnectionStringBuilder is read-only. (Inherited from DbConnectionStringBuilder.) (Overrides DbConnectionStringBuilder:::Item[[[String]]])</td>
</tr>
<tr>
<td>Item</td>
<td>Gets an ICollection that contains the keys in the DbConnectionStringBuilder. (Inherited from DbConnectionStringBuilder.)</td>
</tr>
<tr>
<td>Keepalive</td>
<td>Gets or sets a boolean value that indicates whether logging is enabled.</td>
</tr>
<tr>
<td>MaximumPoolSize</td>
<td>Gets or sets the maximum connection pool setting.</td>
</tr>
<tr>
<td>MinimumPoolSize</td>
<td>Gets the minimum connection pool size.</td>
</tr>
<tr>
<td>OldGuids</td>
<td>Gets or sets the password that should be used to connect with.</td>
</tr>
<tr>
<td>Password</td>
<td>Gets or sets a boolean value that indicates if the password should be persisted in the connection</td>
</tr>
</tbody>
</table>
PipeName  Gets or sets the name of the named pipe that should be used for communicating with MySQL.

Pooling  Gets or sets a boolean value indicating if connection pooling is enabled.

Port  Gets or sets the port number that is used when the socket protocol is being used.

ProcedureCacheSize  Gets or sets the size of the stored procedure cache.

RespectBinaryFlags

Server  Gets or sets the name of the server.

SharedMemoryName  Gets or sets the base name of the shared memory objects used to communicate with MySQL when the shared memory protocol is being used.

SslMode  Indicates whether to use SSL connections and how to handle server certificate errors.

TreatBlobsAsUTF8  Indicates whether the driver should treat binary blobs as UTF8

TreatTinyAsBoolean

UseAffectedRows  Gets or sets a boolean value that indicates whether this connection should use compression.

UseCompression

UseOldSyntax  Gets or sets a boolean value that indicates whether this connection uses the old style (@) parameter markers or the new (?) style.

UsePerformanceMonitor  Gets or sets a boolean value indicating if the permon hooks should be enabled.

UseProcedureBodies

UserID  Gets or sets the user id that should be used to connect with.

UseUsageAdvisor  Gets a ICollection that contains the values in the DbConnectionStringBuilder.

Values  (Inherited from DbConnectionStringBuilder.)
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value that indicates whether this connection will allow commands to send multiple SQL statements in one execution.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
C#

```csharp
public bool AllowBatch { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property AllowBatch As Boolean
```

Visual C++

```cpp
public:
property bool AllowBatch {
    bool get();
    void set (bool value);
}
```
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ConnectionStringBuilder

AllowUserVariables Property

MySqlConnectionStringBuilder Class

See Also

Send Feedback

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool AllowUserVariables { get; set; }

Visual Basic (Declaration)

Public Property AllowUserVariables As Boolean

Visual C++

public:
    property bool AllowUserVariables {
        bool get ();
        void set (bool value);
    }
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value that indicates if zero date time values are supported.

**Namespace:**  [MySQL.Data.MySqlClient](https://mySql.Data.MySqlClient)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Synta

c#

public bool AllowZeroDateTime { get; set; }

Visual Basic (Declaration)

Public Property AllowZeroDateTime As Boolean

Visual C++

public:
property bool AllowZeroDateTime {
    bool get ();
    void set (bool value);
}
See Also

**MySqlConnectionStringBuilder Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
AutoEnlist Property

`MySqlConnectionStringBuilder` Class

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysql.com/)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool AutoEnlist { get; set; }

Visual Basic (Declaration)

Public Property AutoEnlist As Boolean

Visual C++

public:
property bool AutoEnlist {
    bool get ();
    void set (bool value);
}
See Also

**MySqlConnectionStringBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the pattern that matches the columns that should not be treated as UTF8

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public string BlobAsUTF8ExcludePattern { get; set; }

Visual Basic (Declaration)

Public Property BlobAsUTF8ExcludePattern As String

Visual C++

public:
property String^ BlobAsUTF8ExcludePattern {
    String^ get ();
    void set (String^ value);
}

See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the pattern that matches the columns that should be treated as UTF8

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
### Syntax

#### C#

```csharp
public string BlobAsUTF8IncludePattern { get; set; }
```

#### Visual Basic (Declaration)

```vbnet
Public Property BlobAsUTF8IncludePattern As String
```

#### Visual C++

```cpp
public:
    property String^ BlobAsUTF8IncludePattern {
        String^ get ();
        void set (String^ value);
    }
```
See Also

MySQLConnectionStringBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ConnectionStringBuilder...::CertificateFile Property

**MySqlConnectionStringBuilder Class**  See Also  Send Feedback

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string CertificateFile { get; set; }

Visual Basic (Declaration)

Public Property CertificateFile As String

Visual C++

public:
property String^ CertificateFile {
    String^ get ();
    void set (String^ value);
}

See Also

**MySqlConnectionStringBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnectionStringBuilder.CertificatePassword Property

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string CertificatePassword { get; set; }

Visual Basic (Declaration)

Public Property CertificatePassword As String

Visual C++

public:
property String^ CertificatePassword {
    String^ get ();
    void set (String^ value);
}
See Also

MySQLConnectionStringBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnectionStringBuilder::CertificateStoreLocation Property

MySqlConnectionStringBuilder Class  See Also  Send Feedback

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCertificateStoreLocation CertificateStoreLocation { get; }

Visual Basic (Declaration)

Public Property CertificateStoreLocation As MySqlCertificateStoreLocation

Visual C++

public:

property MySqlCertificateStoreLocation CertificateStoreLocation { get ();
void set (MySqlCertificateStoreLocation value); }

}
See Also

**SqlConnectionStringBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnectionStringBuilder:::CertificateThumbprint Property

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/doc/en/)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string CertificateThumbprint { get; set; }

Visual Basic (Declaration)

Public Property CertificateThumbprint As String

Visual C++

public:
property String^ CertificateThumbprint {
    String^ get ();
    void set (String^ value);
}
See Also

SqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the character set that should be used for sending queries to the server.

**Namespace:**  [MySQL.Data.MySqlClient](https://mySql.Data.MySqlClient)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public string CharSet { get; set; }

Visual Basic (Declaration)

Public Property CharSet As String

Visual C++

public:
property String^ CharSet {
    String^ get ();
    void set (String^ value);
}
See Also

**MySqlConnectionStringBuilder Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ConnectionStringBuilder::ConnectionLifeTime Property

Gets or sets the lifetime of a pooled connection.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public uint ConnectionLifeTime { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property ConnectionLifeTime As UInteger
```

**Visual C++**

```cpp
public:
    property unsigned int ConnectionLifeTime {
        unsigned int get ();
        void set (unsigned int value);
    }
```
See Also

**MySqlCommandBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlConnectionStringBuilder...::ConnectionProtocol Property
MySqlConnectionStringBuilder Class  See Also  Send Feedback

Gets or sets the protocol that should be used for communicating with MySQL.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlConnectionProtocol ConnectionProtocol { get; set; }

Visual Basic (Declaration)

Public Property ConnectionProtocol As MySqlConnectionProtocol

Visual C++

public:
property MySqlConnectionProtocol ConnectionProtocol {
    MySqlConnectionProtocol get ();
    void set (MySqlConnectionProtocol value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value indicating if the connection should be reset when retrieved from the pool.

**Namespace:**  [MySQL.Data.MySqlClient](https://example.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public bool ConnectionReset { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property ConnectionReset As Boolean
```

Visual C++

```c++
public:
    property bool ConnectionReset {
        bool get();
        void set(bool value);
    }
```
See Also

MySqlConnectionStringBuilder Class
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the connection timeout.

**Namespace:**  [MySQL.Data.MySqlClient](https://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public uint ConnectionTimeout { get; set; }

Visual Basic (Declaration)

Public Property ConnectionTimeout As UInteger

Visual C++

public:
property unsigned int ConnectionTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value indicating if zero datetime values should be converted to DateTime.MinValue.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public bool ConvertZeroDateTime { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property ConvertZeroDateTime As Boolean
```

Visual C++

```cpp
public:
property bool ConvertZeroDateTime {
    bool get ();
    void set (bool value);
}
```
See Also

MySQLConnectionStringBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnectionStringBuilder...Database Property

**MySqlConnectionStringBuilder Class**  [See Also]  [Send Feedback]

Gets or sets the name of the database the connection should initially connect to.

**Namespace:**  [MySQL.Data.MySqlClient]

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public string Database { get; set; }

Visual Basic (Declaration)

Public Property Database As String

Visual C++

public:
property String^ Database {
    String^ get ();
    void set (String^ value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the default command timeout.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public uint DefaultCommandTimeout { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property DefaultCommandTimeout AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int DefaultCommandTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
```
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ConnectionStringBuilder:: FunctionsReturnString Property

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySql.Data](#) (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool FunctionsReturnString { get; set; }

Visual Basic (Declaration)

Public Property FunctionsReturnString As Boolean

Visual C++

public:
property bool FunctionsReturnString {
    bool get ();
    void set (bool value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value indicating if calls to Prepare() should be ignored.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool IgnorePrepare { get; set; }

Visual Basic (Declaration)

Public Property IgnorePrepare As Boolean

Visual C++

public:
property bool IgnorePrepare {
    bool get ();
    void set (bool value);
}
See Also

MySQLConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ConnectionStringBuilder...::InteractiveSession Property

**Namespace:**  [MySql.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll)  Version: 6.2.2.0
Syntax

C#

public bool InteractiveSession { get; set; }

Visual Basic (Declaration)

Public Property InteractiveSession As Boolean

Visual C++

public:
property bool InteractiveSession {
    bool get ();
    void set (bool value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ConnectionStringBuilder:::Item Property

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  [MySQL.Data](https://www.mysql.com) (in MySQL.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override Object this[
    string keyword
] { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Default Property Item ( _
    keyword As String _
) As Object
```

### Visual C++

```cpp
public:
    virtual property Object^ default[String^ keyword] { 
        Object^ get (String^ keyword) override; 
        void set (String^ keyword, Object^ value) override; 
    }
```

### Parameters

**keyword**  
Type: System::String
See Also

**MySqlConnectionStringBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlConnectionStringBuilder...::Keepalive Property

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public uint Keepalive { get; set; }

Visual Basic (Declaration)

Public Property Keepalive AsUInteger

Visual C++

public:
property unsigned int Keepalive {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlConnectionStringBuilder...:::Logging Property

namespace: MySql.Data.MySqlClient

assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool Logging { get; set; }

Visual Basic (Declaration)

Public Property Logging As Boolean

Visual C++

public:
property bool Logging {
    bool get ();
    void set (bool value);
}
See Also

MYSQLConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the maximum connection pool setting.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

**C#**

public uint MaximumPoolSize { get; set; }

**Visual Basic (Declaration)**

Public Property MaximumPoolSize AsUInteger

**Visual C++**

public:
property unsigned int MaximumPoolSize {
unsigned int get ();
void set (unsigned int value);
}


See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the minimum connection pool size.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public uint MinimumPoolSize { get; set; }

Visual Basic (Declaration)

Public Property MinimumPoolSize AsUInteger

Visual C++

public:
property unsigned int MinimumPoolSize {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
OldGuids Property

MySQL ConnectionStringBuilder Class

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool OldGuids { get; set; }

Visual Basic (Declaration)

Public Property OldGuids As Boolean

Visual C++

public:
property bool OldGuids {
    bool get ();
    void set (bool value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlConnectionStringBuilder::Password Property

Gets or sets the password that should be used to connect with.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string Password { get; set; }

Visual Basic (Declaration)

Public Property Password As String

Visual C++

public:
property String^ Password {
    String^ get ();
    void set (String^ value);
}
See Also

**MySqlConnectionStringBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value that indicates if the password should be persisted in the connection string.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool PersistSecurityInfo { get; set; }

Visual Basic (Declaration)

Public Property PersistSecurityInfo As Boolean

Visual C++

public:
property bool PersistSecurityInfo {
    bool get ();
    void set (bool value);
}
See Also

MySQLConnectionStringBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the name of the named pipe that should be used for communicating with MySQL.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public string PipeName { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property PipeName As String
```

**Visual C++**

```cpp
public:
property String^ PipeName {
    String^ get ();
    void set (String^ value);
}
```
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value indicating if connection pooling is enabled.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySql.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

public bool Pooling { get; set; }

Visual Basic (Declaration)

Public Property Pooling As Boolean

Visual C++

public:
property bool Pooling {
    bool get ();
    void set (bool value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlConnectionStringBuilder Port Property

Gets or sets the port number that is used when the socket protocol is being used.

**Namespace:** MySql.Data.MySqlClient  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public uint Port { get; set; }

Visual Basic (Declaration)

Public Property Port AsUInteger

Visual C++

public:
property unsigned int Port {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

**MySqlConnectionStringBuilder Class**

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the size of the stored procedure cache.

**Namespace:**  [MySql.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public uint ProcedureCacheSize { get; set; }

Visual Basic (Declaration)

Public Property ProcedureCacheSize As UInteger

Visual C++

public:
property unsigned int ProcedureCacheSize {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL ConnectionStringBuilder..::RespectBinaryFlags Property

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool RespectBinaryFlags { get; set; }  

Visual Basic (Declaration)

Public Property RespectBinaryFlags As Boolean

Visual C++

public:
property bool RespectBinaryFlags {
    bool get (){
    void set (bool value);
}
See Also

MySQLConnectionStringBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the name of the server.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

public `string` Server { get; set; }

**Visual Basic (Declaration)**

Public Property Server As `String`

**Visual C++**

public:
property `String^` Server {
`String^` get ();
void set (`String^` value);
}

**Field Value**

The server.
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the base name of the shared memory objects used to communicate with MySQL when the shared memory protocol is being used.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public string SharedMemoryName { get; set; }

Visual Basic (Declaration)

Public Property SharedMemoryName As String

Visual C++

public:
property String^ SharedMemoryName {
    String^ get ();
    void set (String^ value);
}
See Also

**MySqlConnectionStringBuilder Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Indicates whether to use SSL connections and how to handle server certificate errors.

**Namespace:**  [MySql.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlSslMode SslMode { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property SslMode As MySqlSslMode
```

Visual C++

```cpp
public:
property MySqlSslMode SslMode {
    MySqlSslMode get ();
    void set (MySqlSslMode value);
}
```
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnectionStringBuilder.TreatBlobsAsUTF8 Property

Indicates whether the driver should treat binary blobs as UTF8

**Namespace:** MySql.Data.MySqlClient

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool TreatBlobsAsUTF8 { get; set; }

Visual Basic (Declaration)

Public Property TreatBlobsAsUTF8 As Boolean

Visual C++

public:
property bool TreatBlobsAsUTF8 {
    bool get ();
    void set (bool value);
}
See Also

**MySQLConnectionStringBuilder Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnectionStringBuilder:::TreatTinyAsBoolean Property

MySqlConnectionStringBuilder Class  See Also  Send Feedback

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool TreatTinyAsBoolean { get; set; }

Visual Basic (Declaration)

Public Property TreatTinyAsBoolean As Boolean

Visual C++

public:
property bool TreatTinyAsBoolean {
    bool get ();
    void set (bool value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool UseAffectedRows { get; set; }

Visual Basic (Declaration)

Public Property UseAffectedRows As Boolean

Visual C++

public:
property bool UseAffectedRows {
    bool get ();
    void set (bool value);
}
See Also

MySQLConnectionStringBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value that indicates whether this connection should use compression.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool UseCompression { get; set; }

Visual Basic (Declaration)

Public Property UseCompression As Boolean

Visual C++

public:
property bool UseCompression {
    bool get ();
    void set (bool value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value that indicates whether this connection uses the old style (@) parameter markers or the new (?) style.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

[ObsoleteAttribute("Use Old Syntax is no longer needed. See documentation")]
public bool UseOldSyntax { get; set; }

Visual Basic (Declaration)

<ObsoleteAttribute("Use Old Syntax is no longer needed. See documentation")>
Public Property UseOldSyntax As Boolean

Visual C++

[ObsoleteAttribute(L"Use Old Syntax is no longer needed. See documentation")]
public:
property bool UseOldSyntax {
    bool get ();
    void set (bool value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value indicating if the permon hooks should be enabled.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public bool UsePerformanceMonitor { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property UsePerformanceMonitor As Boolean
```

Visual C++

```cpp
public:
property bool UsePerformanceMonitor {
    bool get ();
    void set (bool value);
}
```
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnectionStringBuilder::UseProcedureBodies Property

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public bool UseProcedureBodies { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property UseProcedureBodies As Boolean
```

**Visual C++**

```cpp
public:
property bool UseProcedureBodies {
    bool get ();
    void set (bool value);
}
```
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the user id that should be used to connect with.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public string UserID { get; set; }

Visual Basic (Declaration)

Public Property UserID As String

Visual C++

public:
property String^ UserID {
    String^ get ();
    void set (String^ value);
}
See Also

MySqlConnectionStringBuilder Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a boolean value indicating if the Usage Advisor should be enabled.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQL.Data](#) (in `MySQL.Data.dll`) Version: 6.2.2.0
Syntax

C#

public bool UseUsageAdvisor { get; set; }

Visual Basic (Declaration)

Public Property UseUsageAdvisor As Boolean

Visual C++

public:
property bool UseUsageAdvisor {
    bool get ();
    void set (bool value);
    }

See Also

MySQLConnectionStringBuilder Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents a set of data commands and a database connection that are used to fill a dataset and update a MySQL database. This class cannot be inherited.

**Namespace:** [MySQL.Data.MySqlClient](#)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public sealed class MySqlDataAdapter : DbDataAdapter, IDbDataAdapter, IDataAdapter, ICloneable

Visual Basic (Declaration)

Public NotInheritable Class MySqlDataAdapter _
    Inherits DbDataAdapter _
    Implements IDbDataAdapter, IDataAdapter, ICloneable

Visual C++

public ref class MySqlDataAdapter sealed : public DbDataAdapter, IDbDataAdapter, IDataAdapter, ICloneable
Remarks

The MySQLDataAdapter, serves as a bridge between a DataSet and MySQL for retrieving and saving data. The MySQLDataAdapter provides this bridge by mapping Fill(DataSet), which changes the data in the DataSet to match the data in the data source, and Update(DataSet), which changes the data in the data source to match the data in the DataSet, using the appropriate SQL statements against the data source.

When the MySQLDataAdapter fills a DataSet, it will create the necessary tables and columns for the returned data if they do not already exist. However, primary key information will not be included in the implicitly created schema unless the MissingSchemaAction property is set to AddWithKey. You may also have the MySQLDataAdapter create the schema of the DataSet, including primary key information, before filling it with data using FillSchema(DataTable, SchemaType).

MySQLDataAdapter is used in conjunction with MySqlConnection and MySqlCommand to increase performance when connecting to a MySQL database.

The MySQLDataAdapter also includes the SelectCommand, InsertCommand, DeleteCommand, UpdateCommand, and TableMappings properties to facilitate the loading and updating of data.

When an instance of MySQLDataAdapter is created, the read/write properties are set to initial values. For a list of these values, see the MySQLDataAdapter constructor.

⚠️ Note: Please be aware that the DataColumn class allows only Int16, Int32, and Int64 to have the AutoIncrement property set. If you plan to use autoincrement columns with MySQL, you should consider using signed integer columns.
Examples

The following example creates a **MySQLCommand** and a **MySqlConnection**. The MySqlConnection is opened and set as the **Connection** for the MySqlCommand. The example then calls **ExecuteNonQuery()**, and closes the connection. To accomplish this, the ExecuteNonQuery is passed a connection string and a query string that is a SQL INSERT statement.

**VB.NET**

```
Public Function SelectRows(dataSet As DataSet, connection As String, query As String) As DataSet
    Dim conn As New MySqlConnection(connection)
    Dim adapter As New MySqlDataAdapter()
    adapter.SelectCommand = New MySqlCommand(query, conn)
    adapter.Fill(dataset)
    Return dataset
End Function
```

**C#**

```
public DataSet SelectRows(DataSet dataset, string connection, string q)
{
    MySqlConnection conn = new MySqlConnection(connection);
    MySqlCommand cmd = new MySqlCommand(q, conn);
    adapter.SelectCommand = cmd;
    adapter.Fill(dataset);
    return dataset;
}
```
Inheritance Hierarchy

System..::.Object
System..::.MarshalByRefObject
System.ComponentModel..::.Component
System.Data.Common..::.DataAdapter
System.Data.Common..::.DbDataAdapter
MySql.Data.MySqlClient..::.MySqlDataAdapter
See Also

MySQLDataAdapter Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlDataAdapter` type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlDataAdapter</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddToBatch</strong></td>
<td>Adds a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.idbcommand">IDbCommand</a> to the current batch. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.databindadapter">DbDataAdapter</a>.)</td>
</tr>
<tr>
<td><strong>ClearBatch</strong></td>
<td>Removes all <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.idbcommand">IDbCommand</a> objects from the batch. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.databindadapter">DbDataAdapter</a>.) <strong>Obsolete.</strong></td>
</tr>
<tr>
<td><strong>CloneInternals</strong></td>
<td>Creates a copy of this instance of <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.datatableadapter">DataAdapter</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.databindadapter">DataAdapter</a>.) Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.runtime.interopservices.marshalbyrefobject">MarshalByRefObject</a>.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Initializes a new instance of the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.rowupdatedeventargs">RowUpdatedEventArgs</a> class. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.databindadapter">DbDataAdapter</a>.)</td>
</tr>
<tr>
<td><strong>CreateRowUpdatedEvent</strong></td>
<td>Initializes a new instance of the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.rowupdatingeventargs">RowUpdatingEventArgs</a> class. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.databindadapter">DbDataAdapter</a>.)</td>
</tr>
<tr>
<td><strong>CreateRowUpdatingEvent</strong></td>
<td>Creates a new <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.datatablemappingcollection">DataTableMappingCollection</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.databindadapter">DbDataAdapter</a>.)</td>
</tr>
<tr>
<td><strong>CreateTableMappings</strong></td>
<td></td>
</tr>
</tbody>
</table>
Dispose

Determines whether the specified Object is equal to the current Object.
(Inherited from Object.)

Equals

Executes the current batch.
(Inherited from DbDataAdapter.)

ExecuteBatch

Overloaded.

Fill

Overloaded.

FillSchema

Releases unmanaged resources and performs other cleanup operations before the Component is reclaimed by garbage collection.
(Inherited from Component.)

Finalize

Returns a IDataParameter from one of the commands in the current batch.
(Inherited from DbDataAdapter.)

GetBatchedParameter

Returns information about an individual update attempt within a larger batched update.
(Inherited from DbDataAdapter.)

GetBatchedRecordsAffected

Gets the parameters set by the user when executing an SQL SELECT statement.
(Inherited from DbDataAdapter.)

GetFillParameters

Serves as a hash function for a particular type.
(Inherited from Object.)

GetHashCode

Retrieves the current lifetime
- **GetLifetimeService**
  service object that controls the lifetime policy for this instance.
  (Inherited from MarshalByRefObject.)

- **GetService**
  Returns an object that represents a service provided by the Component or by its Container.
  (Inherited from Component.)

- **GetType**
  Gets the Type of the current instance.
  (Inherited from Object.)

- **HasTableMappings**
  Indicates whether a DataTableMappingCollection has been created.
  (Inherited from DataAdapter.)

- **InitializeBatching**
  Initializes batching for the DbDataAdapter.
  (Inherited from DbDataAdapter.)

- **InitializeLifetimeService**
  Obtains a lifetime service object to control the lifetime policy for this instance.
  (Inherited from MarshalByRefObject.)

- **MemberwiseClone**
  Overloaded.

- **OnFillError**
  Invoked when an error occurs during a Fill.
  (Inherited from DataAdapter.)

- **OnRowUpdated**
  Raises the RowUpdated event of a .NET Framework data provider.
  (Inherited from DbDataAdapter.)

- **OnRowUpdating**
  Raises the RowUpdating event of a .NET Framework
OnRow Updating

Determines whether the AcceptChangesDuringFill property should be persisted.

(Inherited from DataAdapter.)

Reset Fill Load Option

Determines whether the FillLoadOption property should be persisted.

(Inherited from DataAdapter.)

ShouldSerialize Accept Changes During Fill

Determines whether the FillLoadOption property should be persisted.

(Inherited from DataAdapter.)

ShouldSerialize Fill Load Option

Determines whether one or more DataTableMapping objects exist and they should be persisted.

(Inherited from DataAdapter.)

ShouldSerialize Table Mappings

Ends batching for the DbDataAdapter.

(Inherited from DbDataAdapter.)

Terminate Batching

Returns a String containing the name of the Component, if any. This method should not be overridden.

(Inherited from Component.)

ToString

Update

Overloaded.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcceptChangesDuringFill</td>
<td>Gets or sets a value indicating whether <code>AcceptChanges()</code> is called on a <code>DataRow</code> after it is added to the <code>DataTable</code> during any of the Fill operations. (Inherited from <code>DataAdapter</code>.)</td>
</tr>
<tr>
<td>AcceptChangesDuringUpdate ()</td>
<td>Gets or sets whether <code>AcceptChanges()</code> is called during a <code>Update(DataSet)</code>. (Inherited from <code>DataAdapter</code>.)</td>
</tr>
<tr>
<td>CanRaiseEvents</td>
<td>Gets a value indicating whether the component can raise an event. (Inherited from <code>Component</code>.)</td>
</tr>
<tr>
<td>Container</td>
<td>Gets the <code>IContainer</code> that contains the <code>Component</code>. (Inherited from <code>Component</code>.)</td>
</tr>
<tr>
<td>ContinueUpdateOnError</td>
<td>Gets or sets a value that specifies whether to generate an exception when an error is encountered during a row update. (Inherited from <code>DataAdapter</code>.)</td>
</tr>
<tr>
<td>DeleteCommand</td>
<td>Gets or sets a SQL statement or stored procedure used to delete records from the data set.</td>
</tr>
<tr>
<td>DesignMode</td>
<td>Gets a value that indicates whether the <code>Component</code> is currently in design mode. (Inherited from <code>Component</code>.)</td>
</tr>
<tr>
<td>Events</td>
<td>Gets the list of event handlers that are attached to this <code>Component</code>. (Inherited from <code>Component</code>.)</td>
</tr>
<tr>
<td>FillCommandBehavior</td>
<td>Gets or sets the behavior of the command used to fill the data adapter. (Inherited from <code>DbDataAdapter</code>.)</td>
</tr>
</tbody>
</table>
FillLoadOption

Gets or sets the `LoadOption` that determines how the adapter fills the `DataTable` from the `DbDataReader`. (Inherited from `DataAdapter`.)

InsertCommand

Gets or sets a SQL statement or stored procedure used to insert records into the data set.

MissingMappingAction

Determines the action to take when incoming data does not have a matching table or column. (Inherited from `DataAdapter`.)

MissingSchemaAction

Determines the action to take when existing `DataSet` schema does not match incoming data. (Inherited from `DataAdapter`.)

ReturnProviderSpecificTypes

Gets or sets whether the Fill method should return provider-specific values or common CLS-compliant values. (Inherited from `DataAdapter`.)

SelectCommand

Gets or sets a SQL statement or stored procedure used to select records in the data source.

Site

Gets or sets the `ISite` of the `Component`. (Inherited from `Component`.)

TableMappings

Gets a collection that provides the master mapping between a source table and a `DataTable`. (Inherited from `DataAdapter`.)

UpdateBatchSize

(Overrides `DbDataAdapter...:UpdateBatchSize`.)

UpdateCommand

Gets or sets a SQL statement or stored procedure used to updated records in the data source.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposed</td>
<td>Occurs when the component is disposed by a call to the <code>Dispose()</code> method.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Component.)</td>
</tr>
<tr>
<td>FillError</td>
<td>Returned when an error occurs during a fill operation.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DataAdapter.)</td>
</tr>
<tr>
<td>RowUpdated</td>
<td>Occurs during Update after a command is executed against the data source.</td>
</tr>
<tr>
<td></td>
<td>The attempt to update is made, so the event fires.</td>
</tr>
<tr>
<td>RowUpdating</td>
<td>Occurs during Update before a command is executed against the data source.</td>
</tr>
<tr>
<td></td>
<td>The attempt to update is made, so the event fires.</td>
</tr>
</tbody>
</table>
See Also

* [MySqlDataAdapter Class](#)
* [MySql.Data.MySqlClient Namespace](#)

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net
MySqlDataAdapter Constructor

MySqlDataAdapter Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlDataAdapter()()()</td>
<td>Initializes a new instance of the MySqlDataAdapter class.</td>
</tr>
<tr>
<td>MySqlDataAdapter(MySqlCommand)</td>
<td>Initializes a new instance of the MySqlDataAdapter class with the specified MySqlCommand as the SelectCommand property.</td>
</tr>
<tr>
<td>MySqlDataAdapter(String, MySqlConnection)</td>
<td>Initializes a new instance of the MySqlDataAdapter class with a SelectCommand and a MySqlConnection object.</td>
</tr>
<tr>
<td>MySqlDataAdapter(String, String)</td>
<td>Initializes a new instance of the MySqlDataAdapter class with a SelectCommand and a connection string.</td>
</tr>
</tbody>
</table>
See Also

**MySqlDataAdapter Class**
**MySqlDataAdapter Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the MySqlDataAdapter class.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlDataAdapter()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySqlDataAdapter()
Remarks

When an instance of `MySqlDataAdapter` is created, the following read/write properties are set to the following initial values.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MissingMappingAction</code></td>
<td><code>MissingMappingAction.Passthrough</code></td>
</tr>
<tr>
<td><code>MissingSchemaAction</code></td>
<td><code>MissingSchemaAction.Add</code></td>
</tr>
</tbody>
</table>

You can change the value of any of these properties through a separate call to the property.
Examples

The following example creates a MySqlDataAdapter and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateSqlDataAdapter()
Dim conn As MySqlConnection = New MySqlConnection("Data Source=localhost;database=test")
Dim da As MySqlDataAdapter = New MySqlDataAdapter

da.SelectCommand = New MySqlCommand("SELECT id, name FROM mytable", conn)
da.InsertCommand = New MySqlCommand("INSERT INTO mytable (id, name) VALUES (@id, @name)", conn)
da.UpdateCommand = New MySqlCommand("UPDATE mytable SET id=@id, name @name WHERE id=@oldId", conn)
da.DeleteCommand = New MySqlCommand("DELETE FROM mytable WHERE id=@id", conn)
da.InsertCommand.Parameters.Add("@id", SqlDbType.VarChar, 5, "id")
da.InsertCommand.Parameters.Add("@name", SqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@id", SqlDbType.VarChar, 5, "id")
da.UpdateCommand.Parameters.Add("@name", SqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@oldId", SqlDbType.VarChar, 5, "id")
da.DeleteCommand.Parameters.Add("@id", SqlDbType.VarChar, 5, "id")
End Sub
```

**C#**

```csharp
public static void CreateSqlDataAdapter()
{
    MySqlConnection conn = new MySqlConnection("Data Source=localhost;database=test")
    MySqlDataAdapter da = new MySqlDataAdapter();

da.SelectCommand = new MySqlCommand("SELECT id, name FROM mytable", conn);
da.InsertCommand = new MySqlCommand("INSERT INTO mytable (id, name) VALUES (@id, @name)", conn);
da.UpdateCommand = new MySqlCommand("UPDATE mytable SET id=@id, name @name WHERE id=@oldId", conn);
da.DeleteCommand = new MySqlCommand("DELETE FROM mytable WHERE id=@id", conn);
```
da.InsertCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
da.InsertCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "n")

da.UpdateCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
da.UpdateCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "n")
da.UpdateCommand.Parameters.Add("@oldId", MySqlDbType.VarChar, 5, "i")

da.DeleteCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
See Also

**MySqlCommand Class**
**MySqlCommand Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Data Adapter Constructor (MySqlCommand)

**New instance of the** MySqlDataAdapter **class with the specified** MySqlCommand **as the** SelectCommand **property.**

**Namespace:** MySql.Data.MySqlClient
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlDataAdapter(
    MySqlCommand selectCommand
)
```

Visual Basic (Declaration)

```vbnet
Public Sub New ( _
    selectCommand As MySqlCommand _
)
```

Visual C++

```cpp
public:
MySqlDataAdapter(
    MySqlCommand^ selectCommand
)
```

Parameters

selectCommand

Type: `MySql.Data.MySqlClient::MySqlCommand` that is a SQL SELECT statement or stored procedure and is set as the `SelectCommand` property of the `MySqlDataAdapter`. 
**Remarks**

When an instance of `MySqlDataAdapter` is created, the following read/write properties are set to the following initial values.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MissingMappingAction</code></td>
<td><code>MissingMappingAction.Passthrough</code></td>
</tr>
<tr>
<td><code>MissingSchemaAction</code></td>
<td><code>MissingSchemaAction.Add</code></td>
</tr>
</tbody>
</table>

You can change the value of any of these properties through a separate call to the property.

When `SelectCommand` (or any of the other command properties) is assigned to a previously created `MySqlCommand`, the `MySqlCommand` is not cloned. The `SelectCommand` maintains a reference to the previously created `MySqlCommand` object.
Examples

The following example creates a `MySqlDataAdapter` and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateSqlDataAdapter()
    Dim conn As MySqlConnection = New MySqlConnection("Data Source=localhost;
    "database=test")
    Dim cmd As New MySqlCommand("SELECT id, name FROM mytable", conn)
    Dim da As MySqlDataAdapter = New MySqlDataAdapter(cmd)
    da.InsertCommand = New MySqlCommand("INSERT INTO mytable (id, name) "VALUES (@id, @name)", conn)
    da.UpdateCommand = New MySqlCommand("UPDATE mytable SET id=@id, name "WHERE id=@oldId", conn)
    da.DeleteCommand = New MySqlCommand("DELETE FROM mytable WHERE id=@i"
    da.InsertCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
    da.InsertCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
    da.UpdateCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
    da.UpdateCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
    da.UpdateCommand.Parameters.Add("@oldId", MySqlDbType.VarChar, 5, "id")
    da.DeleteCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
End Sub
```

**C#**

```csharp
public static void CreateSqlDataAdapter()
{
    MySqlConnection conn = new MySqlConnection("Data Source=localhost;da
    MySqlCommand cmd = new MySqlCommand("SELECT id, name FROM mytable","n
    MySqlDataAdapter da = new MySqlDataAdapter(cmd);
    da.InsertCommand = new MySqlCommand("INSERT INTO mytable (id, name) "VALUES (@id, @name)", conn);
    da.UpdateCommand = new MySqlCommand("UPDATE mytable SET id=@id, name "WHERE id=@oldId", conn);
    da.DeleteCommand = new MySqlCommand("DELETE FROM mytable WHERE id=@i"
da.InsertCommand.Parameters.Add("@id", SqlDbType.VarChar, 5, "id")
da.InsertCommand.Parameters.Add("@name", SqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@id", SqlDbType.VarChar, 5, "id")
da.UpdateCommand.Parameters.Add("@name", SqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@oldId", SqlDbType.VarChar, 5, "id")
da.DeleteCommand.Parameters.Add("@id", SqlDbType.VarChar, 5, "id")
See Also

MySQLDataAdapter Class
MySQLDataAdapter Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlDataAdapter Constructor (String, MySqlConnection)

Initializes a new instance of the MySqlDataAdapter class with a SelectCommand and a MySqlConnection object.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlDataAdapter(
    string selectCommandText,
    MySqlConnection connection
)

Visual Basic (Declaration)

Public Sub New (_
    selectCommandText As String, _
    connection As MySqlConnection _
)

Visual C++

public:
MySqlDataAdapter(
    String^ selectCommandText,
    MySqlConnection^ connection
)

Parameters

selectCommandText
Type: System::::String
A String that is a SQL SELECT statement or stored procedure to be used by the SelectCommand property of the MySqlDataAdapter.

collection
Type: MySql.Data.MySqlClient:::MySqlConnection
A MySqlConnection that represents the connection.
Remarks

This implementation of the `MySqlDataAdapter` opens and closes a `MySqlConnection` if it is not already open. This can be useful in an application that must call the `Fill(DataSet)` method for two or more `MySqlDataAdapter` objects. If the `MySqlConnection` is already open, you must explicitly call `Close()` or `Dispose(Boolean)` to close it.

When an instance of `MySqlDataAdapter` is created, the following read/write properties are set to the following initial values.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MissingMappingAction</code></td>
<td><code>MissingMappingAction.Passthrough</code></td>
</tr>
<tr>
<td><code>MissingSchemaAction</code></td>
<td><code>MissingSchemaAction.Add</code></td>
</tr>
</tbody>
</table>

You can change the value of any of these properties through a separate call to the property.
Examples

The following example creates a `MySqlDataAdapter` and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateSqlDataAdapter()
    Dim conn As MySqlConnection = New MySqlConnection("Data Source=localhost;database=test")
    Dim da As MySqlDataAdapter = New MySqlDataAdapter("SELECT id, name FROM mytable")

    da.InsertCommand = New MySqlCommand("INSERT INTO mytable (id, name) VALUES (@id, @name)", conn)
    da.UpdateCommand = New MySqlCommand("UPDATE mytable SET id=@id, name WHERE id=@oldId", conn)
    da.DeleteCommand = New MySqlCommand("DELETE FROM mytable WHERE id=@id", conn)

    da.InsertCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
    da.InsertCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
    da.UpdateCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
    da.UpdateCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
    da.UpdateCommand.Parameters.Add("@oldId", MySqlDbType.VarChar, 5, "id")
    da.DeleteCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
End Sub
```

**C#**

```csharp
public static void CreateSqlDataAdapter()
{
    MySqlConnection conn = new MySqlConnection("Data Source=localhost;database=test")
    MySqlDataAdapter da = new MySqlDataAdapter("SELECT id, name FROM mytable")

    da.InsertCommand = new MySqlCommand("INSERT INTO mytable (id, name) VALUES (@id, @name)", conn);
    da.UpdateCommand = new MySqlCommand("UPDATE mytable SET id=@id, name WHERE id=@oldId", conn);
    da.DeleteCommand = new MySqlCommand("DELETE FROM mytable WHERE id=@id", conn);

    da.InsertCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
    da.InsertCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
    da.UpdateCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
    da.UpdateCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
    da.UpdateCommand.Parameters.Add("@oldId", MySqlDbType.VarChar, 5, "id")
}
```
da.InsertCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
da.UpdateCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@oldId", MySqlDbType.VarChar, 5, "id")
da.DeleteCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
}
See Also

**MySqlDataAdapter Class**
**MySqlDataAdapter Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlDataAdapter` class with a `SelectCommand` and a connection string.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlDataAdapter(
    string selectCommandText, 
    string selectConnString
)
```

Visual Basic (Declaration)

```vbnet
Public Sub New ( _
    selectCommandText As String, _
    selectConnString As String _
)
```

Visual C++

```cpp
public:
MySqlDataAdapter(
    String^ selectCommandText, 
    String^ selectConnString
)
```

Parameters

**selectCommandText**
Type: `System::String`
A `String` that is a SQL SELECT statement or stored procedure to be used by the `SelectCommand` property of the `MySqlDataAdapter`.

**selectConnString**
Type: `System::String`
The connection string
Remarks

When an instance of `MySqlDataAdapter` is created, the following read/write properties are set to the following initial values.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MissingMappingAction</code></td>
<td><code>MissingMappingAction.Passthrough</code></td>
</tr>
<tr>
<td><code>MissingSchemaAction</code></td>
<td><code>MissingSchemaAction.Add</code></td>
</tr>
</tbody>
</table>

You can change the value of any of these properties through a separate call to the property.
Examples

The following example creates a `MySqlDataAdapter` and sets some of its properties.

**VB.NET**

```vbnet
Public Sub CreateSqlDataAdapter()
Dim da As MySqlDataAdapter = New MySqlDataAdapter("SELECT id, name FROM mytable")
Dim conn As MySqlConnection = da.SelectCommand.Connection

da.InsertCommand = New MySqlCommand("INSERT INTO mytable (id, name) VALUES (@id, @name)", conn)
da.UpdateCommand = New MySqlCommand("UPDATE mytable SET id=@id, name WHERE id=@oldId", conn)
da.DeleteCommand = New MySqlCommand("DELETE FROM mytable WHERE id=@id")

da.InsertCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
da.InsertCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
da.UpdateCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@oldId", MySqlDbType.VarChar, 5, "id")
da.DeleteCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
End Sub
```

**C#**

```csharp
public static void CreateSqlDataAdapter()
{
    MySqlDataAdapter da = new MySqlDataAdapter("SELECT id, name FROM mytable")
    MySqlConnection conn = da.SelectCommand.Connection;

    da.InsertCommand = new MySqlCommand("INSERT INTO mytable (id, name) VALUES (@id, @name)", conn);
da.UpdateCommand = new MySqlCommand("UPDATE mytable SET id=@id, name WHERE id=@oldId", conn);
da.DeleteCommand = new MySqlCommand("DELETE FROM mytable WHERE id=@id")

da.InsertCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
da.InsertCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
da.UpdateCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "name")
da.UpdateCommand.Parameters.Add("@oldId", MySqlDbType.VarChar, 5, "id")
da.DeleteCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
}```
da.UpdateCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
da.UpdateCommand.Parameters.Add("@name", MySqlDbType.VarChar, 40, "n")
da.UpdateCommand.Parameters.Add("@oldId", MySqlDbType.VarChar, 5, "i")
da.DeleteCommand.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")}
See Also

See Also

MySqlDataAdapter Class
MySqlDataAdapter Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLDataAdapter` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddToBatch</td>
<td>Adds a <a href="#">IDbCommand</a> to the current batch. (Inherited from <a href="#">DbDataAdapter</a>)</td>
</tr>
<tr>
<td>ClearBatch</td>
<td>Removes all <a href="#">IDbCommand</a> objects from the batch. (Inherited from <a href="#">DbDataAdapter</a>)</td>
</tr>
<tr>
<td>CloneInternals</td>
<td>Obsolete. Creates a copy of this instance of <a href="#">DataAdapter</a>. (Inherited from <a href="#">DataAdapter</a>)</td>
</tr>
<tr>
<td>CreateObjRef</td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <a href="#">MarshalByRefObject</a>)</td>
</tr>
<tr>
<td>CreateRowUpdatedEvent</td>
<td>Initializes a new instance of the <a href="#">RowUpdatedEventArgs</a> class. (Inherited from <a href="#">DbDataAdapter</a>)</td>
</tr>
<tr>
<td>CreateRowUpdatingEvent</td>
<td>Initializes a new instance of the <a href="#">RowUpdatingEventArgs</a> class. (Inherited from <a href="#">DbDataAdapter</a>)</td>
</tr>
<tr>
<td>CreateTableMappings</td>
<td>Creates a new <a href="#">DataTableMappingCollection</a>.</td>
</tr>
</tbody>
</table>
Dispose

Overloaded.

Equals

Determines whether the specified Object is equal to the current Object.
(Inherited from Object.)

ExecuteBatch

Executes the current batch.
(Inherited from DbDataAdapter.)

Fill

Overloaded.

FillSchema

Overloaded.

Finalize

Releases unmanaged resources and performs other cleanup operations before the Component is reclaimed by garbage collection.
(Inherited from Component.)

GetBatchedParameter

Returns a IDataParameter from one of the commands in the current batch.
(Inherited from DbDataAdapter.)

GetBatchedRecordsAffected

Returns information about an individual update attempt within a larger batched update.
(Inherited from DbDataAdapter.)

GetFillParameters

Gets the parameters set by the user when executing an SQL SELECT statement.
(Inherited from DbDataAdapter.)

GetHashCode

Serves as a hash function for a particular type.
(Inherited from Object.)

Retrieves the current lifetime
- **GetLifetimeService**
  service object that controls the lifetime policy for this instance.
  (Inherited from MarshalByRefObject.)

- **GetService**
  Returns an object that represents a service provided by the Component or by its Container.
  (Inherited from Component.)

- **GetType**
  Gets the Type of the current instance.
  (Inherited from Object.)

- **HasTableMappings**
  Indicates whether a DataTableMappingCollection has been created.
  (Inherited fromDataAdapter.)

- **InitializeBatching**
  Initializes batching for the DbDataAdapter.
  (Inherited from DbDataAdapter.)

- **InitializeLifetimeService**
  Obtains a lifetime service object to control the lifetime policy for this instance.
  (Inherited from MarshalByRefObject.)

- **MemberwiseClone**
  Overloaded.

- **OnFillError**
  Invoked when an error occurs during a Fill.
  (Inherited from DataAdapter.)

- **OnRowUpdated**
  Raises the RowUpdated event of a .NET Framework data provider.
  (Inherited from DbDataAdapter.)

- **OnRowUpdating**
  Raises the RowUpdating event of a .NET Framework
**OnRowUpdating**

Resets FillLoadOption to its default state and causes Fill(DataSet) to honor AcceptChangesDuringFill. (Inherited from DataAdapter.)

**ResetFillLoadOption**

Determines whether the AcceptChangesDuringFill property should be persisted. (Inherited from DataAdapter.)

**ShouldSerializeAcceptChangesDuringFill**

Determines whether the FillLoadOption property should be persisted. (Inherited from DataAdapter.)

**ShouldSerializeFillLoadOption**

Determines whether one or more DataTableMapping objects exist and they should be persisted. (Inherited from DataAdapter.)

**ShouldSerializeTableMappings**

Ends batching for the DbDataAdapter.

**TerminateBatching**

Returns a String containing the name of the Component, if any. This method should not be overridden. (Inherited from Component.)

**ToString**

Overloaded.

**Update**
See Also

MySQLDataAdapter Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose()()</td>
<td>Releases all resources used by the <a href="#">Component</a>. (Inherited from <a href="#">Component</a>.)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases the unmanaged resources used by the <a href="#">DbDataAdapter</a> and optionally releases the managed resources. (Inherited from <a href="#">DbDataAdapter</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlDataAdapter Class
MySqlDataAdapter Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members
MySQL Connector/Net
MySqlDataAdapter...Fill Method

MySqlDataAdapter Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill(DataSet)</td>
<td>Adds or refreshes rows in the <strong>DataSet</strong>. (Inherited from <strong>DbDataAdapter</strong>.)</td>
</tr>
<tr>
<td>Fill(DataTable)</td>
<td>Adds or refreshes rows in a specified range in the <strong>DataSet</strong> to match those in the data source using the <strong>DataTable</strong> name. (Inherited from <strong>DbDataAdapter</strong>.)</td>
</tr>
<tr>
<td>Fill(DataSet, String)</td>
<td>Adds or refreshes rows in the <strong>DataSet</strong> to match those in the data source using the <strong>DataSet</strong> and <strong>DataTable</strong> names. (Inherited from <strong>DbDataAdapter</strong>.)</td>
</tr>
<tr>
<td>Fill(DataTable, IDataReader)</td>
<td>Adds or refreshes rows in the <strong>DataTable</strong> to match those in the data source using the <strong>DataTable</strong> name and the specified <strong>IDataReader</strong>. (Inherited from <strong>DataAdapter</strong>.)</td>
</tr>
<tr>
<td>Fill(Int32, Int32, array&lt;DataTable&gt;[][])</td>
<td>Adds or refreshes rows in a <strong>DataTable</strong> to match those in the data source starting at the specified record and retrieving up to the specified maximum number of records. (Inherited from <strong>DbDataAdapter</strong>.)</td>
</tr>
<tr>
<td>Fill(DataTable, IDbCommand, CommandBehavior)</td>
<td>Adds or refreshes rows in a <strong>DataTable</strong> to match those in the data source using the specified <strong>DataTable</strong>, <strong>IDbCommand</strong> and <strong>CommandBehavior</strong>. (Inherited from <strong>DbDataAdapter</strong>.)</td>
</tr>
<tr>
<td>Fill(DataSet, Int32, Int32, String)</td>
<td>Adds or refreshes rows in a specified range in the <strong>DataSet</strong> to match those in the data source using the <strong>DataSet</strong> and <strong>DataTable</strong> names. (Inherited from <strong>DbDataAdapter</strong>.)</td>
</tr>
<tr>
<td>Fill(array&lt;DataTable&gt;[][])</td>
<td>Adds or refreshes rows in a specified range in the collection of <strong>DataTable</strong> objects to</td>
</tr>
</tbody>
</table>
**IDataReader, Int32, Int32**

match those in the data source.
(Inherited from **DataAdapter**.)

**Fill(array.DataTable[][], Int32, Int32, IDbCommand, CommandBehavior)**

Adds or refreshes rows in a specified range in the **DataSet** to match those in the data source using the **DataSet** and **DataTable** names.
(Inherited from **DbDataAdapter**.)

**Fill(DataSet, String, IDataReader, Int32, Int32)**

Adds or refreshes rows in a specified range in the **DataSet** to match those in the data source using the **DataSet** and **DataTable** names.
(Inherited from **DataAdapter**.)

**Fill(DataSet, Int32, Int32, String, IDbCommand, CommandBehavior)**

Adds or refreshes rows in a specified range in the **DataSet** to match those in the data source using the **DataSet** and source table names, command string, and command behavior.
(Inherited from **DbDataAdapter**.)
See Also

SqlConnection Class
SqlConnection Members
SqlConnection Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataAdapter::FillSchema Method

MySqlDataAdapter Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FillSchema(DataTable, SchemaType)</td>
<td>Configures the schema of the specified DataTable based on the specified SchemaType. (Inherited from DbDataAdapter.)</td>
</tr>
<tr>
<td>FillSchema(DataSet, SchemaType)</td>
<td>Adds a DataTable named &quot;Table&quot; to the specified DataSet and configures the schema to match that in the data source based on the specified SchemaType. (Inherited from DbDataAdapter.)</td>
</tr>
<tr>
<td>FillSchema(DataSet, SchemaType, String)</td>
<td>Adds a DataTable to the specified DataSet and configures the schema to match that in the data source based upon the specified SchemaType and DataTable. (Inherited from DbDataAdapter.)</td>
</tr>
<tr>
<td>FillSchema(DataTable, SchemaType, String, IDataReader)</td>
<td>Adds a DataTable to the specified DataSet. (Inherited from Adapter.)</td>
</tr>
<tr>
<td>FillSchema(DataTable, SchemaType, IDbCommand, CommandBehavior)</td>
<td>Configures the schema of the specified DataTable based on the specified SchemaType, command string, and CommandBehavior values. (Inherited from DbDataAdapter.)</td>
</tr>
<tr>
<td>FillSchema(DataSet, SchemaType, String, IDataReader)</td>
<td>Adds a DataTable to the specified DataSet. (Inherited from Adapter.)</td>
</tr>
<tr>
<td>FillSchema(DataSet, SchemaType, IDbCommand, String, CommandBehavior)</td>
<td>Adds a DataTable to the specified DataSet and configures the schema to match that in the data source based on the specified SchemaType. (Inherited from DbDataAdapter.)</td>
</tr>
</tbody>
</table>
See Also

[MySqlDataAdapter Class](#)
[MySqlDataAdapter Members](#)
[MySql.Data.MySqlClient Namespace](#)

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlDataAdapter::MemberwiseClone Method

See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MemberwiseClone()</code>()</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone(Boolean)</code></td>
<td>Creates a shallow copy of the current <code>MarshalByRefObject</code> object. (Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySQLDataAdapter Class
MySQLDataAdapter Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update(DataSet)</td>
<td>Calls the respective INSERT, UPDATE, or DELETE statements for each inserted, updated, or deleted row in the specified <strong>DataSet</strong>.</td>
</tr>
<tr>
<td>(Inherited from DbDataAdapter.)</td>
<td></td>
</tr>
<tr>
<td>Update(array&lt;DataRow&gt;[][])</td>
<td>Calls the respective INSERT, UPDATE, or DELETE statements for each inserted, updated, or deleted row in the specified array of <strong>DataRow</strong> objects.</td>
</tr>
<tr>
<td>(Inherited from DbDataAdapter.)</td>
<td></td>
</tr>
<tr>
<td>Update(DataTable)</td>
<td>Calls the respective INSERT, UPDATE, or DELETE statements for each inserted, updated, or deleted row in the specified <strong>DataTable</strong>.</td>
</tr>
<tr>
<td>(Inherited from DbDataAdapter.)</td>
<td></td>
</tr>
<tr>
<td>Update(DataSet, String)</td>
<td>Calls the respective INSERT, UPDATE, or DELETE statements for each inserted, updated, or deleted row in the <strong>DataSet</strong> with the specified <strong>DataTable</strong> name.</td>
</tr>
<tr>
<td>(Inherited from DbDataAdapter.)</td>
<td></td>
</tr>
<tr>
<td>Update(array&lt;DataRow&gt;[][], DataTableMapping)</td>
<td>Calls the respective INSERT, UPDATE, or DELETE statements for each inserted, updated, or deleted row in the specified array of <strong>DataRow</strong> objects.</td>
</tr>
<tr>
<td>(Inherited from DbDataAdapter.)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySQLDataAdapter Class
MySQLDataAdapter Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlDataAdapter` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcceptChangesDuringFill</td>
<td>Gets or sets a value indicating whether <code>AcceptChanges()</code> is called on a <code>DataRow</code> after it is added to the <code>DataTable</code> during any of the Fill operations. (Inherited from <code>DataAdapter</code>.)</td>
</tr>
<tr>
<td>AcceptChangesDuringUpdate ()</td>
<td>Gets or sets whether <code>AcceptChanges()</code> is called during a <code>Update(DataSet)</code>. (Inherited from <code>DataAdapter</code>.)</td>
</tr>
<tr>
<td>CanRaiseEvents</td>
<td>Gets a value indicating whether the component can raise an event. (Inherited from <code>Component</code>.)</td>
</tr>
<tr>
<td>Container</td>
<td>Gets or sets a value that specifies whether to generate an exception when an error is encountered during a row update. (Inherited from <code>DataAdapter</code>.)</td>
</tr>
<tr>
<td>ContinueUpdateOnError</td>
<td>Gets or sets a SQL statement or stored procedure used to delete records from the data set.</td>
</tr>
<tr>
<td>DeleteCommand</td>
<td>Gets a value that indicates whether the <code>Component</code> is currently in design mode. (Inherited from <code>Component</code>.)</td>
</tr>
<tr>
<td>DesignMode</td>
<td>Gets the <code>IContainer</code> that contains the <code>Component</code>.</td>
</tr>
<tr>
<td>Events</td>
<td>Gets the list of event handlers that are attached to this <code>Component</code>. (Inherited from <code>Component</code>.)</td>
</tr>
<tr>
<td>FillCommandBehavior</td>
<td>Gets or sets the behavior of the command used to fill the data adapter. (Inherited from <code>DbDataAdapter</code>.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FillLoadOption</td>
<td>Gets or sets the <strong>LoadOption</strong> that determines how the adapter fills the <strong>DataTable</strong> from the <strong>DbDataReader</strong>. (Inherited from <strong>DataAdapter</strong>.)</td>
</tr>
<tr>
<td>InsertCommand</td>
<td>Gets or sets a SQL statement or stored procedure used to insert records into the data set.</td>
</tr>
<tr>
<td>MissingMappingAction</td>
<td>Determines the action to take when incoming data does not have a matching table or column. (Inherited from <strong>DataAdapter</strong>.)</td>
</tr>
<tr>
<td>MissingSchemaAction</td>
<td>Determines the action to take when existing <strong>DataSet</strong> schema does not match incoming data. (Inherited from <strong>DataAdapter</strong>.)</td>
</tr>
<tr>
<td>ReturnProviderSpecificTypes</td>
<td>Gets or sets whether the Fill method should return provider-specific values or common CLS-compliant values. (Inherited from <strong>DataAdapter</strong>.)</td>
</tr>
<tr>
<td>SelectCommand</td>
<td>Gets or sets a SQL statement or stored procedure used to select records in the data source.</td>
</tr>
<tr>
<td>Site</td>
<td>Gets or sets the <strong>ISite</strong> of the <strong>Component</strong>. (Inherited from <strong>Component</strong>.)</td>
</tr>
<tr>
<td>TableMappings</td>
<td>Gets a collection that provides the master mapping between a source table and a <strong>DataTable</strong>. (Inherited from <strong>DataAdapter</strong>.)</td>
</tr>
<tr>
<td>UpdateBatchSize</td>
<td>(Overrides <strong>DbDataAdapter</strong>:::<strong>UpdateBatchSize</strong>.)</td>
</tr>
<tr>
<td>UpdateCommand</td>
<td>Gets or sets a SQL statement or stored procedure used to updated records in the data source.</td>
</tr>
</tbody>
</table>
See Also

**MySQLDataAdapter Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a SQL statement or stored procedure used to delete records from the data set.

**Namespace:**  [MySQL.Data.MySqlClient](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
class YourClass
{
    public MySqlCommand DeleteCommand { get; set; }
}
```

Visual Basic (Declaration)

```vbnet
Public Property DeleteCommand As MySqlCommand
```

Visual C++

```cpp
public:
    property MySqlCommand^ DeleteCommand
    {
        MySqlCommand^ get ()
        {
            MySqlCommand^ value = ...;
            return value;
        }
        void set (MySqlCommand^ value);
    }
```

Field Value

A `MySqlCommand` used during `Update(DataSet)` to delete records in the database that correspond to deleted rows in the `DataSet`. 
Remarks

During **Update(DataSet)**, if this property is not set and primary key information is present in the **DataSet**, the DeleteCommand can be generated automatically if you set the **SelectCommand** property and use the **MySqlCommandBuilder**. Then, any additional commands that you do not set are generated by the MySqlCommandBuilder. This generation logic requires key column information to be present in the DataSet.

When DeleteCommand is assigned to a previously created **MySqlCommand**, the MySqlCommand is not cloned. The DeleteCommand maintains a reference to the previously created MySqlCommand object.
Examples

The following example creates a `MySqlDataAdapter` and sets the `SelectCommand` and `DeleteCommand` properties. It assumes you have already created a `MySqlConnection` object.

**VB.NET**

```vbnet
Public Shared Function CreateCustomerAdapter(conn As MySqlConnection)
Dim da As MySqlDataAdapter = New MySqlDataAdapter()
Dim cmd As MySqlCommand
Dim parm As MySqlParameter

' Create the SelectCommand.
    cmd = New MySqlCommand("SELECT * FROM mytable WHERE id=@id AND name=
    cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15)
    cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15)

da.SelectCommand = cmd

' Create the DeleteCommand.
    cmd = New MySqlCommand("DELETE FROM mytable WHERE id=@id", conn)
    parm = cmd.Parameters.Add("@id", MySqlDbType.VarChar, 5, "id")
    parm.SourceVersion = DataRowVersion.Original

da.DeleteCommand = cmd

Return da
End Function
```

**C#**

```csharp
public static MySqlDataAdapter CreateCustomerAdapter(MySqlConnection conn) {
    MySqlDataAdapter da = new MySqlDataAdapter();
    MySqlCommand cmd;
    MySqlParameter parm;

    // Create the SelectCommand.
    cmd = new MySqlCommand("SELECT * FROM mytable WHERE id=@id AND name=
```
cmd.Parameters.Add("@id", SqlDbType.VarChar, 15);
cmd.Parameters.Add("@name", SqlDbType.VarChar, 15);

da.SelectCommand = cmd;

// Create the DeleteCommand.
new MySqlCommand("DELETE FROM mytable WHERE id=@id", conn);

parm = cmd.Parameters.Add("@id", SqlDbType.VarChar, 5, "id");
parm.SourceVersion = DataRowVersion.Original;

da.DeleteCommand = cmd;

return da;
See Also

MySqlDataAdapter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a SQL statement or stored procedure used to insert records into the data set.

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:** `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public MySqlConnection InsertCommand { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property InsertCommand As MySqlConnection
```

**Visual C++**

```cpp
public:
property MySqlConnection^ InsertCommand { 
    MySqlConnection^ get ();
    void set (MySqlConnection^ value);
}
```

**Field Value**

A `MySqlCommand` used during `Update(DataSet)` to insert records into the database that correspond to new rows in the `DataSet`. 
Remarks

During **Update(DataSet)**, if this property is not set and primary key information is present in the **DataSet**, the InsertCommand can be generated automatically if you set the **SelectCommand** property and use the **MySqlCommandBuilder**. Then, any additional commands that you do not set are generated by the MySqlCommandBuilder. This generation logic requires key column information to be present in the DataSet.

When InsertCommand is assigned to a previously created **MySqlCommand**, the MySqlCommand is not cloned. The InsertCommand maintains a reference to the previously created MySqlCommand object.

**Note:** If execution of this command returns rows, these rows may be added to the DataSet depending on how you set the **UpdatedRowSource** property of the MySqlCommand object.
Examples

The following example creates a **MySqlDataAdapter** and sets the **SelectCommand** and InsertCommand properties. It assumes you have already created a **MySqlConnection** object.

**VB.NET**

```vbnet
Public Shared Function CreateCustomerAdapter(conn As MySqlConnection)
Dim da As MySqlDataAdapter = New MySqlDataAdapter()
Dim cmd As MySqlCommand
Dim parm As SqlParameter

' Create the SelectCommand.
cmd = New MySqlCommand("SELECT * FROM mytable WHERE id=@id AND name=
cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15)
cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15)
da.SelectCommand = cmd

' Create the InsertCommand.
' cmd = New MySqlCommand("INSERT INTO mytable (id,name) VALUES (@id, @
cmd.Parameters.Add( "@id", MySqlDbType.VarChar, 15, "id" )
cmd.Parameters.Add( "@name", MySqlDbType.VarChar, 15, "name" )
da.InsertCommand = cmd

Return da
End Function
```

**C#**

```csharp
public static MySqlDataAdapter CreateCustomerAdapter(MySqlConnection conn)
{
    MySqlDataAdapter da = new MySqlDataAdapter();
    MySqlCommand cmd;
    SqlParameter parm;

    // Create the SelectCommand.
    cmd = new MySqlCommand("SELECT * FROM mytable WHERE id=@id AND name=
    cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15);
```
cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15);

da.SelectCommand = cmd;

// Create the InsertCommand.
    cmd = new MySqlCommand("INSERT INTO mytable (id,name) VALUES (@id,@n
cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15, "id" )
    cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15, "name")

    da.InsertCommand = cmd;

return da;
}
See Also

MySQLDataAdapter Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a SQL statement or stored procedure used to select records in the data source.

**Namespace:**  [MySQL.Data.MySqlClient](https://example.com)

**Assembly:**  [MySQL.Data](https://example.com) (in [MySQL.Data.dll](https://example.com)) Version: 6.2.2.0
Syntax

C#

public MySqlCommand SelectCommand { get; set; }

Visual Basic (Declaration)

Public Property SelectCommand As MySqlCommand

Visual C++

public:
 property MySqlCommand^ SelectCommand {  
MySqlCommand^ get ();  
void set (MySqlCommand^ value);  
}

Field Value

A MySqlCommand used during Fill(DataSet) to select records from the database for placement in the DataSet.
Remarks

When SelectCommand is assigned to a previously created `MySqlCommand`, the MySqlCommand is not cloned. The SelectCommand maintains a reference to the previously created MySqlCommand object.

If the SelectCommand does not return any rows, no tables are added to the `DataSet`, and no exception is raised.
Examples

The following example creates a `MySqlDataAdapter` and sets the SelectCommand and InsertCommand properties. It assumes you have already created a `MySqlConnection` object.

**VB.NET**

```vbnet
Public Shared Function CreateCustomerAdapter(conn As MySqlConnection)
    Dim da As MySqlDataAdapter = New MySqlDataAdapter()
    Dim cmd As MySqlCommand
    Dim parm As SqlParameter

    ' Create the SelectCommand.
    cmd = New MySqlCommand("SELECT * FROM mytable WHERE id=@id AND name=@name",
                          conn)
    cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15)
    cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15)

    da.SelectCommand = cmd

    ' Create the InsertCommand.
    cmd = New MySqlCommand("INSERT INTO mytable (id,name) VALUES (@id, @name)
                          VALUES (@id, @name)
                          VALUES (@id, @name)",
                          conn)
    cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15, "id")
    cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15, "name")

    da.InsertCommand = cmd

    Return da
End Function
```

**C#**

```csharp
public static MySqlDataAdapter CreateCustomerAdapter(MySqlConnection conn)
{
    MySqlDataAdapter da = new MySqlDataAdapter();
    MySqlCommand cmd;
    SqlParameter parm;

    // Create the SelectCommand.
    cmd = new MySqlCommand("SELECT * FROM mytable WHERE id=@id AND name=
                          VALUES (@id, @name)
                          VALUES (@id, @name)",
                          conn)
    cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15);
```
cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15);

da.SelectCommand = cmd;

// Create the InsertCommand.
cmd = new MySqlCommand("INSERT INTO mytable (id,name) VALUES (@id,@n
cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15, "id" );
cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15, "name" );

da.InsertCommand = cmd;

return da;
}
See Also

 MySqlDataAdapter Class
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL DataAdapter....:::UpdateBatchSize Property

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll)  Version: 6.2.2.0
Syntax

C#
public override int UpdateBatchSize { get; set; }

Visual Basic (Declaration)
Public Overrides Property UpdateBatchSize As Integer

Visual C++
public:
virtual property int UpdateBatchSize {
    int get () override;
    void set (int value) override;
}
See Also

MySqlDataAdapter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a SQL statement or stored procedure used to updated records in the data source.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public MySqlCommand UpdateCommand { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property UpdateCommand As MySqlCommand
```

**Visual C++**

```csharp
public:
property MySqlCommand^ UpdateCommand {
    MySqlCommand^ get ();
    void set (MySqlCommand^ value);
}
```

**Field Value**

A **MySqlCommand** used during **Update(DataSet)** to update records in the database with data from the **DataSet**.
Remarks

During `Update(DataSet)`, if this property is not set and primary key information is present in the `DataSet`, the `UpdateCommand` can be generated automatically if you set the `SelectCommand` property and use the `MySqlCommandBuilder`. Then, any additional commands that you do not set are generated by the `MySqlCommandBuilder`. This generation logic requires key column information to be present in the `DataSet`.

When `UpdateCommand` is assigned to a previously created `MySqlCommand`, the `MySqlCommand` is not cloned. The `UpdateCommand` maintains a reference to the previously created `MySqlCommand` object.

☑️ **Note:** If execution of this command returns rows, these rows may be merged with the `DataSet` depending on how you set the `UpdatedRowSource` property of the `MySqlCommand` object.
Examples

The following example creates a **MySqlDataAdapter** and sets the **SelectCommand** and UpdateCommand properties. It assumes you have already created a **MySqlConnection** object.

**VB.NET**

```vbnet
Public Shared Function CreateCustomerAdapter(conn As MySqlConnection)
    Dim da As MySqlDataAdapter = New MySqlDataAdapter()
    Dim cmd As MySqlCommand
    Dim parm As SqlParameter

    ' Create the SelectCommand.
    cmd = New MySqlCommand("SELECT * FROM mytable WHERE id=@id AND name=@name",
        conn)
    cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15)
    cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15)

    da.SelectCommand = cmd

    ' Create the UpdateCommand.
    cmd = New MySqlCommand("UPDATE mytable SET id=@id, name=@name WHERE id=@oldId",
        conn)
    cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15, "id")
    cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15, "name")
    parm = cmd.Parameters.Add("@oldId", MySqlDbType.VarChar, 15, "id")
    parm.SourceVersion = DataRowVersion.Original

    da.UpdateCommand = cmd

    Return da
End Function
```

**C#**

```csharp
public static MySqlDataAdapter CreateCustomerAdapter(MySqlConnection conn)
{
    MySqlDataAdapter da = new MySqlDataAdapter();
    MySqlCommand cmd;
    SqlParameter parm;
```
// Create the SelectCommand.
cmd = new MySqlCommand("SELECT * FROM mytable WHERE id=@id AND name=

cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15);
cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15);

da.SelectCommand = cmd;

// Create the UpdateCommand.
cmd = new MySqlCommand("UPDATE mytable SET id=@id, name=@name WHERE

cmd.Parameters.Add("@id", MySqlDbType.VarChar, 15, "id" );
cmd.Parameters.Add("@name", MySqlDbType.VarChar, 15, "name" );

parm = cmd.Parameters.Add( "@oldId", MySqlDbType.VarChar, 15, "id" )
parm.SourceVersion = DataRowVersion.Original;

da.UpdateCommand = cmd;

return da;
}
See Also

MySqlDataAdapter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySqlDataAdapter** type exposes the following members.
# Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposed</td>
<td>Occurs when the component is disposed by a call to the Dispose() method.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Component.)</td>
</tr>
<tr>
<td>FillError</td>
<td>Returned when an error occurs during a fill operation.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DataAdapter.)</td>
</tr>
<tr>
<td>RowUpdated</td>
<td>Occurs during Update after a command is executed against the data source.</td>
</tr>
<tr>
<td></td>
<td>The attempt to update is made, so the event fires.</td>
</tr>
<tr>
<td>RowUpdating</td>
<td>Occurs during Update before a command is executed against the data source.</td>
</tr>
<tr>
<td></td>
<td>The attempt to update is made, so the event fires.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataAdapter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Occurs during Update after a command is executed against the data source. The attempt to update is made, so the event fires.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public event MySqlRowUpdatedEventHandler RowUpdated

Visual Basic (Declaration)

Public Event RowUpdated As MySqlRowUpdatedEventHandler

Visual C++

public:
    event MySqlRowUpdatedEventHandler^ RowUpdated {
        void add (MySqlRowUpdatedEventHandler^ value);
        void remove (MySqlRowUpdatedEventHandler^ value);
    }
See Also

**MySqlCommand Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Occurs during Update before a command is executed against the data source. The attempt to update is made, so the event fires.

**Namespace:** [MySQL.Data.MySqlClient](#)  
**Assembly:** MySQL.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

### C#

```csharp
public event MySqlRowUpdatingEventHandler RowUpdating
```

### Visual Basic (Declaration)

```vbnet
Public Event RowUpdating As MySqlRowUpdatingEventHandler
```

### Visual C++

```cpp
public:
  event MySqlRowUpdatingEventHandler^ RowUpdating {
    void add (MySqlRowUpdatingEventHandler^ value);
    void remove (MySqlRowUpdatingEventHandler^ value);
  }
```

See Also

MySqlDataAdapter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Provides a means of reading a forward-only stream of rows from a MySQL database. This class cannot be inherited.

**Namespace:** [MySQL.Data.MySqlClient](#)  
**Assembly:** [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public sealed class MySqlDataReader : DbDataReader, 
IDataReader, IDisposable, IDataRecord

Visual Basic (Declaration)

Public NotInheritable Class MySqlDataReader _
Inherits DbDataReader _
Implements IDataReader, IDisposable, IDataRecord

Visual C++

public ref class MySqlDataReader sealed : public DbDataReader, 
IDataReader, IDisposable, IDataRecord
Remarks

To create a MySQLDataReader, you must call the `ExecuteReader()` method of the `MySqlCommand` object, rather than directly using a constructor.

While the MySqlDataReader is in use, the associated `MySqlConnection` is busy serving the MySqlDataReader, and no other operations can be performed on the MySqlConnection other than closing it. This is the case until the `Close()` method of the MySqlDataReader is called.

`IsClosed` and `RecordsAffected` are the only properties that you can call after the MySqlDataReader is closed. Though the RecordsAffected property may be accessed at any time while the MySqlDataReader exists, always call Close before returning the value of RecordsAffected to ensure an accurate return value.

For optimal performance, MySqlDataReader avoids creating unnecessary objects or making unnecessary copies of data. As a result, multiple calls to methods such as `GetValue(Int32)` return a reference to the same object. Use caution if you are modifying the underlying value of the objects returned by methods such as GetValue.
Examples

The following example creates a `MySqlConnection`, a `MySqlCommand`, and a `MySqlDataReader`. The example reads through the data, writing it out to the console. Finally, the example closes the `MySqlDataReader`, then the `MySqlConnection`.

**VB.NET**

```vbnet
Public Sub ReadMyData(myConnString As String)
Dim mySelectQuery As String = "SELECT OrderID, CustomerID FROM Orders"
Dim myConnection As New MySqlConnection(myConnString)
Dim myCommand As New MySqlCommand(mySelectQuery, myConnection)
myConnection.Open()
Dim myReader As MySqlDataReader
myReader = myCommand.ExecuteReader()
' Always call Read before accessing data.
While myReader.Read()
    Console.WriteLine((myReader.GetInt32(0) & ", " & myReader.GetString(1)))
End While
' always call Close when done reading.
myReader.Close()
' Close the connection when done with it.
myConnection.Close()
End Sub 'ReadMyData
```

**C#**

```csharp
public void ReadMyData(string myConnString) {
    string mySelectQuery = "SELECT OrderID, CustomerID FROM Orders";
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    MySqlCommand myCommand = new MySqlCommand(mySelectQuery, myConnection);
    myConnection.Open();
    MySqlDataReader myReader;
    myReader = myCommand.ExecuteReader();
    // Always call Read before accessing data.
    while (myReader.Read()) {
        Console.WriteLine(myReader.GetInt32(0) + ", " + myReader.GetString(1));
    }
    // always call Close when done reading.
    myReader.Close();
    // Close the connection when done with it.
    myConnection.Close();
}
```
Inheritance Hierarchy

System::Object
System::MarshalByRefObject
System.Data.Common::DbDataReader
MySql.Data.MySqlClient::MySqlDataReader
See Also

MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlDataReader` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Close</strong></td>
<td>Closes the MySqlDataReader object. (Overrides <code>DbDataReader::Close()</code>.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetBoolean</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetByte</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetBytes</strong></td>
<td>Reads a stream of bytes from the specified column offset into the buffer as an array starting at the given buffer offset. (Overrides <code>DbDataReader::GetBytes(Int32, Int64, array&lt;Byte&gt;[], Int32, Int32)</code>.)</td>
</tr>
<tr>
<td><strong>GetChar</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetChars</strong></td>
<td>Reads a stream of characters from the specified column offset into the buffer as an array starting at the given buffer offset. (Overrides <code>DbDataReader::GetChars(Int32, Int64, array&lt;Char&gt;[], Int32, Int32)</code>.)</td>
</tr>
<tr>
<td><strong>GetData</strong></td>
<td>Returns a <code>DbDataReader</code> object for the requested column ordinal. (Inherited from <code>DbDataReader</code>.)</td>
</tr>
</tbody>
</table>

**Gets the name of the source data type.**
- **GetDataTypeName**  (Overrides `DbDataReader::GetDataTypeName(Int32)`)
- **GetDateTime**  Overloaded.
- **GetDbDataReader**  Returns a `DbDataReader` object for the requested column ordinal that can be overridden with a provider-specific implementation.
  (Inherited from `DbDataReader`.)
- **GetDecimal**  Overloaded.
- **GetDouble**  Overloaded.
- **GetEnumerator**  Returns an `IEnumerator` that iterates through the `MySqlDataReader`.
  (Overrides `DbDataReader::GetEnumerator()`)
- **GetFieldType**  Gets the Type that is the data type of the object.
  (Overrides `DbDataReader::GetFieldType(Int32)`)
- **GetFloat**  Overloaded.
- **GetGuid**  Overloaded.
- **GetHashCode**  Serves as a hash function for a particular type.
  (Inherited from `Object`.)
- **GetInt16**  Overloaded.
- **GetInt32**  Overloaded.
- **GetInt64**  Overloaded.
- **GetLifetimeService**  Retrieves the current lifetime service object that controls the lifetime policy for this instance.
  (Inherited from `MarshalByRefObject`.)
- **GetMySqlDateTime**  Overloaded.
- **GetMySqlDecimal**  Overloaded.
- **GetName**  Gets the name of the specified column.
  (Overrides `DbDataReader::GetName(Int32)`)
- **GetOrdinal**  Gets the column ordinal, given the name of the column.
  (Overrides `DbDataReader::GetOrdinal(String)`
  Returns the provider-specific field type of the specified column.
  (Inherited from `DbDataReader`.)
- **GetProviderSpecificValue**: Gets the value of the specified column as an instance of `Object`. (Inherited from `DbDataReader`.)

- **GetProviderSpecificValues**: Gets all provider-specific attribute columns in the collection for the current row. (Inherited from `DbDataReader`.)

- **GetSByte**: Overloaded. Returns a `DataTable` that describes the column metadata of the MySqlDataReader. (Overrides `DbDataReader`:::GetSchemaTable().)

- **GetSchemaTable**: Overloaded. Gets the `Type` of the current instance. (Inherited from `Object`.)

- **GetString**: Overloaded. Gets all `DateTime` columns in the collection for the current row. (Overrides `DbDataReader`:::GetValues(array<`Object>[]).)

- **GetTimeSpan**: Overloaded. Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from `MarshalByRefObject`.)

- **GetType**: Gets the value of the specified column in its native format. (Overrides `DbDataReader`:::GetValues(array<`Object>[]).)

- **GetValue**: Overloaded. Gets all attribute columns in the collection for the current row. (Overrides `DbDataReader`:::GetValues(array<`Object>[]).)

- **GetValues**: Overloaded. Advances the data reader to the next result, when reading the results of batch SQL statements. (Overrides `DbDataReader`:::NextResult().)

- **InitializeLifetimeService**: Advances the MySqlDataReader to the next
Read

record.
(Overrides DbDataReader:::Read().)

ToString

Returns a String that represents the current Obj
(Inherited from Object.)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth</td>
<td>Gets a value indicating the depth of nesting for the current row. This method is not supported currently and always returns 0. (Overrides <code>DbDataReader::Depth</code>.)</td>
</tr>
<tr>
<td>FieldCount</td>
<td>Gets the number of columns in the current row. (Overrides <code>DbDataReader::FieldCount</code>.)</td>
</tr>
<tr>
<td>HasRows</td>
<td>Gets a value indicating whether the MySqlDataReader contains one or more rows. (Overrides <code>DbDataReader::HasRows</code>.)</td>
</tr>
<tr>
<td>IsClosed</td>
<td>Gets a value indicating whether the data reader is closed. (Overrides <code>DbDataReader::IsClosed</code>.)</td>
</tr>
<tr>
<td>Item</td>
<td>Overloaded. Gets the number of rows changed, inserted, or deleted by execution of the SQL statement. (Overrides <code>DbDataReader::RecordsAffected</code>.)</td>
</tr>
<tr>
<td>RecordsAffected</td>
<td>Gets the number of fields in the <code>DbDataReader</code> that are not hidden. (Inherited from <code>DbDataReader</code>.)</td>
</tr>
<tr>
<td>VisibleFieldCount</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlDataReader` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️ Close</td>
<td>Closes the MySqlDataReader object. (Overrides <code>DbDataReader..::.Close()</code>.)</td>
</tr>
<tr>
<td>✔️ CreateObjRef</td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td>✔️ Dispose</td>
<td>Overloaded. Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>✔️ Equals</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>✔️ Finalize</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>✔️ GetBoolean</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>✔️ GetByte</td>
<td>Reads a stream of bytes from the specified column offset into the buffer an array starting at the given buffer offset. (Overrides <code>DbDataReader..::.GetBytes(Int32, Int64, array&lt;Byte&gt;[][], Int32, Int32).</code>)</td>
</tr>
<tr>
<td>✔️ GetBytes</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>✔️ GetChar</td>
<td>Reads a stream of characters from the specified column offset into the buffer as an array starting at the given buffer offset. (Overrides <code>DbDataReader..::.GetChars(Int32, Int64, array&lt;Char&gt;[][], Int32, Int32).</code>)</td>
</tr>
<tr>
<td>✔️ GetChars</td>
<td>Returns a <code>DbDataReader</code> object for the requested column ordinal. (Inherited from <code>DbDataReader</code>.)</td>
</tr>
<tr>
<td>✔️ GetData</td>
<td>Gets the name of the source data type.</td>
</tr>
</tbody>
</table>
**GetDataTypeName**
(Overrides `DbDataReader::GetDataTypeName(Int32)`.)

**GetDateTime**
Overloaded.

**GetDbDataReader**
Returns a `DbDataReader` object for the requested column ordinal that can be overridden with a provider-specific implementation.
(Inherited from `DbDataReader`.)

**GetDecimal**
Overloaded.

**GetDouble**
Overloaded.

**GetEnumerator**
Returns an `IEnumerator` that iterates through the `MySqlDataReader`.
(Overrides `DbDataReader::GetEnumerator()`.)

**GetFieldType**
Gets the Type that is the data type of the object.
(Overrides `DbDataReader::GetFieldType(Int32)`.)

**GetFloat**
Overloaded.

**GetGuid**
Overloaded.

**GetHashCode**
Serves as a hash function for a particular type.
(Inherited from `Object`.)

**GetInt16**
Overloaded.

**GetInt32**
Overloaded.

**GetInt64**
Overloaded.

**GetLifetimeService**
Retrieves the current lifetime service object that controls the lifetime policy for this instance.
(Inherited from `MarshalByRefObject`.)

**GetMySqlDateTime**
Overloaded.

**GetMySqlDecimal**
Overloaded.

**GetName**
Gets the name of the specified column.
(Overrides `DbDataReader::GetName(Int32)`.)

**GetOrdinal**
Gets the column ordinal, given the name of the column.
(Overrides `DbDataReader::GetOrdinal(String)`)

Returns the provider-specific field type of the specified column.
(Inherited from `DbDataReader`.)
**GetProviderSpecificValue**: Gets the value of the specified column as an instance of `Object`.  
(Inherited from `DbDataReader`.)

**GetProviderSpecificValues**: Gets all provider-specific attribute columns in the collection for the current row.  
(Inherited from `DbDataReader`.)

**GetSByte**: Overloaded.  
Returns a `DataTable` that describes the column metadata of the MySqlDataReader.  
(Overrides `DbDataReader..::.GetSchemaTable(0)`.)

**GetSchemaTable**: Overloaded.

**GetString**

**GetTimeSpan**

**GetType**: Gets the `Type` of the current instance.  
(Inherited from `Object`.)

**GetUInt16**

**GetUInt32**

**GetUInt64**

**GetValue**: Gets the value of the specified column in its native format.  
(Overrides `DbDataReader..::.GetValue(Int32)`.)

**GetValues**: Gets all attribute columns in the collection for the current row.  
(Overrides `DbDataReader..::.GetValues(array<Object>[])`.)

**InitializeLifetimeService**: Obtains a lifetime service object to control the lifetime policy for this instance.  
(Inherited from `MarshalByRefObject`.)

**IsDBNull**: Gets a value indicating whether the column contains non-existent or missing values.  
(Overrides `DbDataReader..::.IsDBNull(Int32)`.)

**MemberwiseClone**: Overloaded.

**NextResult**: Advances the data reader to the next result, when reading the results of batch SQL statements.  
(Overrides `DbDataReader..::.NextResult()`.)

Advances the MySqlDataReader to the next
Read

(Overrrides DbDataReader..::.Read())

ToString

Returns a String that represents the current Obj
(Inherited from Object.)
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Closes the MySqlDataReader object.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public override void Close()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Close
```

**Visual C++**

```cpp
public:
virtual void Close() override
```

### Implements

```csharp
IDataReader::Close();
IDataReader::Close();
```
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataReader... Dispose Method

MySqlDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose()()</td>
<td>Releases all resources used by the current instance of the <code>DbDataReader</code> class. (Inherited from <code>DbDataReader</code>.)</td>
</tr>
<tr>
<td>Dispose(Boolean)</td>
<td>Releases the managed resources used by the <code>DbDataReader</code> and optionally releases the unmanaged resources. (Inherited from <code>DbDataReader</code>.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlDataReader Class**
**MySqlDataReader Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlDataReader...:.GetBoolean Method

MySqlDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoolean(Int32)</td>
<td>Gets the value of the specified column as a Boolean. (Overrides DbDataReader::GetBoolean(Int32).)</td>
</tr>
<tr>
<td>GetBoolean(String)</td>
<td>Gets the value of the specified column as a Boolean.</td>
</tr>
</tbody>
</table>
See Also

MySQLDataReader Class
MySQLDataReader Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a Boolean.

**Namespace:** MySql.Data.MySqlClient

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public override bool GetBoolean( int i )
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function GetBoolean( _
    i As Integer _
) As Boolean
```

**Visual C++**

```cpp
public:
virtual bool GetBoolean( _
    int i ) override
```

**Parameters**

`i`

Type: `System::::Int32`

**Return Value**

**Implements**

`IDataRecord::::GetBoolean(Int32)`

`IDataRecord::::GetBoolean(Int32)`
See Also

**MySqlDataReader Class**  
**GetBoolean Overload**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a Boolean.

**Namespace:**  [MySQL.Data.MySqlClient](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool GetBoolean(
    string name
)

Visual Basic (Declaration)

Public Function GetBoolean ( _
    name As String _
) As Boolean

Visual C++

public:
    bool GetBoolean(
        String^ name
    )

Parameters

name
    Type: System:::String

Return Value
See Also

MySqlParameter Class
GetBoolean Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDataReader::GetByte Method

MySQLDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetByte(Int32)</td>
<td>Gets the value of the specified column as a byte. (Overrides DbDataReader.GetByte(Int32).)</td>
</tr>
<tr>
<td>GetByte(String)</td>
<td>Gets the value of the specified column as a byte.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a byte.

**Namespace:** [MySQL.Data.MySqlClient](#)  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override byte GetByte(
    int i
)

Visual Basic (Declaration)

Public Overrides Function GetByte ( _
    i As Integer _
) As Byte

Visual C++

public:
    virtual unsigned char GetByte(
        int i
    ) override

Parameters

i

    Type: System:::Int32

Return Value

Implements

IDataRecord:::GetByte(Int32)
IDataRecord:::GetByte(Int32)
See Also

MySqlDataReader Class
GetByte Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLDataReader Class

::

::

::

::

Gets the value of the specified column as a byte.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public byte GetByte(
    string name
)

Visual Basic (Declaration)

Public Function GetByte (_
    name As String _
) As Byte

Visual C++

public:
    unsigned char GetByte(
        String^ name
    )

Parameters

name
    Type: System::String

Return Value
See Also

MySqlDataReader Class
GetByte Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Reads a stream of bytes from the specified column offset into the buffer an array starting at the given buffer offset.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override long GetBytes(
    int i,
    long fieldOffset,
    byte[] buffer,
    int bufferoffset,
    int length
)

Visual Basic (Declaration)

Public Overrides Function GetBytes ( _
    i As Integer, _
    fieldOffset As Long, _
    buffer As Byte(), _
    bufferoffset As Integer, _
    length As Integer _
) As Long

Visual C++

public:
virtual long long GetBytes(
    int i,
    long long fieldOffset,
    array<unsigned char>^ buffer,
    int bufferoffset,
    int length
) override

Parameters

i
    Type: System::::Int32
    The zero-based column ordinal.

fieldOffset
Type: `System::Int64`
The index within the field from which to begin the read operation.

`buffer`
Type: `array< System::Byte >[]`[]
The buffer into which to read the stream of bytes.

`bufferoffset`
Type: `System::Int32`
The index for buffer to begin the read operation.

`length`
Type: `System::Int32`
The maximum length to copy into the buffer.

**Return Value**

The actual number of bytes read.

**Implements**

`IDataRecord::GetBytes(Int32, Int64, array<Byte>[][], Int32, Int32)`

`IDataRecord::GetBytes(Int32, Int64, array<Byte>[][], Int32, Int32)`
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataReader...: GetChar Method

MySqlDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>[GetChar(Int32)]</code></td>
<td>Gets the value of the specified column as a single character.</td>
</tr>
<tr>
<td></td>
<td>(Overrides [DbDataReader.GetChar(Int32)].)</td>
</tr>
<tr>
<td><code>[GetChar(String)]</code></td>
<td>Gets the value of the specified column as a single character.</td>
</tr>
</tbody>
</table>
See Also

MySqlConnection Class
MySqlConnection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a single character.

**Namespace:**  [MySQL.Data.MySqlClient](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override char GetChar(
    int i
)

Visual Basic (Declaration)

Public Overrides Function GetChar ( _
    i As Integer _
) As Char

Visual C++

public:
    virtual wchar_t GetChar(
        int i
    ) override

Parameters

i
    Type: System::::Int32

Return Value

Implements

IDataRecord,...:::GetChar(Int32)
IDataRecord,...:::GetChar(Int32)
See Also

MySQLDataReader Class
GetChar Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a single character.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySql.Data](#) (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public char GetChar(
    string name
)

Visual Basic (Declaration)

Public Function GetChar ( _
    name As String _
) As Char

Visual C++

public: 
wchar_t GetChar(
    String^ name
)

Parameters

name
    Type: System::String

Return Value
See Also

**MySqlDataReader Class**
**GetChar Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlDataReader..::.GetChars Method

MySqlDataReader Class  See Also  Send Feedback

Reads a stream of characters from the specified column offset into the buffer as an array starting at the given buffer offset.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override long GetChars(
    int i,
    long fieldoffset,
    char[] buffer,
    int bufferoffset,
    int length
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function GetChars ( _
    i As Integer, _
    fieldoffset As Long, _
    buffer As Char(), _
    bufferoffset As Integer, _
    length As Integer _
) As Long
```

Visual C++

```cpp
public:
virtual long long GetChars(
    int i,
    long long fieldoffset,
    array<wchar_t>^ buffer,
    int bufferoffset,
    int length
) override
```

Parameters

i
Type: `System::::Int32`

fieldoffset
Type: `System::::Int64`
buffer
  Type: array< System::Char >[]

bufferoffset
  Type: System::Int32

length
  Type: System::Int32

Return Value

Implements

IDataRecord::GetChars(Int32, Int64, array<Char>[][], Int32, Int32)
IDataRecord::GetChars(Int32, Int64, array<Char>[][], Int32, Int32)
See Also

MySQLDataReader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the name of the source data type.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/productsconnector/)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override string GetDataTypeName(
        int i
    )
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function GetDataTypeName ( _
        i As Integer _
    ) As String
```

Visual C++

```cpp
public:
virtual String^ GetDataTypeName(
        int i
    ) override
```

Parameters

i

Type: System::::Int32

Return Value

Implements

IDataRecord:::GetDataTypeName(Int32)
IDataRecord:::GetDataTypeName(Int32)
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlDataReader...::GetDateTime Method

MySqlDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetDateTime(Int32)</code></td>
<td>Gets the value of the specified column as a <code>DateTime</code> object. (Overrides <code>DbDataReader...::GetDateTime(Int32)</code>.)</td>
<td></td>
</tr>
<tr>
<td><code>GetDateTime(String)</code></td>
<td>Gets the value of the specified column as a <code>DateTime</code> object.</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySQLDataReader Class
MySQLDataReader Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a DateTime object.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override DateTime GetDateTime(int i)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function GetDateTime ( _
    i As Integer _
) As DateTime
```

Visual C++

```cpp
public:
virtual DateTime GetDateTime(
    int i
) override
```

Parameters

i

Type: System::::Int32
The zero-based column ordinal.

Return Value

The value of the specified column.

Implements

IDataRecord:::GetDateTime(Int32)
IDataRecord:::GetDateTime(Int32)
Remarks

No conversions are performed; therefore, the data retrieved must already be a DateTime object.

Call IsDBNull to check for null values before calling this method.

✔ Note:

MySQL allows date columns to contain the value '0000-00-00' and datetime columns to contain the value '0000-00-00 00:00:00'. The DateTime structure cannot contain or represent these values. To read a datetime value from a column that might contain zero values, use GetMySqlDateTime(Int32).

The behavior of reading a zero datetime column using this method is defined by the ZeroDateTimeBehavior connection string option. For more information on this option, please refer to ConnectionString.
See Also

 MySqlDataReader Class
 DateTime Overload
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a DateTime object.

**Namespace:** MySql.Data.MySqlClient  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public DateTime GetDateTime(string column)

Visual Basic (Declaration)

Public Function GetDateTime(_
    column As String _
) As DateTime

Visual C++

public:
    DateTime GetDateTime(_
        String^ column
    )

Parameters

column
    Type: System::String
    The column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a DateTime object.

Call IsDBNull to check for null values before calling this method.

 ✓ Note:

MySql allows date columns to contain the value '0000-00-00' and datetime columns to contain the value '0000-00-00 00:00:00'. The DateTime structure cannot contain or represent these values. To read a datetime value from a column that might contain zero values, use GetMySqlDateTime(Int32).

The behavior of reading a zero datetime column using this method is defined by the ZeroDateTimeBehavior connection string option. For more information on this option, please refer to ConnectionString.
See Also

MySqlDataReader Class
GetDateTime Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

**MySqlDataReader**...::GetDecimal Method

**MySqlDataReader Class**  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDecimal(Int32)</td>
<td>Gets the value of the specified column as a Decimal object. (Overrides DbDataReader::GetDecimal(Int32).)</td>
</tr>
<tr>
<td>GetDecimal(String)</td>
<td>Gets the value of the specified column as a Decimal object.</td>
</tr>
</tbody>
</table>
See Also

[MYSQLDataReader Class]
[MYSQLDataReader Members]
[MYSQL.Data.MySqlClient Namespace]

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a `Decimal` object.

**Namespace:**  `[Sqoop.Data.MySqlClient]`  
**Assembly:**  `Sql.Data (in Sql.Data.dll) Version: 6.2.2.0`
Syntax

C#

```csharp
public override decimal GetDecimal(
    int i
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function GetDecimal ( _
    i As Integer _
) As Decimal
```

Visual C++

```cpp
public:
    virtual Decimal GetDecimal(  
        int i
    ) override
```

Parameters

i

Type: System::Int32
The zero-based column ordinal

Return Value

The value of the specified column.

Implements

IDataRecord::GetDecimal(Int32)
IDataRecord::GetDecimal(Int32)
Remarks

No conversions are performed; therefore, the data retrieved must already be a Decimal object.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlDataReader Class**
**GetDecimal Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a `Decimal` object.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C# public decimal GetDecimal(string column) Visual Basic (Declaration) Public Function GetDecimal ( _ column As String _ ) As Decimal Visual C++ public: Decimal GetDecimal( String^ column )

Parameters

column Type: System::String The column name

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a **Decimal** object.

Call IsDBNull to check for null values before calling this method.
See Also

MySqlDataReader Class
GetDecimal Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataReader::GetDouble Method

MySqlDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDouble(Int32)</td>
<td>Gets the value of the specified column as a double-precision floating point number. (Overrides DbDataReader::GetDouble(Int32).)</td>
</tr>
<tr>
<td>GetDouble(String)</td>
<td>Gets the value of the specified column as a double-precision floating point number.</td>
</tr>
</tbody>
</table>
See Also

MySQLDataReader Class
MySQLDataReader Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlDataReader Class
See Also  Send Feedback

Gets the value of the specified column as a double-precision floating point number.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override double GetDouble(
    int i
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function GetDouble ( _
    i As Integer _
) As Double
```

Visual C++

```cpp
public:
virtual double GetDouble(
    int i
) override
```

Parameters

i

Type: System:::Int32
The zero-based column ordinal.

Return Value

The value of the specified column.

Implements

IDataRecord:::GetDouble(Int32)
IDataRecord:::GetDouble(Int32)
Remarks

No conversions are performed; therefore, the data retrieved must already be a **Double** object.

Call IsDBNull to check for null values before calling this method.
See Also

SqlConnection
GetDouble Overload
SqlConnection Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a double-precision floating point number.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public double GetDouble(
    string column
)

Visual Basic (Declaration)

Public Function GetDouble ( _
    column As String _
) As Double

Visual C++

public:
    double GetDouble(
        String^ column
    )

Parameters

column
    Type: System::String
    The column name

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a **Double** object.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlDataReader Class**
**GetDouble Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
_Returns an _IEnumerator_ that iterates through the _MySQLDataReader_.

**Namespace:**  _MySQL.Data.MySqlClient_

**Assembly:**  _MySQL.Data_ (in _MySQL.Data.dll_)  Version: 6.2.2.0
**Syntax**

**C#**

```
public override IEnumerator GetEnumerator()
```

**Visual Basic (Declaration)**

```
Public Overrides Function GetEnumerator As IEnumerator
```

**Visual C++**

```
public: 
virtual GetEnumerator^ GetEnumerator() override
```

**Return Value**

**Implements**

```
IEnumerator..::.GetEnumerator()
```
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlDataReader Class

Gets the Type that is the data type of the object.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override Type GetFieldType(
    int i
)

Visual Basic (Declaration)

Public Overrides Function GetFieldType ( _
    i As Integer _
) As Type

Visual C++

public:
virtual Type^ GetFieldType(
    int i
) override

Parameters

i

Type: System::::Int32

Return Value

Implements

IDataRecord:::GetFieldType(Int32)
IDataRecord:::GetFieldType(Int32)
See Also

**MySqlDataReader Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataReader...:.GetFloat Method

MySqlDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetFloat(Int32)</td>
<td>Gets the value of the specified column as a single-precision floating point number. (Overrides <code>DbDataReader::GetFloat(Int32)</code>.)</td>
</tr>
<tr>
<td>GetFloat(String)</td>
<td>Gets the value of the specified column as a single-precision floating point number.</td>
</tr>
</tbody>
</table>
See Also

MySQLDataReader Class
MySQLDataReader Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic  Visual C++
MySQL Connector/Net
MySqlDataReader...:.GetFloat Method (Int32)
MySqlDataReader Class  See Also  Send Feedback

Gets the value of the specified column as a single-precision floating point number.

**Namespace:**  MySql.Data.MySqlClient
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override float GetFloat(
    int i
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function GetFloat ( _
    i As Integer _
) As Single
```

Visual C++

```cpp
public:
    virtual float GetFloat( 
    int i
) override
```

Parameters

i

Type: `System::::Int32`
The zero-based column ordinal.

Return Value

The value of the specified column.

Implements

`IDataRecord::::GetFloat(Int32)`
`IDataRecord::::GetFloat(Int32)`
Remarks

No conversions are performed; therefore, the data retrieved must already be a Float object.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlDataReader Class**
**GetFloat Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a single-precision floating point number.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public float GetFloat(
    string column
)

Visual Basic (Declaration)

Public Function GetFloat ( _
    column As String _
) As Single

Visual C++

public:
    float GetFloat(
        String^ column
    )

Parameters

column
    Type: System::String
    The column name

Return Value

The value of the specified column.
**Remarks**

No conversions are performed; therefore, the data retrieved must already be a **Float** object.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlDataReader Class**  
**GetFloat Overload**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C# | Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net
MySQLDataReader::GetGuid Method

See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetGuid(Int32)</td>
<td>(Overrides DbDataReader::GetGuid(Int32).)</td>
</tr>
<tr>
<td>GetGuid(String)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlParameter Class
MySqlParameter Members
SqlTransaction Class
SqlTransaction Methods

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic  Visual C++  MySQL Connector/Net  MySqlDataReader...::.GetGuid Method (Int32)  MySqlDataReader Class  See Also  Send Feedback

**Namespace:**  [MySQL.Data.MySqlClient]  
**Assembly:**  MySql.Data (in MySql.Data.dll)  Version: 6.2.2.0
Syntax

C#

public override Guid GetGuid(
    int i
)

Visual Basic (Declaration)

Public Overrides Function GetGuid ( _
    i As Integer _
) As Guid

Visual C++

public:
    virtual Guid GetGuid(
    int i
    ) override

Parameters

i

Type: System:::Int32

Implements

IDataRecord:::GetGuid(Int32)
IDataRecord:::GetGuid(Int32)
See Also

**MySqlParameter Class**
**GetGuid Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataReader...::GetGuid Method (String)

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public Guid GetGuid(
    string column
)

Visual Basic (Declaration)

Public Function GetGuid ( _
    column As String _
) As Guid

Visual C++

public:
    Guid GetGuid(
        String^ column
    )

Parameters

column
    Type: System::String
See Also

MySqlDataReader Class
GetGuid Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataReader::GetInt16 Method

MySqlDataReader Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInt16(Int32)</td>
<td>Gets the value of the specified column as a 16-bit signed integer.</td>
</tr>
<tr>
<td>(Overrides ...GetInt16(Int32).)</td>
<td></td>
</tr>
<tr>
<td>GetInt16(String)</td>
<td>Gets the value of the specified column as a 16-bit signed integer.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 16-bit signed integer.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override short GetInt16(
    int i
)

Visual Basic (Declaration)

Public Overrides Function GetInt16 ( _
    i As Integer _
) As Short

Visual C++

public:
virtual short GetInt16(
    int i
) override

Parameters

i

Type: System:::Int32
The zero-based column ordinal.

Return Value

The value of the specified column.

Implements

IDataRecord:::GetInt16(Int32)
IDataRecord:::GetInt16(Int32)
Remarks

No conversions are performed; therefore, the data retrieved must already be a **16 bit integer** value.

Call IsDBNull to check for null values before calling this method.
See Also

MySqlDataReader Class
GetInt16 Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 16-bit signed integer.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySql.Data](#) (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public short GetInt16(
    string column
)

Visual Basic (Declaration)

Public Function GetInt16 ( _
    column As String _
) As Short

Visual C++

public:
short GetInt16(
    String^ column
)

Parameters

column
    Type: System::String
    The column name

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 16-bit integer value.

Call IsDBNull to check for null values before calling this method.
See Also

MySQLDataReader Class
GetInt16 Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlDataReader Class

See Also
Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetInt32(Int32)</code></td>
<td>Gets the value of the specified column as a 32-bit signed integer.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbDataReader..::.GetInt32(Int32)</code>.)</td>
</tr>
<tr>
<td><code>GetInt32(String)</code></td>
<td>Gets the value of the specified column as a 32-bit signed integer.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 32-bit signed integer.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override int GetInt32(
    int i
)

Visual Basic (Declaration)

Public Overrides Function GetInt32 ( _
    i As Integer _
) As Integer

Visual C++

public:
virtual int GetInt32(
    int i
) override

Parameters

i

Type: System::::Int32
The zero-based column ordinal.

Return Value

The value of the specified column.

Implements

IDataRecord::::GetInt32(Int32)
IDataRecord::::GetInt32(Int32)
Remarks

No conversions are performed; therefore, the data retrieved must already be a 32-bit integer value.

Call IsDBNull to check for null values before calling this method.
See Also

 MySqlDataReader Class
 GetInt32 Overload
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 32-bit signed integer.

**Namespace:**  MySQL.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public int GetInt32(
    string column
)

Visual Basic (Declaration)

Public Function GetInt32 ( _
    column As String _
) As Integer

Visual C++

public:
    int GetInt32(
        String^ column
    )

Parameters

column
    Type: System::String
    The column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 32-bit integer value.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlDataReader Class**
**GetInt32 Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL DataReader....GetInt64 Method

MySQLDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInt64(Int32)</td>
<td>Gets the value of the specified column as a 64-bit signed integer. (Overrides DbDataReader..:::GetInt64(Int32).)</td>
</tr>
<tr>
<td>GetInt64(String)</td>
<td>Gets the value of the specified column as a 64-bit signed integer.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 64-bit signed integer.

**Namespace:**  [MySQL.Data.MySqlClient](https://example.com)

**Assembly:**  [MySql.Data](https://example.com) (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override long GetInt64(
    int i
)

Visual Basic (Declaration)

Public Overrides Function GetInt64 ( _
    i As Integer _) _
) As Long

Visual C++

public:
virtual long long GetInt64( 
    int i
) override

Parameters

i

Type: System::::Int32
The zero-based column ordinal.

Return Value

The value of the specified column.

Implements

IDataRecord::::GetInt64(Int32)
IDataRecord::::GetInt64(Int32)
Remarks

No conversions are performed; therefore, the data retrieved must already be a 64 bit integer value.

Call IsDBNull to check for null values before calling this method.
See Also

MySqlCommand Class
GetInt64 Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 64-bit signed integer.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public long GetInt64(
    string column
)
```

### Visual Basic (Declaration)

```vbnet
Public Function GetInt64 ( _
    column As String _
) As Long
```

### Visual C++

```cpp
public:
long long GetInt64(
    String^ column
)
```

## Parameters

- **column**
  - Type: System::::String
  - The column name.

## Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 64-bit integer value.

Call IsDBNull to check for null values before calling this method.
See Also

MySQLDataReader Class
GetInt64 Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetMySqlDateTime(Int32)</td>
<td>Gets the value of the specified column as a MySqlDateTime object.</td>
</tr>
<tr>
<td>GetMySqlDateTime(String)</td>
<td>Gets the value of the specified column as a MySqlDateTime object.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a `MySQLDateTime` object.

**Namespace:**  `MySQL.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in MySql.Data.dll)  Version: 6.2.2.0
Syntax

C#

public MySqlDateTime GetMySqlDateTime(
    int column
)

Visual Basic (Declaration)

Public Function GetMySqlDateTime ( _
    column As Integer _
) As MySqlDateTime

Visual C++

public:
    MySqlDateTime GetMySqlDateTime(
        int column
    )

Parameters

column
    Type: System::::Int32
    The zero-based column ordinal or column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a DateTime object.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlCommand Class**
**GetMySqlDateTime Overload**
**MySQL.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic  Visual C++  MySQL Connector/Net
MySQLDataReader...:..GetMySqlDateTime Method (String)
MySQLDataReader Class  See Also  Send Feedback

Gets the value of the specified column as a MySQLDateTime object.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlDateTime GetMySqlDateTime(string column)

Visual Basic (Declaration)

Public Function GetMySqlDateTime ( _
    column As String _
) As MySqlDateTime

Visual C++

public:
    MySqlDateTime GetMySqlDateTime(
        String^ column
    )

Parameters

column
    Type: System::String
    The zero-based column ordinal or column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a DateTime object.

Call IsDBNull to check for null values before calling this method.
See Also

MySqlDataReader Class
GetMySqlDateTime Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDataReader...: GetMySqlDecimal Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetMySqlDecimal(Int32)</code></td>
<td></td>
</tr>
<tr>
<td><code>GetMySqlDecimal(String)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL DataReader...::GetMySqlDecimal Method (Int32)

MySQLDataReader Class  See Also  Send Feedback

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlDecimal GetMySqlDecimal(int i)

Visual Basic (Declaration)

Public Function GetMySqlDecimal ( _
    i As Integer _
) As MySqlDecimal

Visual C++

public:
MySqlDecimal GetMySqlDecimal(int i)

Parameters

i

Type: System::::Int32
See Also

MySQLDataReader Class
GetMySqlDecimal Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySQLDataReader...::GetMySqlDecimal Method (String)

MySQLDataReader Class  See Also  Send Feedback

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public MySqlDecimal GetMySqlDecimal(string column)
```

### Visual Basic (Declaration)

```vbnet
Public Function GetMySqlDecimal(column As String) As MySqlDecimal
```

### Visual C++

```c++
public:
MySqlDecimal GetMySqlDecimal(
    String^ column
)
```

## Parameters

column

  Type: System::::String
See Also

MySqlDataReader Class
GetMySqlDecimal Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the name of the specified column.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override string GetName(
    int i
) {
}

Visual Basic (Declaration)

Public Overrides Function GetName ( _
    i As Integer _
) As String

Visual C++

public:
    virtual String^ GetName(
        int i
    ) override

Parameters

i

Type: System::::Int32

Return Value

Implements

IDataRecord:::GetName(Int32)
IDataRecord:::GetName(Int32)
See Also

* [MySqlDataReader Class](#)
* [MySql.Data.MySqlClient Namespace](#)

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDataReader.GetOrdinal Method

**Namespace:**  [MySQL.Data.MySqlClient](https://my.cnki.com.cn/Article_en/CJFDTotal-MyCQ202107224655665.shtml)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0

Gets the column ordinal, given the name of the column.
### Syntax

#### C#

```csharp
public override int GetOrdinal(
    string name
)
```

#### Visual Basic (Declaration)

```vbnet
Public Overrides Function GetOrdinal ( _
    name As String _
) As Integer
```

#### Visual C++

```cpp
public:
virtual int GetOrdinal(
    String^ name
) override
```

### Parameters

- **name**
  - Type: `System::String`

### Return Value

### Implements

- `IDataRecord::GetOrdinal(String)`
- `IDataRecord::GetOrdinal(String)`
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataReader...: GetSByte Method

MySqlDataReader Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSByte(Int32)</td>
<td>Gets the value of the specified column as a sbyte.</td>
</tr>
<tr>
<td>GetSByte(String)</td>
<td>Gets the value of the specified column as a sbyte.</td>
</tr>
</tbody>
</table>
See Also

MySqlParameter Class
MySqlParameter Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLDataReader..::.GetSByte Method (Int32)

Gets the value of the specified column as a sbyte.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public sbyte GetSByte(
    int i
)
```

Visual Basic (Declaration)

```vbnet
Public Function GetSByte ( _
    i As Integer _
) As SByte
```

Visual C++

```cpp
public:
    signed char GetSByte( int i
)
```

Parameters

i

Type: System::::Int32

Return Value
See Also

MySqlDataReader Class
GetSByte Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a sbyte.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public sbyte GetSByte(
    string name
)

Visual Basic (Declaration)

Public Function GetSByte ( _
    name As String _
) As SByte

Visual C++

public:
    signed char GetSByte(
        String^ name
    )

Parameters

name

Type: System::String

Return Value
See Also

MySQLDataReader Class
GetSByte Overload
 MySqlCommand.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns a DataTable that describes the column metadata of the MySqlDataReader.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override DataTable GetSchemaTable()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function GetSchemaTable As DataTable
```

**Visual C++**

```cpp
public:
virtual DataTable^ GetSchemaTable() override
```

**Return Value**

**Implements**

```csharp
IDataReader::GetSchemaTable()
IDataReader::GetSchemaTable()
```
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetString(Int32)</td>
<td>Gets the value of the specified column as a <code>String</code> object. (Overrides <code>DbDataReader::GetString(Int32)</code>.)</td>
</tr>
<tr>
<td>GetString(String)</td>
<td>Gets the value of the specified column as a <code>String</code> object.</td>
</tr>
</tbody>
</table>
See Also

MySQLDataReader Class
MySQLDataReader Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a `String` object.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override string GetString(
  int i
)

Visual Basic (Declaration)

Public Overrides Function GetString (_
  i As Integer _
) As String

Visual C++

public:
virtual String^ GetString(
  int i
) override

Parameters

i

Type: System::::Int32
The zero-based column ordinal.

Return Value

The value of the specified column.

Implements

IDataRecord::::GetString(Int32)
IDataRecord::::GetString(Int32)
Remarks

No conversions are performed; therefore, the data retrieved must already be a `String` object.

Call `IsDBNull` to check for null values before calling this method.
See Also

MySqlDataReader Class
GetString Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a **String** object.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public string GetString(
    string column
)

Visual Basic (Declaration)

Public Function GetString ( _
    column As String _
) As String

Visual C++

public:
String^ GetString(
    String^ column
)

Parameters

column
    Type: System::String
    The column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 
**String** object.

Call IsDBNull to check for null values before calling this method.
See Also

MySqlDataReader Class
GetString Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDataReader...::GetTimeSpan Method

**See Also**

MySQLDataReader Class

**Send Feedback**
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetTimeSpan(Int32)</code></td>
<td>Gets the value of the specified column as a <code>TimeSpan</code> object.</td>
</tr>
<tr>
<td><code>GetTimeSpan(String)</code></td>
<td>Gets the value of the specified column as a <code>TimeSpan</code> object.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a `TimeSpan` object.

**Namespace:**  [MySQL.Data.MySqlClient](#)
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public TimeSpan GetTimeSpan(
    int column
)

Visual Basic (Declaration)

Public Function GetTimeSpan (_
    column As Integer _
) As TimeSpan

Visual C++

public:
    __declspec(dllimport) TimeSpan GetTimeSpan(
        __int32 column
    )

Parameters

column
    Type: System::Int32
    The zero-based column ordinal or column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a *Time* value.

Call IsDBNull to check for null values before calling this method.
See Also

MySqlDataReader Class
GetTimeSpan Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a **TimeSpan** object.

**Namespace:**  [MySQL.Data.MySqlClient](https://github.com/enums/MySQL.Data)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public TimeSpan GetTimeSpan(
    string column
)

Visual Basic (Declaration)

Public Function GetTimeSpan ( _
    column As String _
) As TimeSpan

Visual C++

public:
    TimeSpan GetTimeSpan(
        String^ column
    )

Parameters

column
    Type: System::String
    The zero-based column ordinal or column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a Time value.

Call IsDBNull to check for null values before calling this method.
See Also

MySQLDataReader Class
GetTimeSpan Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDataReader...::GetUInt16 Method

MySQLDataReader Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetUInt16(Int32)</td>
<td>Gets the value of the specified column as a 16-bit unsigned integer.</td>
</tr>
<tr>
<td>GetUInt16(String)</td>
<td>Gets the value of the specified column as a 16-bit unsigned integer.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C# □ Visual Basic
□ Visual C++
MySQL Connector/Net
MySqlDataReader.GetUInt16 Method (Int32)
MySqlDataReader Class See Also Send Feedback

Gets the value of the specified column as a 16-bit unsigned integer.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public ushort GetUInt16(
          int column
)

Visual Basic (Declaration)

Public Function GetUInt16 ( _
            column As Integer _
) As UShort

Visual C++

public:
    unsigned short GetUInt16(
          int column
)

Parameters

column
    Type: System::::Int32
    The zero-based column ordinal or column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 16 bit unsigned integer value.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlDataReader Class**
**GetUInt16 Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 16-bit unsigned integer.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public ushort GetUInt16(
    string column
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function GetUInt16 ( _
    column As String _
) As UShort
```

**Visual C++**

```cpp
public:
    unsigned short GetUInt16(
        String^ column
    )
```

**Parameters**

column

Type: `System::String`

The zero-based column ordinal or column name.

**Return Value**

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 16 bit unsigned integer value.

Call IsDBNull to check for null values before calling this method.
See Also

MySqlDataReader Class
GetUInt16 Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlDataReader...:.GetUInt32 Method

MySqlDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetUInt32(Int32)</td>
<td>Gets the value of the specified column as a 32-bit unsigned integer.</td>
</tr>
<tr>
<td>GetUInt32(String)</td>
<td>Gets the value of the specified column as a 32-bit unsigned integer.</td>
</tr>
</tbody>
</table>
See Also

**MySqlDataReader Class**
**MySqlDataReader Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 32-bit unsigned integer.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public uint GetUInt32(
    int column
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function GetUInt32 ( _
    column As Integer _
) As UInteger
```

**Visual C++**

```cpp
public:
unsigned int GetUInt32(
    int column
)
```

**Parameters**

column

Type: `System::::Int32`
The zero-based column ordinal or column name.

**Return Value**

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 32 bit unsigned integer value.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlDataReader Class**
**GetUInt32 Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 32-bit unsigned integer.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public uint GetUInt32(
    string column
)

Visual Basic (Declaration)

Public Function GetUInt32 ( _
    column As String _
) As UInteger

Visual C++

public:
    unsigned int GetUInt32(
        String^ column
    )

Parameters

column
    Type: System::String
    The zero-based column ordinal or column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 32 bit unsigned integer value.

Call IsDBNull to check for null values before calling this method.
See Also

MySQLDataReader Class
GetUInt32 Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlDataReader...::GetUInt64 Method

MySqlDataReader Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetUInt64(Int32)</td>
<td>Gets the value of the specified column as a 64-bit unsigned integer.</td>
</tr>
<tr>
<td>GetUInt64(String)</td>
<td>Gets the value of the specified column as a 64-bit unsigned integer.</td>
</tr>
</tbody>
</table>
See Also

MySqlDataReader Class
MySqlDataReader Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column as a 64-bit unsigned integer.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public ulong GetUInt64(
    int column
)
```

Visual Basic (Declaration)

```vbnet
Public Function GetUInt64 ( _
    column As Integer _
) As ULong
```

Visual C++

```cpp
public:
    unsigned long long GetUInt64(
        int column
    )
```

Parameters

column

Type: `System::::Int32`

The zero-based column ordinal or column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 64-bit unsigned integer value.

Call IsDBNull to check for null values before calling this method.
See Also

MySqlDataReader Class
GetUInt64 Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlDataReader::GetUInt64 Method (String)

Gets the value of the specified column as a 64-bit unsigned integer.

**Namespace:** MySql.Data.MySqlClient  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public ulong GetUInt64(
    string column
)

Visual Basic (Declaration)

Public Function GetUInt64 ( _
    column As String _
) As ULong

Visual C++

public:
    unsigned long long GetUInt64(
        String^ column
    )

Parameters

column
    Type: System::String
    The zero-based column ordinal or column name.

Return Value

The value of the specified column.
Remarks

No conversions are performed; therefore, the data retrieved must already be a 64 bit unsigned integer value.

Call IsDBNull to check for null values before calling this method.
See Also

**MySqlDataReader Class**

**GetUInt64 Overload**

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of the specified column in its native format.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override Object GetValue(
    int i
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function GetValue( _
    i As Integer _
) As Object
```

Visual C++

```cpp
public:
    virtual Object^ GetValue(
        int i
    ) override
```

Parameters

- **i**
  
  Type: `System::::Int32`

Return Value

Implements

- `IDataRecord::::GetValue(Int32)`
- `IDataRecord::::GetValue(Int32)`
See Also

MySQLDataReader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets all attribute columns in the collection for the current row.

**Namespace:**  [MySQL.Data.MySqlClient](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override int GetValues(
    Object[] values
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function GetValues ( _
    values As Object() _
) As Integer
```

### Visual C++

```cpp
public:
    virtual int GetValues(
        array<Object^>^ values
    ) override
```

## Parameters

- **values**
  - Type: array<`System::Object`>[]

## Return Value

## Implements

- `IDataRecord::GetValues(array<Object>[][])`
- `IDataRecord::GetValues(array<Object>[][])`
See Also

**MySqlDataReader Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a value indicating whether the column contains non-existent or missing values.

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override bool IsDBNull(int i)

Visual Basic (Declaration)

Public Overrides Function IsDBNull(_
   i As Integer _) _
As Boolean

Visual C++

public:
virtual bool IsDBNull(_
   int i _
) override

Parameters

i

Type: System::::Int32

Return Value

Implements

IDataRecord:::IsDBNull(Int32)
IDataRecord:::IsDBNull(Int32)
See Also

MySQLDataReader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDataReader...:MemberwiseClone Method

See Also

Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberwiseClone()</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone(Boolean)</td>
<td>Creates a shallow copy of the current MarshalByRefObject object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from MarshalByRefObject.)</td>
</tr>
</tbody>
</table>
See Also

MySQLDataReader Class
MySQLDataReader Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Advances the data reader to the next result, when reading the results of batch SQL statements.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override bool NextResult()

Visual Basic (Declaration)

Public Overrides Function NextResult As Boolean

Visual C++

public:
virtual bool NextResult() override

ReturnValue

Implements

IDataReader:::NextResult()() postDataReader:::NextResult()()
See Also

MySQLDataReader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Advances the MySqlDataReader to the next record.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override bool Read()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function Read As Boolean
```

**Visual C++**

```cpp
public:
virtual bool Read() override
```

**ReturnValue**

**Implements**

- `IDataReader::Read()`
- `IDataReader::Read()`
See Also

MySQLDataReader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySqlDataReader** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Depth</strong></td>
<td>Gets a value indicating the depth of nesting for the current row. This method is not supported currently and always returns 0.</td>
</tr>
<tr>
<td><strong>FieldCount</strong></td>
<td>Gets the number of columns in the current row.</td>
</tr>
<tr>
<td><strong>HasRows</strong></td>
<td>Gets a value indicating whether the MySqlDataReader contains one or more rows.</td>
</tr>
<tr>
<td><strong>IsClosed</strong></td>
<td>Gets a value indicating whether the data reader is closed.</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>RecordsAffected</strong></td>
<td>Gets the number of rows changed, inserted, or deleted by execution of the SQL statement.</td>
</tr>
<tr>
<td><strong>VisibleFieldCount</strong></td>
<td>Gets the number of fields in the MySqlDataReader that are not hidden.</td>
</tr>
</tbody>
</table>

(Overrides DbDataReader:::Depth.)

(Overrides DbDataReader:::FieldCount.)

(Overrides DbDataReader:::HasRows.)

(Overrides DbDataReader:::IsClosed.)

(Overrides DbDataReader:::RecordsAffected.)

(Inherited from DbDataReader.)
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a value indicating the depth of nesting for the current row. This method is not supported currently and always returns 0.

**Namespace:** [MySql.Data.MySqlClient](https://github.com/mysqlb/connector-net)  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override int Depth { get; }

Visual Basic (Declaration)

Public OverridesReadOnly Property Depth As Integer

Visual C++

public:
virtual property int Depth {
    int get () override;
}

Implements

IDataReader::Depth
IDataReader::Depth
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the number of columns in the current row.

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public override int FieldCount { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides ReadOnly Property FieldCount As Integer
```

**Visual C++**

```cpp
public:
virtual property int FieldCount {
    int get () override;
}
```

### Implements

- `IDataRecord::FieldCount`
- `IDataRecord::<::FieldCount`
See Also

MySQLDataReader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a value indicating whether the MySqlDataReader contains one or more rows.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override bool HasRows { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property HasRows As Boolean

Visual C++

public:
virtual property bool HasRows {
    bool get () override;
}

See Also

MySQLDataReader Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a value indicating whether the data reader is closed.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override bool IsClosed { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property IsClosed As Boolean

Visual C++

public:
    virtual property bool IsClosed {
        bool get () override;
    }

Implements

IDataReader:::IsClosed
IDataReader:::IsClosed
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySQLDataReader...::Item Property

MySQLDataReader Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item[['Int32']]</td>
<td>Overloaded. Gets the value of a column in its native format. In C#, this property is the indexer for the MySqlDataReader class. (Overrides DbDataReader..:::Item[['Int32']].)</td>
</tr>
<tr>
<td>Item[['String']]</td>
<td>Gets the value of a column in its native format. [C#] In C#, this property is the indexer for the MySqlDataReader class. (Overrides DbDataReader..:::Item[['String']].)</td>
</tr>
</tbody>
</table>
See Also

MySQLDataReader Class
MySQLDataReader Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Overloaded. Gets the value of a column in its native format. In C#, this property is the indexer for the MySqlDataReader class.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override Object this[
    int i
] { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Default Property Item ( _
    i As Integer _
) As Object

Visual C++

public:
virtual property Object^ default[int i] { 
    Object^ get (int i) override;
}

Parameters

i
    Type: System:::Int32

Implements

IDataRecord:::Item[{{Int32}}]
IDataRecord:::Item[{{Int32}}]
See Also

MySqlDataReader Class
Item Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the value of a column in its native format. [C#] In C#, this property is the indexer for the MySqlDataReader class.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override Object this[
    string name
] { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Default Property Item ( _
    name As String _
) As Object

Visual C++

public:
    virtual property Object^ default[String^ name] {
        Object^ get (String^ name) override;
    }

Parameters

name
    Type: System:::String

Implements

IDataRecord:::Item[](String])
IDataRecord:::Item[](String])
See Also

MySQLDataReader Class
Item Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the number of rows changed, inserted, or deleted by execution of the SQL statement.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll)  Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override int RecordsAffected { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides ReadOnly Property RecordsAffected As Integer
```

**Visual C++**

```cpp
public:
virtual property int RecordsAffected {
    int get () override;
}
```

**Implements**

- `IDataReader::RecordsAffected`
- `IDataReader::RecordsAffected`
See Also

MySqlDataReader Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Specifies MySQL specific data type of a field, property, for use in a

**MySqlParameter**.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public enum MySqlDbType

Visual Basic (Declaration)

Public Enumeration MySqlDbType

Visual C++

public enum class MySqlDbType
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal</td>
<td>A fixed precision and scale numeric value between -1038 (-1) and 1038 (-1).</td>
</tr>
<tr>
<td>Byte</td>
<td>The signed range is -128 to 127. The unsigned range is 0 to 255.</td>
</tr>
<tr>
<td>Int16</td>
<td>A 16-bit signed integer. The signed range is -32768 to 32767. The unsigned range is 0 to 65535.</td>
</tr>
<tr>
<td>Int24</td>
<td>Specifies a 24 (3 byte) signed or unsigned value.</td>
</tr>
<tr>
<td>Int32</td>
<td>A 32-bit signed integer</td>
</tr>
<tr>
<td>Int64</td>
<td>A 64-bit signed integer.</td>
</tr>
<tr>
<td>Single</td>
<td>A small (single-precision) floating-point number. Allowable values are -3.402823466E+38 to -1.175494351E-38, 0, and 1.175494351E-38 to 3.402823466E+38.</td>
</tr>
<tr>
<td>Float</td>
<td>Double</td>
</tr>
<tr>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Double</td>
<td>A normal-size (double-precision) floating-point number. Allowable values are -1.7976931348623157E+308 to -2.2250738585072014E-308, 0, and 2.2250738585072014E-308 to 1.7976931348623157E+308.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>A timestamp. The range is '1970-01-01 00:00:00' to sometime in the year 2037</td>
</tr>
<tr>
<td>Date</td>
<td>Date The supported range is '1000-01-01' to '9999-12-31'.</td>
</tr>
<tr>
<td>Time</td>
<td>The range is '-838:59:59' to '838:59:59'.</td>
</tr>
<tr>
<td>DateTime</td>
<td>DateTime The supported range is '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. <strong>Obsolete.</strong></td>
</tr>
<tr>
<td>Datetime</td>
<td>Datetime The supported range is '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.</td>
</tr>
<tr>
<td>Year</td>
<td>A year in 2- or 4-digit format (default is 4-digit). The allowable values are 1901 to 2155, 0000 in the 4-digit year format, and 1970-2069 if you use the 2-digit format (70-69).</td>
</tr>
<tr>
<td>NewDate</td>
<td><strong>Obsolete</strong> Use Datetime or Date type</td>
</tr>
<tr>
<td>VarString</td>
<td>A variable-length string containing 0 to 65535 characters</td>
</tr>
<tr>
<td>Bit</td>
<td>Bit-field data type</td>
</tr>
<tr>
<td>NewDecimal</td>
<td>New Decimal</td>
</tr>
<tr>
<td>Enum</td>
<td>An enumeration. A string object that can have only one value, chosen from the list of values 'value1', 'value2',..., NULL or the special &quot;&quot; error value. An ENUM can have a maximum of 65535 distinct values</td>
</tr>
<tr>
<td>Set</td>
<td>A set. A string object that can have zero or more values, each of which must be chosen from the list of values 'value1', 'value2',... A SET can have a maximum of 64 members.</td>
</tr>
<tr>
<td>TinyBlob</td>
<td>A binary column with a maximum length of 255 (2^8 - 1) characters</td>
</tr>
<tr>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>MediumBlob</strong></td>
<td>A binary column with a maximum length of 16777215 (2^24 - 1) bytes.</td>
</tr>
<tr>
<td><strong>LongBlob</strong></td>
<td>A binary column with a maximum length of 4294967295 or 4G (2^32 - 1) bytes.</td>
</tr>
<tr>
<td><strong>Blob</strong></td>
<td>A binary column with a maximum length of 65535 (2^16 - 1) bytes.</td>
</tr>
<tr>
<td><strong>VarChar</strong></td>
<td>A variable-length string containing 0 to 255 bytes.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td>A fixed-length string.</td>
</tr>
<tr>
<td><strong>Geometry</strong></td>
<td>Geometric (GIS) data type.</td>
</tr>
<tr>
<td><strong>UByte</strong></td>
<td>Unsigned 8-bit value.</td>
</tr>
<tr>
<td><strong>UInt16</strong></td>
<td>Unsigned 16-bit value.</td>
</tr>
<tr>
<td><strong>UInt24</strong></td>
<td>Unsigned 24-bit value.</td>
</tr>
<tr>
<td><strong>UInt32</strong></td>
<td>Unsigned 32-bit value.</td>
</tr>
<tr>
<td><strong>UInt64</strong></td>
<td>Unsigned 64-bit value.</td>
</tr>
<tr>
<td><strong>Binary</strong></td>
<td>Fixed length binary string.</td>
</tr>
<tr>
<td><strong>VarBinary</strong></td>
<td>Variable length binary string.</td>
</tr>
<tr>
<td><strong>TinyText</strong></td>
<td>A text column with a maximum length of 255 (2^8 - 1) characters.</td>
</tr>
<tr>
<td><strong>MediumText</strong></td>
<td>A text column with a maximum length of 16777215 (2^24 - 1) characters.</td>
</tr>
<tr>
<td><strong>LongText</strong></td>
<td>A text column with a maximum length of 4294967295 or 4G (2^32 - 1) characters.</td>
</tr>
<tr>
<td><strong>Text</strong></td>
<td>A text column with a maximum length of 65535 (2^16 - 1) characters.</td>
</tr>
<tr>
<td><strong>Guid</strong></td>
<td>A guid column</td>
</tr>
</tbody>
</table>
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Specifies the connection types supported

**Namespace:** [MySQL.Data.MySqlClient](#)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public enum MySqlDriverType

Visual Basic (Declaration)

Public Enumeration MySqlDriverType

Visual C++

public enum class MySqlDriverType
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native</td>
<td>Use TCP/IP sockets.</td>
</tr>
<tr>
<td>Client</td>
<td>Use client library.</td>
</tr>
<tr>
<td>Embedded</td>
<td>Use MySQL embedded server.</td>
</tr>
</tbody>
</table>
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Collection of error codes that can be returned by the server

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysqlconnector.net)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public class MySqlError

Visual Basic (Declaration)

Public Class MySqlError

Visual C++

public ref class MySqlError
Inheritance Hierarchy

System..::.Object

MySql.Data.MySqlClient..::.MySqlError
See Also

MySQLError Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLError** type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQLError</td>
<td></td>
</tr>
</tbody>
</table>
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Error code</td>
</tr>
<tr>
<td>Level</td>
<td>Error level</td>
</tr>
<tr>
<td>Message</td>
<td>Error message</td>
</tr>
</tbody>
</table>
See Also

**MySqlError Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlError(
    string level,
    int code,
    string message
)

Visual Basic (Declaration)

Public Sub New (_
    level As String, _
    code As Integer, _
    message As String _
)

Visual C++

public:
MySqlError(
    String^ level,
    int code,
    String^ message
)

Parameters

level
    Type: System:::String

code
    Type: System:::Int32

message
    Type: System:::String
See Also

MySQLError Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlError` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySQLError Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlError` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Error code</td>
</tr>
<tr>
<td>Level</td>
<td>Error level</td>
</tr>
<tr>
<td>Message</td>
<td>Error message</td>
</tr>
</tbody>
</table>
See Also

MySQLError Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Error code

**Namespace:**  [MySQL.Data.MySqlClient](https://docs.microsoft.com/en-us/mysqlconnector/)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Code { get; }

Visual Basic (Declaration)

Public ReadOnly Property Code As Integer

Visual C++

public:
property int Code {
    int get();
}

See Also

MySqlError Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Error and Level Property

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string Level { get; }

Visual Basic (Declaration)

Public ReadOnly Property Level As String

Visual C++

public:
property String^ Level {
    String^ get ();
}
See Also

[link to MySqlError Class]
[link to MySql.Data.MySqlClient Namespace]

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Error message

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string Message { get; }

Visual Basic (Declaration)

Public ReadOnly Property Message As String

Visual C++

public:
property String^ Message {
    String^ get ();
}


See Also

MySQLError Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Provides a reference to error codes returned by MySQL.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
## Syntax

**C#**

    public enum MySqlErrorCode

**Visual Basic (Declaration)**

    Public Enumeration MySqlErrorCode

**Visual C++**

    public enum class MySqlErrorCode
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
<tr>
<td>DuplicateKey</td>
<td>There is already a key with the given values.</td>
</tr>
<tr>
<td>KeyNotFound</td>
<td>The specified key was not found.</td>
</tr>
<tr>
<td>UnableToConnectToHost</td>
<td>Given when the connection is unable to successfully connect to host.</td>
</tr>
<tr>
<td>AccessDenied</td>
<td>Normally returned when an incorrect password is given.</td>
</tr>
<tr>
<td>UnknownDatabase</td>
<td></td>
</tr>
<tr>
<td>DuplicateKeyName</td>
<td>Duplicate Key Name</td>
</tr>
<tr>
<td>DuplicateKeyEntry</td>
<td>Duplicate Key Entry</td>
</tr>
<tr>
<td>HostNotPrivileged</td>
<td>The given host is not allowed to connect</td>
</tr>
<tr>
<td>AnonymousUser</td>
<td>The anonymous user is not allowed to connect</td>
</tr>
<tr>
<td>PasswordNotAllowed</td>
<td>The given password is not allowed</td>
</tr>
<tr>
<td>PasswordNoMatch</td>
<td>The given password does not match</td>
</tr>
<tr>
<td>TableAccessDenied</td>
<td></td>
</tr>
<tr>
<td>ColumnAccessDenied</td>
<td></td>
</tr>
<tr>
<td>IllegalGrantForTable</td>
<td></td>
</tr>
<tr>
<td>NoSuchTable</td>
<td></td>
</tr>
<tr>
<td>NonExistingTableGrant</td>
<td></td>
</tr>
<tr>
<td>PacketTooLarge</td>
<td>An attempt was made to send or receive a packet larger than max_allowed_packet_size</td>
</tr>
</tbody>
</table>
See Also

**MySqlCommand**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The exception that is thrown when MySQL returns an error. This class cannot be inherited.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

[SerializableAttribute]
public sealed class MySqlException : DbException

Visual Basic (Declaration)

<SerializableAttribute>_Public NotInheritable Class MySqlException _
    Inherits DbException

Visual C++

[SerializableAttribute]
public ref class MySqlException sealed : public DbException
Remarks

This class is created whenever the MySql Data Provider encounters an error generated from the server.

Any open connections are not automatically closed when an exception is thrown. If the client application determines that the exception is fatal, it should close any open MySqlDataReader objects or MySqlConnection objects.
Examples

The following example generates a MySqlException due to a missing server, and then displays the exception.

**VB.NET**

```
Public Sub ShowException()
Dim mySelectQuery As String = "SELECT column1 FROM table1"
Dim myConnection As New MySqlConnection ("Data Source=localhost;Database=Sample;"
Dim myCommand As New MySqlCommand(mySelectQuery, myConnection)

Try
    myCommand.Connection.Open()
Catch e As MySqlException
    MessageBox.Show( e.Message )
End Try
End Sub
```

**C#**

```
public void ShowException()
{
    string mySelectQuery = "SELECT column1 FROM table1";
    MySqlConnection myConnection =
    new MySqlConnection("Data Source=localhost;Database=Sample;");
    MySqlCommand myCommand = new MySqlCommand(mySelectQuery,myConnection)

    try
    {
        myCommand.Connection.Open();
    }
    catch (MySqlException e)
    {
        MessageBox.Show( e.Message );
    }
}
```
Inheritance Hierarchy

System..::.Object
   System..::.Exception
      System..::.SystemException
         System.Runtime.InteropServices..::.ExternalException
            System.Data.Common..::.DbException
               MySql.Data.MySqlClient..::.MySqlException
See Also

MySqlException Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlException` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <strong>SerializationInfo</strong> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.) Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>ErrorCode</strong></td>
<td>Gets the HRESULT of the error. (Inherited from ExternalException.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HResult</strong></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Number</strong></td>
<td>Gets a number that identifies the type of error.</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

MySqlException Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlException` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Equals**    | Determines whether the specified [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object) is equal to the current [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object).
|               | (Inherited from [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object).) |
| **Finalize**  | Allows an [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object) to attempt to free resources and perform other cleanup operations before the [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object) is reclaimed by garbage collection.
|               | (Inherited from [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object).) |
| **GetBaseException** | When overridden in a derived class, returns the [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception) that is the root cause of one or more subsequent exceptions. |
|               | (Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| **GetHashCode** | Serves as a hash function for a particular type. When overridden in a derived class, sets the [SerializationInfo](https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo) with information about the exception. |
|               | (Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| **GetObjectData** | Gets the runtime type of the current instance. |
|               | (Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| **GetType**   | Creates a shallow copy of the current [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object). |
|               | (Inherited from [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object).) |
| **MemberwiseClone** | Creates and returns a string representation of the current exception. |
|               | (Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| **ToString**  |                                                                                   |
|               |                                                                                   |
See Also

MySqlException Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>ErrorCode</strong></td>
<td>Gets the HRESULT of the error. (Inherited from <strong>ExternalException</strong>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>HResult</strong></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <strong>Exception</strong> instance that caused the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>Number</strong></td>
<td>Gets a number that identifies the type of error.</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
</tbody>
</table>
See Also

*MySqlException Class*
*MySql.Data.MySqlClient Namespace*

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a number that identifies the type of error.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Number { get; }

Visual Basic (Declaration)

Public ReadOnly Property Number As Integer

Visual C++

public:
property int Number {
    int get ();
}

See Also

MySqlException Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLHelper Class

Members  See Also  Send Feedback

Helper class that makes it easier to work with the provider.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#
public sealed class MySqlHelper

Visual Basic (Declaration)
Public NotInheritable Class MySqlHelper

Visual C++
public ref class MySqlHelper sealed
Inheritance Hierarchy

System..:::Object
MySQL.Data.MySqlClient..:::MySqlHelper
See Also

**MySqlHelper Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySqlHelper** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DoubleQuoteString</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Escapes the string. Executes a single SQL command and returns the first row of the resultset. A new MySqlConnection object is created, opened, and closed during this method.</td>
</tr>
<tr>
<td><strong>EscapeString</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ExecuteDataRow</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ExecuteDataset</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ExecuteNonQuery</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ExecuteReader</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ExecuteScalar</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>UpdateDataSet</strong></td>
<td>Updates the given table with data from the given <code>DataSet</code></td>
</tr>
</tbody>
</table>
See Also

*MySQLHelper Class*
*MySQL.Data.MySqlClient Namespace*

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlHelper` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoubleQuoteString</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Escapes the string.</td>
</tr>
<tr>
<td>EscapeString</td>
<td>Executes a single SQL command and returns the first row of the resultset. A new MySqlConnection object is created, opened, and closed during this method.</td>
</tr>
<tr>
<td>ExecuteDataRow</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExecuteDataset</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExecuteNonQuery</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExecuteReader</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>UpdateDataSet</td>
<td>Updates the given table with data from the given DataSet.</td>
</tr>
</tbody>
</table>
See Also

 MySqlHelper Class
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

public static string DoubleQuoteString(  
    string value
)

**Visual Basic (Declaration)**

Public Shared Function DoubleQuoteString (  
    value As String
) As String

**Visual C++**

public:
static string^ DoubleQuoteString(  
    string^ value
)

### Parameters

value
  
  Type: System::String
See Also

MySqlHelper Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Escapes the string.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static string EscapeString( 
    string value
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function EscapeString ( _
    value As String _
) As String
```

**Visual C++**

```cpp
public:
static String^ EscapeString( 
    String^ value
)
```

**Parameters**

`value`

Type: `System::String`

The string to escape

**Return Value**

The string with all quotes escaped.
See Also

MySqlCommand Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes a single SQL command and returns the first row of the resultset. A new MySqlConnection object is created, opened, and closed during this method.

**Namespace:** MySql.Data.MySqlClient  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public static DataRow ExecuteDataRow(
    string connectionString,
    string commandText,
    params SqlParameter[] parms)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function ExecuteDataRow (_
    connectionString As String, _
    commandText As String, _
    ParamArray parms As SqlParameter() _
) As DataRow
```

**Visual C++**

```cpp
public:
static DataRow^ ExecuteDataRow(
    String^ connectionString, _
    String^ commandText, _
    ... array<MySqlParameter^>^ parms
)
```

### Parameters

**connectionString**
Type: `System::::String`
Settings to be used for the connection

**commandText**
Type: `System::::String`
Command to execute

**parms**
Type: `array<MySQL.Data.MySqlClient::::MySqlParameter>[]()[[]]`
Parameters to use for the command

Return Value

DataRow containing the first row of the resultset
See Also

MySQLHelper Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net
MySqlHelper...::ExecuteDataset Method

 MySQLHelper Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecuteDataset(MySqlConnection, String)</td>
<td>Executes a single SQL command and returns the resultset in a <strong>DataSet</strong>. The state of the <strong>MySqlConnection</strong> object remains unchanged after execution of this method.</td>
</tr>
<tr>
<td>ExecuteDataset(String, String)</td>
<td>Executes a single SQL command and returns the resultset in a <strong>DataSet</strong>. A new <strong>MySqlConnection</strong> object is created, opened, and closed during this method.</td>
</tr>
<tr>
<td>ExecuteDataset(MySqlConnection, String, array&lt;MySqlParameter&gt;[] []])</td>
<td>Executes a single SQL command and returns the resultset in a <strong>DataSet</strong>. The state of the <strong>MySqlConnection</strong> object remains unchanged after execution of this method.</td>
</tr>
<tr>
<td>ExecuteDataset(String, String, array&lt;MySqlParameter&gt;[] []])</td>
<td>Executes a single SQL command and returns the resultset in a <strong>DataSet</strong>. A new <strong>MySqlConnection</strong> object is created, opened, and closed during this method.</td>
</tr>
</tbody>
</table>
See Also

MySQLHelper Class
MySQLHelper Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes a single SQL command and returns the resultset in a `DataSet`. The state of the `MySQLConnection` object remains unchanged after execution of this method.

**Namespace: **`MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
C#

```csharp
public static DataSet ExecuteDataset(
    MySqlConnection connection,
    string commandText
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function ExecuteDataset (_
    connection As MySqlConnection, _
    commandText As String _
) As DataSet
```

Visual C++

```cpp
public:
static DataSet^ ExecuteDataset(
    MySqlConnection^ connection, 
    String^ commandText
)
```

Parameters

- **connection**
  - Type: **System::::SqlConnection** object to use

- **commandText**
  - Type: **System::::String**
  - Command to execute

Return Value

**DataSet** containing the resultset
See Also

SqlConnection Class
ExecuteDataset Overload
SqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes a single SQL command and returns the resultset in a `DataSet`. A new `MySqlConnection` object is created, opened, and closed during this method.

**Namespace:** [MySQL.Data.MySqlClient](https://www.red-gate.com/technet/articles/data-access/using-the-mysql-helper-class/)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static DataSet ExecuteDataset(
    string connectionString,
    string commandText
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function ExecuteDataset ( _
    connectionString As String, _
    commandText As String _
) As DataSet
```

**Visual C++**

```cpp
public:
static DataSet^ ExecuteDataset(  
    String^ connectionString,  
    String^ commandText
)
```

**Parameters**

- **connectionString**
  - Type: `System::::String`
  - Settings to be used for the connection

- **commandText**
  - Type: `System::::String`
  - Command to execute

**Return Value**

- `DataSet` containing the resultset
See Also

**MySqlHelper Class**  
**ExecuteDataset Overload**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes a single SQL command and returns the resultset in a **DataSet**. The state of the **MySQLConnection** object remains unchanged after execution of this method.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static DataSet ExecuteDataset(
    MySqlConnection connection,
    string commandText,
    params SqlParameter[] commandParameters
)
``` 

Visual Basic (Declaration)

```vbnet
Public Shared Function ExecuteDataset ( _
    connection As MySqlConnection, _
    commandText As String, _
    ParamArray commandParameters As SqlParameter() _
) As DataSet
``` 

Visual C++

```cpp
public:
static DataSet^ ExecuteDataset( 
    MySqlConnection^ connection, 
    String^ commandText,  
    ... array<SqlParameter^>^ commandParameters 
)
``` 

Parameters

connection
Type: MySql.Data.MySqlClient::MySqlConnection
MySQLConnection object to use

commandText
Type: System::String
Command to execute

commandParameters
Type: array< MySql.Data.MySqlClient::::MySqlParameter >[](0)
Parameters to use for the command

Return Value

**DataSet** containing the resultset
See Also

MySQLHelper Class
ExecuteDataset Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes a single SQL command and returns the resultset in a `DataSet`. A new `MySqlConnection` object is created, opened, and closed during this method.

**Namespace:** [MySql.Data.MySqlClient](https://www.mysql.com)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public static DataSet ExecuteDataset(
    string connectionString,
    string commandText,
    params MySqlParameter[] commandParameters
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function ExecuteDataset (_
    connectionString As String, _
    commandText As String, _
    ParamArray commandParameters As MySqlParameter() _
) As DataSet
```

### Visual C++

```cpp
public:
static DataSet^ ExecuteDataset(
    String^ connectionString, String^ commandText,
    ... array<MySqlParameter^>^ commandParameters
)
```

## Parameters

**connectionString**
- Type: `System::String`
- Settings to be used for the connection

**commandText**
- Type: `System::String`
- Command to execute

**commandParameters**
- Type: `array< MySql.Data.MySqlClient::MySqlParameter >::[]`
Parameters to use for the command

**Return Value**

*DataSet* containing the resultset
See Also

**MySqlHelper Class**
**ExecuteDataset Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySqlHelper...::ExecuteNonQuery Method

MySQLHelper Class  See Also  Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecuteNonQuery(MySqlConnection, String, array&lt;MySqlParameter[]&gt;[])</td>
<td>Executes a single command against a MySQL database. The MySqlConnection is assumed to be open when the method is called and remains open after the method completes.</td>
</tr>
<tr>
<td>ExecuteNonQuery(String, String, array&lt;MySqlParameter[]&gt;[])</td>
<td>Executes a single command against a MySQL database. A new MySqlConnection is created using the ConnectionString given.</td>
</tr>
</tbody>
</table>
See Also

[MySqlHelper Class](#)
[MySqlHelper Members](#)
[MySql.Data.MySqlClient Namespace](#)

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
**MySqlHelper..::.ExecuteNonQuery Method (MySqlConnection, String, array<MySqlParameter>[][])**

**MySqlHelper Class  See Also  Send Feedback**

Executes a single command against a MySQL database. The **MySQLConnection** is assumed to be open when the method is called and remains open after the method completes.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static int ExecuteNonQuery(
    MySqlConnection connection,
    string commandText,
    params SqlParameter[] commandParameters
)

Visual Basic (Declaration)

Public Shared Function ExecuteNonQuery ( _
    connection As MySqlConnection, _
    commandText As String, _
    ParamArray commandParameters As SqlParameter() _
) As Integer

Visual C++

public:
static int ExecuteNonQuery(
    MySqlConnection^ connection,
    String^ commandText,
    ... array<SqlDbType>^ commandParameters
)

Parameters

connection
    Type: MySql.Data.MySqlClient::MySqlConnection
    MySqlConnection object to use

commandText
    Type: System::String
    SQL command to be executed

commandParameters
    Type: array<MySql.Data.MySqlClient::SqlParameter>[]
Array of `MySqlParameter` objects to use with the command.

Return Value
See Also

**MySqlHelper Class**  
**ExecuteNonQuery Overload**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Helper Class

ExecuteNonQuery Method (String, String, array<MySqlParameter>[])[]()

Executes a single command against a MySQL database. A new MySQLConnection is created using the ConnectionString given.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static int ExecuteNonQuery(
    string connectionString,
    string commandText,
    params MySqlParameter[] parms
)

Visual Basic (Declaration)

Public Shared Function ExecuteNonQuery ( _
    connectionString As String, _
    commandText As String, _
    ParamArray parms As MySqlParameter() _
) As Integer

Visual C++

public:
static int ExecuteNonQuery(
    String^ connectionString,
    String^ commandText,
    ... array<MySqlParameter^>^ parms
)

Parameters

connectionString
Type: System:::String
ConnectionString to use

commandText
Type: System:::String
SQL command to be executed

parms
Type: array< MySql.Data.MySqlClient:::MySqlParameter >[]()}
Array of `MySqlParameter` objects to use with the command.

**Return Value**
See Also

MySqlHelper Class
ExecuteNonQuery Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net
```
MySqlHelper..:::ExecuteReader Method
```
MySQLHelper Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecuteReader(String, String)</td>
<td>Executes a single command against a MySQL database.</td>
</tr>
<tr>
<td>ExecuteReader(String, String, array&lt;SqlParameter&gt;[][])</td>
<td>Executes a single command against a MySQL database.</td>
</tr>
</tbody>
</table>
See Also

MySQLHelper Class
MySQLHelper Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes a single command against a MySQL database.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static MySqlDataReader ExecuteReader(
    string connectionString,
    string commandText
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function ExecuteReader ( _
    connectionString As String, _
    commandText As String _
) As MySqlDataReader
```

**Visual C++**

```cpp
public:
    static MySqlDataReader^ ExecuteReader(
        String^ connectionString, 
        String^ commandText 
    )
```

**Parameters**

**connectionString**
Type: `System::String`
Settings to use for this command

**commandText**
Type: `System::String`
Command text to use

**Return Value**

`MySqlDataReader` object ready to read the results of the command
See Also

MySQLHelper Class
ExecuteReader Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes a single command against a MySQL database.

**Namespace**:  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly**:  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static MySqlDataReader ExecuteReader(
    string connectionString,
    string commandText,
    params SqlParameter[] commandParameters
)

Visual Basic (Declaration)

Public Shared Function ExecuteReader (_
    connectionString As String, _
    commandText As String, _
    ParamArray commandParameters As MySqlParameter() _
) As MySqlDataReader

Visual C++

public:
static MySqlDataReader^ ExecuteReader(
    String^ connectionString, 
    String^ commandText, 
    ... array<MySqlParameter^>^ commandParameters
)

Parameters

collectionString
    Type: System::String
    Settings to use for this command

commandText
    Type: System::String
    Command text to use

commandParameters
    Type: array< MySql.Data.MySqlClient::MySqlParameter >[]()[]
Array of `MySqlParameter` objects to use with the command

**Return Value**

`MySqlDataReader` object ready to read the results of the command
See Also

MYSQLHelper Class
ExecuteReader Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlHelper...::ExecuteScalar Method

MySqlHelper Class  See Also  Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ExecuteScalar(MySqlConnection, String)</code></td>
<td>Execute a single command against a MySQL database.</td>
</tr>
<tr>
<td><code>ExecuteScalar(String, String)</code></td>
<td>Execute a single command against a MySQL database.</td>
</tr>
<tr>
<td><code>ExecuteScalar(MySqlConnection, String, array&lt;MySqlParameter&gt;[][])</code></td>
<td>Execute a single command against a MySQL database.</td>
</tr>
<tr>
<td><code>ExecuteScalar(String, String, array&lt;MySqlParameter&gt;[][])</code></td>
<td>Execute a single command against a MySQL database.</td>
</tr>
</tbody>
</table>
See Also

MySqlHelper Class
MySqlHelper Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLHelper...::ExecuteScalar Method (MySqlConnection, String)

MySQLHelper Class  See Also  Send Feedback

Execute a single command against a MySQL database.

**Namespace:**  **MySQL.Data.MySqlClient**

**Assembly:**  **MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0**
Syntax

C#

public static Object ExecuteScalar(
    MySqlConnection connection,
    string commandText
)

Visual Basic (Declaration)

Public Shared Function ExecuteScalar ( _
    connection As MySqlCommand, _
    commandText As String _
) As Object

Visual C++

public:
static Object^ ExecuteScalar(
    MySqlConnection^ connection,
    String^ commandText
)

Parameters

collection
    Type: MySql.Data.MySqlClient::MySqlConnection
    MySqlConnection object to use

commandText
    Type: System::String
    Command text to use for the command

Return Value

The first column of the first row in the result set, or a null reference if the result set is empty.
See Also

**MySqlHelper Class**
**ExecuteScalar Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Execute a single command against a MySQL database.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public static Object ExecuteScalar(
    string connectionString,
    string commandText
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function ExecuteScalar (_
    connectionString As String, _
    commandText As String _
) As Object
```

### Visual C++

```cpp
public:
static Object^ ExecuteScalar(
    String^ connectionString, 
    String^ commandText
)
```

## Parameters

### connectionString
- Type: `System::String`
- Settings to use for the update

### commandText
- Type: `System::String`
- Command text to use for the update

## Return Value

The first column of the first row in the result set, or a null reference if the result set is empty.
See Also

MySQLHelper Class
ExecuteScalar Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Execute a single command against a MySQL database.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.net)

**Assembly:**  [MySQL.Data](https://www.mysql.net) (in MySQL.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public static Object ExecuteScalar(
    MySqlConnection connection,
    string commandText,
    params SqlParameter[] commandParameters
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function ExecuteScalar ( _
    connection As MySqlConnection, _
    commandText As String, _
    ParamArray commandParameters As SqlParameter() _
) As Object
```

**Visual C++**

```cpp
public:
static Object^ ExecuteScalar(
    MySqlConnection^ connection,
    String^ commandText,
    ... array<MySqlParameter^>^ commandParameters
)
```

### Parameters

- **connection**
  - Type: `MySql.Data.MySqlClient::MySqlConnection` object to use

- **commandText**
  - Type: `System::String` Command text to use for the command

- **commandParameters**
  - Type: `array<MySql.Data.MySqlClient::MySqlParameter>[]`
Parameters to use for the command

**Return Value**

The first column of the first row in the result set, or a null reference if the result set is empty.
See Also

**MySqlHelper Class**
**ExecuteScalar Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Execute a single command against a MySQL database.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static Object ExecuteScalar(
    string connectionString,
    string commandText,
    params SqlParameter[] commandParameters
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function ExecuteScalar ( _
    connectionString As String, _
    commandText As String, _
    ParamArray commandParameters As SqlParameter() _
) As Object
```

**Visual C++**

```cpp
public:
static Object^ ExecuteScalar(
    String^ connectionString, 
    String^ commandText, 
    ... array<MySQLParameter^>^ commandParameters
)
```

**Parameters**

- **connectionString**
  - Type: System::.String
  - Settings to use for the command

- **commandText**
  - Type: System::.String
  - Command text to use for the command

- **commandParameters**
  - Type: array<MySQL.Data.MySqlClient::.SqlParameter>[]()[]
Parameters to use for the command

**Return Value**

The first column of the first row in the result set, or a null reference if the result set is empty.
See Also

**MySqlHelper Class**
**ExecuteScalar Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Updates the given table with data from the given `DataSet`

**Namespace:**  [MySql.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public static void UpdateDataSet(
    string connectionString,
    string commandText,
    DataSet ds,
    string tablename
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Sub UpdateDataSet (_
    connectionString As String, _
    commandText As String, _
    ds As DataSet, _
    tablename As String _
)
```

**Visual C++**

```cpp
public:
  static void UpdateDataSet(
    String^ connectionString,
    String^ commandText,
    DataSet^ ds,
    String^ tablename
  )
```

### Parameters

**connectionString**  
Type: `System::String`  
Settings to use for the update

**commandText**  
Type: `System::String`  
Command text to use for the update
ds
  Type: System.Data:::DataSet
  DataSet containing the new data to use in the update
tablename
  Type: System:::String
  Tablename in the dataset to update
See Also

MySQLHelper Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Provides data for the InfoMessage event. This class cannot be inherited.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll)  
**Version:**  6.2.2.0
Syntax

C#

public class MySqlInfoMessageEventArgs : EventArgs

Visual Basic (Declaration)

Public Class MySqlInfoMessageEventArgs_
    Inherits EventArgs

Visual C++

public ref class MySqlInfoMessageEventArgs : public EventArgs
Inheritance Hierarchy

[System,::,Object]
[System,::,EventArgs]
See Also

MySQLInfoMessageEventArgs Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The MySqlInfoMessageEventArgs type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="MySqlInfoMessageEventArgs" /></td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>errors</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlInfoMessageEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public MySqlInfoMessageEventArgs()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
MySqlInfoMessageEventArgs()
```
See Also

MySQLInfoMessageEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlInfoMessageEventArgs` type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌐 errors</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySQLInfoMessageEventArgs Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL InfoMessageEventArgs::errors Field

MySQLInfoMessageEventArgs Class  See Also  Send Feedback

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlError[] errors

Visual Basic (Declaration)

Public errors As MySqlError()

Visual C++

public:
array<MySqlError>^ errors
See Also

MySqlInfoMessageEventArgs Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The MySQLInfoMessageEventArgs type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

MySqlInfoMessageEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents the method that will handle the `InfoMessage` event of a `MySqlConnection`.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  [MySQLData](#) (in MySQL.Data.dll)  
**Version:**  6.2.2.0
Syntax

C#

```csharp
public delegate void MySqlInfoMessageEventHandler(
    Object sender,
    MySqlInfoMessageEventArgs args
)
```

Visual Basic (Declaration)

```vbnet
Public Delegate Sub MySqlInfoMessageEventHandler (_
    sender As Object, _
    args As MySqlInfoMessageEventArgs _
)
```

Visual C++

```cpp
public delegate void MySqlInfoMessageEventHandler(
    Object^ sender,
    MySqlInfoMessageEventArgs^ args
)
```

Parameters

sender
Type: `System::Object`

args
Type: `MySql.Data.MySqlClient::MySqlInfoMessageEventArgs`
See Also

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents a parameter to a `MySqlCommand`, and optionally, its mapping to `DataSet` columns. This class cannot be inherited.

**Namespace:**  MySQL.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

**C#**

```csharp
public sealed class MySqlParameter : DbParameter,
    IDbDataParameter, IDataParameter, ICloneable
```

**Visual Basic (Declaration)**

```vbnet
Public NotInheritable Class MySqlParameter _
    Inherits DbParameter _
    Implements IDbDataParameter, IDataParameter, ICloneable
```

**Visual C++**

```cpp
public ref class MySqlParameter sealed : public DbParameter,
    IDbDataParameter, IDataParameter, ICloneable
```
Inheritance Hierarchy

System...Object
System...MarshalByRefObject
System.Data.Common...DbParameter
MySql.Data.MySqlClient...MySqlParameter
See Also

 MyersParameter Members
 MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlParameter` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MySqlParameter</code></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clone</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetLifetimeService</strong></td>
<td>Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ResetDbType</strong></td>
<td>Resets the DbType property to its original settings. (Overrides DbParameter..:::ResetDbType()())</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Overridden. Gets a string containing the ParameterName. (Overrides Object..:::ToString()())</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbType</td>
<td>Gets or sets the DbType of the parameter. (Overrides DbType.)</td>
</tr>
<tr>
<td>Direction</td>
<td>Gets or sets a value indicating whether the parameter is input-only, output-only, bidirectional, or a stored procedure return value parameter. As of MySql version 4.1 and earlier, input-only is the only valid choice. (Overrides DbType::Direction.)</td>
</tr>
<tr>
<td>IsNullable</td>
<td>Gets or sets a value indicating whether the parameter accepts null values. (Overrides DbType::IsNullable.)</td>
</tr>
<tr>
<td>MySqlDbType</td>
<td>Gets or sets the MySqlDbType of the parameter.</td>
</tr>
<tr>
<td>ParameterName</td>
<td>Gets or sets the name of the MySqlParameter. (Overrides DbType::ParameterName.)</td>
</tr>
<tr>
<td>PossibleValues</td>
<td>Returns the possible values for this parameter if this parameter is of type SET or ENUM. Returns null otherwise.</td>
</tr>
<tr>
<td>Precision</td>
<td>Gets or sets the maximum number of digits used to represent the Value property.</td>
</tr>
<tr>
<td>Scale</td>
<td>Gets or sets the number of decimal places to which Value is resolved.</td>
</tr>
<tr>
<td>Size</td>
<td>Gets or sets the maximum size, in bytes, of the data within the column. (Overrides DbType::Size.)</td>
</tr>
<tr>
<td>SourceColumn</td>
<td>Gets or sets the name of the source column that is mapped to the DataSet and used for loading or returning the Value. (Overrides DbType::SourceColumn.)</td>
</tr>
<tr>
<td></td>
<td>Sets or gets a value which indicates whether the source column is nullable. This allows</td>
</tr>
</tbody>
</table>
**SourceColumnNullMapping**

*DbCommandBuilder* to correctly generate Update statements for nullable columns.

(Overrides *DbParameter...::SourceColumnNullMapping.*)

**SourceVersion**

Gets or sets the *DataRowVersion* to use when loading *Value*.

(Overrides *DbParameter...::SourceVersion.*)

**Value**

Gets or sets the value of the parameter.

(Overrides *DbParameter...::Value.*)
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MySqlParameter</strong></td>
<td>Initializes a new instance of the <strong>MySqlParameter</strong> class.</td>
</tr>
<tr>
<td><strong>MySqlParameter(String, MySqlDbType, Int32, ParameterDirection, Boolean, Byte, Byte, String, DataRowVersion, Object)</strong></td>
<td>Initializes a new instance of the <strong>MySqlParameter</strong> class with the parameter name, the type of the parameter, the size of the parameter, a <strong>ParameterDirection</strong>, the precision of the parameter, the scale of the parameter, the source column, a <strong>DataRowVersion</strong> to use, and the value of the parameter.</td>
</tr>
<tr>
<td><strong>MySqlParameter(String, MySqlDbType)</strong></td>
<td>Initializes a new instance of the <strong>MySqlParameter</strong> class with the parameter name and the data type.</td>
</tr>
<tr>
<td><strong>MySqlParameter(String, Object)</strong></td>
<td>Initializes a new instance of the <strong>MySqlParameter</strong> class with the parameter name and a value of the new <strong>MySqlParameter</strong>.</td>
</tr>
<tr>
<td><strong>MySqlParameter(String, MySqlDbType, Int32)</strong></td>
<td>Initializes a new instance of the <strong>MySqlParameter</strong> class with the parameter name, the <strong>MySqlDbType</strong>, and the size.</td>
</tr>
<tr>
<td><strong>MySqlParameter(String, MySqlDbType, Int32, String)</strong></td>
<td>Initializes a new instance of the <strong>MySqlParameter</strong> class with the parameter name, the <strong>MySqlDbType</strong>, the size, and the source column name.</td>
</tr>
</tbody>
</table>
See Also

MySqlParameter Class
MySqlParameter Members
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlParameter` class.

**Namespace:**  `MySQL.Data.MySqlClient`

**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.0
**Syntax**

**C#**

```csharp
public MySqlParameter()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
MySqlParameter()
```
See Also

MySqlParameter Class
MySqlParameter Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlParameter Constructor (String, SqlDbType, Int32, ParameterDirection, Boolean, Byte, Byte, String, DataRowVersion, Object)

Initializes a new instance of the `SqlParameter` class with the parameter name, the type of the parameter, the size of the parameter, a `ParameterDirection`, the precision of the parameter, the scale of the parameter, the source column, a `DataRowVersion` to use, and the value of the parameter.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data (in MySql.Data.dll) Version: 6.2.2.0`
**Syntax**

### C#

```csharp
public MySqlParameter(
    string parameterName,
    SqlDbType dbType,
    int size,
    ParameterDirection direction,
    bool isNullable,
    byte precision,
    byte scale,
    string sourceColumn,
    DataRowVersion sourceVersion,
    Object value)
```

### Visual Basic (Declaration)

```vbnet
Public Sub New (_
    parameterName As String, _
    dbType As SqlDbType, _
    size As Integer, _
    direction As ParameterDirection, _
    isNullable As Boolean, _
    precision As Byte, _
    scale As Byte, _
    sourceColumn As String, _
    sourceVersion As DataRowVersion, _
    value As Object _
)
```

### Visual C++

```cpp
public:
    MySqlParameter(
        String^ parameterName,
        SqlDbType dbType,
        int size,
        ParameterDirection direction,
        bool isNullable,
        unsigned char precision,
        unsigned char scale,
        String^ sourceColumn,
```
```csharp
DataRowVersion sourceVersion,
Object^ value
)

**Parameters**

`parameterName`
- Type: `System:::String`
  - The name of the parameter to map.

`dbType`
- Type: `MySql.Data.MySqlClient:::MySqlDbType`
  - One of the `MySqlDbType` values.

`size`
- Type: `System:::Int32`
  - The length of the parameter.

`direction`
- Type: `System.Data:::ParameterDirection`
  - One of the `ParameterDirection` values.

`isNullable`
- Type: `System:::Boolean`
  - true if the value of the field can be null, otherwise false.

`precision`
- Type: `System:::Byte`
  - The total number of digits to the left and right of the decimal point to which `Value` is resolved.

`scale`
- Type: `System:::Byte`
  - The total number of decimal places to which `Value` is resolved.

`sourceColumn`
- Type: `System:::String`
  - The name of the source column.
sourceVersion
Type: System.Data::DataRowVersion
One of the DataRowVersion values.

value
Type: System::Object
An Object that is the value of the MySqlConnection.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System::ArgumentException</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlParameter Class
MySqlParameter Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlParameter` class with the parameter name and the data type.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySql.Data](#) (in [MySql.Data.dll](#)) Version: 6.2.2.0
**Syntax**

C#

```csharp
public MySqlParameter(
    string parameterName,
    MySqlDbType dbType
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New (_
    parameterName As String, _
    dbType As MySqlDbType _
)
```

**Visual C++**

```cpp
public:
MySqlParameter(
    String^ parameterName,
    MySqlDbType dbType
)
```

**Parameters**

**parameterName**
Type: `System::String`
The name of the parameter to map.

**dbType**
Type: `MySql.Data.MySqlClient::MySqlDbType`
One of the `MySqlDbType` values.
See Also

MySqlParameter Class
MySqlParameter Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlParameter` class with the parameter name and a value of the new `MySqlParameter`.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public MySqlParameter(
    string parameterName,
    Object value
)

Visual Basic (Declaration)

Public Sub New (_
    parameterName As String, _
    value As Object _
)

Visual C++

public:
MySqlParameter(
    String^ parameterName,
    Object^ value
)

Parameters

parameterName
    Type: System::String
    The name of the parameter to map.

value
    Type: System::Object
    An Object that is the value of the MySqlParameter.
See Also

MySqlParameter Class
MySqlParameter Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlParameter` class with the parameter name, the `MySqlDbType`, and the size.

**Namespace:**  `MySQL.Data.MySqlClient`  
**Assembly:**  `MySql.Data (in MySql.Data.dll)`  Version: 6.2.2.0
## Syntax

### C#

```csharp
public SqlParameter(
    string parameterName,
    MySqlDbType dbType,
    int size
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub New (_
    ByVal parameterName As String, _
    ByVal dbType As MySqlDbType, _
    ByVal size As Integer _
)
```

### Visual C++

```cpp
public: 
SqlParameter( 
    String^ parameterName, 
    MySqlDbType dbType, 
    int size 
)
```

## Parameters

**parameterName**
- Type: `System::String`
- The name of the parameter to map.

**dbType**
- Type: `MySql.Data.MySqlClient::MySqlDbType`
- One of the `MySqlDbType` values.

**size**
- Type: `System::Int32`
The length of the parameter.
See Also

MySqlParameter Class
MySqlParameter Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlParameter` class with the parameter name, the `MySqlDbType`, the size, and the source column name.

**Namespace:** `MySQL.Data.MySqlClient`  
**Assembly:** `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public SqlParameter(
    string parameterName,
    MySqlDbType dbType,
    int size,
    string sourceColumn
)

Visual Basic (Declaration)

Public Sub New (_
    parameterName As String, _
    dbType As MySqlDbType, _
    size As Integer, _
    sourceColumn As String _
)

Visual C++

public:
    MySqlParameter(
        String^ parameterName,
        MySqlDbType dbType,
        int size,
        String^ sourceColumn
    )

Parameters

parameterName
    Type: System::String
    The name of the parameter to map.

dbType
    Type: MySql.Data.MySqlClient::MySqlDbType
    One of the MySqlDbType values.
size
  Type: System::Int32
  The length of the parameter.

sourceColumn
  Type: System::String
  The name of the source column.
See Also

MySqlParameter Class
MySqlParameter Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlParameter` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clone</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetLifetimeService</strong></td>
<td>Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td><strong>InitializeLifetimeService</strong></td>
<td>Overloaded. Resets the DbType property to its original settings. (Overrides DbParameter..:::ResetDbType()() Q.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Overridden. Gets a string containing the ParameterName. (Overrides Object..:::ToString()() Q.)</td>
</tr>
<tr>
<td><strong>ResetDbType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Parameter Clone Method

See Also
Send Feedback

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlParameter Clone()

Visual Basic (Declaration)

Public Function Clone As MySqlParameter

Visual C++

public:
MySqlParameter^ Clone()
See Also

MySQLParameter Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net
MySqlParameter...MemberwiseClone Method

MySqlParameter Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] MemberwiseClone()()</td>
<td>Creates a shallow copy of the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>![ ] MemberwiseClone(Boolean) MarshalByRefObject</td>
<td>Creates a shallow copy of the current <a href="#">MarshalByRefObject</a>. (Inherited from <a href="#">MarshalByRefObject</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlParameter Class
MySqlParameter Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Resets the **DbType** property to its original settings.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  `MySQL.Data` (in `MySQL.Data.dll`) Version: 6.2.2.0
Syntax

C#

public override void ResetDbType()

Visual Basic (Declaration)

Public Overrides Sub ResetDbType

Visual C++

public:
virtual void ResetDbType() override
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLParameter...::ToString Method

Overridden. Gets a string containing the ParameterName.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public override string ToString()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function ToString As String
```

**Visual C++**

```cpp
public:
    virtual String^ ToString() override
```

**Return Value**
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlParameter` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbType</td>
<td>Gets or sets the DbType of the parameter. (Overrides DbParameter...:DbType.) Gets or sets a value indicating whether the parameter is input-only, output-only, bidirectional, or a stored procedure return value parameter. As of MySql version 4.1 and earlier, input-only is the only valid choice. (Overrides DbParameter...:Direction.)</td>
</tr>
<tr>
<td>Direction</td>
<td>Gets or sets a value indicating whether the parameter accepts null values. (Overrides DbParameter...:IsNullable.)</td>
</tr>
<tr>
<td>IsNullable</td>
<td>Gets or sets the MySqlDbType of the parameter.</td>
</tr>
<tr>
<td>MySqlDbType</td>
<td>Gets or sets the name of the MySqlParameter. (Overrides DbParameter...:ParameterName.) Returns the possible values for this parameter if this parameter is of type SET or ENUM. Returns null otherwise.</td>
</tr>
<tr>
<td>ParameterName</td>
<td>Gets or sets the maximum number of digits used to represent the Value property.</td>
</tr>
<tr>
<td>PossibleValues</td>
<td>Gets or sets the number of decimal places to which Value is resolved.</td>
</tr>
<tr>
<td>Precision</td>
<td>Gets or sets the maximum size, in bytes, of the data within the column. (Overrides DbParameter...:Size.)</td>
</tr>
<tr>
<td>Scale</td>
<td>Gets or sets the name of the source column that is mapped to the DataSet and used for loading or returning the Value. (Overrides DbParameter...:SourceColumn.) Sets or gets a value which indicates whether the source column is nullable. This allows</td>
</tr>
</tbody>
</table>
SourceColumnNullMapping DbCommandBuilder to correctly generate Update statements for nullable columns. (Overrides DbParameter...::SourceColumnNullMapping.)

SourceVersion Gets or sets the DataRowVersion to use when loading Value. (Overrides DbParameter...::SourceVersion.)

Value Gets or sets the value of the parameter. (Overrides DbParameter...::Value.)
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C# Visual Basic
Visual C++
MySQL Connector/Net

**MySqlParameter**.DbType Property

**MySqlParameter** Class  [See Also](#)  [Send Feedback](#)

Gets or sets the DbType of the parameter.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override DbType DbType { get; set; }

Visual Basic (Declaration)

Public Overrides Property DbType As DbType

Visual C++

public:
virtual property DbType DbType {
        DbType get () override;
        void set (DbType value) override;
}

Implements

IDataParameter..:::.DbType
IDataParameter..:::.DbType
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a value indicating whether the parameter is input-only, output-only, bidirectional, or a stored procedure return value parameter. As of MySql version 4.1 and earlier, input-only is the only valid choice.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override ParameterDirection Direction { get; set; }

Visual Basic (Declaration)

Public Overrides Property Direction As ParameterDirection

Visual C++

public:
virtual property ParameterDirection Direction {
ParameterDirection get () override;
void set (ParameterDirection value) override;
}

Implements

IDataParameter..:::Direction
IDataParameter..:::Direction
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets a value indicating whether the parameter accepts null values.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override bool IsNullable { get; set; }

Visual Basic (Declaration)

Public Overrides Property IsNullable As Boolean

Visual C++

public:
virtual property bool IsNullable {
    bool get () override;
    void set (bool value) override;
}

Implements

IDataParameter::IsNullual
IDataParameter::IsNullual
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the MySqlDbType of the parameter.

**Namespace:**  [MySQL.Data.MySqlClient](https://github.com/MySql/MySQL Connector/Net)

**Assembly:**  [MySQL.Data](https://github.com/MySql/MySQL Connector/Net) (in MySQL.Data.dll) Version: 6.2.2.0
## Syntax

**C#**

```csharp
public MySqlDbType MySqlDbType { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property MySqlDbType As MySqlDbType
```

**Visual C++**

```cpp
public:
property MySqlDbType MySqlDbType {
    MySqlDbType get ();
    void set (MySqlDbType value);
}
```
See Also

 MySqlParameter Class
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the name of the MySqlParameter.

**Namespace:**  [MySql.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public override string ParameterName { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Property ParameterName As String
```

**Visual C++**

```cpp
public:
virtual property String^ ParameterName {
    String^ get () override;
    void set (String^ value) override;
}
```

### Implements

- `IDataParameter::::ParameterName`
- `IDataParameter::::ParameterName`
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns the possible values for this parameter if this parameter is of type SET or ENUM. Returns null otherwise.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public IList PossibleValues { get; internal set; }

Visual Basic (Declaration)

Public Property PossibleValues As IList

Visual C++

public:
property IList^ PossibleValues {
    IList^ get ();
    void set (IList^ value);
}
See Also

**MySqlParameter Class**

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the maximum number of digits used to represent the Value property.

**Namespace:** [MySQL.Data.MySqlClient](https://www.mysql.com/productsconnector-net)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public byte Precision { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Precision As Byte
```

**Visual C++**

```cpp
public:
virtual property unsigned char Precision {
    unsigned char get () sealed;
    void set (unsigned char value) sealed;
}
```

**Implements**

```cpp
IDbDataParameter:::Precision
```
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the number of decimal places to which `Value` is resolved.

**Namespace:**  [MySql.Data.MySqlClient](https://github.com/MySqlConnector/MySqlConnector)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public byte Scale { get; set; }
```

**Visual Basic (Declaration)**

Public Property Scale As Byte

**Visual C++**

```cpp
public:
virtual property unsigned char Scale {
    unsigned char get () sealed;
    void set (unsigned char value) sealed;
}
```

**Implements**

IDbDataParameter::::Scale
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the maximum size, in bytes, of the data within the column.

**Namespace:** [MySQL.Data.MySqlClient](#)  
**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override int Size { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Property Size As Integer
```

### Visual C++

```cpp
public:
virtual property int Size {
    int get () override;
    void set (int value) override;
}
```

### Implements

- `IDbDataParameter::Size`
- `IDbDataParameter::Size`
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the name of the source column that is mapped to the `DataSet` and used for loading or returning the `Value`.

**Namespace:**  [MySql.Data.MySqlClient](https://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override string SourceColumn { get; set; }

Visual Basic (Declaration)

Public Overrides Property SourceColumn As String

Visual C++

public:
virtual property String^ SourceColumn {
  String^ get () override;
  void set (String^ value) override;
}

Implements

IDataParameter::<SourceColumn
IDataParameter::<SourceColumn
See Also

- MySqlConnection Class
- MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Sets or gets a value which indicates whether the source column is nullable. This allows `DbCommandBuilder` to correctly generate Update statements for nullable columns.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public override bool SourceColumnNullMapping { get; set; }

Visual Basic (Declaration)

Public Overrides Property SourceColumnNullMapping As Boolean

Visual C++

public:
virtual property bool SourceColumnNullMapping {
    bool get () override;
    void set (bool value) override;
}
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the `DataRowVersion` to use when loading `Value`.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override DataRowVersion SourceVersion { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Property SourceVersion As DataRowVersion
```

**Visual C++**

```cpp
public:
virtual property DataRowVersion SourceVersion {
    DataRowVersion get () override;
    void set (DataRowVersion value) override;
}
```

**Implements**

IDataParameter::<Version>
IDataParameter::<Version>
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the value of the parameter.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override Object Value { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Property Value As Object
```

**Visual C++**

```c++
public:
virtual property Object^ Value {
    Object^ get () override;
    void set (Object^ value) override;
}
```

**Implements**

```csharp
IDataParameter:::Value
IDataParameter:::Value
```
See Also

MySqlParameter Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents a collection of parameters relevant to a `MySqlCommand` as well as their respective mappings to columns in a `DataSet`. This class cannot be inherited.

**Namespace:**  `MySQL.Data.MySqlClient`  
**Assembly:**  `MySQL.Data` (in `MySQL.Data.dll`)  
**Version:**  6.2.2.0
**Syntax**

C#

public sealed class MySqlParameterCollection : DbParameterCollection

**Visual Basic (Declaration)**

Public NotInheritable Class MySqlParameterCollection
    Inherits DbParameterCollection

**Visual C++**

public ref class MySqlParameterCollection sealed : public DbParameterCollection
Remarks

The number of the parameters in the collection must be equal to the number of parameter placeholders within the command text, or an exception will be generated.
Examples

The following example creates multiple instances of `MySqlParameter` through the `MySqlParameterCollection` collection within the `MySqlDataAdapter`. These parameters are used to select data within the data source and place the data in the `DataSet`. This code assumes that a `DataSet` and a `MySqlDataAdapter` have already been created with the appropriate schema, commands, and connection.

**VB.NET**

```vbnet
Public Sub AddParameters()
' ...
' create myDataSet and myDataAdapter
' ...
End Sub 'AddSqlParameters
```

**C#**

```csharp
public void AddSqlParameters()
{
    // ...
    // create myDataSet and myDataAdapter
    // ...

}
```
Inheritance Hierarchy

System..:::Object
System..:::MarshalByRefObject
System.Data.Common..:::DbParameterCollection
MySQL.Data.MySqlClient..:::MySqlParameterCollection
See Also

MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlParameterCollection` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Overloaded. Adds an array of values to the end of the <strong>MySqlParameterCollection</strong>. (Overrides <strong>DbParameterCollection</strong>:::<strong>AddRange</strong>(<em>Array</em>).)</td>
</tr>
<tr>
<td><strong>AddRange</strong></td>
<td>Adds an array of values to the end of the <strong>MySqlParameterCollection</strong>. (Overrides <strong>DbParameterCollection</strong>:::<strong>AddRange</strong>(<em>Array</em>).)</td>
</tr>
<tr>
<td><strong>AddWithValue</strong></td>
<td>Removes all items from the collection. (Overrrides <strong>DbParameterCollection</strong>:::<strong>Clear</strong>() Q.) Overloaded.</td>
</tr>
<tr>
<td><strong>Clear</strong></td>
<td>Gets a value indicating whether a <strong>MySqlParameter</strong> exists in the collection. Copies <strong>MySqlParameter</strong> objects from the <strong>MySqlParameterCollection</strong> to the specified array. (Overrides <strong>DbParameterCollection</strong>:::<strong>CopyTo</strong>(<em>Array</em>, <strong>Int32</strong>).)</td>
</tr>
<tr>
<td><strong>Contents</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <strong>MarshalByRefObject</strong>.)</td>
</tr>
<tr>
<td><strong>CopyTo</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.) Allows an <strong>Object</strong> to attempt to free resources and perform other cleanup operations before the <strong>Object</strong> is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.) Returns an enumerator that iterates through the <strong>MySqlParameterCollection</strong>.</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>(Overrides DbParameterCollection..:::GetEnumerator())</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetLifetimeService</td>
<td>Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>GetParameter</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Gets the location of a MySqlParameter in the collection.</td>
</tr>
<tr>
<td>InitializeLifetimeService</td>
<td>Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>Insert</td>
<td>(Overrides DbParameterCollection..:::Insert(Int32, Object).)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the specified MySqlParameter from the collection. (Overrides DbParameterCollection..:::Remove(Object).)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Removes the specified MySqlParameter from the collection.</td>
</tr>
<tr>
<td>SetParameter</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong></td>
<td>Gets the number of MySqlCommand objects in the collection. (Overrides DbParameterCollection..::.Count.)</td>
</tr>
<tr>
<td><strong>IsFixedSize</strong></td>
<td>Gets a value that indicates whether the MySqlCommandCollection has a fixed size. (Overrides DbParameterCollection..::.IsFixedSize.)</td>
</tr>
<tr>
<td><strong>IsReadOnly</strong></td>
<td>Gets a value that indicates whether the MySqlCommandCollection is read-only. (Overrides DbParameterCollection..::.IsReadOnly.)</td>
</tr>
<tr>
<td><strong>IsSynchronized</strong></td>
<td>Gets a value that indicates whether the MySqlCommandCollection is synchronized. (Overrides DbParameterCollection..::.IsSynchronized.)</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>Overloaded. Gets the MySqlCommand with a specified attribute. [C#] In C#, this property is the indexer for the MySqlCommandCollection class. Gets an object that can be used to synchronize access to the MySqlCommandCollection. (Overrides DbParameterCollection..::.SyncRoot.)</td>
</tr>
<tr>
<td><strong>SyncRoot</strong></td>
<td>Gets an object that can be used to synchronize access to the MySqlCommandCollection. (Overrides DbParameterCollection..::.SyncRoot.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlParameterCollection Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlParameterCollection` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Overloaded. Adds an array of values to the end of the <strong>MySqlParameterCollection</strong>. (Overrides <strong>DbParameterCollection</strong>:::<strong>AddRange</strong>(Array).)</td>
</tr>
<tr>
<td><strong>AddRange</strong></td>
<td>Adds an array of values to the end of the <strong>MySqlParameterCollection</strong>. (Overrides <strong>DbParameterCollection</strong>:::<strong>AddRange</strong>(Array).)</td>
</tr>
<tr>
<td><strong>AddWithValue</strong></td>
<td>Removes all items from the collection. (Overloads <strong>DbParameterCollection</strong>:::<strong>Clear</strong>().)</td>
</tr>
<tr>
<td><strong>Clear</strong></td>
<td>Overloaded. Gets a value indicating whether a <strong>MySqlParameter</strong> exists in the collection.</td>
</tr>
<tr>
<td><strong>Contains</strong></td>
<td>Copies <strong>MySqlParameter</strong> objects from the <strong>MySqlParameterCollection</strong> to the specified array. (Overloads <strong>DbParameterCollection</strong>:::<strong>CopyTo</strong>(Array, Int32).)</td>
</tr>
<tr>
<td><strong>CopyTo</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <strong>MarshalByRefObject</strong>.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Allows an <strong>Object</strong> to attempt to free resources and perform other cleanup operations before the <strong>Object</strong> is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Returns an enumerator that iterates through the <strong>MySqlParameterCollection</strong>.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>(Overrides <code>DbParameterCollection..::.GetEnumerator()()</code>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetLifetimeService</td>
<td>Retrieves the current lifetime service object that controls the lifetime</td>
</tr>
<tr>
<td></td>
<td>policy for this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td>GetParameter</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>IndexOf</td>
<td>Gets the location of a <code>MySqlParameter</code> in the collection.</td>
</tr>
<tr>
<td></td>
<td>Obtains a lifetime service object to control the lifetime policy for this</td>
</tr>
<tr>
<td></td>
<td>instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td>Insert</td>
<td>(Overrides <code>DbParameterCollection..::.Insert(Int32, Object)</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the specified <code>MySqlParameter</code> from the collection.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbParameterCollection..::.Remove(Object)</code>.)</td>
</tr>
<tr>
<td>RemoveAt</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>SetParameter</td>
<td>Removes the specified <code>MySqlParameter</code> from the collection.</td>
</tr>
<tr>
<td></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlParameterCollection Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(String, MySqlDbType)</td>
<td>Adds a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.mysqldb.mysqldbparameter?view=netcore31">MySqlParameter</a> to the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.mystoredprocedures.mystoredprocedurescollection?view=netcore31">MySqlParameterCollection</a> given the parameter name and the data type. <strong>Obsolete.</strong></td>
</tr>
<tr>
<td>Add(String, MySqlDbType, Int32)</td>
<td>Adds a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.mysqldb.mysqldbparameter?view=netcore31">MySqlParameter</a> to the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.mystoredprocedures.mystoredprocedurescollection?view=netcore31">MySqlParameterCollection</a> with the parameter name, the data type, and the column length.</td>
</tr>
<tr>
<td>Add(String, MySqlDbType, Int32, String)</td>
<td>Adds a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.mysqldb.mysqldbparameter?view=netcore31">MySqlParameter</a> to the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.mystoredprocedures.mystoredprocedurescollection?view=netcore31">MySqlParameterCollection</a> with the parameter name, the data type, the column length, and the source column name.</td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Adds the specified `MySqlParameter` object to the `MySqlParameterCollection`.

**Namespace:**  `MySql.Data.MySqlClient`

**Assembly:**  `MySql.Data (in MySql.Data.dll) Version: 6.2.2.0`
**Syntax**

**C#**

```csharp
public MySqlParameter Add(
    MySqlParameter value
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Add (
    value As MySqlParameter
) As MySqlParameter
```

**Visual C++**

```cpp
public:
    MySqlParameter^ Add(
        MySqlParameter^ value
    )
```

**Parameters**

`value`

Type: `MySql.Data.MySqlClient::MySqlParameter`

The `MySqlParameter` to add to the collection.

**Return Value**

The newly added `MySqlParameter` object.
See Also

MySqlParameterCollection Class
Add Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlParameterCollection..::..Add Method (Object)

**MySqlParameterCollection Class**  See Also  Send Feedback

Adds the specified *MySqlParameter* object to the *MySqlParameterCollection*.

**Namespace:**  MySql.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll)  Version: 6.2.2.0
## Syntax

**C#**

```csharp
public override int Add(
    Object value
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function Add ( _
    value As Object _
) As Integer
```

**Visual C++**

```cpp
public:
virtual int Add(
    Object^ value
) override
```

## Parameters

value

Type: `System::Object`

The `MySqlParameter` to add to the collection.

## Return Value

The index of the new `MySqlParameter` object.

## Implements

`IList::Add(Object)`
See Also

MySqlParameterCollection Class
Add Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Adds a `MySqlParameter` to the `MySqlParameterCollection` given the parameter name and the data type.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data (in MySql.Data.dll)`  
Version: 6.2.2.0
### Syntax

#### C#

```csharp
public `MySqlParameter` Add(
    `string` parameterName,
    `MySqlDbType` dbType
)
```

#### Visual Basic (Declaration)

```vbnet
Public Function Add ( _
    ByVal parameterName As `String`, _
    ByVal dbType As `MySqlDbType` _
) As `MySqlParameter`
```

#### Visual C++

```cpp
public: `MySqlParameter`^ Add(
    `String`^ parameterName,
    `MySqlDbType` dbType
)
```

### Parameters

**parameterName**

Type: `System::String`

The name of the parameter.

**dbType**

Type: `MySql.Data.MySqlClient::MySqlDbType`

One of the `MySqlDbType` values.

### Return Value

The newly added `MySqlParameter` object.
See Also

**MySqlParameterCollection Class**
**Add Overload**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlParameterCollection::Add Method (String, Object)

MySqlParameterCollection Class  See Also  Send Feedback

Adds a MySqlParameter to the MySqlParameterCollection given the specified parameter name and value.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

```csharp
[ObsoleteAttribute("Add(String parameterName, Object value) has been deprecated. Use AddWithValue(String parameterName, Object value)")]
public MySqlParameter Add(string parameterName, Object value)
```

**Visual Basic**

```vbnet
<ObsoleteAttribute("Add(String parameterName, Object value) has been deprecated. Use AddWithValue(String parameterName, Object value)")>
Public Function Add(parameterName As String, value As Object) As MySqlParameter
```

**Visual C++**

```cpp
[ObsoleteAttribute(L"Add(String parameterName, Object value) has been deprecated. Use AddWithValue(String parameterName, Object value)")]
public: 
MySqlParameter^ Add(String^ parameterName, Object^ value)
```

**Parameters**

- `parameterName`
  - **Type:** `System::String`
  - The name of the parameter.

- `value`
  - **Type:** `System::Object`
  - The **Value** of the `MySqlParameter` to add to the collection.

**Return Value**
The newly added *MySqlParameter* object.
See Also

MySqlParameterCollection Class
Add Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Parameter Collection...:: Add Method (String, MySqlDbType, Int32)  

MySQL Parameter Collection Class  See Also  Send Feedback

Adds a **MySqlParameter** to the **MySqlParameterCollection** with the parameter name, the data type, and the column length.

**Namespace:**  [MySQL.Data.MySqlClient]

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public SqlParameter Add(string parameterName, MySqlDbType dbType, int size)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Add(_
    ByVal parameterName As String, _
    ByVal dbType As MySqlDbType, _
    ByVal size As Integer) As MySqlParameter
```

**Visual C++**

```cpp
public: 
    MySqlParameter^ Add(
        String^ parameterName, 
        MySqlDbType dbType, 
        int size)
```

**Parameters**

**parameterName**

Type: System::String

The name of the parameter.

**dbType**

Type: MySql.Data.MySqlClient::MySqlDbType

One of the MySqlDbType values.

**size**

Type: System::Int32
The length of the column.

**Return Value**

See Also

MySqlParameterCollection Class
Add Overload
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Parameter Collection:

**Add Method** (String, MySqlDbType, Int32, String)

This method adds a `MySqlParameter` to the `MySqlParameterCollection` with the parameter name, the data type, the column length, and the source column name.

- **Namespace:** `MySql.Data.MySqlClient`
- **Assembly:** `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
## Syntax

### C#

```csharp
public MySqlConnection Add(
    string parameterName,
    MySqlDbType dbType,
    int size,
    string sourceColumn
)
```

### Visual Basic (Declaration)

```vbnet
Public Function Add (_,
    parameterName As String, _
    dbType As MySqlDbType, _
    size As Integer, _
    sourceColumn As String _
) As MySqlConnection
```

### Visual C++

```cpp
public:
    MySqlConnection^ Add(
        String^ parameterName,
        MySqlDbType dbType,
        int size,
        String^ sourceColumn
    )
```

## Parameters

- **parameterName**
  - Type: `System::::String`
  - The name of the parameter.

- **dbType**
  - Type: `MySql.Data.MySqlClient::MySqlDbType`
  - One of the `MySqlDbType` values.
size
   Type: System::Int32
   The length of the column.

sourceColumn
   Type: System::String
   The name of the source column.

**Return Value**

The newly added MySqlParameter object.
See Also

MySqlParameterCollection Class
Add Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlParameterCollection Class

MySqlParameterCollection..::..AddRange Method

Adds an array of values to the end of the MySQLParameterCollection.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override void AddRange(
    Array values
)

Visual Basic (Declaration)

Public Overrides Sub AddRange ( _
    values As Array _
)

Visual C++

public:
    virtual void AddRange(
        Array^ values
    ) override

Parameters

values
    Type: System::Array
See Also

**MySqlParameterCollection Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlParameterCollection..::.AddWithValue Method

MySqlParameterCollection Class  See Also  Send Feedback

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public MySqlConnection AddWithValue(
    string parameterName,
    Object value
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function AddWithValue ( _
    parameterName As String, _
    value As Object _
) As MySqlConnection
```

**Visual C++**

```cpp
public:
MySqlConnection^ AddWithValue(
    String^ parameterName, 
    Object^ value
)
```

**Parameters**

**parameterName**
  Type: **System::String**

**value**
  Type: **System::Object**
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlParameterCollection.Clear Method

Removes all items from the collection.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#
public override void Clear()

Visual Basic (Declaration)
Public Overrides Sub Clear

Visual C++
public:
virtual void Clear() override

Implements

IList:::Clear()()}
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Contains Method

MySqlParameterCollection Class  See Also  Send Feedback

Gets a value indicating whether a MySqlParameter exists in the collection.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Contains(Object)</code></td>
<td>Gets a value indicating whether a <code>MySqlParameter</code> exists in the collection. (Overrides <code>DbParameterCollection::Contains(Object)</code>.)</td>
</tr>
<tr>
<td><code>Contains(String)</code></td>
<td>Gets a value indicating whether a <code>MySqlParameter</code> with the specified parameter name exists in the collection. (Overrides <code>DbParameterCollection::Contains(String)</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic  Visual C++  MySQL Connector/Net

MySqlParameterCollection..::.Contains Method (Object)

MySqlParameterCollection Class  See Also  Send Feedback

Gets a value indicating whether a MySqlParameter exists in the collection.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
# Syntax

## C#

```csharp
public override bool Contains(
    Object value
)
```

## Visual Basic (Declaration)

```vbnet
Public Overrides Function Contains ( _
    value As Object _
) As Boolean
```

## Visual C++

```cpp
public:
    virtual bool Contains(
        Object^ value
    ) override
```

## Parameters

- **value**
  - Type: `System::Object`
  - The value of the `MySqlParameter` object to find.

## Return Value

- true if the collection contains the `MySqlParameter` object; otherwise, false.

## Implements

`IList::Contains(Object)`
See Also

MySqlParameterCollection Class
Contains Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Parameter Collection...::Contains Method (String)

**MySqlParameterCollection Class**  **See Also**  **Send Feedback**

GETS a value indicating whether a **MySqlParameter** with the specified parameter name exists in the collection.

**Namespace:**  **MySQL.Data.MySqlClient**  **Assembly:**  **MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0**
Syntax

C#

```csharp
public override bool Contains(string parameterName)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function Contains ( _
    parameterName As String _
) As Boolean
```

Visual C++

```cpp
public:
virtual bool Contains(
    String^ parameterName
) override
```

Parameters

parameterName

Type: `System::::String`

The name of the `MySqlParameter` object to find.

Return Value

true if the collection contains the parameter; otherwise, false.

Implements

`IDataParameterCollection::::Contains(String)`
See Also

MySqlParameterCollection Class
Contains Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Copies `MySqlParameter` objects from the `MySqlParameterCollection` to the specified array.

**Namespace:**  [MySQL.Data.MySqlClient](http://example.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public override void CopyTo(
    Array array,
    int index
)

Visual Basic (Declaration)

Public Overrides Sub CopyTo ( _
    array As Array, _
    index As Integer _
)

Visual C++

public:
virtual void CopyTo(
    Array^ array,
    int index
) override

Parameters

array
    Type: System::Array

index
    Type: System::Int32

Implements

ICollection::CopyTo(Array, Int32)
See Also

MySQLParameterCollection Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLParameterCollection..::.GetEnumerator Method

_Returns an enumerator that iterates through the [MySqlParameterCollection](https://example.com).

**Namespace:**  [MySql.Data.MySqlClient](https://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override IEnumerator GetEnumerator()

Visual Basic (Declaration)

Public Overrides Function GetEnumerator As IEnumerable

Visual C++

public:
virtual IEnumerable^ GetEnumerator() override

ReturnValue

Implements

IEnumerable::GetEnumerator()()
See Also

**MySqlParameterCollection Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLParameterCollection...:::getParameter Method

MySQLParameterCollection Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetParameter(Int32)</code></td>
<td>Returns the <a href="#">DbParameter</a> object at the specified index in the collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DbParameterCollection</a>.)</td>
</tr>
<tr>
<td><code>GetParameter(String)</code></td>
<td>Returns <a href="#">DbParameter</a> the object with the specified name.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DbParameterCollection</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the location of a `MySqlParameter` in the collection.
**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IndexOf(Object)</code></td>
<td>Gets the location of a <a href="https://example.com">MySqlParameter</a> in the collection. (Overrides DbParameterCollection..::..IndexOf(Object).)</td>
</tr>
<tr>
<td><code>IndexOf(String)</code></td>
<td>Gets the location of the <a href="https://example.com">MySqlParameter</a> in the collection with a specific parameter name. (Overrides DbParameterCollection..::..IndexOf(String).)</td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the location of a `MySqlParameter` in the collection.

**Namespace:**  `MySQL.Data.MySqlClient`  
**Assembly:**  `MySql.Data (in MySql.Data.dll) Version: 6.2.2.0`
Syntax

C#

public override int IndexOf(
    Object value
)

Visual Basic (Declaration)

Public Overrides Function IndexOf ( _
    value As Object _
) As Integer

Visual C++

public:
    virtual int IndexOf(
        Object^ value
    ) override

Parameters

value
    Type: System::Object
    The MySqlParameter object to locate.

Return Value

The zero-based location of the MySqlParameter in the collection.

Implements

IList::IndexOf(Object)
See Also

MySqlParameterCollection Class
IndexOf Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the location of the `MySqlParameter` in the collection with a specific parameter name.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySQL.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override int IndexOf(
    string parameterName
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function IndexOf ( _
    parameterName As String _
) As Integer
```

Visual C++

```cpp
public:
virtual int IndexOf(
    String^ parameterName
) override
```

Parameters

parameterName
  Type: System::String
  The name of the MySqlParameter object to retrieve.

Return Value

The zero-based location of the MySqlParameter in the collection.

Implements

IDataParameterCollection::IndexOf(String)
See Also

MySqlParameterCollection Class
IndexOf Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Inserts a SqlParameter into the collection at the specified index.

**Namespace:**  [MySQL.Data.MySqlClient](https://github.com/mysql/mysql-connector-net)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override void Insert(
    int index,
    Object value
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub Insert ( _
    index As Integer, _
    value As Object _
)
```

Visual C++

```cpp
public:
    virtual void Insert(
        int index,
        Object^ value
    ) override
```

Parameters

- **index**
  - Type: `System::::Int32`

- **value**
  - Type: `System::::Object`

Implements

- `IList::::Insert(Int32, Object)`
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  
Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
MySQL Connector/Net

MySqlParameterCollection::MemberwiseClone Method

MySqlParameterCollection Class  See Also  Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="MemberwiseClone()" /></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">Object</a>.)</td>
<td></td>
</tr>
<tr>
<td>![MemberwiseClone(Boolean)]</td>
<td>Creates a shallow copy of the current <a href="#">MarshalByRefObject</a> object.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">MarshalByRefObject</a>.)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Removes the specified MySqlCommand from the collection.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override void Remove(
    Object value
)

Visual Basic (Declaration)

Public Overrides Sub Remove ( _
    value As Object _
)

Visual C++

public:
virtual void Remove(
    Object^ value
) override

Parameters

value
    Type: System:::Object

Implements

IList:::Remove(Object)
See Also

**MySqlParameterCollection Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Removes the specified `MySqlParameter` from the collection.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveAt(Int32)</td>
<td>Removes the specified <code>MySqlParameter</code> from the collection using a specific index. (Overrides <code>DbParameterCollection::RemoveAt(Int32)</code>.)</td>
</tr>
<tr>
<td>RemoveAt(String)</td>
<td>Removes the specified <code>MySqlParameter</code> from the collection using the parameter name. (Overrides <code>DbParameterCollection::RemoveAt(String)</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Removes the specified `MySqlParameter` from the collection using a specific index.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

```csharp
public override void RemoveAt(
    int index
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub RemoveAt (_
    index As Integer _
)
```

Visual C++

```cpp
public:
virtual void RemoveAt(
    int index
) override
```

Parameters

index

Type: `System::::Int32`

The zero-based index of the parameter.

Implements

`IList:::RemoveAt(Int32)`
See Also

MySqlParameterCollection Class
RemoveAt Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Removes the specified *MySQLParameter* from the collection using the parameter name.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public override void RemoveAt(string parameterName)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub RemoveAt ( _
  parameterName As String _
)
```

**Visual C++**

```csharp
public:
virtual void RemoveAt(
  String^ parameterName
) override
```

### Parameters

**parameterName**

- **Type**: `System.String`
- The name of the `MySqlParameter` object to retrieve.

### Implements

`IDataParameterCollection::RemoveAt(String)`
See Also

**MySqlParameterCollection Class**  
**RemoveAt Overload**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlParameterCollection::SetParameter Method

<MySqlParameterCollection Class>  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetParameter(Int32, DbParameter)</td>
<td>Sets the DbParameter object at the specified index to a new value.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DbParameterCollection.)</td>
</tr>
<tr>
<td>SetParameter(String, DbParameter)</td>
<td>Sets the DbParameter object with the specified name to a new value.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DbParameterCollection.)</td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlParameterCollection` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong></td>
<td>Gets the number of <code>MySqlParameter</code> objects in the collection.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbParameterCollection:::Count</code>.)</td>
</tr>
<tr>
<td><strong>IsFixedSize</strong></td>
<td>Gets a value that indicates whether the <code>MySqlParameterCollection</code> has a fixed size.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbParameterCollection:::IsFixedSize</code>.)</td>
</tr>
<tr>
<td><strong>IsReadOnly</strong></td>
<td>Gets a value that indicates whether the <code>MySqlParameterCollection</code> is read-only.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbParameterCollection:::IsReadOnly</code>.)</td>
</tr>
<tr>
<td><strong>IsSynchronized</strong></td>
<td><code>MySqlParameterCollection</code> is synchronized.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbParameterCollection:::IsSynchronized</code>.)</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Gets the <code>MySqlParameter</code> with a specified attribute.</td>
</tr>
<tr>
<td></td>
<td>[C#] In C#, this property is the indexer for the <code>MySqlParameterCollection</code> class.</td>
</tr>
<tr>
<td><strong>SyncRoot</strong></td>
<td>Gets an object that can be used to synchronize access to the <code>MySqlParameterCollection</code>.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbParameterCollection:::SyncRoot</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the number of MySqlParameter objects in the collection.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)  
**Assembly:**  [MySql.Data](https://www.mysql.com) (in MySql.Data.dll)  
**Version:**  6.2.2.0
Syntax

C#

public override int Count { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property Count As Integer

Visual C++

public:
virtual property int Count {
    int get () override;
}

Implements

ICollection::Count
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a value that indicates whether the **MySqlParameterCollection** has a fixed size.

**Namespace:**  [MySQL.Data.MySqlClient](#)
**Assembly:**  [MySQL.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

public override bool IsFixedSize { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property IsFixedSize As Boolean

Visual C++

public:
virtual property bool IsFixedSize {
    bool get () override;
}

Implements

IList::. IsFixedSize
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlParameterCollection...:::IsReadOnly Property

Gets a value that indicates whether the `MySqlParameterCollection` is read-only.

**Namespace:**  [MySQL.Data.MySqlClient](https://example.com/mysql-data-mysqlclient)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#
public override bool IsReadOnly { get; }  

Visual Basic (Declaration)
Public Overrides ReadOnly Property IsReadOnly As Boolean

Visual C++
public:
virtual property bool IsReadOnly {
    bool get () override;
}

Implements

IList:::IsReadOnly
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a value that indicates whether the `MySqlParameterCollection` is synchronized.

**Namespace:**  [MySql.Data.MySqlClient](https://www.connectionpool.com/)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override bool IsSynchronized { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property IsSynchronized As Boolean

Visual C++

public:
virtual property bool IsSynchronized {
    bool get () override;
}

Implements

ICollection..::.IsSynchronized
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the `MySqlParameter` with a specified attribute. [C#] In C#, this property is the indexer for the `MySqlParameterCollection` class.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Item[Int32]</code></td>
<td>Gets the <code>MySqlParameter</code> at the specified index.</td>
</tr>
<tr>
<td><code>Item[String]</code></td>
<td>Gets the <code>MySqlParameter</code> with the specified name.</td>
</tr>
</tbody>
</table>
See Also

MySqlParameterCollection Class
MySqlParameterCollection Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the `MySqlParameter` at the specified index.

**Namespace:**  MySQL.Data.MySqlClient  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlParameter this[int index]
{
    get;
    set;
}
```

Visual Basic (Declaration)

```vbnet
Public Default Property Item ( _
index As Integer _
) As MySqlParameter
```

Visual C++

```cpp
public:
property MySqlParameter^ default[int index] {
    MySqlParameter^ get (int index);
    void set (int index, MySqlParameter^ value);
}
```

Parameters

index
Type: System::::Int32
See Also

MySqlParameterCollection Class
Item Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the `MySqlParameter` with the specified name.

**Namespace:**  `MySQL.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public SqlParameter this[string name]
{
    get;
    set;
}

Visual Basic (Declaration)

Public Default Property Item ( _
    name As String _
) As SqlParameter

Visual C++

public:
    property SqlParameter^ default[String^ name] {
        SqlParameter^ get (String^ name);
        void set (String^ name, SqlParameter^ value);
    }

Parameters

name
    Type: System:::String
See Also

MySqlParameterCollection Class
Item Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
DB::SyncRoot Property

Gets an object that can be used to synchronize access to the
MySQLParameterCollection.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

public override Object SyncRoot { get; }

**Visual Basic (Declaration)**

Public Overrides ReadOnly Property SyncRoot As Object

**Visual C++**

public:
virtual property Object^ SyncRoot {
    Object^ get () override;
}

**Implements**

ICollection::<::<SyncRoot
See Also

MySqlParameterCollection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Provides data for the RowUpdated event. This class cannot be inherited.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public sealed class MySqlRowUpdatedEventArgs : RowUpdatedEventArgs

Visual Basic (Declaration)

Public NotInheritable Class MySqlRowUpdatedEventArgs
    Inherits RowUpdatedEventArgs

Visual C++

public ref class MySqlRowUpdatedEventArgs sealed : public RowUpdatedEventArgs
Inheritance Hierarchy

System:::Object
System:::EventArgs
System.Data.Common:::RowUpdatedEventArgs
MySql.Data.MySqlClient:::MySqlRowUpdatedEventArgs
See Also

MySqlRowUpdatedEventArgs Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlRowUpdatedEventArgs` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlRowUpdatedEventArgs</td>
<td>Initializes a new instance of the MySqlRowUpdatedEventArgs class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyToRows</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the Object is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command</strong></td>
<td>Gets or sets the MySqlCommand executed when Update is called.</td>
</tr>
<tr>
<td><strong>Errors</strong></td>
<td>Gets any errors generated by the .NET Framework data provider when the Command was executed. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>RecordsAffected</strong></td>
<td>Gets the number of rows changed, inserted, or deleted by execution of the SQL statement. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>Row</strong></td>
<td>Gets the DataRow sent through an Update(DataSet). (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>RowCount</strong></td>
<td>Gets the number of rows processed in a batch of updated records. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>StatementType</strong></td>
<td>Gets the type of SQL statement executed. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Gets the UpdateStatus of the Command property. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>TableMapping</strong></td>
<td>Gets the DataTableMapping sent through an Update(DataSet). (Inherited from RowUpdatedEventArgs.)</td>
</tr>
</tbody>
</table>
See Also

MySqlRowUpdatedEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySQLRowUpdatedEventArgs Constructor

MySQLRowUpdatedEventArgs Class  See Also  Send Feedback

Initializes a new instance of the MySqlRowUpdatedEventArgs class.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public MySqlRowUpdatedEventArgs(
    DataRow row,
    IDbCommand command,
    StatementType statementType,
    DataTableMapping tableMapping
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub New (_
    row As DataRow, _
    command As IDbCommand, _
    statementType As StatementType, _
    tableMapping As DataTableMapping _
)
```

### Visual C++

```cpp
public:
MySqlRowUpdatedEventArgs(
    DataRow^ row,
    IDbCommand^ command,
    StatementType statementType,
    DataTableMapping^ tableMapping
)
```

## Parameters

### row

Type: `System.Data::::DataRow`

The `DataRow` sent through an `Update(DataSet)`.

### command

Type: `System.Data::::IDbCommand`

The `IDbCommand` executed when `Update(DataSet)` is called.
statementType
Type: System.Data:::StatementType
One of the StatementType values that specifies the type of query executed.

tableMapping
Type: System.Data.Common:::DataTableMapping
The DataTableMapping sent through an Update(DataSet).
See Also

**MySqlRowUpdatedEventArgs Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLRowUpdatedEventArgs` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyToRows</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Equals</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlRowUpdatedEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL RowUpdatedEventArgs CopyToRows Method

MySQLRowUpdatedEventArgs Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyToRows(array&lt;DataRow&gt;[][])</td>
<td>Copies references to the modified rows into the provided array. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td>CopyToRows(array&lt;DataRow&gt;[][], Int32)</td>
<td>Copies references to the modified rows into the provided array. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
</tbody>
</table>
See Also

**See Also**

MySQLRowUpdatedEventArgs Class  
MySQLRowUpdatedEventArgs Members  
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLRowUpdatedEventArgs** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command</strong></td>
<td>Gets or sets the MySqlCommand executed when Update is called.</td>
</tr>
<tr>
<td><strong>Errors</strong></td>
<td>Gets any errors generated by the .NET Framework data provider when the Command was executed. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>RecordsAffected</strong></td>
<td>Gets the number of rows changed, inserted, or deleted by execution of the SQL statement. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>RowCount</strong></td>
<td>Gets the number of rows processed in a batch of updated records. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>StatementType</strong></td>
<td>Gets the type of SQL statement executed. (Inherited from RowUpdatedEventArgs.)</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Gets the UpdateStatus of the Command property. (Inherited from Command.)</td>
</tr>
<tr>
<td><strong>TableMapping</strong></td>
<td>Gets the DataTableMapping sent through an Update(DataSet). (Inherited from RowUpdatedEventArgs.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlRowUpdatedEventArgs Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlRowUpdatedEventArgs Class

Gets or sets the MySqlCommand executed when Update is called.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCommand Command { get; }

Visual Basic (Declaration)

Public ReadOnly Property Command As MySqlCommand

Visual C++

public:
property MySqlCommand^ Command {
    MySqlCommand^ get ();
}
}
See Also

MySqlRowUpdatedEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents the method that will handle the RowUpdated event of a MySqlDataAdapter.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public delegate void MySqlRowUpdatedEventHandler(
    Object sender,
    MySqlRowUpdatedEventArgs e
)
```

**Visual Basic (Declaration)**

```vbnet
Public Delegate Sub MySqlRowUpdatedEventHandler ( _
    sender As Object, _
    e As MySqlRowUpdatedEventArgs _
)
```

**Visual C++**

```c++
public delegate void MySqlRowUpdatedEventHandler(
    Object^ sender,
    MySqlRowUpdatedEventArgs^ e
)
```

**Parameters**

- **sender**
  - Type: `System::Object`

- **e**
  - Type: `MySql.Data.MySqlClient::MySqlRowUpdatedEventArgs`
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Provides data for the RowUpdating event. This class cannot be inherited.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

public sealed class MySqlRowUpdatingEventArgs : RowUpdatingEventArgs

**Visual Basic (Declaration)**

Public NotInheritable Class MySqlRowUpdatingEventArgs
    Inherits RowUpdatingEventArgs

**Visual C++**

public ref class MySqlRowUpdatingEventArgs sealed : public RowUpdatingEventArgs
Inheritance Hierarchy

System..::.Object
System..::.EventArgs
System.Data.Common..::.RowUpdatingEventArgs
MySQL.Data.MySqlClient..::.MySqlRowUpdatingEventArgs
See Also

MySqlRowUpdatingEventArgs Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlRowUpdatingEventArgs` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlRowUpdatingEventArgs</td>
<td>Initializes a new instance of the MySqlRowUpdatingEventArgs class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BaseCommand</strong></td>
<td>Gets or sets the IDbCommand object for an instance of this class. (Inherited from RowUpdatingEventArgs.)</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>Gets or sets the MySqlCommand to execute when performing the Update. Gets any errors generated by the .NET Framework data provider when the Command executes. (Inherited from RowUpdatingEventArgs.)</td>
</tr>
<tr>
<td><strong>Errors</strong></td>
<td>Gets the DataRow that will be sent to the server as part of an insert, update, or delete operation. (Inherited from RowUpdatingEventArgs.)</td>
</tr>
<tr>
<td><strong>Row</strong></td>
<td>Gets the type of SQL statement to execute. (Inherited from RowUpdatingEventArgs.) Gets the UpdateStatus of the Command property. (Inherited from RowUpdatingEventArgs.)</td>
</tr>
<tr>
<td><strong>StatementType</strong></td>
<td>Gets or sets the UpdateStatus of the Command property. (Inherited from RowUpdatingEventArgs.) Gets the DataTableMapping to send through the Update(DataSet). (Inherited from RowUpdatingEventArgs.)</td>
</tr>
<tr>
<td><strong>TableMapping</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlRowUpdatingEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the MySqlRowUpdatingEventArgs class.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlRowUpdatingEventArgs(
    DataRow row,
    IDbCommand command,
    StatementType statementType,
    DataTableMapping tableMapping
)

Visual Basic (Declaration)

Public Sub New (_
    row As DataRow, _
    command As IDbCommand, _
    statementType As StatementType, _
    tableMapping As DataTableMapping _
)

Visual C++

public:
    MySqlRowUpdatingEventArgs(
        DataRow^ row,
        IDbCommand^ command,
        StatementType statementType,
        DataTableMapping^ tableMapping
    )

Parameters

row
    Type: System.Data:::DataRow
    The DataRow to Update(DataSet).

command
    Type: System.Data:::IDbCommand
    The IDbCommand to execute during Update(DataSet).
statementType
Type: System.Data.:::StatementType
One of the StatementType values that specifies the type of query executed.

tableMapping
Type: System.Data.Common.:::DataTableMapping
The DataTableMapping sent through an Update(DataSet).
See Also

MySqlRowUpdatingEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlRowUpdatingEventArgs` type exposes the following members.
<table>
<thead>
<tr>
<th><strong>Methods</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Equals</td>
</tr>
<tr>
<td>Finalize</td>
</tr>
<tr>
<td>GetHashCode</td>
</tr>
<tr>
<td>GetType</td>
</tr>
<tr>
<td>MemberwiseClone</td>
</tr>
<tr>
<td>ToString</td>
</tr>
</tbody>
</table>
See Also

MySqlRowUpdatingEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlRowUpdatingEventArgs` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BaseCommand</code></td>
<td>Gets or sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.idbcommand">IDbCommand</a> object for an instance of this class.</td>
</tr>
<tr>
<td><code>command</code></td>
<td>Gets or sets the MySqlCommand to execute when performing the Update.</td>
</tr>
<tr>
<td><code>Errors</code></td>
<td>Gets any errors generated by the .NET Framework data provider when the <code>Command</code> executes. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.rowupdatingeventargs">RowUpdatingEventArgs</a>.)</td>
</tr>
<tr>
<td><code>Row</code></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.datarow">DataRow</a> that will be sent to the server as part of an insert, update, or delete operation. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.rowupdatingeventargs">RowUpdatingEventArgs</a>.)</td>
</tr>
<tr>
<td><code>StatementType</code></td>
<td>Gets the type of SQL statement to execute. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.rowupdatingeventargs">RowUpdatingEventArgs</a>.)</td>
</tr>
<tr>
<td><code>Status</code></td>
<td>Gets or sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.rowupdatingstatus">UpdateStatus</a> of the <code>Command</code> property. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.rowupdatingeventargs">RowUpdatingEventArgs</a>.)</td>
</tr>
<tr>
<td><code>TableMapping</code></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.datatablemapping">DataTableMapping</a> to send through the <code>Update(DataSet)</code>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.rowupdatingeventargs">RowUpdatingEventArgs</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlRowUpdatingEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL RowUpdatingEventArgs Command Property

Gets or sets the MySqlCommand to execute when performing the Update.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlCommand Command { get; set; }

Visual Basic (Declaration)

Public Property Command As MySqlCommand

Visual C++

public:
property MySqlCommand^ Command { 
    MySqlCommand^ get ();
    void set (MySqlCommand^ value);
}
See Also

MySqlRowUpdatingEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents the method that will handle the `RowUpdating` event of a `MySqlCommandAdapter`.

**Namespace:** `System.Data.SqlClient`

Syntax

C#

```csharp
public delegate void MySqlRowUpdatingEventHandler(
    Object sender,
    MySqlRowUpdatingEventArgs e
)
```

Visual Basic (Declaration)

```vbnet
Public Delegate Sub MySqlRowUpdatingEventHandler ( _
    sender As Object, _
    e As MySqlRowUpdatingEventArgs _
)
```

Visual C++

```c++
public delegate void MySqlRowUpdatingEventHandler( 
    Object^ sender,
    MySqlRowUpdatingEventArgs^ e
)
```

Parameters

sender
  Type: System::Object

e
  Type: MySql.Data.MySqlClient::MySqlRowUpdatingEventArgs
See Also

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Provides a class capable of executing a SQL script containing multiple SQL statements including CREATE PROCEDURE statements that require changing the delimiter

**Namespace:** MySql.Data.MySqlClient  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#
public class MySqlScript

Visual Basic (Declaration)
Public Class MySqlScript

Visual C++
public ref class MySqlScript
Inheritance Hierarchy

System...:::Object
MySql.Data.MySqlClient...:::MySqlScript
See Also

MySqlScript Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlScript` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlScript</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td>Executes this instance.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td><strong>Connection</strong></td>
<td>Gets or sets the connection.</td>
</tr>
<tr>
<td><strong>Delimiter</strong></td>
<td>Gets or sets the delimiter.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Gets or sets the query.</td>
</tr>
</tbody>
</table>
# Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>ScriptCompleted</td>
<td></td>
</tr>
<tr>
<td>StatementExecuted</td>
<td></td>
</tr>
</tbody>
</table>
See Also

**MySqlScript Class**  
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MySqlScript()</code></td>
<td>Initializes a new instance of the <code>MySqlScript</code> class.</td>
</tr>
<tr>
<td><code>MySqlScript(MySqlConnection)</code></td>
<td>Initializes a new instance of the <code>MySqlScript</code> class.</td>
</tr>
<tr>
<td><code>MySqlScript(String)</code></td>
<td>Initializes a new instance of the <code>MySqlScript</code> class.</td>
</tr>
<tr>
<td><code>MySqlScript(MySqlConnection, String)</code></td>
<td>Initializes a new instance of the <code>MySqlScript</code> class.</td>
</tr>
</tbody>
</table>
See Also

MySqlScript Class
MySqlScript Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlScript` class.

**Namespace:**  [MySql.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlScript()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySqlScript()
See Also

MySqlScript Class
MySqlScript Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlScript` class.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySQL.Data](#) (in `MySQL.Data.dll`) Version: 6.2.2.0
Syntax

C#

public MySqlScript(
    MySqlConnection connection
)

Visual Basic (Declaration)

Public Sub New (_
    connection As MySqlConnection _
)

Visual C++

public:
MySqlScript(
    MySqlConnection^ connection
)

Parameters

connection
Type: MySql.Data.MySqlClient::MySqlConnection
The connection.
See Also

MySqlScript Class
MySqlScript Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlScript` class.

**Namespace:**  `MySql.Data.MySqlClient`  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public MySqlScript(
    string query
)

Visual Basic (Declaration)

Public Sub New (_
    query As String _
)

Visual C++

public:
MySqlScript(
    String^ query
)

Parameters

query
    Type: System::String
    The query.
See Also

MySqlScript Class
MySqlScript Overload
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLScript Constructor (MySqlConnection, String)

**MySQLScript Class**  [See Also]  [Send Feedback]

Initializes a new instance of the **MySQLScript** class.

**Namespace:**  MySql.Data.MySqlClient

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlScript(
    MySqlConnection connection,
    string query
)

Visual Basic (Declaration)

Public Sub New (_
    connection As MySqlConnection, _
    query As String _
)

Visual C++

public:
MySqlScript(
    MySqlConnection^ connection,
    String^ query
)

Parameters

connection
    Type: MySql.Data.MySqlClient::MySqlConnection
    The connection.

query
    Type: System::String
    The query.
See Also

MySQLScript Class
MySQLScript Overload
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Script Type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td>Executes this instance.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlScript Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Executes this instance.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Execute()

Visual Basic (Declaration)

Public Function Execute As Integer

Visual C++

public:
    int Execute()

Return Value

The number of statements executed as part of the script.
See Also

MySqlScript Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlScript` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection</strong></td>
<td>Gets or sets the connection.</td>
</tr>
<tr>
<td><strong>Delimiter</strong></td>
<td>Gets or sets the delimiter.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Gets or sets the query.</td>
</tr>
</tbody>
</table>
See Also

MySqlScript Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the connection.

**Namespace:**  [MySql.Data.MySqlClient](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public MySqlConnection Connection { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property Connection As MySqlConnection
```

### Visual C++

```cpp
public:
property MySqlConnection^ Connection {
    MySqlConnection^ get ();
    void set (MySqlConnection^ value);
}
```

## Field Value

The connection.
See Also

MySQLScript Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the delimiter.

**Namespace:** [MySQL.Data.MySqlClient](https://github.com/mysql/connector-net)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string Delimiter { get; set; }

Visual Basic (Declaration)

Public Property Delimiter As String

Visual C++

public:
property String^ Delimiter {
String^ get ();
void set (String^ value);
}

Field Value

The delimiter.
See Also

MySqlScript Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the query.

**Namespace:** [MySQL.Data.MySqlClient](https://example.com)
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public string Query { get; set; }

Visual Basic (Declaration)

Public Property Query As String

Visual C++

public:
    property String^ Query {
        String^ get ();
        void set (String^ value);
    }

Field Value

The query.
See Also

MySQLScript Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlScript` type exposes the following members.
# Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>ScriptCompleted</td>
<td></td>
</tr>
<tr>
<td>StatementExecuted</td>
<td></td>
</tr>
</tbody>
</table>
See Also

**MySqlScript Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

public event MySqlScriptErrorEventHandler Error

**Visual Basic (Declaration)**

Public Event Error As MySqlScriptErrorEventHandler

**Visual C++**

public:

  event MySqlScriptErrorEventHandler^ Error {
    void add (MySqlScriptErrorEventHandler^ value);
    void remove (MySqlScriptErrorEventHandler^ value);
  }
See Also

MySQLScript Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLScript...:::ScriptCompleted Event

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public event EventHandler ScriptCompleted
```

Visual Basic (Declaration)

```vbnet
Public Event ScriptCompleted As EventHandler
```

Visual C++

```cpp
public:
    event EventHandler^ ScriptCompleted {
        void add (EventHandler^ value);
        void remove (EventHandler^ value);
    }
```
See Also

MySqlScript Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
MySQL Connector/Net
MySQLScript...:::StatementExecuted Event
MySQLScript Class  See Also  Send Feedback

**Namespace:**  MySQL.Data.MySqlClient
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public event MySqlStatementExecutedEventHandler StatementExecuted

Visual Basic (Declaration)

Public Event StatementExecuted As MySqlStatementExecutedEventHandler

Visual C++

public:
    event MySqlStatementExecutedEventHandler^ StatementExecuted {
        void add (MySqlStatementExecutedEventHandler^ value);
        void remove (MySqlStatementExecutedEventHandler^ value);
    }

See Also

MySQLScript Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public class MySqlScriptErrorEventArgs : MySqlScriptEventArgs

Visual Basic (Declaration)

Public Class MySqlScriptErrorEventArgs
    Inherits MySqlScriptEventArgs

Visual C++

public ref class MySqlScriptErrorEventArgs : public MySqlScriptEvent
Inheritance Hierarchy

System..:::.Object
System..:::.EventArgs
   MySql.Data.MySqlClient..:::.MySqlScriptEventArgs
   MySql.Data.MySqlClient..:::.MySqlScriptErrorEventArgs
See Also

MySqlScriptErrorEventArgs Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlScriptEventArgs` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlScriptErrorEventArgs</td>
<td>Initializes a new instance of the MySqlScriptErrorEventArgs class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exception</strong></td>
<td>Gets the exception.</td>
</tr>
<tr>
<td><strong>Ignore</strong></td>
<td>Gets or sets a value indicating whether this MySqlScriptErrorEventArgs is ignore.</td>
</tr>
<tr>
<td><strong>Line</strong></td>
<td>Gets the line.</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>(Inherited from MySqlScriptEventArgs.)</td>
</tr>
<tr>
<td><strong>StatementText</strong></td>
<td>Gets the statement text.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from MySqlScriptEventArgs.)</td>
</tr>
</tbody>
</table>
See Also

 MySqlScriptErrorEventArgs Class
 MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes a new instance of the `MySqlScriptErrorEventArgs` class.

**Namespace:** `MySQL.Data.MySqlClient`  
**Assembly:** `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public MySqlScriptErrorEventArgs(
    Exception exception
)

Visual Basic (Declaration)

Public Sub New (_
    exception As Exception _
)

Visual C++

public:
MySqlScriptErrorEventArgs(
    Exception^ exception
)

Parameters

exception
    Type: System::Exception
    The exception.
See Also

MySqlScriptErrorEventArgs Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlScriptEventArgs` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlScriptErrorEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLScriptEventArgs` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exception</strong></td>
<td>Gets the exception.</td>
</tr>
<tr>
<td><strong>Ignore</strong></td>
<td>Gets or sets a value indicating whether this MySqlScriptErrorEventArgs is ignore.</td>
</tr>
<tr>
<td><strong>Line</strong></td>
<td>Gets the line.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from MySqlScriptEventArgs.)</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>Gets the position.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from MySqlScriptEventArgs.)</td>
</tr>
<tr>
<td><strong>StatementText</strong></td>
<td>Gets the statement text.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from MySqlScriptEventArgs.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlScriptErrorEventArgs Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlScriptErrorEventArgs Class  See Also  Send Feedback

Gets the exception.

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

#### C#

```csharp
public Exception Exception { get; }
```

#### Visual Basic (Declaration)

```
Public ReadOnly Property Exception As Exception
```

#### Visual C++

```
public:
property Exception^ Exception { Exception^ get ();
}
```

### Field Value

The exception.
See Also

MySqlCommandEventArgs Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLScriptErrorEventArgs::Ignore Property

Gets or sets a value indicating whether this MySQLScriptErrorEventArgs is ignore.

Namespace: MySql.Data.MySqlClient
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public bool Ignore { get; set; }

Visual Basic (Declaration)

Public Property Ignore As Boolean

Visual C++

public:
    property bool Ignore {
        bool get ();
        void set (bool value);
    }

Field Value

true if ignore; otherwise, false.
See Also

`MySqlScriptErrorEventArgs Class`
`MySql.Data.MySqlClient Namespace`

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public delegate void MySqlScriptErrorEventHandler(
    Object sender,
    MySqlScriptErrorEventArgs args
)
```

Visual Basic (Declaration)

```vbnet
Public Delegate Sub MySqlScriptErrorEventHandler ( _
    sender As Object,
    args As MySqlScriptErrorEventArgs _
)
```

Visual C++

```cpp
public delegate void MySqlScriptErrorEventHandler(
    Object^ sender,
    MySqlScriptErrorEventArgs^ args
)
```

Parameters

sender
Type: `System::Object`

args
Type: `MySql.Data.MySqlClient::MySqlScriptErrorEventArgs`
See Also

**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public class MySqlScriptEventArgs : EventArgs
```

**Visual Basic (Declaration)**

```vbnet
Public Class MySqlScriptEventArgs
    Inherits EventArgs
```

**Visual C++**

```cpp
public ref class MySqlScriptEventArgs : public EventArgs
```
Inheritance Hierarchy

System::Object
  System::EventArgs
    MySql.Data.MySqlClient::MySqlScriptEventArgs
      MySql.Data.MySqlClient::MySqlScriptErrorEventArgs
See Also

MySqlScriptEventArgs Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlScriptEventArgs` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlScriptEventArgs</td>
<td></td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Gets the line.</td>
</tr>
<tr>
<td>Position</td>
<td>Gets the position.</td>
</tr>
<tr>
<td>StatementText</td>
<td>Gets the statement text.</td>
</tr>
</tbody>
</table>
See Also

MySqlScriptEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLScriptEventArgs Constructor

**Namespace:** MySql.Data.MySqlClient

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public MySqlScriptEventArgs()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
MySqlScriptEventArgs()
```
See Also

MySQLScriptEventArgs Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlScriptEventArgs` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlCommandEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The MySqlScriptEventArgs type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Gets the line.</td>
</tr>
<tr>
<td>Position</td>
<td>Gets the position.</td>
</tr>
<tr>
<td>StatementText</td>
<td>Gets the statement text.</td>
</tr>
</tbody>
</table>
See Also

MySqlScriptEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Wins the line.

**Namespace:**  [MySQL.Data.MySqlClient](#)
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public int Line { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Line As Integer
```

**Visual C++**

```cpp
public:
    property int Line {
        int get ();
    }
```

**Field Value**

The line.
See Also

MySQLScriptEventArgs Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the position.

**Namespace:**  [MySQL.Data.MySqlClient](https://www.mysql.com/)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public int Position { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Position As Integer
```

**Visual C++**

```cpp
public:
property int Position {
    int get ();
}
```

**Field Value**

The position.
See Also

MySqlScriptEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the statement text.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

public string StatementText { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property StatementText As String

**Visual C++**

public:
    property String^ StatementText {
        String^ get ();
    }

**Field Value**

The statement text.
See Also

MySqlCommandEventArg
MySqlCommandEventArgs Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
SSL options for connection.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public enum MySqlSslMode

Visual Basic (Declaration)

Public Enumeration MySqlSslMode

Visual C++

public enum class MySqlSslMode
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Do not use SSL.</td>
</tr>
<tr>
<td>Preferred</td>
<td>Use SSL, if server supports it.</td>
</tr>
<tr>
<td>Prefered</td>
<td>Use SSL, if server supports it.</td>
</tr>
<tr>
<td>Required</td>
<td>Always use SSL. Deny connection if server does not support SSL. Do not perform server certificate validation.</td>
</tr>
<tr>
<td>VerifyCA</td>
<td>Always use SSL. Validate server SSL certificate, but different host name mismatch.</td>
</tr>
<tr>
<td>VerifyFull</td>
<td>Always use SSL and perform full certificate validation.</td>
</tr>
</tbody>
</table>
See Also

`MySql.Data.MySqlClient Namespace`

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public delegate void MySqlStatementExecutedEventHandler(
    Object sender,
    MySqlScriptEventArgs args
)

Visual Basic (Declaration)

Public Delegate Sub MySqlStatementExecutedEventHandler ( _
    sender As Object, _
    args As MySqlScriptEventArgs _
)

Visual C++

public delegate void MySqlStatementExecutedEventHandler(
    Object^ sender,
    MySqlScriptEventArgs^ args
)

Parameters

sender
  Type: System:::Object

args
  Type: MySql.Data.MySqlClient:::MySqlScriptEventArgs
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#
public class MySqlTrace

Visual Basic (Declaration)
Public Class MySqlTrace

Visual C++
public ref class MySqlTrace
Inheritance Hierarchy

System..:::Object
 MySql.Data.MySqlClient..:::MySqlTrace
See Also

MySqlTrace Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLTrace` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlTrace</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Listeners" /></td>
<td>Listeners</td>
</tr>
<tr>
<td><img src="image" alt="Switch" /></td>
<td>Switch</td>
</tr>
</tbody>
</table>
See Also

MySqlTrace Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLTrace Constructor

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlTrace()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySqlTrace()
See Also

**MySqlTrace Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLTrace` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlTrace Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLTrace** type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listeners</td>
<td></td>
</tr>
<tr>
<td>Switch</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlTrace Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Trace...:.Listeners Property

Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static TraceListenerCollection Listeners { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property Listeners As TraceListenerCollection

Visual C++

public:
static property TraceListenerCollection^ Listeners {
    TraceListenerCollection^ get ();
}


See Also

MySQLTrace Class
MySQL.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLTrace Class

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static SourceSwitch Switch { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Shared Property Switch As SourceSwitch
```

Visual C++

```cpp
public:
static property SourceSwitch^ Switch {
    SourceSwitch^ get ();
    void set (SourceSwitch^ value);
}
```
See Also

**MySqlTrace Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public enum MySqlTraceEventType

Visual Basic (Declaration)

Public Enumeration MySqlTraceEventType

Visual C++

public enum class MySqlTraceEventType
# Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionOpened</td>
<td></td>
</tr>
<tr>
<td>ConnectionClosed</td>
<td></td>
</tr>
<tr>
<td>QueryOpened</td>
<td></td>
</tr>
<tr>
<td>ResultOpened</td>
<td></td>
</tr>
<tr>
<td>ResultClosed</td>
<td></td>
</tr>
<tr>
<td>QueryClosed</td>
<td></td>
</tr>
<tr>
<td>StatementPrepared</td>
<td></td>
</tr>
<tr>
<td>StatementExecuted</td>
<td></td>
</tr>
<tr>
<td>StatementClosed</td>
<td></td>
</tr>
<tr>
<td>NonQuery</td>
<td></td>
</tr>
<tr>
<td>UsageAdvisorWarning</td>
<td></td>
</tr>
<tr>
<td>Warning</td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>QueryNormalized</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Represents a SQL transaction to be made in a MySQL database. This class cannot be inherited.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  [MySQL.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

public sealed class MySqlTransaction : DbTransaction

Visual Basic (Declaration)

Public NotInheritable Class MySqlTransaction
    Inherits DbTransaction

Visual C++

public ref class MySqlTransaction sealed : public DbTransaction
Remarks

The application creates a MySqlConnection object by calling `BeginTransaction()` on the MySqlConnection object. All subsequent operations associated with the transaction (for example, committing or aborting the transaction), are performed on the MySqlTransaction object.
Examples

The following example creates a `MySqlConnection` and a `MySqlTransaction`. It also demonstrates how to use the `BeginTransaction()`, `Commit()` and `Rollback()` methods.

**VB.NET**

```vbnet
Public Sub RunTransaction(myConnString As String)
Dim myConnection As New MySqlConnection(myConnString)
myConnection.Open()

Dim myCommand As MySqlCommand = myConnection.CreateCommand()
Dim myTrans As MySqlTransaction

' Start a local transaction
myTrans = myConnection.BeginTransaction()
' Must assign both transaction object and connection
' to Command object for a pending local transaction
myCommand.Connection = myConnection
myCommand.Transaction = myTrans

Try
myCommand.CommandText = "Insert into Region (RegionID, RegionDescription) VALUES (100, 'Description')"
myCommand.ExecuteNonQuery()
myCommand.CommandText = "Insert into Region (RegionID, RegionDescription) VALUES (101, 'Description')"
myCommand.ExecuteNonQuery()
myTrans.Commit()
Console.WriteLine("Both records are written to database.")
Catch e As Exception
Try
myTrans.Rollback()
Catch ex As MySqlException
If Not myTrans.Connection Is Nothing Then
Console.WriteLine("An exception of type " & ex.GetType().ToString() & " was encountered while attempting to roll back the transaction.")
End If
End Try

Console.WriteLine("An exception of type " & e.GetType().ToString() & " was encountered while inserting the data.")
Finally
myConnection.Close()
End Try
```
public void RunTransaction(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();

    MySqlCommand myCommand = myConnection.CreateCommand();
    MySqlTransaction myTrans;

    // Start a local transaction
    myTrans = myConnection.BeginTransaction();
    // Must assign both transaction object and connection
    // to Command object for a pending local transaction
    myCommand.Connection = myConnection;
    myCommand.Transaction = myTrans;

    try
    {
        myCommand.CommandText = "Insert into Region (RegionID, RegionDescription)
Values (100, 'Region Description');"
        myCommand.ExecuteNonQuery();
        myCommand.CommandText = "Insert into Region (RegionID, RegionDescription)
Values (101, 'Region Description');"
        myCommand.ExecuteNonQuery();
        myTrans.Commit();
        Console.WriteLine("Both records are written to database.");
    }
    catch (Exception e)
    {
        try
        {
            myTrans.Rollback();
        }
        catch (MySqlException ex)
        {
            if (myTrans.Connection != null)
            {
                Console.WriteLine("An exception of type " + ex.GetType() + 
" was encountered while attempting to roll back the transaction.");
            }
        }
    }

    Console.WriteLine("An exception of type " + e.GetType() + 
" was encountered while inserting the data.");
    Console.WriteLine("Neither record was written to database.");
}
finally
{
myConnection.Close();
}
}
Inheritance Hierarchy

System..:::Object
  System..:::MarshalByRefObject
    System.Data.Common..:::DbTransaction
      MySql.Data.MySqlClient..:::MySqlTransaction
See Also

MySqlTransaction Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlTransaction` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Commit</strong></td>
<td>Commits the database transaction.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbTransaction::&lt;&gt;::Commit()</code>.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Creates an object that contains all the relevant information required to</td>
</tr>
<tr>
<td></td>
<td>generate a proxy used to communicate with a remote object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetLifetimeService</strong></td>
<td>Retrieves the current lifetime service object that controls the lifetime</td>
</tr>
<tr>
<td></td>
<td>policy for this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>InitializeLifetimeService</strong></td>
<td>Obtains a lifetime service object to control the</td>
</tr>
<tr>
<td></td>
<td>lifetime policy for this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Rollback</strong></td>
<td>Rolls back a transaction from a pending state.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>DbTransaction::&lt;&gt;::Rollback()</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Gets the <a href="#">MySqlConnection</a> object associated with the transaction, or a null reference (Nothing in Visual Basic) if the transaction is no longer valid.</td>
</tr>
<tr>
<td>DbConnection</td>
<td>Specifies the <a href="#">DbConnection</a> object associated with the transaction.</td>
</tr>
<tr>
<td>IsolationLevel</td>
<td>Specifies the <a href="#">IsolationLevel</a> for this transaction.</td>
</tr>
</tbody>
</table>

*(Inherited from [DbTransaction](#).)* *(Overrides [DbTransaction::IsolationLevel](#).)*
See Also

MySqlTransaction Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLTransaction** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Commit</strong></td>
<td>Commits the database transaction. (Overrides <code>DbTransaction..:::Commit()</code>.)</td>
</tr>
<tr>
<td><strong>CreateObjRef</strong></td>
<td>Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetLifetimeService</strong></td>
<td>Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>InitializeLifetimeService</strong></td>
<td>Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from <code>MarshalByRefObject</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Rollback</strong></td>
<td>Rolls back a transaction from a pending state. (Overrides <code>DbTransaction..:::Rollback()</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlTransaction Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Commits the database transaction.

**Namespace:**  [MySQL.Data.MySqlClient](#)  
**Assembly:**  `MySql.Data` (in `MySql.Data.dll`) Version: 6.2.2.0
Syntax

C#

public override void Commit()

Visual Basic (Declaration)

Public Overrides Sub Commit

Visual C++

public:
virtual void Commit() override

Implements

IDbTransaction...::Commit()()
Remarks

The **Commit** method is equivalent to the MySQL SQL statement `COMMIT`. 
Examples

The following example creates a MySqlConnection and a MySqlTransaction. It also demonstrates how to use the BeginTransaction(), Commit(), and Rollback methods.

VB.NET

Public Sub RunSqlTransaction(myConnString As String)
Dim myConnection As New MySqlConnection(myConnString)
myConnection.Open()

Dim myCommand As MySqlCommand = myConnection.CreateCommand()
Dim myTrans As MySqlTransaction

' Start a local transaction
myTrans = myConnection.BeginTransaction()

' Must assign both transaction object and connection
' to Command object for a pending local transaction
myCommand.Connection = myConnection
myCommand.Transaction = myTrans

Try
myCommand.CommandText = "Insert into mytable (id, desc) VALUES (100,
myCommand.ExecuteNonQuery()
myCommand.CommandText = "Insert into mytable (id, desc) VALUES (101,
myCommand.ExecuteNonQuery()
myTrans.Commit()
Console.WriteLine("Success."")
Catch e As Exception
Try
myTrans.Rollback()
Catch ex As MySqlException
If Not myTrans.Connection Is Nothing Then
Console.WriteLine("An exception of type " & ex.GetType().ToString()
" was encountered while attempting to roll back the transaction.")
End If
End Try

Console.WriteLine("An exception of type " & e.GetType().ToString() &
" was encountered while inserting the data.")
Console.WriteLine("Neither record was written to database.")
Finally
myConnection.Close()
public void RunSqlTransaction(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();

    MySqlCommand myCommand = myConnection.CreateCommand();
    MySqlTransaction myTrans;

    // Start a local transaction
    myTrans = myConnection.BeginTransaction();
    // Must assign both transaction object and connection
    // to Command object for a pending local transaction
    myCommand.Connection = myConnection;
    myCommand.Transaction = myTrans;

    try
    {
        myCommand.CommandText = "Insert into mytable (id, desc) VALUES (100,"
        myCommand.ExecuteNonQuery();
        myCommand.CommandText = "Insert into mytable (id, desc) VALUES (101,"
        myCommand.ExecuteNonQuery();
        myTrans.Commit();
        Console.WriteLine("Both records are written to database.");
    }
    catch(Exception e)
    {
        try
        {
            myTrans.Rollback();
        }
        catch (MySqlException ex)
        {
            if (myTrans.Connection != null)
            {
                Console.WriteLine("An exception of type " + ex.GetType() +
                " was encountered while attempting to roll back the transaction.");
            }
        }
    }

    Console.WriteLine("An exception of type " + e.GetType() +
    " was encountered while inserting the data.");
    Console.WriteLine("Neither record was written to database.");
}
finally
{ myConnection.Close(); }
}
See Also

MySqlTransaction Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlTransaction....Dispose Method

See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="function" alt="Dispose()" /></td>
<td>Releases the unmanaged resources used by the DbTransaction. (Inherited from DbTransaction.)</td>
</tr>
<tr>
<td><img src="function" alt="Dispose(Boolean)" /></td>
<td>Releases the unmanaged resources used by the DbTransaction and optionally releases the managed resources. (Inherited from DbTransaction.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlTransaction Class**
**MySqlTransaction Members**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Transaction:::MemberwiseClone Method

MySQLTransaction Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberwiseClone()()</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone(Boolean)</td>
<td>Creates a shallow copy of the current MarshalByRefObject object. (Inherited from MarshalByRefObject.)</td>
</tr>
</tbody>
</table>
See Also

MySqlTransaction Class
MySqlTransaction Members
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Rolls back a transaction from a pending state.

Namespace:  MySQL.Data.MySqlClient  
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override void Rollback()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub Rollback
```

**Visual C++**

```cpp
public:
virtual void Rollback() override
```

**Implements**

```csharp
IDbTransaction::Rollback()
```
Remarks

The Rollback method is equivalent to the MySQL statement ROLLBACK. The transaction can only be rolled back from a pending state (after BeginTransaction has been called, but before Commit is called).
Examples

The following example creates a `MySqlConnection` and a `MySqlTransaction`. It also demonstrates how to use the `BeginTransaction()`, `Commit()`, and Rollback methods.

**VB.NET**

```vbnet
Public Sub RunSqlTransaction(myConnString As String)
    Dim myConnection As New MySqlConnection(myConnString)
    myConnection.Open()

    Dim myCommand As MySqlCommand = myConnection.CreateCommand()
    Dim myTrans As MySqlTransaction

    ' Start a local transaction
    myTrans = myConnection.BeginTransaction()

    ' Must assign both transaction object and connection
    ' to Command object for a pending local transaction
    myCommand.Connection = myConnection
    myCommand.Transaction = myTrans

    Try
        myCommand.CommandText = "Insert into mytable (id, desc) VALUES (100,
            myCommand.ExecuteNonQuery()
        myCommand.CommandText = "Insert into mytable (id, desc) VALUES (101,
            myCommand.ExecuteNonQuery()
        myTrans.Commit()
        Console.WriteLine("Success."")
    Catch e As Exception
        Try
            myTrans.Rollback()
        Catch ex As MySqlException
            If Not myTrans.Connection Is Nothing Then
                Console.WriteLine("An exception of type ", e.GetType().ToString() & 
                    "was encountered while attempting to roll back the transaction.")
            End If
        End Try

        Console.WriteLine("An exception of type ", e.GetType().ToString() &  "was encountered while inserting the data.")
    Console.WriteLine("Neither record was written to database.")
    Finally
    myConnection.Close()
```

www.example.com
public void RunSqlTransaction(string myConnString)
{
    MySqlConnection myConnection = new MySqlConnection(myConnString);
    myConnection.Open();

    MySqlCommand myCommand = myConnection.CreateCommand();
    MySqlTransaction myTrans;

    // Start a local transaction
    myTrans = myConnection.BeginTransaction();
    // Must assign both transaction object and connection
    // to Command object for a pending local transaction
    myCommand.Connection = myConnection;
    myCommand.Transaction = myTrans;

    try
    {
        myCommand.CommandText = "Insert into mytable (id, desc) VALUES (100,
        myCommand.ExecuteNonQuery();
        myCommand.CommandText = "Insert into mytable (id, desc) VALUES (101,
        myCommand.ExecuteNonQuery();
        myTrans.Commit();
        Console.WriteLine("Both records are written to database.");
    }
    catch(Exception e)
    {
        try
        {
            myTrans.Rollback();
        }
        catch (MySqlException ex)
        {
            if (myTrans.Connection != null)
            {
                Console.WriteLine("An exception of type " + ex.GetType() +
                " was encountered while attempting to roll back the transaction.");
            }
        }
    }
    finally
    {
        Console.WriteLine("An exception of type " + e.GetType() +
        " was encountered while inserting the data.");
        Console.WriteLine("Neither record was written to database.");
    }
}
{  
myConnection.Close();  
}  
}
See Also

MySqlTransaction Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlTransaction` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.mysqlconnection?view=netcore31">MySqlConnection</a> object associated with the transaction, or a null reference (Nothing in Visual Basic) if the transaction is no longer valid.</td>
</tr>
<tr>
<td><strong>DbConnection</strong></td>
<td>Specifies the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.dbconnection?view=netcore31">DbConnection</a> object associated with the transaction. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.dbtransaction?view=netcore31">DbTransaction</a>.)</td>
</tr>
<tr>
<td><strong>IsolationLevel</strong></td>
<td>Specifies the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.isolationlevel?view=netcore31">IsolationLevel</a> for this transaction. (Overrides <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.dbtransaction?view=netcore31">DbTransaction</a>.)</td>
</tr>
</tbody>
</table>
See Also

**MySqlTransaction Class**
**MySql.Data.MySqlClient Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the `MySqlConnection` object associated with the transaction, or a null reference (Nothing in Visual Basic) if the transaction is no longer valid.

**Namespace:**  [MySql.Data.MySqlClient](https://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public MySqlConnection Connection { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Connection As MySqlConnection
```

**Visual C++**

```cpp
public:
property MySqlConnection^ Connection {
    MySqlConnection^ get ();
}
```

**Field Value**

The `MySqlConnection` object associated with this transaction.
Remarks

A single application may have multiple database connections, each with zero or more transactions. This property enables you to determine the connection object associated with a particular transaction created by `BeginTransaction()`.
See Also

MySqlTransaction Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Specifies the IsolationLevel for this transaction.

**Namespace:**  [MySQL.Data.MySqlClient](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override IsolationLevel IsolationLevel { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property IsolationLevel As IsolationLevel

Visual C++

public:
virtual property IsolationLevel IsolationLevel {
    IsolationLevel get () override;
}

Field Value

The IsolationLevel for this transaction. The default is ReadCommitted.

Implements

IDbTransaction::IsolationLevel
Remarks

Parallel transactions are not supported. Therefore, the IsolationLevel applies to the entire transaction.
See Also

MySqlConnection Class
MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.MySqlClient
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public enum UsageAdvisorWarningFlags

Visual Basic (Declaration)

Public Enumeration UsageAdvisorWarningFlags

Visual C++

public enum class UsageAdvisorWarningFlags
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoIndex</td>
<td></td>
</tr>
<tr>
<td>BadIndex</td>
<td></td>
</tr>
<tr>
<td>SkippedRows</td>
<td></td>
</tr>
<tr>
<td>SkippedColumns</td>
<td></td>
</tr>
<tr>
<td>FieldConversion</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySql.Data.MySqlClient Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  □ Visual Basic  □ Visual C++
MySQL Connector/Net
MySQL.Data.MySqlClient.Properties Namespace
Send Feedback
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources</td>
<td>A strongly-typed resource class, for looking up localized</td>
</tr>
<tr>
<td></td>
<td>strings, etc.</td>
</tr>
</tbody>
</table>

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
A strongly-typed resource class, for looking up localized strings, etc.

Namespace:  **MySQL.Data.MySqlClient.Properties**
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public class Resources

Visual Basic (Declaration)

Public Class Resources

Visual C++

public ref class Resources
Inheritance Hierarchy

System::Object
MySql.Data.MySqlClient.Properties::Resources
The `Resources` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>AdapterIsNull</td>
<td>Looks up a localized string similar to Improper MySqlCommandBuilder state: adapter is null.</td>
</tr>
<tr>
<td>AdapterSelectIsNull</td>
<td>Looks up a localized string similar to Improper MySqlCommandBuilder state: adapter's SelectCommand is null.</td>
</tr>
<tr>
<td>AttemptToAccessBeforeRead</td>
<td>Looks up a localized string similar to Invalid attempt to access a field before calling Read().</td>
</tr>
<tr>
<td>BadVersionFormat</td>
<td>Looks up a localized string similar to Version string not in acceptable format.</td>
</tr>
<tr>
<td>BufferCannotBeNull</td>
<td>Looks up a localized string similar to The buffer cannot be null.</td>
</tr>
<tr>
<td>BufferNotLargeEnough</td>
<td>Looks up a localized string similar to Buffer is not large enough.</td>
</tr>
<tr>
<td>CancelNeeds50</td>
<td>Looks up a localized string similar to Canceling an executing query requires MySQL 5.0 or higher.</td>
</tr>
<tr>
<td>CancelNotSupported</td>
<td>Looks up a localized string similar to Canceling an active query is only supported on MySQL 5.0.0 and above.</td>
</tr>
</tbody>
</table>
CBMultiTableNotSupported

similar to MySqlCommandBuilder does not support multi-table statements.

Looks up a localized string similar to MySqlCommandBuilder cannot operate on tables with no unique or key columns.

CBNoKeyColumn

Looks up a localized string similar to Chaos isolation level is not supported.

ChaosNotSupported

Looks up a localized string similar to The CommandText property has not been properly initialized..

CommandTextNotInitialized

Looks up a localized string similar to The connection is already open..

ConnectionAlreadyOpen

Looks up a localized string similar to Connection unexpectedly terminated..

ConnectionBroken

Looks up a localized string similar to Connection must be valid and open.

ConnectionMustBeOpen

Looks up a localized string similar to The connection is not open..

ConnectionNotOpen

Looks up a localized string similar to The connection property has not been set or is null..

ConnectionNotSet

CouldNotFindColumnName

Looks up a localized string similar to Could not find specified column in
**CountCannotBeNegative**

Looks up a localized string similar to Count cannot be negative.

**CSNoSetLength**

Looks up a localized string similar to SetLength is not a valid operation on CompressedStream.

**Culture**

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

**DataNotInSupportedFormat**

Looks up a localized string similar to The given value was not in a supported format..

**DataReaderOpen**

Looks up a localized string similar to There is already an open DataReader associated with this Connection which must be closed first..

**DefaultEncodingNotFound**

Looks up a localized string similar to The default connection encoding was not found. Please report this as a bug along with your connection string and system details..

**DistributedTxnNotSupported**

MySQL Connector/Net does not currently support distributed transactions..

**ErrorCreatingSocket**

Looks up a localized string similar to Error creating
socket connection.
Looks up a localized string similar to Fatal error encountered during command execution..
Looks up a localized string similar to Fatal error encountered during data read..
Looks up a localized string similar to Fatal error encountered attempting to read the resultset..
Looks up a localized string similar to From index and length use more bytes than from contains.
Looks up a localized string similar to From index must be a valid index inside the from buffer.
Looks up a localized string similar to Call to GetHostEntry failed after {0} while querying for hostname '{1}':
SocketErrorCode={2},
ErrorCode={3},
NativeErrorCode={4}..
Looks up a localized string similar to Retrieving procedure metadata for {0} from server..
Looks up a localized string similar to Value has an unsupported format..
Looks up a localized string
Incorrect Transmission

Index And Length Too Big

Index Must Be Valid

Invalid Column Ordinal

Invalid Connection String Value

Invalid Proc Name

Invalid Value For Boolean

Keyword No Null

Keyword Not Supported

similar to An incorrect response was received from the server.

Looks up a localized string similar to An incorrect response was received from the server.

Looks up a localized string similar to An incorrect response was received from the server.

Looks up a localized string similar to Index and length use more bytes than to has room for.

Looks up a localized string similar to Index and length use more bytes than to has room for.

Looks up a localized string similar to Index must be a valid position in the buffer.

Looks up a localized string similar to Index must be a valid position in the buffer.

Looks up a localized string similar to You have specified an invalid column ordinal.

Looks up a localized string similar to You have specified an invalid column ordinal.

Looks up a localized string similar to The requested value '{0}' is invalid for the given keyword '{1}'.

Looks up a localized string similar to The requested value '{0}' is invalid for the given keyword '{1}'.

Looks up a localized string similar to You have specified an invalid column ordinal.

Looks up a localized string similar to You have specified an invalid column ordinal.

Looks up a localized string similar to Procedure or function '{0}' cannot be found in database '{1}'.

Looks up a localized string similar to Procedure or function '{0}' cannot be found in database '{1}'.

Looks up a localized string similar to ' {0} ' is an illegal value for a boolean option.

Looks up a localized string similar to ' {0} ' is an illegal value for a boolean option.

Looks up a localized string similar to Keyword does not allow null values.

Looks up a localized string similar to Keyword does not allow null values.

Looks up a localized string similar to Keyword not supported.

Looks up a localized string similar to Keyword not supported.

Looks up a localized string similar to ACCESSIBLE ADD ALL ALTER ANALYZE AND AS ASC ASENSITIVE BEFORE
MoreThanOneOPRow

Looks up a localized string similar to INTERNAL ERROR: More than one output parameter row detected..
MultipleConnectionsInTransactionNotSupported

Looks up a localized string similar to Multiple simultaneous connections or connections with different connection strings inside the same transaction are not currently supported.

NamedPipeNoSeek

Looks up a localized string similar to NamedPipeStream does not support seeking.

NamedPipeNoSetLength

Looks up a localized string similar to NamedPipeStream doesn't support SetLength.

NextResultIsClosed

Looks up a localized string similar to Invalid attempt to call NextResult when the reader is closed.

NoBodiesAndTypeNotSet

Looks up a localized string similar to When calling stored procedures and 'Use Procedure Bodies' is false, all parameters must have their type explicitly set.

NoNestedTransactions

Looks up a localized string similar to Nested transactions are not supported.

NoServerSSLSupport

Looks up a localized string similar to The host {0} does not support SSL connections.

ObjectDisposed

Looks up a localized string similar to The object is not open or has been
Looked up a localized string similar to Offset cannot be negative.

Looked up a localized string similar to Offset must be a valid position in buffer.

Looked up a localized string similar to Parameter '{0}' has already been defined..

Looked up a localized string similar to Parameter cannot have a negative value.

Looked up a localized string similar to Parameter cannot be null.

Looked up a localized string similar to Parameter is invalid..

Looked up a localized string similar to Parameter '{0}' must be defined..

Looked up a localized string similar to Parameter '{0}' was not found during prepare..

Looked up a localized string similar to Password must be valid and contain length characters.

Looked up a localized string similar to This category includes a series of counters for MySQL..

Looked up a localized string similar to .NET Data.
Provider for MySQL.
Looks up a localized string similar to The number of times a procedures metadata had to be queried from the server..
Looks up a localized string similar to Hard Procedure Queries.
Looks up a localized string similar to The number of times a procedures metadata was retrieved from the client-side cache..
Looks up a localized string similar to Soft Procedure Queries.
Looks up a localized string similar to same name are not supported..
Looks up a localized string similar to Packets larger than max_allowed_packet are not allowed..
Looks up a localized string similar to Reading from the stream has failed..
Looks up a localized string similar to Invalid attempt to read a prior column using SequentialAccess.
Returns the cached ResourceManager instance used by this class.
Looks up a localized string similar to Routine '{0}' cannot be found. Either
check the spelling or make sure you have sufficient rights to execute the routine..

Looks up a localized string similar to Connector/Net no longer supports server versions prior to 4.1.

Looks up a localized string similar to Socket streams do not support seeking.

Looks up a localized string similar to Retrieving procedure metadata for {0} from procedure cache..

Looks up a localized string similar to Stored procedures are not supported on this version of MySQL.

Looks up a localized string similar to The stream has already been closed.

Looks up a localized string similar to The stream does not support reading.

Looks up a localized string similar to The stream does not support writing.

Looks up a localized string similar to Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding..
TimeoutGettingConnection

Looks up a localized string similar to error connecting: Timeout expired. The timeout period elapsed prior to obtaining a connection from the pool. This may have occurred because all pooled connections were in use and max pool size was reached.

TraceCloseConnection

Looks up a localized string similar to {0}: Connection Closed.

TraceErrorMoreThanMaxValueConnections

Looks up a localized string similar to Unable to trace. There are more than Int32.MaxValue connections in use.

TraceFetchError

Looks up a localized string similar to {0}: Error encountered during row fetch. Number = {1}, Message={2}.

TraceOpenConnection

Looks up a localized string similar to {0}: Connection Opened: connection string = '{1}'.

TraceOpenResultError

Looks up a localized string similar to {0}: Error encountered attempting to open result: Number={1}, Message={2}.

TraceQueryDone

Looks up a localized string similar to {0}: Query Closed.
TraceQueryNormalized

Looks up a localized string similar to {0}: Query Normalized: {2}.

TraceQueryOpened

Looks up a localized string similar to {0}: Query Opened: {2}.

TraceResult

Looks up a localized string similar to {0}: Resultset Opened: field(s) = {1}, affected rows = {2}, inserted id = {3}.

TraceResultClosed

Looks up a localized string similar to {0}: Resultset Closed. Total rows={1}, skipped rows={2}, size (bytes)={3}.

TraceSetDatabase

Looks up a localized string similar to {0}: Set Database: {1}.

TraceStatementClosed

Looks up a localized string similar to {0}: Statement closed: statement id = {1}.

TraceStatementExecuted

Looks up a localized string similar to {0}: Statement executed: statement id = {1}.

TraceStatementPrepared

Looks up a localized string similar to {0}: Statement prepared: sql='{1}', statement id={2}.

TraceUAWarningBadIndex

Usage Advisor Warning: Query is using a bad index.

TraceUAWarningFieldConversion

Usage Advisor Warning: The field '{2}' was converted
to the following types: {3}.

Looks up a localized string similar to {0}: Usage Advisor Warning: Query does not use an index.

Looks up a localized string similar to {0}: Usage Advisor Warning: The following columns were not accessed: {2}.

Looks up a localized string similar to {0}: Usage Advisor Warning: Skipped {2} rows. Consider a more focused query..

Looks up a localized string similar to {0}: MySql Warning: Level={1}, Code={2}, Message={3}.

Looks up a localized string similar to Unable to connect to any of the specified MySQL hosts..

Looks up a localized string similar to Unable to derive stored routine parameters. The 'Parameters' information schema table is not available and access to the stored procedure body has been disabled..

Looks up a localized string similar to An error occurred attempting to enumerate the user-defined functions. Do you have SELECT
privileges on the mysql.func table.

Looks up a localized string similar to Unable to execute stored procedure '{0}'.

Looks up a localized string similar to Unable to execute stored procedure '{0}'..

Looks up a localized string similar to Unable to retrieve stored routine parameters. Either grant access to the routine or add the 'Use Procedure Bodies=false' option to your connection string..

Looks up a localized string similar to Unable to retrieve stored procedure metadata for routine '{0}'. Either grant SELECT privilege to mysql.proc for this user or use "use procedure bodies=false" with your connection string..

Looks up a localized string similar to Unable to start a second async operation while one is running..

Looks up a localized string similar to Unix sockets are not supported on Windows.

Looks up a localized string
ValueNotSupportedForGuid
similar to The requested column value could not be treated as or converted to a Guid..

WriteToStreamFailed
Looks up a localized string similar to Writing to the stream failed..

WrongParameterName
Looks up a localized string similar to Parameter '{0}' is not found but a parameter with the name '{1}' is found. Parameter names must include the leading parameter marker..
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The Resources type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The Resources type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AdapterIsNull</code></td>
<td>Looks up a localized string similar to Improper MySqlCommandBuilder state: adapter is null.</td>
</tr>
<tr>
<td><code>AdapterSelectIsNull</code></td>
<td>Looks up a localized string similar to Improper MySqlCommandBuilder state: adapter's SelectCommand is null.</td>
</tr>
<tr>
<td><code>AttemptToAccessBeforeRead</code></td>
<td>Looks up a localized string similar to Invalid attempt to access a field before calling Read().</td>
</tr>
<tr>
<td><code>BadVersionFormat</code></td>
<td>Looks up a localized string similar to Version string not in acceptable format.</td>
</tr>
<tr>
<td><code>BufferCannotBeNull</code></td>
<td>Looks up a localized string similar to The buffer cannot be null.</td>
</tr>
<tr>
<td><code>BufferNotLargeEnough</code></td>
<td>Looks up a localized string similar to Buffer is not large enough.</td>
</tr>
<tr>
<td><code>CancelNeeds50</code></td>
<td>Looks up a localized string similar to Canceling an executing query requires MySQL 5.0 or higher..</td>
</tr>
<tr>
<td><code>CancelNotSupported</code></td>
<td>Looks up a localized string similar to Canceling an active query is only supported on MySQL 5.0.0 and above.</td>
</tr>
</tbody>
</table>
CBMultiTableNotSupported

MySqlCommandBuilder does not support multi-table statements.

Looks up a localized string similar to

CBNoKeyColumn

MySqlCommandBuilder cannot operate on tables with no unique or key columns.

Looks up a localized string similar to

ChaosNotSupported

Chaos isolation level is not supported.

Looks up a localized string similar to

CommandTextNotInitialized

The CommandText property has not been properly initialized..

Looks up a localized string similar to

ConnectionAlreadyOpen

The connection is already open..

Looks up a localized string similar to

ConnectionBroken

Connection unexpectedly terminated..

Looks up a localized string similar to

ConnectionMustBeOpen

Connection must be valid and open.

Looks up a localized string similar to

ConnectionNotFound

The connection is not open..

Looks up a localized string similar to

ConnectionNotSet

The connection property has not been set or is null..

Looks up a localized string similar to

CouldNotFindColumnName

Could not find specified column in
CountCannotBeNegative

CSNoSetLength

Culture

DataNotInSupportedFormat

DataReaderOpen

DefaultEncodingNotFound

DistributedTxnNotSupported

ErrorCreatingSocket

results.
Looks up a localized string similar to Count cannot be negative.
Looks up a localized string similar to SetLength is not a valid operation on CompressedStream.
Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.
Looks up a localized string similar to The given value was not in a supported format..
Looks up a localized string similar to There is already an open DataReader associated with this Connection which must be closed first..
Looks up a localized string similar to The default connection encoding was not found. Please report this as a bug along with your connection string and system details..
Looks up a localized string similar to MySQL Connector/Net does not currently support distributed transactions..
Looks up a localized string similar to Error creating
socket connection.
Looks up a localized string similar to Fatal error encountered during command execution..

FatalErrorDuringRead
Looks up a localized string similar to Fatal error encountered during data read..

FatalErrorReadingResult
Looks up a localized string similar to Fatal error encountered attempting to read the resultset..

FromAndLengthTooBig
Looks up a localized string similar to From index and length use more bytes than from contains.

FromIndexMustBeValid
Looks up a localized string similar to From index must be a valid index inside the from buffer.

GetHostEntryFailed
Looks up a localized string similar to Call to GetHostEntry failed after {0} while querying for hostname '{1}':
SocketErrorCode={2}, ErrorCode={3}, NativeErrorCode={4}..

HardProcQuery
Looks up a localized string similar to Retrieving procedure metadata for {0} from server..

ImproperValueFormat
Looks up a localized string similar to Value has an unsupported format..

Looks up a localized string
IncorrectTransmission similar to An incorrect response was received from the server..

IndexAndLengthTooBig Looks up a localized string similar to Index and length use more bytes than to has room for.

IndexMustBeValid Looks up a localized string similar to Index must be a valid position in the buffer.

InvalidColumnOrdinal Looks up a localized string similar to You have specified an invalid column ordinal..

InvalidConnectionStringValue Looks up a localized string similar to The requested value '{0}' is invalid for the given keyword '{1}'..

InvalidProcName Looks up a localized string similar to Procedure or function '{0}' cannot be found in database '{1}'..

InvalidValueForBoolean Looks up a localized string similar to '{0}' is an illegal value for a boolean option..

KeywordNoNull Looks up a localized string similar to Keyword does not allow null values..

KeywordNotSupported Looks up a localized string similar to Keyword not supported..

Looks up a localized string similar to ACCESSIBLE ADD ALL ALTER ANALYZE AND AS ASC ASSENSITIVE BEFORE
Looks up a localized string similar to INTERNAL ERROR: More than one output parameter row detected.
Looks up a localized string similar to *Multiple simultaneous connections or connections with different connection strings inside the same transaction are not currently supported.*

Looks up a localized string similar to *NamedPipeStream does not support seeking.*

Looks up a localized string similar to *NamedPipeStream doesn't support SetLength.*

Looks up a localized string similar to *Invalid attempt to call NextResult when the reader is closed.*

Looks up a localized string similar to *When calling stored procedures and 'Use Procedure Bodies' is false, all parameters must have their type explicitly set.*

Looks up a localized string similar to *Nested transactions are not supported.*

Looks up a localized string similar to *The host {0} does not support SSL connections.*

Looks up a localized string similar to *The object is not open or has been*
disposed..
Looks up a localized string similar to Offset cannot be negative.

OffsetMustBeValid
Looks up a localized string similar to Offset must be a valid position in buffer.

ParameterAlreadyDefined
Looks up a localized string similar to Parameter '{0}' has already been defined..

ParameterCannotBeNegative
Looks up a localized string similar to Parameter cannot have a negative value.

ParameterCannotBeNull
Looks up a localized string similar to Parameter cannot be null.

ParameterIsInvalid
Looks up a localized string similar to Parameter is invalid..

ParameterMustBeDefined
Looks up a localized string similar to Parameter '{0}' must be defined..

ParameterNotFoundDuringPrepare
Looks up a localized string similar to Parameter '{0}' was not found during prepare..

PasswordMustHaveLegalChars
Looks up a localized string similar to Password must be valid and contain length characters.

PerfMonCategoryHelp
Looks up a localized string similar to This category includes a series of counters for MySQL..

PerfMonCategoryName
Looks up a localized string similar to .NET Data
PerfMonHardProcHelp

Looks up a localized string similar to The number of times a procedures metadata had to be queried from the server..

PerfMonHardProcName

Looks up a localized string similar to Hard Procedure Queries.

PerfMonSoftProcHelp

Looks up a localized string similar to The number of times a procedures metadata was retrieved from the client-side cache..

PerfMonSoftProcName

Looks up a localized string similar to Soft Procedure Queries.

ProcAndFuncSameName

Looks up a localized string similar to same name are not supported..

QueryTooLarge

Looks up a localized string similar to Packets larger than max_allowed_packet are not allowed..

ReadFromStreamFailed

Looks up a localized string similar to Reading from the stream has failed..

ReadingPriorColumnUsingSeqAccess

Returns the cached ResourceManager instance used by this class.

ResourceManager

Looks up a localized string similar to Routine '{0}' cannot be found. Either

Provider for MySQL.
RoutineNotFound

check the spelling or make sure you have sufficient rights to execute the routine..

ServerTooOld

Looks up a localized string similar to Connector/Net no longer supports server versions prior to 4.1.

SocketNoSeek

Looks up a localized string similar to Socket streams do not support seeking.

SoftProcQuery

Looks up a localized string similar to Retrieving procedure metadata for {0} from procedure cache..

SPNotSupported

Looks up a localized string similar to Stored procedures are not supported on this version of MySQL.

StreamAlreadyClosed

Looks up a localized string similar to The stream has already been closed.

StreamNoRead

Looks up a localized string similar to The stream does not support reading.

StreamNoWrite

Looks up a localized string similar to The stream does not support writing.

Timeout

Looks up a localized string similar to Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding..
Looks up a localized string similar to error connecting: Timeout expired. The timeout period elapsed prior to obtaining a connection from the pool. This may have occurred because all pooled connections were in use and max pool size was reached.

Looks up a localized string similar to {0}: Connection Closed.

Looks up a localized string similar to Unable to trace. There are more than Int32.MaxValue connections in use.

Looks up a localized string similar to {0}: Error encountered during row fetch. Number = {1}, Message={2}.

Looks up a localized string similar to {0}: Connection Opened: connection string = '{1}.'

Looks up a localized string similar to {0}: Error encountered attempting to open result: Number={1}, Message={2}.

Looks up a localized string similar to {0}: Query Closed.
TraceQueryNormalized

similar to \{0\}: Query Normalized: \{2\}.

Looks up a localized string similar to \{0\}: Query Opened: \{2\}.

TraceQueryOpened

Looks up a localized string similar to \{0\}: ResultSet Opened: field(s) = \{1\},
affected rows = \{2\}, inserted id = \{3\}.

TraceResult

Looks up a localized string similar to \{0\}: ResultSet Closed. Total rows=\{1\}, skipped rows=\{2\}, size (bytes)=\{3\}.

TraceResultClosed

Looks up a localized string similar to \{0\}: Set Database: \{1\}.

TraceSetDatabase

Looks up a localized string similar to \{0\}: Statement closed: statement id = \{1\}.

TraceStatementClosed

Looks up a localized string similar to \{0\}: Statement executed: statement id = \{1\}.

TraceStatementExecuted

Looks up a localized string similar to \{0\}: Statement prepared: sql='\{1\}',
statement id=\{2\}.

TraceStatementPrepared

Looks up a localized string similar to \{0\}: Usage Advisor Warning: Query is using a bad index.

TraceUAWarningBadIndex

Looks up a localized string similar to \{0\}: Usage Advisor Warning: The field '{2}' was converted
TraceUAWarningNoIndex

Looks up a localized string similar to {0}: Usage Advisor Warning: Query does not use an index.

TraceUAWarningSkippedColumns

Looks up a localized string similar to {0}: Usage Advisor Warning: The following columns were not accessed: {2}.

TraceUAWarningSkippedRows

Looks up a localized string similar to {0}: Usage Advisor Warning: Skipped {2} rows. Consider a more focused query..

TraceWarning

Looks up a localized string similar to {0}: MySql Warning: Level={1}, Code={2}, Message={3}.

UnableToConnectToHost

Looks up a localized string similar to Unable to connect to any of the specified MySQL hosts..

UnableToDeriveParameters

Looks up a localized string similar to Unable to derive stored routine parameters. The 'Parameters' information schema table is not available and access to the stored procedure body has been disabled..

UnableToEnumerateUDF

Looks up a localized string similar to An error occurred attempting to enumerate the user-defined functions.

Do you have SELECT
privileges on the mysql.func table.

Looks up a localized string similar to Unable to execute stored procedure '{0}'..

Looks up a localized string similar to There was an error parsing the foreign key definition..

Looks up a localized string similar to Unable to retrieve stored routine parameters. Either grant access to the routine or add the 'Use Procedure Bodies=false' option to your connection string..

Looks up a localized string similar to Unable to retrieve stored procedure metadata for routine '{0}'. Either grant SELECT privilege to mysql.proc for this user or use "use procedure bodies=false" with your connection string..

Looks up a localized string similar to Unable to start a second async operation while one is running..

Looks up a localized string similar to Unix sockets are not supported on Windows.
similar to The requested column value could not be treated as or conveted to a Guid..

Looks up a localized string similar to Writing to the stream failed..

Looks up a localized string similar to Parameter '{0}' is not found but a parameter with the name '{1}' is found. Parameter names must include the leading parameter marker..
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Improper MySqlCommandBuilder state: adapter is null.

**Namespace:** [MySQL.Data.MySqlClient.Properties](#)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string AdapterIsNull { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property AdapterIsNull As String

Visual C++

public:
static property String^ AdapterIsNull {
    String^ get ();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Improper MySqlCommandBuilder state: adapter's SelectCommand is null.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string AdapterSelectIsNull { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property AdapterSelectIsNull As String

Visual C++

public:
static property String^ AdapterSelectIsNull { 
    String^ get () ;
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Attempts to access a field before calling Read().

**Namespace:** [MySql.Data.MySqlClient.Properties](#)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string AttemptToAccessBeforeRead { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property AttemptToAccessBeforeRead As String

Visual C++

public:
static property String^ AttemptToAccessBeforeRead {
  String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Version string not in acceptable format.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

public static string BadVersionFormat { get; }

**Visual Basic (Declaration)**

Public Shared ReadOnly Property BadVersionFormat As String

**Visual C++**

public:
static property String^ BadVersionFormat {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
BufferCannotBeNull Property

Syntax

C#

public static string BufferCannotBeNull { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property BufferCannotBeNull As String

Visual C++

public:
static property String^ BufferCannotBeNull {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Buffer is not large enough.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string BufferNotLargeEnough { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property BufferNotLargeEnough As String

Visual C++

public:
static property String^ BufferNotLargeEnough {
    String^ get ();
}
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Canceling an executing query requires MySQL 5.0 or higher..


**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string CancelNeeds50 { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property CancelNeeds50 As String

Visual C++

public:
static property String^ CancelNeeds50 {
    String^ get ();
}
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Canceling an active query is only supported on MySQL 5.0.0 and above.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public static string CancelNotSupported { get; }
```

### Visual Basic (Declaration)

```vbnet
Public Shared Readonly Property CancelNotSupported As String
```

### Visual C++

```cpp
public:
static property String^ CancelNotSupported { 
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to MySqlCommandBuilder does not support multi-table statements.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string CBMultiTableNotSupported { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property CBMultiTableNotSupported As String

Visual C++

public:
static property String^ CBMultiTableNotSupported {
        String^ get ();
    }
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to MySqlCommandBuilder cannot operate on tables with no unique or key columns.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string CBNoKeyColumn { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property CBNoKeyColumn As String

Visual C++

public:
static property String^ CBNoKeyColumn {
    String^ get ();
}

}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Chaos isolation level is not supported.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#
public static string ChaosNotSupported { get; }

Visual Basic (Declaration)
Public Shared ReadOnly Property ChaosNotSupported As String

Visual C++
public:
static property String^ ChaosNotSupported { 
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The CommandText property has not been properly initialized.


**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string CommandTextNotInitialized { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property CommandTextNotInitialized As String

Visual C++

public:
static property String^ CommandTextNotInitialized {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The connection is already open..

**Namespace:** [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ConnectionAlreadyOpen { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ConnectionAlreadyOpen As String

Visual C++

public:
static property String^ ConnectionAlreadyOpen { 
    String^ get (); 
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Connection unexpectedly terminated..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ConnectionBroken { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ConnectionBroken As String

Visual C++

public:
static property String^ ConnectionBroken {
    String^ get ();
}
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Connection must be valid and open.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public static string ConnectionMustBeOpen { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property ConnectionMustBeOpen As String
```

**Visual C++**

```cpp
public:
static property String^ ConnectionMustBeOpen { 
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The connection is not open.

**Namespace:**  MySql.Data.MySqlClient.Properties  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ConnectionNotOpen { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ConnectionNotOpen As String

Visual C++

public:
static property String^ ConnectionNotOpen { String^ get (); }
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The connection property has not been set or is null..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](https://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ConnectionNotSet { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ConnectionNotSet As String

Visual C++

public:
static property String^ ConnectionNotSet {
    String^ get();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Could not find specified column in results.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string CouldNotFindColumnName { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property CouldNotFindColumnName As String

Visual C++

public:
static property String^ CouldNotFindColumnName {
    String^ get ();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Count cannot be negative.

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string CountCannotBeNegative { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property CountCannotBeNegative As String

Visual C++

public:
static property String^ CountCannotBeNegative {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to SetLength is not a valid operation on CompressedStream.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  [MySQL.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

```csharp
public static string CSNoSetLength { get; }
```

Visual Basic (Declaration)

Public Shared ReadOnly Property CSNoSetLength As String

Visual C++

```c++
public:
static property String^ CSNoSetLength {
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static CultureInfo Culture { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Shared Property Culture As CultureInfo
```

Visual C++

```cpp
public:
static property CultureInfo^ Culture { 
    CultureInfo^ get ();
    void set (CultureInfo^ value);
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The given value was not in a supported format.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll)  
Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static string DataNotInSupportedFormat { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Readonly Property DataNotInSupportedFormat As String
```

**Visual C++**

```cpp
public:
static property String^ DataNotInSupportedFormat {
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to There is already an open DataReader associated with this Connection which must be closed first..

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public static string DataReaderOpen { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property DataReaderOpen As String
```

**Visual C++**

```cpp
public:
static property String^ DataReaderOpen {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
DefaultEncodingNotFound Property

Looks up a localized string similar to The default connection encoding was not found. Please report this as a bug along with your connection string and system details..

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string DefaultEncodingNotFound { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property DefaultEncodingNotFound As String

Visual C++

public:
static property String^ DefaultEncodingNotFound {
    String^ get ();
}

}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to MySQL Connector/Net does not currently support distributed transactions..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)
**Assembly:**  [MySql.Data](#) (in [MySql.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

public static string DistributedTxnNotSupported { get; }

Visual Basic (Declaration)

Public Shared Readonly Property DistributedTxnNotSupported As String

Visual C++

public:
static property String^ DistributedTxnNotSupported {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Error creating socket connection.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ErrorCreatingSocket { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ErrorCreatingSocket As String

Visual C++

public:
static property String^ ErrorCreatingSocket {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Fatal error encountered during command execution..

**Namespace:** [MySQL.Data.MySqlClient.Properties](#)

**Assembly:** [MySQL.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
Syntax

**C#**

```csharp
public static string FatalErrorDuringExecute { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property FatalErrorDuringExecute As String
```

**Visual C++**

```cpp
public:
static property String^ FatalErrorDuringExecute { 
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Fatal error encountered during data read..

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  [MySql.Data](#) (in [MySql.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

public static string FatalErrorDuringRead { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property FatalErrorDuringRead As String

Visual C++

public:
static property String^ FatalErrorDuringRead { 
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Fatal error encountered attempting to read the resultset.

**Namespace:** MySql.Data.MySqlClient.Properties  
**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string FatalErrorReadingResult { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property FatalErrorReadingResult As String

Visual C++

public:
static property String^ FatalErrorReadingResult { 
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to `From` index and length use more bytes than from contains.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string FromAndLengthTooBig { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property FromAndLengthTooBig As String

Visual C++

public:
static property String^ FromAndLengthTooBig {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to From index must be a valid index inside the from buffer.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string FromIndexMustBeValid { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property FromIndexMustBeValid As String

Visual C++

public:
static property String^ FromIndexMustBeValid {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Call to GetHostEntry failed after {0} while querying for hostname '{1}': SocketErrorCode={2}, ErrorCode={3}, NativeErrorCode={4}..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string GetHostEntryFailed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property GetHostEntryFailed As String

Visual C++

public:
static property String^ GetHostEntryFailed {
    String^ get ();
}

}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Retrieving procedure metadata for `{0}` from server.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static string HardProcQuery { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Readonly Property HardProcQuery As String
```

**Visual C++**

```cpp
public:
static property String^ HardProcQuery {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Value has an unsupported format..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ImproperValueFormat { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ImproperValueFormat As String

Visual C++

public:
static property String^ ImproperValueFormat {
    String^ get ();
}

See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to An incorrect response was received from the server.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](https://www.mysql.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string IncorrectTransmission { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property IncorrectTransmission As String

Visual C++

public:
static property String^ IncorrectTransmission {
    String^ get ();
}

}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Index and length use more bytes than to has room for.

Namespace:  MySQL.Data.MySqlClient.Properties
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string IndexAndLengthTooBig { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property IndexAndLengthTooBig As String

Visual C++

public:
static property String^ IndexAndLengthTooBig {
    String^ get ()
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Index must be a valid position in the buffer.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string IndexMustBeValid { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property IndexMustBeValid As String

Visual C++

public:
static property String^ IndexMustBeValid {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
InvalidColumnOrdinal Property

Looks up a localized string similar to You have specified an invalid column ordinal..

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string InvalidColumnOrdinal { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property InvalidColumnOrdinal As String

Visual C++

public:
static property String^ InvalidColumnOrdinal {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The requested value '{0}' is invalid for the given keyword '{1}'.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string InvalidConnectionStringValue { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property InvalidConnectionStringValue As String

Visual C++

public:
static property String^ InvalidConnectionStringValue {
    String^ get ();
}

See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Procedure or function '{0}' cannot be found in database '{1}'.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](https://mylbrace.net)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string InvalidProcName { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property InvalidProcName As String

Visual C++

public:
static property String^ InvalidProcName {
    String^ get ()
}
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
InvalidValueForBoolean Property

Resources..::.InvalidValueForBoolean Property

Looks up a localized string similar to '{0}' is an illegal value for a boolean option..

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string InvalidValueForBoolean { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property InvalidValueForBoolean As String

Visual C++

public:
static property String^ InvalidValueForBoolean { 
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Keyword does not allow null values.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string KeywordNoNull { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property KeywordNoNull As String

Visual C++

public:
static property String^ KeywordNoNull {
    String^ get ();
}

}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
KeywordNotSupported Property

Looks up a localized string similar to Keyword not supported..

**Namespace:** [MySql.Data.MySqlClient.Properties](#)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string KeywordNotSupported { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared Readonly Property KeywordNotSupported As String
```

Visual C++

```cpp
public:
static property String^ KeywordNotSupported { 
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to ACCESSIBLE ADD ALL ALTER ANALYZE AND AS ASC ASENSITIVE BEFORE BETWEEN BIGINT BINARY BLOB BOTH BY CALL CASCADE CASE CHANGE CHAR CHARACTER CHECK COLLATE COLUMN CONDITION CONNECTION CONSTRAINT CONTINUE CONVERT CREATE CROSS CURRENT_DATE CURRENT_TIME CURRENT_TIMESTAMP CURRENT_USER CURSOR DATABASE DATABASES DAY_HOUR DAY_MICROSECOND DAY_MINUTE DAY_SECOND DEC DECIMAL DECLARE DEFAULT DELAYED DELETE DESC DESCRIBE DETERMINISTIC DISTINCT DISTINCTROW DIV DOUBLE DROP DUAL EACH ELSE ELSEIF ENCLOSED ESCAPED EXISTS EXIT EXP [rest of string was truncated]";.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string keywords { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property keywords As String

Visual C++

public:
static property String^ keywords {
    String^ get ();
}
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to INTERNAL ERROR: More than one output parameter row detected.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string MoreThanOneOPRow { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property MoreThanOneOPRow As String

Visual C++

public:
static property String^ MoreThanOneOPRow {
    String^ get ();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Multiple simultaneous connections or connections with different connection strings inside the same transaction are not currently supported.

**Namespace:**  MySql.Data.MySqlClient.Properties
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string MultipleConnectionsInTransactionNotSupported {

Visual Basic (Declaration)

Public Shared ReadOnly Property MultipleConnectionsInTransactionNotSupported

Visual C++

public:
static property String^ MultipleConnectionsInTransactionNotSupported

String^ get ()
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to NamedPipeStream does not support seeking.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll)  Version: 6.2.2.0
Syntax

C#

public static string NamedPipeNoSeek { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property NamedPipeNoSeek As String

Visual C++

public:
static property String^ NamedPipeNoSeek {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to NamedPipeStream doesn't support SetLength.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string NamedPipeNoSetLength { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property NamedPipeNoSetLength As String

Visual C++

public:
static property String^ NamedPipeNoSetLength {{
    String^ get ();
}}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Invalid attempt to call NextResult when the reader is closed.

**Namespace:**  MySQL.Data.MySqlClient.Properties  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string NextResultIsClosed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property NextResultIsClosed As String

Visual C++

public:
static property String^ NextResultIsClosed {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to When calling stored procedures and 'Use Procedure Bodies' is false, all parameters must have their type explicitly set.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string NoBodiesAndTypeNotSet { get; } 
```

Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property NoBodiesAndTypeNotSet As String 
```

Visual C++

```cpp
public:
static property String^ NoBodiesAndTypeNotSet {
    String^ get ();
} 
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Nested transactions are not supported..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string NoNestedTransactions { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property NoNestedTransactions As String

Visual C++

public:
static property String^ NoNestedTransactions {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The host {0} does not support SSL connections..

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string NoServerSSLSupport { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property NoServerSSLSupport As String

Visual C++

public:
static property String^ NoServerSSLSupport { String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The object is not open or has been disposed..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ObjectDisposed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ObjectDisposed As String

Visual C++

public:
static property String^ ObjectDisposed {
    String^ get ();
}

See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Offset cannot be negative.

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string OffsetCannotBeNegative { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property OffsetCannotBeNegative As String

Visual C++

public:
static property String^ OffsetCannotBeNegative {
    String^ get();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Offset must be a valid position in buffer.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static string OffsetMustBeValid { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property OffsetMustBeValid As String
```

**Visual C++**

```cpp
public:
static property String^ OffsetMustBeValid { 
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Parameter '{0}' has already been defined..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string ParameterAlreadyDefined { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property ParameterAlreadyDefined As String
```

Visual C++

```cpp
public:
static property String^ ParameterAlreadyDefined { 
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Parameter cannot have a negative value.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string ParameterCannotBeNegative { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property ParameterCannotBeNegative As String
```

Visual C++

```csharp
public:
static property String^ ParameterCannotBeNegative {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Parameter cannot be null.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

**C#**

```csharp
public static string ParameterCannotBeNull { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property ParameterCannotBeNull As String
```

**Visual C++**

```cpp
public:
static property String^ ParameterCannotBeNull {
    String^ get ();
}
```
See Also

- Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Parameter is invalid..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public static string ParameterIsInvalid { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property ParameterIsInvalid As String
```

**Visual C++**

```cpp
public:
static property String^ ParameterIsInvalid {
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
ParameterMustBeDefined Property

assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0

Looks up a localized string similar to Parameter '{0}' must be defined..
**Syntax**

C#

```csharp
public static string ParameterMustBeDefined { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property ParameterMustBeDefined As String
```

Visual C++

```cpp
public:
static property String^ ParameterMustBeDefined {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
ParameterNotFoundDuringPrepare Property

Looks up a localized string similar to Parameter '{0}' was not found during prepare..

**Namespace:**  MySql.Data.MySqlClient.Properties

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ParameterNotFoundDuringPrepare { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ParameterNotFoundDuringPrepare As String

Visual C++

public:
static property String^ ParameterNotFoundDuringPrepare {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Password must be valid and contain length characters.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#
public static string PasswordMustHaveLegalChars { get; }

Visual Basic (Declaration)
Public Shared ReadOnly Property PasswordMustHaveLegalChars As String

Visual C++
public:
static property String^ PasswordMustHaveLegalChars {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to This category includes a series of counters for MySQL..

**Namespace**: [MySQL.Data.MySqlClient.Properties](#)

**Assembly**: MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
### Syntax

#### C#

```csharp
public static string PerfMonCategoryHelp { get; }
```

#### Visual Basic (Declaration)

```vbnet
Public Shared Readonly Property PerfMonCategoryHelp As String
```

#### Visual C++

```cpp
public:
static property String^ PerfMonCategoryHelp {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to .NET Data Provider for MySQL.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string PerfMonCategoryName { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PerfMonCategoryName As String

Visual C++

public:
static property String^ PerfMonCategoryName { 
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The number of times a procedures metadata had to be queried from the server..

**Namespace:** [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:** [MySQL.Data (in MySQL.Data.dll)](#)  
**Version:** 6.2.2.0
Syntax

C#

public static string PerfMonHardProcHelp { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PerfMonHardProcHelp As String

Visual C++

public:
static property String^ PerfMonHardProcHelp { 
    String^ get (); 
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Hard Procedure Queries.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string PerfMonHardProcName { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PerfMonHardProcName As String

Visual C++

public:
static property String^ PerfMonHardProcName {
    String^ get ();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The number of times a procedures metadata was retrieved from the client-side cache..

**Namespace:**  [MySql.Data.MySqlClient.Properties](https://www.mysql.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string PerfMonSoftProcHelp { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PerfMonSoftProcHelp As String

Visual C++

public:
static property String^ PerfMonSoftProcHelp {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Soft Procedure Queries.

**Namespace:** [MySql.Data.MySqlClient.Properties](https://www.mysql.com)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string PerfMonSoftProcName { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PerfMonSoftProcName As String

Visual C++

public:
static property String^ PerfMonSoftProcName { 
    String^ get();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to same name are not supported..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ProcAndFuncSameName { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ProcAndFuncSameName As String

Visual C++

public:
static property String^ ProcAndFuncSameName {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Packets larger than max_allowed_packet are not allowed.

**Namespace:** [MySQL.Data.MySqlClient.Properties](https://example.com/namespace)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string QueryTooLarge { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property QueryTooLarge As String

Visual C++

public:
static property String^ QueryTooLarge {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Reading from the stream has failed..

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ReadFromStreamFailed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ReadFromStreamFailed As String

Visual C++

public:
static property String^ ReadFromStreamFailed { 
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

Resources.

ReadingPriorColumnUsingSeqAccess Property


Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0

Looks up a localized string similar to Invalid attempt to read a prior column using SequentialAccess.
Syntax

C#

public static string ReadingPriorColumnUsingSeqAccess { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ReadingPriorColumnUsingSeqAccess As

Visual C++

public:
static property String^ ReadingPriorColumnUsingSeqAccess {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns the cached ResourceManager instance used by this class.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static ResourceManager ResourceManager { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Readonly Property ResourceManager As ResourceManager
```

**Visual C++**

```cpp
public:
static property ResourceManager^ ResourceManager { 
ResourceManager^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Routine '{0}' cannot be found. Either check the spelling or make sure you have sufficient rights to execute the routine.

Syntax

C#

public static string RoutineNotFound { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property RoutineNotFound As String

Visual C++

public:
static property String^ RoutineNotFound {
    String^ get ();
}

See Also

Resources Class
StyleSheet

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Connector/Net no longer supports server versions prior to 4.1.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ServerTooOld { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ServerTooOld As String

Visual C++

public:
static property String^ ServerTooOld { 
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Socket streams do not support seeking.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string SocketNoSeek { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property SocketNoSeek As String

Visual C++

public:
static property String^ SocketNoSeek {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Retrieving procedure metadata for \{0\} from procedure cache..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](https://www.mysql.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string SoftProcQuery { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property SoftProcQuery As String

Visual C++

public:
static property String^ SoftProcQuery {
    String^ get ();
}


See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Stored procedures are not supported on this version of MySQL.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string SPNotSupported { get; }

Visual Basic (Declaration)

Public Shared Readonly Property SPNotSupported As String

Visual C++

public:
static property String^ SPNotSupported { 
    String^ get ();
}


See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
 Looks up a localized string similar to The stream has already been closed.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string StreamAlreadyClosed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property StreamAlreadyClosed As String

Visual C++

public:
static property String^ StreamAlreadyClosed {
    String^ get ();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The stream does not support reading.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string StreamNoRead { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property StreamNoRead As String

Visual C++

public:
static property String^ StreamNoRead {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The stream does not support writing.

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string StreamNoWrite { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property StreamNoWrite As String

Visual C++

public:
static property String^ StreamNoWrite {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Timeout Property

Looks up a localized string similar to Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding.

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public static string Timeout { get; }
```

### Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property Timeout As String
```

### Visual C++

```cpp
public:
    static property String^ Timeout { 
        String^ get ();
    }
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to error connecting: Timeout expired. The timeout period elapsed prior to obtaining a connection from the pool. This may have occurred because all pooled connections were in use and max pool size was reached.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  [MySql.Data](#) (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TimeoutGettingConnection { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TimeoutGettingConnection As String

Visual C++

public:
static property String^ TimeoutGettingConnection {
    String^ get ();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Connection Closed.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string TraceCloseConnection { get; }
```

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceCloseConnection As String

Visual C++

```c++
public:
static property String^ TraceCloseConnection {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unable to trace. There are more than Int32.MaxValue connections in use..

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceErrorMoreThanMaxValueConnections { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceErrorMoreThanMaxValueConnections

Visual C++

public:
static property String^ TraceErrorMoreThanMaxValueConnections {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Error encountered during row fetch. Number = {1}, Message={2}.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceFetchError { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceFetchError As String

Visual C++

public:
static property String^ TraceFetchError {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to \{0\}: Connection Opened: connection string = \'{1}\'.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](https://www.mysql.com)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

class CSharpExample
{
    public static string TraceOpenConnection { get; }
}

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceOpenConnection As String

Visual C++

public:
static property String^ TraceOpenConnection {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Error encountered attempting to open result: Number={1}, Message={2}.

**Namespace:** [MySql.Data.MySqlClient.Properties](#)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string TraceOpenResultError { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property TraceOpenResultError As String
```

Visual C++

```cpp
public:
static property String^ TraceOpenResultError {  
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to `{0}`: Query Closed.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceQueryDone { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceQueryDone As String

Visual C++

public:
static property String^ TraceQueryDone {
    String^ get ();
}

}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Query Normalized: {2}.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceQueryNormalized { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceQueryNormalized As String

Visual C++

public:
static property String^ TraceQueryNormalized { 
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to \{0\}: Query Opened: \{2\}.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  [MySQL.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

public static string TraceQueryOpened { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceQueryOpened As String

Visual C++

public:
static property String^ TraceQueryOpened {
    String^ get();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Resultset Opened: field(s) = {1}, affected rows = {2}, inserted id = {3}.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceResult { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceResult As String

Visual C++

public:
static property String^ TraceResult {
    String^ get ();
}


See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to \{0\}: Resultset Closed. Total rows=\{1\}, skipped rows=\{2\}, size (bytes)=\{3\}.

**Namespace:** [MySQL.Data.MySqlClient.Properties](https://example.com)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceResultClosed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceResultClosed As String

Visual C++

public:
static property String^ TraceResultClosed {
    String^ get ();
}

See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to \{0\}: Set Database: \{1\}.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceSetDatabase { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceSetDatabase As String

Visual C++

public:
static property String^ TraceSetDatabase { 
    String^ get ();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Statement closed: statement id = {1}.

**Namespace:**  MySQL.Data.MySqlClient.Properties
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceStatementClosed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceStatementClosed As String

Visual C++

public:
static property String^ TraceStatementClosed { 
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Statement executed: statement id = {1}.


**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceStatementExecuted { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceStatementExecuted As String

Visual C++

public:
static property String^ TraceStatementExecuted {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Statement prepared: sql='{1}', statement id={2}.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:**  MySQL.Data (in MySQL.Data.dll)  
Version: 6.2.2.0
Syntax

C#

```csharp
public static string TraceStatementPrepared { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property TraceStatementPrepared As String
```

Visual C++

```cpp
public:
static property String^ TraceStatementPrepared {
    String^ get ();
}
```
See Also

[Resources Class]

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Usage Advisor Warning: Query is using a bad index.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public static string TraceUAWarningBadIndex { get; }
```

**Visual Basic (Declaration)**

```
Public Shared ReadOnly Property TraceUAWarningBadIndex As String
```

**Visual C++**

```cpp
public:
static property String^ TraceUAWarningBadIndex {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Usage Advisor Warning: The field '{2}' was converted to the following types: {3}.

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceUAWarningFieldConversion { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceUAWarningFieldConversion As String

Visual C++

public:
static property String^ TraceUAWarningFieldConversion {
    String^ get ();
}

See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Usage Advisor Warning: Query does not use an index.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string TraceUAWarningNoIndex { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared Readonly Property TraceUAWarningNoIndex As String
```

Visual C++

```cpp
public:
static property String^ TraceUAWarningNoIndex {
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Usage Advisor Warning: The following columns were not accessed: {2}.

**Namespace:** [MySQL.Data.MySqlClient.Properties](#)

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceUAWarningSkippedColumns { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceUAWarningSkippedColumns As Stri

Visual C++

public:
static property String^ TraceUAWarningSkippedColumns {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to {0}: Usage Advisor Warning: Skipped {2} rows. Consider a more focused query..

**Namespace:** [MySql.Data.MySqlClient.Properties](#)

**Assembly:** MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static string TraceUAWarningSkippedRows { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property TraceUAWarningSkippedRows As String
```

**Visual C++**

```cpp
public:
static property String^ TraceUAWarningSkippedRows {
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to \{0\}: MySql Warning: Level=\{1\}, Code=\{2\}, Message=\{3\}.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string TraceWarning { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property TraceWarning As String

Visual C++

public:
static property String^ TraceWarning {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
UnableToConnectToHost Property

Resources..<...Resources Class  See Also  Send Feedback

Looks up a localized string similar to Unable to connect to any of the specified MySQL hosts..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public static string UnableToConnectToHost { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property UnableToConnectToHost As String
```

**Visual C++**

```cpp
public:
static property String^ UnableToConnectToHost {
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
UnableToDeriveParameters Property

Resources...:::UnableToDeriveParameters Property

Resources Class  See Also  Send Feedback

Looks up a localized string similar to Unable to derive stored routine parameters. The 'Parameters' information schema table is not available and access to the stored procedure body has been disabled..

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string UnableToDeriveParameters { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UnableToDeriveParameters As String

Visual C++

public:
static property String^ UnableToDeriveParameters {
    String^ get ();
}
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
UnableToEnumerateUDF Property

Looks up a localized string similar to An error occurred attempting to enumerate the user-defined functions. Do you have SELECT privileges on the mysql.func table?

Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string UnableToEnumerateUDF { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UnableToEnumerateUDF As String

Visual C++

public:
static property String^ UnableToEnumerateUDF {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unable to execute stored procedure '{0}'..

**Namespace:**  [MySql.Data.MySqlClient.Properties]

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string UnableToExecuteSP { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UnableToExecuteSP As String

Visual C++

public:
static property String^ UnableToExecuteSP {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
UnableToParseFK Property

Looks up a localized string similar to There was an error parsing the foreign key definition..

**Namespace:** MySQL.Data.MySqlClient.Properties

**Assembly:** MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string UnableToParseFK { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property UnableToParseFK As String
```

Visual C++

```cpp
public:
static property String^ UnableToParseFK {
    String^ get();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Unable to retrieve parameters. Either grant access to the routine or add the 'Use Procedure Bodies=false' option to your connection string.

Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public static string UnableToRetrieveParameters { get; }
```

### Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property UnableToRetrieveParameters As String
```

### Visual C++

```csharp
public:
static property String^ UnableToRetrieveParameters {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unable to retrieve stored procedure metadata for routine '{0}'. Either grant SELECT privilege to mysql.proc for this user or use "use procedure bodies=false" with your connection string.

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  [MySQL.Data](#) (in [MySQL.Data.dll](#)) Version: 6.2.2.0
## Syntax

### C#

```csharp
public static string UnableToRetrieveSProcData { get; }
```

### Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property UnableToRetrieveSProcData As String
```

### Visual C++

```cpp
public:
static property String^ UnableToRetrieveSProcData {
    String^ get ();
}
```
See Also

Resources Class
MySQL.Data.MySqlClient.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
UnableToStartSecondAsyncOp Property

Looks up a localized string similar to Unable to start a second async operation while one is running..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static string UnableToStartSecondAsyncOp { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Readonly Property UnableToStartSecondAsyncOp As String
```

**Visual C++**

```csharp
public:
static property String^ UnableToStartSecondAsyncOp {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unix sockets are not supported on Windows.

**Namespace:** [MySQL.Data.MySqlClient.Properties](#)  
**Assembly:** [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public static string UnixSocketsNotSupported { get; }
```

**Visual Basic (Declaration)**

```
Public Shared ReadOnly Property UnixSocketsNotSupported As String
```

**Visual C++**

```c++
public:
static property String^ UnixSocketsNotSupported {
    String^ get ();
}
```
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The requested column value could not be treated as or converted to a Guid..

**Namespace:**  [MySQL.Data.MySqlClient.Properties](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string ValueNotSupportedForGuid { get; }

Visual Basic (Declaration)

Public Shared Readonly Property ValueNotSupportedForGuid As String

Visual C++

public:
static property String^ ValueNotSupportedForGuid {
    String^ get ();
}

}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Writing to the stream failed..

**Namespace:**  [MySql.Data.MySqlClient.Properties](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string WriteToStreamFailed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property WriteToStreamFailed As String

Visual C++

public:
static property String^ WriteToStreamFailed {
    String^ get ();
}
See Also

Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Parameter '{0}' is not found but a parameter with the name '{1}' is found. Parameter names must include the leading parameter marker..

**Namespace:**  [MySql.Data.MySqlClient.Properties]

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public static string WrongParameterName { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property WrongParameterName As String

Visual C++

public:
static property String^ WrongParameterName {
    String^ get ();
}
C#
Visual Basic
Visual C++
MySQL Connector/Net
MySql.Data.Types Namespace
Send Feedback
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlConversionException</td>
<td>Summary description for MySqlConversionException.</td>
</tr>
</tbody>
</table>
### Structures

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlDateTime</td>
<td></td>
</tr>
<tr>
<td>MySqlDecimal</td>
<td></td>
</tr>
</tbody>
</table>

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Summary description for MySqlConversionException.

**Namespace:**  [MySql.Data.Types](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

[SerializableAttribute]
public class MySqlConversionException : Exception

Visual Basic (Declaration)

<SerializableAttribute> 
Public Class MySqlConversionException 
Inherits Exception

Visual C++

[SerializableAttribute]
public ref class MySqlConversionException : public Exception
Inheritance Hierarchy

- **System..:::Object**
- **System..:::Exception**
- **MySql.Data.Types..:::MySqlConversionException**
See Also

MySQLConversionException Members
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlConversionException` type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlConversionException</td>
<td>Ctor</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception.  (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception.  (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HResult</strong></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception.  (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception.  (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception.  (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error.  (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown.  (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception.  (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

MySqlConversionException Class
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLConversionException Constructor

**Namespace**: [MySQL.Data.Types](#)
**Assembly**: MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public MySqlConversionException(
    string msg
)
```

Visual Basic (Declaration)

```vbnet
Public Sub New ( _
    msg As String _
)
```

Visual C++

```cpp
public:
MySqlConversionException(
    String^ msg
)
```

Parameters

msg

Type: System::String
See Also

MySqlConversionException Class
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlConversionException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object.equals">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is reclaimed by garbage collection. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.) Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySQLConversionException Class
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlConversionException` type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>HResult</strong></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <a href="#">Exception</a> instance that caused the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
</tbody>
</table>
See Also

MySqlConversionException Class
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL DateTime Structure

Namespace: MySql.Data.Types
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public struct MySqlDateTime : IConvertible,
    IComparable

Visual Basic (Declaration)

Public Structure MySqlDateTime _
    Implements IConvertible, IComparable

Visual C++

public value class MySqlDateTime : IConvertible,
    IComparable
See Also

MySqlDateTime Members
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlDateTime` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MySqlDateTime</strong></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <strong>Object</strong> to attempt to free resources and perform other cleanup operations before the <strong>Object</strong> is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetDateTime</strong></td>
<td>Returns this value as a DateTime (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a MySQL specific string representation of this value. (Overrides <strong>ValueType</strong>.::<strong>ToString</strong>().)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Explicit</td>
<td></td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day</td>
<td>Returns the day portion of this datetime</td>
</tr>
<tr>
<td>Hour</td>
<td>Returns the hour portion of this datetime</td>
</tr>
<tr>
<td>IsNull</td>
<td>Returns true if this datetime object has a null value</td>
</tr>
<tr>
<td>IsValidDateTime</td>
<td>Indicates if this object contains a value that can be represented as a DateTime</td>
</tr>
<tr>
<td>Millisecond</td>
<td>Retrieves the millisecond value of this object.</td>
</tr>
<tr>
<td>Minute</td>
<td>Returns the minute portion of this datetime</td>
</tr>
<tr>
<td>Month</td>
<td>Returns the month portion of this datetime</td>
</tr>
<tr>
<td>Second</td>
<td>Returns the second portion of this datetime</td>
</tr>
<tr>
<td>Value</td>
<td>Retrieves the value of this <code>MySqlDateTime</code> as a DateTime object.</td>
</tr>
<tr>
<td>Year</td>
<td>Returns the year portion of this datetime</td>
</tr>
</tbody>
</table>
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MySqlDateTime(MySqlDateTime)</code></td>
<td>Constructs a new <code>MySqlDateTime</code> object by copying the current value of the given object.</td>
</tr>
<tr>
<td><code>MySqlDateTime(DateTime)</code></td>
<td>Constructs a new <code>MySqlDateTime</code> object by using values from the given <code>DateTime</code> object.</td>
</tr>
<tr>
<td><code>MySqlDateTime(String)</code></td>
<td>Enables the construction of a <code>MySqlDateTime</code> object by parsing a string.</td>
</tr>
<tr>
<td><code>MySqlDateTime(Int32, Int32, Int32, Int32, Int32)</code></td>
<td>Constructs a new <code>MySqlDateTime</code> object by setting the individual time properties to the given values.</td>
</tr>
</tbody>
</table>
See Also

MySqlDateTime Structure
MySqlDateTime Members
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Constructs a new **MySQLDateTime** object by copying the current value of the given object.

**Namespace:**  [MySQL.Data.Types](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public MySqlDateTime(
    MySqlDateTime mdt
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New (_
    mdt As MySqlDateTime _
)
```

**Visual C++**

```cpp
public:
MySqlDateTime(
    MySqlDateTime mdt
)
```

### Parameters

`mdt`  
Type: `MySql.Data.Types::MySqlDateTime`  
The `MySqlDateTime` object to copy.
See Also

MySQLDateTime Structure
MySQLDateTime Overload
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Constructs a new `MySqlDateTime` object by using values from the given `DateTime` object.

**Namespace:**  [MySql.Data.Types](https://example.com/namespace)

**Assembly:**  [MySql.Data](https://example.com/assembly) (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlDateTime(
    DateTime dt
)

Visual Basic (Declaration)

Public Sub New ( _
    dt As DateTime _
)

Visual C++

public:
MySqlDateTime(
    DateTime dt
)

Parameters

dt

Type: System::DateTime
The DateTime object to copy.
See Also

MySqlDateTime Structure
MySqlDateTime Overload
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Enables the construction of a **MySQLDateTime** object by parsing a string.

**Namespace:**  [MySQL.Data.Types](#)  
**Assembly:**  MySql.Data (in MySql.Data.dll)  
**Version:**  6.2.2.0
Syntax

C#

public MySqlDateTime(
    string dateTime
)

Visual Basic (Declaration)

Public Sub New ( _
    dateTime As String _
)

Visual C++

public:
MySqlDateTime(
    String^ dateTime
)

Parameters

dateTime
    Type: System:::String
See Also

MySqlDateTime Structure
MySqlDateTime Overload
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Date/Time Constructor (Int32, Int32, Int32, Int32, Int32, Int32)

Constructs a new **MySQLDateTime** object by setting the individual time properties to the given values.

**Namespace:**  MySql.Data.Types

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public MySqlDateTime(
    int year,
    int month,
    int day,
    int hour,
    int minute,
    int second
)

Visual Basic (Declaration)

Public Sub New ( _
    year As Integer, _
    month As Integer, _
    day As Integer, _
    hour As Integer, _
    minute As Integer, _
    second As Integer _
)

Visual C++

public:
MySqlDateTime(
    int year,
    int month,
    int day,
    int hour,
    int minute,
    int second
)

Parameters

year
    Type: System::::Int32
    The year to use.
month
   Type: System..::: Int32
   The month to use.

day
   Type: System..::: Int32
   The day to use.

hour
   Type: System..::: Int32
   The hour to use.

minute
   Type: System..::: Int32
   The minute to use.

second
   Type: System..::: Int32
   The second to use.
See Also

MySQLDateTime Structure
MySQLDateTime Overload
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLDateTime` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <strong>Object</strong> to attempt to free resources and perform other cleanup operations before the <strong>Object</strong> is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetDateTime</strong></td>
<td>Returns this value as a DateTime</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a MySQL specific string representation of this value (Overrides <strong>ValueType,</strong>:::<strong>ToString()</strong>).</td>
</tr>
</tbody>
</table>
## Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlDateTime Structure
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace: MySql.Data.Types
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static explicit operator DateTime (MySqlDateTime val)
```

Visual Basic (Declaration)

```vbnet
Public Shared Narrowing Operator CType (_val As MySqlDateTime _) As DateTime
```

Visual C++

```cpp
static explicit operator DateTime (MySqlDateTime val)
```

Parameters

val

Type: `MySql.Data.Types::MySqlDateTime`

Return Value
See Also

MySqlDateTime Structure
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDateTime..::.GetDateTime Method

MySQLDateTime Structure  See Also  Send Feedback

Returns this value as a DateTime

Namespace:  MySql.Data.Types
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public DateTime GetDateTime()

Visual Basic (Declaration)

Public Function GetDateTime As DateTime

Visual C++

public:
    DateTime GetDateTime()
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLDateTime...::ToString Method

**MySQLDateTime Structure**  **See Also**  **Send Feedback**

Returns a MySQL specific string representation of this value

**Namespace:**  [MySQL.Data.Types](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override string ToString()

Visual Basic (Declaration)

Public Overrides Function ToString As String

Visual C++

public:
virtual String^ ToString() override
See Also

MySqlDateTime Structure
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlDateTime` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Day</strong></td>
<td>Returns the day portion of this datetime</td>
</tr>
<tr>
<td><strong>Hour</strong></td>
<td>Returns the hour portion of this datetime</td>
</tr>
<tr>
<td><strong>IsNull</strong></td>
<td>Returns true if this datetime object has a null value</td>
</tr>
<tr>
<td><strong>IsValidDateTime</strong></td>
<td>Indicates if this object contains a value that can be represented as a DateTime</td>
</tr>
<tr>
<td><strong>Millisecond</strong></td>
<td>Retrieves the millisecond value of this object.</td>
</tr>
<tr>
<td><strong>Minute</strong></td>
<td>Returns the minute portion of this datetime</td>
</tr>
<tr>
<td><strong>Month</strong></td>
<td>Returns the month portion of this datetime</td>
</tr>
<tr>
<td><strong>Second</strong></td>
<td>Returns the second portion of this datetime</td>
</tr>
<tr>
<td><strong>Value</strong></td>
<td>Retrieves the value of this <code>MySqlDateTime</code> as a DateTime object.</td>
</tr>
<tr>
<td><strong>Year</strong></td>
<td>Returns the year portion of this datetime</td>
</tr>
</tbody>
</table>
See Also

MySqlDateTime Structure
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns the day portion of this datetime

**Namespace:**  [MySQL.Data.Types](#)

**Assembly:**  [MySQL.Data](#) (in MySQL.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Day { get; set; }

Visual Basic (Declaration)

Public Property Day As Integer

Visual C++

public:
    property int Day {
        int get ();
        void set (int value);
    }
See Also

MySqlDateTime Structure
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL DateTime...:::Hour Property

MySQLDateTime Structure  See Also  Send Feedback

Returns the hour portion of this datetime

Namespace:  MySql.Data.Types
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Hour { get; set; }

Visual Basic (Declaration)

Public Property Hour As Integer

Visual C++

public:
property int Hour {
    int get ();
    void set (int value);
}


See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLDateTime...::isNull Property
MySQLDateTime Structure    See Also    Send Feedback

Returns true if this datetime object has a null value

Namespace:    MySql.Data.Types
Assembly:    MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public bool IsNull { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property IsNull As Boolean
```

**Visual C++**

```c++
public:
virtual property bool IsNull {
    bool get () sealed;
}
```
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Indicates if this object contains a value that can be represented as a DateTime

**Namespace:**  [MySQL.Data.Types](#)

**Assembly:**  MySQL.Data (in MySQL.Data.dll) Version: 6.2.2.0
# Syntax

**C#**

```csharp
public bool IsValidDateTime { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property IsValidDateTime As Boolean
```

**Visual C++**

```csharp
public:

    property bool IsValidDateTime {
        bool get ();
    }
```
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Retrieves the millisecond value of this object.

**Namespace:** [MySQL.Data.Types](#)  
**Assembly:** [MySql.Data](#) (in [MySql.Data.dll](#)) Version: 6.2.2.0
Syntax

C#

public int Millisecond { get; set; }

Visual Basic (Declaration)

Public Property Millisecond As Integer

Visual C++

public:
property int Millisecond {
    int get ();
    void set (int value);
}
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLDateTime...::Minute Property

MySQLDateTime Structure  See Also  Send Feedback

Returns the minute portion of this datetime

Namespace:  MySql.Data.Types
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public int Minute { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Minute As Integer
```

**Visual C++**

```cpp
public:
property int Minute {
    int get ()
    void set (int value);
}
```
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns the month portion of this datetime

Namespace:  MySql.Data.Types
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public int Month { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Month As Integer
```

**Visual C++**

```cpp
public:
property int Month {
    int get ();
    void set (int value);
}
```
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLDateTime..Second Property

MySQLDateTime Structure  See Also  Send Feedback

Returns the second portion of this datetime

**Namespace:**  [MySQL.Data.Types](#)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Second { get; set; }

Visual Basic (Declaration)

Public Property Second As Integer

Visual C++

public:
property int Second {
    int get ();
    void set (int value);
}
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Retrieves the value of this `MySqlDateTime` as a `DateTime` object.

**Namespace:**  [MySQL.Data.Types](http://example.com)

**Assembly:**  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public DateTime Value { get; }

Visual Basic (Declaration)

Public ReadOnly Property Value As DateTime

Visual C++

public:
property DateTime Value {
    DateTime get ();
}

See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns the year portion of this datetime

**Namespace:**  [MySQL.Data.Types](#)

**Assembly:**  MySqL.Data (in MySqL.Data.dll) Version: 6.2.2.0
Syntax

C#

public int Year { get; set; }

Visual Basic (Declaration)

Public Property Year As Integer

Visual C++

public:
property int Year {
    int get ();
    void set (int value);
}
See Also

MySQLDateTime Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Data.Types
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public struct MySqlDecimal

Visual Basic (Declaration)

Public Structure MySqlDecimal

Visual C++

public value class MySqlDecimal
See Also

MySQLDecimal Members
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySqlDecimal** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from <em>ValueType</em>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <em>Object</em> to attempt to free resources and perform other cleanup operations before the <em>Object</em> is reclaimed by garbage collection. (Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance. (Inherited from <em>ValueType</em>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <em>Type</em> of the current instance. (Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <em>Object</em>. (Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Overides <em>ValueType</em>::<em>ToString</em>())</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>IsNull</td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td></td>
</tr>
</tbody>
</table>
The **MySQLDecimal** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Indicates whether this instance and a specified object are equal.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>ValueType</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>ValueType</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToDouble</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Overrides <code>ValueType::&lt;&gt;::ToString()</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySQLDecimal Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Decimal...:::toDouble Method

MySQLDecimal Structure  See Also  Send Feedback

Namespace:  MySql.Data.Types
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

public double ToDouble()

**Visual Basic (Declaration)**

Public Function ToDouble As Double

**Visual C++**

public:

double ToDouble()
See Also

MySQLDecimal Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDecimal::ToString Method

Namespace: MySql.Data.Types
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public override string ToString()

Visual Basic (Declaration)

Public Overrides Function ToString As String

Visual C++

public:
virtual String^ ToString() override
See Also

MySQLDecimal Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlDecimal` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsNull</td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySqlDecimal Structure
MySql.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLDecimal..:::IsNull Property

**Namespace:**  [MySQL.Data.Types](#)
**Assembly:**  MySQL.Data (in MySql.Data.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public bool IsNull { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property IsNull As Boolean
```

**Visual C++**

```cpp
public:
virtual property bool IsNull {
   bool get () sealed;
}
```
See Also

MySQLDecimal Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace: MySql.Data.Types
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
Syntax

C#

public byte Precision { get; set; }

Visual Basic (Declaration)

Public Property Precision As Byte

Visual C++

public:
property unsigned char Precision {
    unsigned char get ();
    void set (unsigned char value);
}
See Also

MySQLDecimal Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLDecimal..::.Scale Property

Namespace:  MySql.Data.Types
Assembly:  MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public byte Scale { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Scale As Byte
```

**Visual C++**

```cpp
public:
property unsigned char Scale {
  unsigned char get ();
  void set (unsigned char value);
}
```
See Also

MySQLDecimal Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLDecimal..::..Value Property

Namespace: MySql.Data.Types
Assembly: MySql.Data (in MySql.Data.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public decimal Value { get; }
```

### Visual Basic (Declaration)

```
Public ReadOnly Property Value As Decimal
```

### Visual C++

```c++
public:
    property Decimal Value {
        Decimal get ();
    }
```
See Also

MySQLDecimal Structure
MySQL.Data.Types Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌟 SchemaManager</td>
<td></td>
</tr>
</tbody>
</table>

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
SchemaManager Class

**Namespace:** [MySql.Web.Common](https://example.com)
**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#
public static class SchemaManager

Visual Basic (Declaration)
Public NotInheritable Class SchemaManager

Visual C++
public ref class SchemaManager abstract sealed
Inheritance Hierarchy

System::Object
   MySql.Web.Common::SchemaManager
See Also

SchemaManager Members
MySql.Web.Common Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `SchemaManager` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Gets the most recent version of the schema.</td>
</tr>
</tbody>
</table>
See Also

SchemaManager Class
MySql.Web.Common Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `SchemaManager` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Gets the most recent version of the schema.</td>
</tr>
</tbody>
</table>
See Also

SchemaManager Class
MySql.Web.Common Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
SchemaManager..:.Version Property

Gets the most recent version of the schema.

**Namespace:**  [MySQL.Web.Common](#)  
**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public static int Version { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property Version As Integer

Visual C++

public:
static property int Version {
    int get ();
}

Field Value

The most recent version number of the schema.
See Also

SchemaManager Class
MySql.Web.Common Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQLProfileProvider</td>
<td></td>
</tr>
</tbody>
</table>

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Web.Profile
Syntax

C#

public class MySQLProfileProvider : ProfileProvider

Visual Basic (Declaration)

Public Class MySQLProfileProvider
    Inherits ProfileProvider

Visual C++

public ref class MySQLProfileProvider : public ProfileProvider
Inheritance Hierarchy

System..::.Object
   System.Configuration.Provider..::.ProviderBase
      System.Configuration..::.SettingsProvider
         System.Web.Profile..::.ProfileProvider
            MySql.Web.Profile..::.MySQLProfileProvider
See Also

MySQLProfileProvider Members
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLProfileProvider` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQLProfileProvider</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeleteInactiveProfiles</strong></td>
<td>When overridden in a derived class, deletes all user-profile data which the last activity date occurred before. (Overrides <code>ProfileProvider...::DeleteInactiveProfiles</code>.</td>
</tr>
<tr>
<td><strong>DeleteProfiles</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>FindInactiveProfilesByUserName</strong></td>
<td>When overridden in a derived class, retrieves profile information which the last activity date occurred on or before name matches the specified user name. (Overrides <code>ProfileProvider...::FindInactiveProfilesByUserName</code>.</td>
</tr>
<tr>
<td><strong>FindProfilesByUserName</strong></td>
<td>When overridden in a derived class, retrieves profile information which the user name matches the specified user names. (Overrides <code>ProfileProvider...::FindProfilesByUserName</code>.</td>
</tr>
<tr>
<td><strong>GetAllInactiveProfiles</strong></td>
<td>When overridden in a derived class, retrieves user-profile data source for profiles in which the last activity date occurred before specified date. (Overrides <code>ProfileProvider...::GetAllInactiveProfiles</code>.</td>
</tr>
<tr>
<td><strong>GetAllProfiles</strong></td>
<td>When overridden in a derived class, retrieves the data source. (Overrides <code>ProfileProvider...::GetAllProfiles</code>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>GetNumberOfInactiveProfiles</strong></td>
<td>When overridden in a derived class, returns last activity date occurred on or before the specified date. (Overrides ProfileProvider...:::GetNumberOfInactiveProfiles.)</td>
</tr>
<tr>
<td><strong>GetPropertyValues</strong></td>
<td>Returns the collection of settings property values for the specified application instance and settings property group. (Overrides SettingsProvider...:::GetPropertyValues.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Initialize</strong></td>
<td>Initializes the provider. (Overrides ProviderBase...:::Initialize.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>SetPropertyValues</strong></td>
<td>Sets the values of the specified group of property settings. (Overrides SettingsProvider...:::SetPropertyValues.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ApplicationName</strong></td>
<td>Gets or sets the name of the currently running application. (Overrides <code>SettingsProvider..::ApplicationName</code>.)</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Gets a brief, friendly description suitable for display in administrative tools or other user interfaces (UIs). (Inherited from <code>ProviderBase</code>.)</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Gets the friendly name used to refer to the provider during configuration. (Inherited from <code>ProviderBase</code>.)</td>
</tr>
</tbody>
</table>
See Also

MySQLProfileProvider Class
MySQL.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Web.Profile
Syntax

C#

public MySQLProfileProvider()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySQLProfileProvider()
See Also

MySQLProfileProvider Class
MySQL.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLProfileProvider` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeleteInactiveProfiles</strong></td>
<td>When overridden in a derived class, deletes all user-profile data which the last activity date occurred before the specified date. (Overrides <strong>ProfileProvider..::.DeleteInactiveProfiles</strong>.)</td>
</tr>
<tr>
<td><strong>DeleteProfiles</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <strong>Object</strong> to attempt to free resources and perform other cleanup operations before the <strong>Object</strong> is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>FindInactiveProfilesByUserName</strong></td>
<td>When overridden in a derived class, retrieves profile information which the last activity date occurred on or before the specified date. (Overrides <strong>ProfileProvider..::.FindInactiveProfilesByUserName</strong>.)</td>
</tr>
<tr>
<td><strong>FindProfilesByUserName</strong></td>
<td>When overridden in a derived class, retrieves profile information which the user name matches the specified user name. (Overrides <strong>ProfileProvider..::.FindProfilesByUserName</strong>.)</td>
</tr>
<tr>
<td><strong>GetAllInactiveProfiles</strong></td>
<td>When overridden in a derived class, retrieves user-profile data source for profiles in which the last activity date occurred on or before the specified date. (Overrides <strong>ProfileProvider..::.GetAllInactiveProfiles</strong>.)</td>
</tr>
<tr>
<td><strong>GetAllProfiles</strong></td>
<td>When overridden in a derived class, retrieves the data source. (Overrides <strong>ProfileProvider..::.GetAllProfiles</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
**GetNumberOfInactiveProfiles**

When overridden in a derived class, returns the number of profiles last activity date occurred on or before the specified date. (Overrides `ProfileProvider..::.GetNumberOfInactiveProfiles(DateTime)`.)

**GetPropertyValues**

Returns the collection of settings property values for the specified application instance and settings property group. (Overrides `SettingsProvider..::.GetPropertyValues(SettingsContext, SettingsPropertyCollection)`.)

**GetType**

Gets the `Type` of the current instance. (Inherited from `Object`.)

**Initialize**

 Initializes the provider. (Overrides `ProviderBase..::.Initialize(String)`.)

**MemberwiseClone**

Creates a shallow copy of the current `Object`. (Inherited from `Object`.)

**SetPropertyValues**

Sets the values of the specified group of property settings. (Overrides `SettingsProvider..::.SetPropertyValues(SettingsContext, SettingsPropertyValueCollection)`.)

**ToString**

Returns a `String` that represents the current instance. (Inherited from `Object`.)
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
When overridden in a derived class, deletes all user-profile data for profiles in which the last activity date occurred before the specified date.

**Namespace:** [MySql.Web.Profile](#)
**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
**Syntax**

### C#

```csharp
public override int DeleteInactiveProfiles(
    ProfileAuthenticationOption authenticationOption,
    DateTime userInactiveSinceDate
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function DeleteInactiveProfiles ( _
    authenticationOption As ProfileAuthenticationOption, _
    userInactiveSinceDate As DateTime _
) As Integer
```

### Visual C++

```cpp
public:
    virtual int DeleteInactiveProfiles(
        ProfileAuthenticationOption authenticationOption,
        DateTime userInactiveSinceDate
    ) override
```

**Parameters**

- **authenticationOption**
  
  Type: `System.Web.Profile..::.ProfileAuthenticationOption`
  
  One of the `ProfileAuthenticationOption` values, specifying whether anonymous, authenticated, or both types of profiles are deleted.

- **userInactiveSinceDate**
  
  Type: `System::::DateTime`
  
  A `DateTime` that identifies which user profiles are considered inactive. If the `LastActivityDate` value of a user profile occurs on or before this date and time, the profile is considered inactive.

**Return Value**
The number of profiles deleted from the data source.
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySQLProfileProvider::DeleteProfiles Method

MySQLProfileProvider Class  See Also  Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **DeleteProfiles(array<String>[][])** | When overridden in a derived class, deletes profile properties and information for profiles that match the supplied list of user names. (Overrides ProfileProvider::<DeleteProfiles(array<
| **DeleteProfiles(ProfileInfoCollection)** | When overridden in a derived class, deletes profile properties and information for the supplied profiles. (Overrides ProfileProvider::<DeleteProfiles(Profile


See Also

MySQLProfileProvider Class
MySQLProfileProvider Members
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
When overridden in a derived class, deletes profile properties and information for profiles that match the supplied list of user names.

**Namespace:** [MySQL.Web.Profile](#)  
**Assembly:** MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override int DeleteProfiles(
    string[] usernames
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function DeleteProfiles (_
    usernames As String() _
) As Integer
```

### Visual C++

```cpp
public:
virtual int DeleteProfiles(
    array<String^[>^ usernames
) override
```

## Parameters

**usernames**

Type: array of System::String[]

A string array of user names for profiles to be deleted.

## Return Value

The number of profiles deleted from the data source.
See Also

MySQLProfileProvider Class
DeleteProfiles Overload
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
When overridden in a derived class, deletes profile properties and information for the supplied list of profiles.

**Namespace:**  [MySQL.Web.Profile](https://example.com)

**Assembly:**  [MySQL.Web](https://example.com) (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public override int DeleteProfiles(
    ProfileInfoCollection profiles
)

Visual Basic (Declaration)

Public Overrides Function DeleteProfiles ( _
    profiles As ProfileInfoCollection _
) As Integer

Visual C++

public:
    virtual int DeleteProfiles( 
        ProfileInfoCollection^ profiles 
    ) override

Parameters

profiles
    Type: System.Web.Profile::.ProfileInfoCollection
        A ProfileInfoCollection of information about profiles that are to be deleted.

Return Value

The number of profiles deleted from the data source.
See Also

MySQLProfileProvider Class
DeleteProfiles Overload
 MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Profile Provider...:::FindInactiveProfilesByUserName Method

MySQL Profile Provider Class See Also Send Feedback

When overridden in a derived class, retrieves profile information for profiles in which the last activity date occurred on or before the specified date and the user name matches the specified user name.

**Namespace:** MySql.Web.Profile

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override ProfileInfoCollection FindInactiveProfilesByUserName(ProfileAuthenticationOption authenticationOption, string usernameToMatch, DateTime userInactiveSinceDate, int pageIndex, int pageSize, out int totalRecords)

Visual Basic (Declaration)

Public Overrides Function FindInactiveProfilesByUserName ( _
    authenticationOption As ProfileAuthenticationOption, _
    usernameToMatch As String, _
    userInactiveSinceDate As DateTime, _
    pageIndex As Integer, _
    pageSize As Integer, _
    <OutAttribute> ByRef totalRecords As Integer _
) As ProfileInfoCollection

Visual C++

public:
    virtual ProfileInfoCollection^ FindInactiveProfilesByUserName( ProfileAuthenticationOption authenticationOption, 
        String^ usernameToMatch, 
        DateTime userInactiveSinceDate, 
        int pageIndex, 
        int pageSize, 
        [OutAttribute] int% totalRecords
    ) override

Parameters

authenticationOption
    Type: System.Web.Profile..::..ProfileAuthenticationOption
    One of the ProfileAuthenticationOption values, specifying whether
anonymous, authenticated, or both types of profiles are returned.

usernameToMatch
  Type: System::String
  The user name to search for.

userInactiveSinceDate
  Type: System::DateTime
  A DateTime that identifies which user profiles are considered inactive. If the LastActivityDate value of a user profile occurs on or before this date and time, the profile is considered inactive.

pageIndex
  Type: System::Int32
  The index of the page of results to return.

pageSize
  Type: System::Int32
  The size of the page of results to return.

totalRecords
  Type: System::Int32
  When this method returns, contains the total number of profiles.

Return Value

A ProfileInfoCollection containing user profile information for inactive profiles where the user name matches the supplied usernameToMatch parameter.
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
When overridden in a derived class, retrieves profile information for profiles in which the user name matches the specified user names.

**Namespace:**  [MySQL.Web.Profile](https://example.com)  
**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public override ProfileInfoCollection FindProfilesByUserName(
    ProfileAuthenticationOption authenticationOption,
    string usernameToMatch,
    int pageIndex,
    int pageSize,
    out int totalRecords
)

Visual Basic (Declaration)

Public Overrides Function FindProfilesByUserName ( _
    authenticationOption As ProfileAuthenticationOption, _
    usernameToMatch As String, _
    pageIndex As Integer, _
    pageSize As Integer, _
    <OutAttribute> ByRef totalRecords As Integer _
) As ProfileInfoCollection

Visual C++

public:
    virtual ProfileInfoCollection^ FindProfilesByUserName(
        ProfileAuthenticationOption authenticationOption,
        String^ usernameToMatch,
        int pageIndex,
        int pageSize,
        [OutAttribute] int% totalRecords
    ) override

Parameters

authenticationOption
    Type: System.Web.Profile..::.ProfileAuthenticationOption
    One of the ProfileAuthenticationOption values, specifying whether anonymous, authenticated, or both types of profiles are returned.
usernameToMatch
   Type: System::String
   The user name to search for.

pageIndex
   Type: System::Int32
   The index of the page of results to return.

pageSize
   Type: System::Int32
   The size of the page of results to return.

totalRecords
   Type: System::Int32 %
   When this method returns, contains the total number of profiles.

Return Value

A ProfileInfoCollection containing user-profile information for profiles where the user name matches the supplied usernameToMatch parameter.
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
When overridden in a derived class, retrieves user-profile data from the data source for profiles in which the last activity date occurred on or before the specified date.

Namespace:  MySQL.Web.Profile
Syntax

C#

public override ProfileInfoCollection GetAllInactiveProfiles(
    ProfileAuthenticationOption authenticationOption,
    DateTime userInactiveSinceDate,
    int pageIndex,
    int pageSize,
    out int totalRecords
)

Visual Basic (Declaration)

Public Overrides Function GetAllInactiveProfiles ( _
    authenticationOption As ProfileAuthenticationOption, _
    userInactiveSinceDate As DateTime, _
    pageIndex As Integer, _
    pageSize As Integer, _
    <OutAttribute> ByRef totalRecords As Integer _
) As ProfileInfoCollection

Visual C++

public:
virtual ProfileInfoCollection^ GetAllInactiveProfiles(
    ProfileAuthenticationOption authenticationOption,
    DateTime userInactiveSinceDate,
    int pageIndex,
    int pageSize,
    [OutAttribute] int% totalRecords
) override

Parameters

authenticationOption
Type: System.Web.Profile..::.ProfileAuthenticationOption
One of the ProfileAuthenticationOption values, specifying whether anonymous, authenticated, or both types of profiles are returned.
userInactiveSinceDate
   Type: System..::..DateTime
   A DateTime that identifies which user profiles are considered inactive. If the LastActivityDate of a user profile occurs on or before this date and time, the profile is considered inactive.

pageIndex
   Type: System..::..Int32
   The index of the page of results to return.

pageSize
   Type: System..::..Int32
   The size of the page of results to return.

totalRecords
   Type: System..::..Int32 %
   When this method returns, contains the total number of profiles.

**Return Value**

A ProfileInfoCollection containing user-profile information about the inactive profiles.
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
When overridden in a derived class, retrieves user profile data for all profiles in the data source.

**Namespace:**  MySql.Web.Profile  
**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override ProfileInfoCollection GetAllProfiles(
    ProfileAuthenticationOption authenticationOption,
    int pageIndex,
    int pageSize,
    out int totalRecords
)

Visual Basic (Declaration)

Public Overrides Function GetAllProfiles (_
    authenticationOption As ProfileAuthenticationOption, _
    pageIndex As Integer, _
    pageSize As Integer, _
    <OutAttribute> ByRef totalRecords As Integer _
) As ProfileInfoCollection

Visual C++

public:
    virtual ProfileInfoCollection^ GetAllProfiles(
    ProfileAuthenticationOption authenticationOption,
    int pageIndex,
    int pageSize,
    [OutAttribute] int% totalRecords
) override

Parameters

authenticationOption
    Type: System.Web.Profile...::ProfileAuthenticationOption
    One of the ProfileAuthenticationOption values, specifying whether
    anonymous, authenticated, or both types of profiles are returned.

pageIndex
    Type: System...::Int32
    The index of the page of results to return.
pageSize
   Type: System::Int32
   The size of the page of results to return.

totalRecords
   Type: System::Int32 %
   When this method returns, contains the total number of profiles.

Return Value

A ProfileInfoCollection containing user-profile information for all profiles in the data source.
See Also

MySQLProfileProvider Class
MySQL.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
When overridden in a derived class, returns the number of profiles in which the last activity date occurred on or before the specified date.

**Namespace:**  [MySql.Web.Profile](https://example.com)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override int GetNumberOfInactiveProfiles(
    ProfileAuthenticationOption authenticationOption,
    DateTime userInactiveSinceDate
)

Visual Basic (Declaration)

Public Overrides Function GetNumberOfInactiveProfiles ( _
    authenticationOption As ProfileAuthenticationOption, _
    userInactiveSinceDate As DateTime _
) As Integer

Visual C++

public:
virtual int GetNumberOfInactiveProfiles(
    ProfileAuthenticationOption authenticationOption,
    DateTime userInactiveSinceDate
) override

Parameters

authenticationOption
Type: System.Web.Profile::ProfileAuthenticationOption
One of the ProfileAuthenticationOption values, specifying whether anonymous, authenticated, or both types of profiles are returned.

userInactiveSinceDate
Type: System::DateTime
A DateTime that identifies which user profiles are considered inactive. If the LastActivityDate of a user profile occurs on or before this date and time, the profile is considered inactive.

Return Value
The number of profiles in which the last activity date occurred on or before the specified date.
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Profile Provider...::GetPropertyValues Method

MySQLProfileProvider Class  See Also  Send Feedback

Returns the collection of settings property values for the specified application instance and settings property group.

Namespace:  MySql.Web.Profile
Syntax

C#

```csharp
public override SettingsPropertyValueCollection GetPropertyValues(
    SettingsContext context,
    SettingsPropertyCollection collection
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function GetPropertyValues ( _
    context As SettingsContext, _
    collection As SettingsPropertyCollection _
) As SettingsPropertyValueCollection
```

**Visual C++**

```cpp
public:
    virtual SettingsPropertyValueCollection^ GetPropertyValues(
        SettingsContext^ context,
        SettingsPropertyCollection^ collection
    ) override
```

**Parameters**

context

Type: `System.Configuration::::SettingsContext`
A `SettingsContext` describing the current application use.

collection

Type: `System.Configuration::::SettingsPropertyCollection`
A `SettingsPropertyCollection` containing the settings property group whose values are to be retrieved.

**Return Value**

A `SettingsPropertyValueCollection` containing the values for the specified
settings property group.
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLProfileProvider...:::Initialize Method
MySQLProfileProvider Class  See Also  Send Feedback

Initializes the provider.

**Namespace:**  [MySQL.Web.Profile](#)
**Assembly:**  [MySQL.Web](#) (in MySQL.Web.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override void Initialize(
    string name,
    NameValueCollection config
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Sub Initialize (_
    name As String, _
    config As NameValueCollection _
)
```

### Visual C++

```cpp
public:
virtual void Initialize(
    String^ name,
    NameValueCollection^ config
) override
```

## Parameters

**name**

Type: `System::String`

The friendly name of the provider.

**config**

Type: `System.Collections.Specialized::NameValueCollection`

A collection of the name/value pairs representing the provider-specific attributes specified in the configuration for this provider.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System..::.ArgumentNullException</td>
<td>The name of the provider is null.</td>
</tr>
<tr>
<td></td>
<td>The name of the provider has a length of zero.</td>
</tr>
<tr>
<td></td>
<td>An attempt is made to call Initialize(String, NameValueCollection) on a provider after the provider has already been initialized.</td>
</tr>
</tbody>
</table>
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Sets the values of the specified group of property settings.

**Namespace:**  [MySQL.Web.Profile](#)
**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override void SetPropertyValues(
    SettingsContext context,
    SettingsPropertyValueCollection collection
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub SetPropertyValues ( _
    context As SettingsContext, _
    collection As SettingsPropertyValueCollection _
)
```

Visual C++

```cpp
public:
virtual void SetPropertyValues(
    SettingsContext^ context,
    SettingsPropertyValueCollection^ collection
) override
```

Parameters

category
  Type: `System.Configuration:::SettingsContext`
  A `SettingsContext` describing the current application usage.

collection
  Type: `System.Configuration:::SettingsPropertyValueCollection`
  A `SettingsPropertyValueCollection` representing the group of property settings to set.
See Also

MySQLProfileProvider Class
MySQL.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLProfileProvider` type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationName</td>
<td>Gets or sets the name of the currently running application. (Overrides SettingsProvider..:::ApplicationName.)</td>
</tr>
<tr>
<td>Description</td>
<td>Gets a brief, friendly description suitable for display in administrative tools or other user interfaces (UIs). (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the friendly name used to refer to the provider during configuration. (Inherited from ProviderBase.)</td>
</tr>
</tbody>
</table>
See Also

MySQLProfileProvider Class
MySql.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Profile Provider Class

This property gets or sets the name of the currently running application.

**Namespace:** MySql.Web.Profile

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override string ApplicationName { get; set; }

Visual Basic (Declaration)

Public Overrides Property ApplicationName As String

Visual C++

public:
virtual property String^ ApplicationName {
    String^ get () override;
    void set (String^ value) override;
}

Field Value

Return Value

A String that contains the application's shortened name, which does not contain a full path or extension, for example, SimpleAppSettings.
See Also

MySQLProfileProvider Class
MySQL.Web.Profile Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
# Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="images/resource_icon.png" alt="Resources" /> Resources</td>
<td>A strongly-typed resource class, for looking up localized strings, etc.</td>
</tr>
</tbody>
</table>

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
A strongly-typed resource class, for looking up localized strings, etc.

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public class Resources

Visual Basic (Declaration)

Public Class Resources

Visual C++

public ref class Resources
Inheritance Hierarchy

System...:::Object
MySql.Web.Properties...:::Resources
See Also

Resources Members
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `Resources` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <strong>Object</strong> to attempt to free resources and perform other cleanup operations before the <strong>Object</strong> is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CannotDeleteAPopulatedRole</td>
<td>Looks up a localized string similar to Cannot delete a populated role.</td>
</tr>
<tr>
<td>CannotRetrieveHashedPasswords</td>
<td>EnablePasswordRetrieval to true when PasswordFormat is Hashed is not supported.</td>
</tr>
<tr>
<td>CannotUnencodeHashedPwd</td>
<td>Looks up a localized string similar to Cannot unencode a hashed password.</td>
</tr>
<tr>
<td>ChangePasswordCanceled</td>
<td>Looks up a localized string similar to Change password operation was canceled.</td>
</tr>
<tr>
<td>Culture</td>
<td>Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.</td>
</tr>
<tr>
<td>ErrorInitOfMembershipProvider</td>
<td>Looks up a localized string similar to There was an error during membership provider initilization.</td>
</tr>
<tr>
<td>ErrorInitOfRoleProvider</td>
<td>Looks up a localized string similar to There was an error during role provider initilization.</td>
</tr>
<tr>
<td>ErrorInitProfileProvider</td>
<td>Looks up a localized string similar to There was an error during profile provider initilization.</td>
</tr>
<tr>
<td>ErrorResettingPassword</td>
<td>Looks up a localized string similar to There was an error resetting the password.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IllegalRoleName</td>
<td>Looks up a localized string similar to Role names must not be null or empty..</td>
</tr>
<tr>
<td>IllegalUserName</td>
<td>Looks up a localized string similar to User names must not be null or empty..</td>
</tr>
<tr>
<td>IncorrectPasswordAnswer</td>
<td>Looks up a localized string similar to Incorrect password answer..</td>
</tr>
<tr>
<td>InvalidCharactersInUserName</td>
<td>Looks up a localized string similar to Invalid characters in user name..</td>
</tr>
<tr>
<td>MissingOrWrongSchema</td>
<td>Looks up a localized string similar to Unable to initialize provider. Missing or incorrect schema..</td>
</tr>
<tr>
<td>MonoDoesNotSupportHash</td>
<td>Looks up a localized string similar to The mono runtime did not support hashed passwords. Please use clear or encrypted passwords..</td>
</tr>
<tr>
<td>NewPasswordValidationFailed</td>
<td>Looks up a localized string similar to Change password canceled due to New password validation failure..</td>
</tr>
<tr>
<td>NotEnoughNonAlphaNumericInPwd</td>
<td>Looks up a localized string similar to Non alpha numeric characters in '{0}' needs to be greater than or equal to '{1}'..</td>
</tr>
<tr>
<td>PasswordAnswerInvalid</td>
<td>Looks up a localized string similar to Password answer supplied is invalid..</td>
</tr>
<tr>
<td>PasswordNotLongEnough</td>
<td>Looks up a localized string similar to The length of parameter '{0}' needs to be greater or equal to '{1}'..</td>
</tr>
</tbody>
</table>
PasswordQuestionInvalid

类似密码问题提供的内容无效。

PasswordRequiredForReset

查找与密码答案相似的本地化字符串，用于密码重置。

PasswordResetCanceledNotValid

查找与重置密码被取消相似的本地化字符串，由于密码验证失败。

PasswordResetNotEnabled

查找与密码重置未启用相似的本地化字符串。

PasswordRetrievalNotEnabled

查找与密码检索未启用相似的本地化字符串。

ProfileUpdateFailed

查找与配置更新失败相似的本地化字符串。

ResourceManager

返回用于此类的缓存的ResourceManager实例。

RoleNameAlreadyExists

查找与角色名称已经存在相似的本地化字符串。

RoleNameNotFound

查找与角色名称未找到相似的本地化字符串。

schema1

类似CREATE TABLE mysql_Membership(`PKID` varchar(36) NOT NULL, Username varchar(255) NOT NULL, ApplicationName varchar(255) NOT NULL, Email varchar(128) NOT NULL, Comment varchar(255) default NULL, Password varchar(128) NOT NULL, PasswordQuestion varchar(255) default NULL, ...
PasswordAnswer varchar(255) default NULL, IsApproved tinyint(1) default NULL, LastActivityDate datetim [rest of string was truncated]"

Looks up a localized string similar to ALTER TABLE mysql_Membership ADD PasswordKey char(32) AFTER Password, ADD PasswordFormat tinyint AFTER PasswordKey, CHANGE Email Email VARCHAR(128), COMMENT='2'; .

Looks up a localized string similar to ALTER TABLE my_aspnet_Membership CONVERT TO CHARACTER SET DEFAULT; ALTER TABLE my_aspnet_Roles CONVERT TO CHARACTER SET DEFAULT; ALTER TABLE my_aspnet_UsersInRoles CONVERT TO CHARACTER SET DEFAULT; UPDATE my_aspnet_SchemaVersion SET version=4 WHERE version=3; .

Looks up a localized string similar to ALTER TABLE my_aspnet_Sessions CONVERT TO CHARACTER SET DEFAULT; ALTER TABLE my_aspnet_Sessions MODIFY SessionItems LONGBLOB; UPDATE
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnableToCreateApplication</td>
<td>Looks up a localized string similar to Unable to create application..</td>
</tr>
<tr>
<td>UnableToCreateUser</td>
<td>Looks up a localized string similar to Unable to create user..</td>
</tr>
<tr>
<td>UnableToLockOutUser</td>
<td>Looks up a localized string similar to Unable to lock out user..</td>
</tr>
<tr>
<td>UnableToRetrieveProfileData</td>
<td>Looks up a localized string similar to Unable to retrieve profile data from database..</td>
</tr>
<tr>
<td>UnableToUpdateFailureCount</td>
<td>Looks up a localized string similar to Unable to update failure count. Membership database may be corrupt..</td>
</tr>
<tr>
<td>UnsupportedPasswordFormat</td>
<td>Looks up a localized string similar to Unsupported password format..</td>
</tr>
<tr>
<td>UserIsAlreadyInRole</td>
<td>Looks up a localized string similar to User is already in role..</td>
</tr>
<tr>
<td>UserIsLockedOut</td>
<td>Looks up a localized string similar to The supplied user is locked out..</td>
</tr>
<tr>
<td>UsernameNotFound</td>
<td>Looks up a localized string similar to Username not found..</td>
</tr>
<tr>
<td>UserNotInRole</td>
<td>Looks up a localized string similar to User not in role..</td>
</tr>
<tr>
<td>ValidatePasswordCanceled</td>
<td>Looks up a localized string similar to The validate password operation was canceled..</td>
</tr>
</tbody>
</table>
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The Resources type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **Resources** type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="flag.png" alt="" /> CannotDeleteAPopulatedRole</td>
<td>Looks up a localized string similar to Cannot delete a populated role.</td>
</tr>
<tr>
<td><img src="flag.png" alt="" /> CannotRetrieveHashedPasswords</td>
<td>Looks up a localized string similar to Setting EnablePasswordRetrieval to true when PasswordFormat is Hashed is not supported.</td>
</tr>
<tr>
<td><img src="flag.png" alt="" /> CannotUnencodeHashedPwd</td>
<td>Looks up a localized string similar to Cannot unencode a hashed password.</td>
</tr>
<tr>
<td><img src="flag.png" alt="" /> ChangePasswordCanceled</td>
<td>Looks up a localized string similar to Change password operation was canceled.</td>
</tr>
<tr>
<td><img src="flag.png" alt="" /> Culture</td>
<td>Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.</td>
</tr>
<tr>
<td><img src="flag.png" alt="" /> ErrorInitOfMembershipProvider</td>
<td>Looks up a localized string similar to There was an error during membership provider initilization.</td>
</tr>
<tr>
<td><img src="flag.png" alt="" /> ErrorInitOfRoleProvider</td>
<td>Looks up a localized string similar to There was an error during role provider initilization.</td>
</tr>
<tr>
<td><img src="flag.png" alt="" /> ErrorInitProfileProvider</td>
<td>Looks up a localized string similar to There was an error during profile provider initilization.</td>
</tr>
<tr>
<td><img src="flag.png" alt="" /> ErrorResettingPassword</td>
<td>Looks up a localized string similar to There was an error resetting the password.</td>
</tr>
</tbody>
</table>
**IllegalRoleName** 
Looks up a localized string similar to Role names must not be null or empty..

**IllegalUserName** 
Looks up a localized string similar to User names must not be null or empty..

**IncorrectPasswordAnswer** 
Looks up a localized string similar to Incorrect password answer..

**InvalidCharactersInUserName** 
Looks up a localized string similar to Invalid characters in user name..

**MissingOrWrongSchema** 
Looks up a localized string similar to Unable to initialize provider. Missing or incorrect schema..

**MonoDoesNotSupportHash** 
Looks up a localized string similar to The mono runtime did not support hashed passwords. Please use clear or encrypted passwords..

**NewPasswordValidationFailed** 
Looks up a localized string similar to Change password canceled due to New password validation failure..

**NotEnoughNonAlphaNumericInPwd** 
Looks up a localized string similar to Non alpha numeric characters in '{0}' needs to be greater than or equal to '{1}'..

**PasswordAnswerInvalid** 
Looks up a localized string similar to Password answer supplied is invalid..

**PasswordNotLong Enough** 
Looks up a localized string similar to The length of parameter '{0}' needs to be greater or equal to '{1}'..

**Looks up a localized string**
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PasswordQuestionInvalid</td>
<td>similar to Password question supplied is invalid..</td>
</tr>
<tr>
<td>PasswordRequiredForReset</td>
<td>Looks up a localized string similar to Password answer required for password reset..</td>
</tr>
<tr>
<td>PasswordResetCanceledNotValid</td>
<td>Looks up a localized string similar to Reset password canceled due to password validation failure..</td>
</tr>
<tr>
<td>PasswordResetNotEnabled</td>
<td>Looks up a localized string similar to Password Reset is not enabled..</td>
</tr>
<tr>
<td>PasswordRetrievalNotEnabled</td>
<td>Looks up a localized string similar to Password Retrieval Not Enabled..</td>
</tr>
<tr>
<td>ProfileUpdateFailed</td>
<td>Looks up a localized string similar to Profile update failed..</td>
</tr>
<tr>
<td>ResourceManager</td>
<td>Returns the cached ResourceManager instance used by this class.</td>
</tr>
<tr>
<td>RoleNameAlreadyExists</td>
<td>Looks up a localized string similar to Role name already exists..</td>
</tr>
<tr>
<td>RoleNameNotFound</td>
<td>Looks up a localized string similar to Role name not found..</td>
</tr>
<tr>
<td>schema1</td>
<td>Looks up a localized string similar to CREATE TABLE mysql_Membership(<code>PKID</code> varchar(36) NOT NULL, Username varchar(255) NOT NULL, ApplicationName varchar(255) NOT NULL, Email varchar(128) NOT NULL, Comment varchar(255) default NULL, Password varchar(128) NOT NULL, PasswordQuestion varchar(255) default NULL,</td>
</tr>
</tbody>
</table>
PasswordAnswer varchar(255)
default NULL, IsApproved
tinyint(1) default NULL,
LastActivityDate datetim [rest of
string was truncated];

Looks up a localized string
similar to ALTER TABLE
mysql_Membership ADD
PasswordKey char(32) AFTER
Password, ADD PasswordFormat
tinyint AFTER PasswordKey,
CHANGE Email Email
VARCHAR(128),
COMMENT=‘2’; .

Looks up a localized string
similar to ALTER TABLE
my_aspnet_Membership
CONVERT TO CHARACTER
SET DEFAULT; ALTER TABLE
my_aspnet_Roles CONVERT TO
CHARACTER SET DEFAULT;
ALTER TABLE
my_aspnet_UsersInRoles
CONVERT TO CHARACTER
SET DEFAULT; UPDATE
my_aspnet_SchemaVersion SET
version=4 WHERE version=3; .

Looks up a localized string
similar to ALTER TABLE
my_aspnet_Sessions CONVERT
TO CHARACTER SET
DEFAULT; ALTER TABLE
my_aspnet_Sessions MODIFY
SessionItems LONGBLOB;
UPDATE
my_aspnet_SchemaVersion SET version=6;

UnableToCreateApplication
Looks up a localized string similar to Unable to create application..

UnableToCreateUser
Looks up a localized string similar to Unable to create user..

UnableToLockOutUser
Looks up a localized string similar to Unable to lock out user..

UnableToRetrieveProfileData
Looks up a localized string similar to Unable to retrieve profile data from database..

UnableToUpdateFailureCount
Looks up a localized string similar to Unable to update failure count. Membership database may be corrupt..

UnsupportedPasswordFormat
Looks up a localized string similar to Unsupported password format..

UserIsAlreadyInRole
Looks up a localized string similar to User is already in role..

UserIsLockedOut
Looks up a localized string similar to The supplied user is locked out..

UsernameNotFound
Looks up a localized string similar to Username not found..

UserNotInRole
Looks up a localized string similar to User not in role..

ValidatePasswordCanceled
Looks up a localized string similar to The validate password operation was canceled..
See Also

Resources Class
 MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Cannot delete a populated role..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  [MySQL.Web](#) (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string CannotDeleteAPopulatedRole { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property CannotDeleteAPopulatedRole As String

Visual C++

public:
static property String^ CannotDeleteAPopulatedRole {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
CannotRetrieveHashedPasswords Property

Looks up a localized string similar to Setting EnablePasswordRetrieval to true when PasswordFormat is Hashed is not supported..

**Namespace:**  MySql.Web.Properties

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string CannotRetrieveHashedPasswords { get; }

Visual Basic (Declaration)

Public Shared Readonly Property CannotRetrieveHashedPasswords As Str

Visual C++

public:
static property String^ CannotRetrieveHashedPasswords {
    String^ get ();
}

See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Cannot unencode a hashed password..

**Namespace:** MySql.Web.Properties

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string CannotUnencodeHashedPwd { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property CannotUnencodeHashedPwd As String

Visual C++

public:
static property String^ CannotUnencodeHashedPwd {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Change password operation was canceled..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  [MySQL.Web](#) (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string ChangePasswordCanceled { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ChangePasswordCanceled As String

Visual C++

public:
static property String^ ChangePasswordCanceled {
  String^ get ();
}

}
See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

**Namespace:**  [MySql.Web.Properties](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static CultureInfo Culture { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Property Culture As CultureInfo
```

**Visual C++**

```cpp
public:
static property CultureInfo^ Culture {
    CultureInfo^ get ();
    void set (CultureInfo^ value);
}
```
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to There was an error during membership provider initialization..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string ErrorInitOfMembershipProvider { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ErrorInitOfMembershipProvider As Str

Visual C++

public:
static property String^ ErrorInitOfMembershipProvider {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
ErrorInitOfRoleProvider Property

Looks up a localized string similar to There was an error during role provider initialization..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

**C#**

public static string ErrorInitOfRoleProvider { get; }

**Visual Basic (Declaration)**

Public Shared ReadOnly Property ErrorInitOfRoleProvider As String

**Visual C++**

public:
static property String^ ErrorInitOfRoleProvider {
    String^ get ()
}
See Also

- Resources Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to There was an error during profile provider initialization.

**Namespace:** [MySql.Web.Properties](#)

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public static string ErrorInitProfileProvider { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property ErrorInitProfileProvider As String
```

**Visual C++**

```cpp
public:
static property String^ ErrorInitProfileProvider {
    String^ get ();
}
```
See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to "There was an error resetting the password."

**Namespace:**  MySql.Web.Properties

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string ErrorResettingPassword { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ErrorResettingPassword As String

Visual C++

public:
static property String^ ErrorResettingPassword {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Role names must not be null or empty.


**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string IllegalRoleName { get; }

Visual Basic (Declaration)

Public Shared Readonly Property IllegalRoleName As String

Visual C++

public:
static property String^ IllegalRoleName {
    String^ get();
}

See Also

Resources Class  
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to User names must not be null or empty.

**Namespace:**  [MySQL.Web.Properties](#)  
**Assembly:**  MySQL.Web (in MySQL.Web.dll)  
**Version:**  6.2.2.0
**Syntax**

**C#**

public static string IllegalUserName { get; }

**Visual Basic (Declaration)**

Public Shared ReadOnly Property IllegalUserName As String

**Visual C++**

public:
static property String^ IllegalUserName {
    String^ get ();
}

See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Incorrect password answer..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string IncorrectPasswordAnswer { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property IncorrectPasswordAnswer As String

Visual C++

public:
static property String^ IncorrectPasswordAnswer {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
InvalidCharactersInUserName Property

Looks up a localized string similar to Invalid characters in user name..

**Namespace:** [MySql.Web.Properties](#)

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string InvalidCharactersInUserName { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property InvalidCharactersInUserName As String

Visual C++

public:
static property String^ InvalidCharactersInUserName {
    String^ get ();
}


See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unable to initialize provider. Missing or incorrect schema..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string MissingOrWrongSchema { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property MissingOrWrongSchema As String

Visual C++

public:
static property String^ MissingOrWrongSchema {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The mono runtime did not support hashed passwords. Please use clear or encrypted passwords..

**Namespace:**  [MySql.Web.Properties](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string MonoDoesNotSupportHash { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property MonoDoesNotSupportHash As String

Visual C++

public:
static property String^ MonoDoesNotSupportHash {
    String^ get ();
}

See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Change password canceled due to New password validation failure..

**Namespace:**  MySQL.Web.Properties  
**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string NewPasswordValidationFailed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property NewPasswordValidationFailed As String

Visual C++

public:
static property String^ NewPasswordValidationFailed {
    String^ get ();
}

See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
NotEnoughNonAlphaNumericInPwd Property

Looks up a localized string similar to Non alpha numeric characters in '{0}' needs to be greater than or equal to '{1}'..

**Namespace:**  MySql.Web.Properties

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string NotEnoughNonAlphaNumericInPwd { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property NotEnoughNonAlphaNumericInPwd As String

Visual C++

public:
static property String^ NotEnoughNonAlphaNumericInPwd {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Password answer supplied is invalid..

**Namespace:**  [MySql.Web.Properties](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string PasswordAnswerInvalid { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PasswordAnswerInvalid As String

Visual C++

public:
static property String^ PasswordAnswerInvalid {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
PasswordNotLongEnough Property

Looks up a localized string similar to The length of parameter '{0}' needs to be greater or equal to '{1}'.

Namespace: MySql.Web.Properties
Syntax

C#

public static string PasswordNotLongEnough { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PasswordNotLongEnough As String

Visual C++

public:
static property String^ PasswordNotLongEnough {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Password question supplied is invalid..

**Namespace:**  MySql.Web.Properties

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public static string PasswordQuestionInvalid { get; }
```

Visual Basic (Declaration)

```vbnet
Public Shared ReadOnly Property PasswordQuestionInvalid As String
```

Visual C++

```cpp
public:  
static property String^ PasswordQuestionInvalid {  
    String^ get ();  
}
```
See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Password answer required for password reset.

Namespace:  MySql.Web.Properties
# Syntax

**C#**

```csharp
public static string PasswordRequiredForReset { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property PasswordRequiredForReset As String
```

**Visual C++**

```cpp
public:
static property String^ PasswordRequiredForReset {
    String^ get ();
}
```
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
PasswordResetCanceledNotValid Property

Looks up a localized string similar to Reset password canceled due to password validation failure.

Namespace: MySql.Web.Properties
Syntax

C#

public static string PasswordResetCanceledNotValid { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PasswordResetCanceledNotValid As String

Visual C++

public:
static property String^ PasswordResetCanceledNotValid {
    String^ get();
}

See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Password Reset is not enabled..

**Namespace:**  [MySQL.Web.Properties](#)  
**Assembly:**  MySQL.Web (in MySQL.Web.dll)  Version: 6.2.2.0
Syntax

C#

public static string PasswordResetNotEnabled { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PasswordResetNotEnabled As String

Visual C++

public:
static property String^ PasswordResetNotEnabled {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Password Retrieval Not Enabled.


**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string PasswordRetrievalNotEnabled { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property PasswordRetrievalNotEnabled As String

Visual C++

public:
static property String^ PasswordRetrievalNotEnabled {
    String^ get();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Profile update failed.

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  MySQL.Web (in MySQL.Web.dll)  Version: 6.2.2.0
Syntax

C#

public static string ProfileUpdateFailed { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ProfileUpdateFailed As String

Visual C++

public:
static property String^ ProfileUpdateFailed {
    String^ get ()
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Returns the cached ResourceManager instance used by this class.

**Namespace:**  [MySql.Web.Properties](#)
**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static ResourceManager ResourceManager { get; }

Visual Basic (Declaration)

Public Shared ReadOnly PropertyResourceManager As ResourceManager

Visual C++

public:
static property ResourceManager^ ResourceManager { ResourceManager^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Role name already exists..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  [MySQL.Web](#) (in [MySQL.Web.dll](#)) Version: 6.2.2.0
Syntax

C#

public static string RoleNameAlreadyExists { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property RoleNameAlreadyExists As String

Visual C++

public:
static property String^ RoleNameAlreadyExists { 
    String^ get ();
    }
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Role name not found..

**Namespace:**  [MySQL.Web.Properties]

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string RoleNameNotFound { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property RoleNameNotFound As String

Visual C++

public:
static property String^ RoleNameNotFound {
    String^ get ();
}
}
See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to CREATE TABLE
mysql_Membership(`PKID` varchar(36) NOT NULL, Username varchar(255)
NOT NULL, ApplicationName varchar(255) NOT NULL, Email varchar(128)
NOT NULL, Comment varchar(255) default NULL, Password varchar(128)
NOT NULL, PasswordQuestion varchar(255) default NULL, PasswordAnswer
varchar(255) default NULL, IsApproved tinyint(1) default NULL,
LastActivityDate datetime [rest of string was truncated]";

**Namespace:**  [MySQL.Web.Properties](https://www.mysql.com/)
**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string schema1 { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property schema1 As String

Visual C++

public:
static property String^ schema1 {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to ALTER TABLE mysql_Membership ADD PasswordKey char(32) AFTER Password, ADD PasswordFormat tinyint AFTER PasswordKey, CHANGE Email Email VARCHAR(128), COMMENT='2';.

**Namespace:**  [MySql.Web.Properties](#)  
**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string schema2 { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property schema2 As String

Visual C++

public:
static property String^ schema2 { String^ get ();
}


See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace:  MySql.Web.Properties
Syntax

C#

public static string schema3 { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property schema3 As String

Visual C++

public:
static property String^ schema3 {
    String^ get ();
}

See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to ALTER TABLE my_aspnet_Membership CONVERT TO CHARACTER SET DEFAULT; ALTER TABLE my_aspnet_Roles CONVERT TO CHARACTER SET DEFAULT; ALTER TABLE my_aspnet_UsersInRoles CONVERT TO CHARACTER SET DEFAULT; UPDATE my_aspnet_SchemaVersion SET version=4 WHERE version=3;


**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string schema4 { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property schema4 As String

Visual C++

public:
static property String^ schema4 {
    String^ get ();
}

}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
☐ Visual C++
MySQL Connector/Net
Resources:::schema5 Property

Namespace:  MySql.Web.Properties
**Syntax**

**C#**

```csharp
public static string schema5 { get; }
```

**Visual Basic (Declaration)**

Public Shared ReadOnly Property schema5 As String

**Visual C++**

```cpp
public:
static property String^ schema5 {
    String^ get ();
}
```
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to ALTER TABLE my_aspnet_Sessions CONVERT TO CHARACTER SET DEFAULT; ALTER TABLE my_aspnet_Sessions MODIFY SessionItems LONGBLOB; UPDATE my_aspnet_SchemaVersion SET version=6;.

**Namespace:**  [MySql.Web.Properties](#)
**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string schema6 { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property schema6 As String

Visual C++

public:
static property String^ schema6 {
    String^ get ();
}

See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
UnableToCreateApplication Property

Namespace: MySql.Web.Properties
Syntax

C#

public static string UnableToCreateApplication { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UnableToCreateApplication As String

Visual C++

public:
static property String^ UnableToCreateApplication {
    String^ get ();
}
See Also

Resources Class
 MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unable to create user..

**Namespace:**  MySql.Web.Properties

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string UnableTo CreateUser { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UnableTo CreateUser As String

Visual C++

public:
static property String^ UnableTo CreateUser {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unable to lock out user..

**Namespace:** [MySQL.Web.Properties](#)  
**Assembly:** MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string UnableToLockOutUser { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UnableToLockOutUser As String

Visual C++

public:
static property String^ UnableToLockOutUser {
    String^ get ();
}

See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
UnableToRetrieveProfileData Property


Syntax

C#

public static string UnableToRetrieveProfileData { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UnableToRetrieveProfileData As String

Visual C++

public:
static property String^ UnableToRetrieveProfileData {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unable to update failure count. Membership database may be corrupt.

**Namespace:** MySql.Web.Properties
**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
## Syntax

**C#**

```csharp
public static string UnableToUpdateFailureCount { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property UnableToUpdateFailureCount As String
```

**Visual C++**

```cpp
public:
static property String^ UnableToUpdateFailureCount {
    String^ get ();
}
```
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Unsupported password format..

**Namespace:** MySql.Web.Properties  
**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
**Syntax**

**C#**

public static string UnsupportedPasswordFormat { get; }

**Visual Basic (Declaration)**

Public Shared Readonly Property UnsupportedPasswordFormat As String

**Visual C++**

public:
static property String^ UnsupportedPasswordFormat {
    String^ get ();
}
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to User is already in role..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  MySQL.Web (in MySQL.Web.dll)  Version: 6.2.2.0
Syntax

C#

public static string UserIsAlreadyInRole { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UserIsAlreadyInRole As String

Visual C++

public:
static property String^ UserIsAlreadyInRole {
    String^ get ();
}
See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The supplied user is locked out..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  [MySQL.Web](#) (in [MySQL.Web.dll](#)) Version: 6.2.2.0
## Syntax

**C#**

```csharp
public static string UserIsLockedOut { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ReadOnly Property UserIsLockedOut As String
```

**Visual C++**

```c++
public:
static property String^ UserIsLockedOut {
    String^ get ();
}
```
See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to Username not found..

**Namespace:**  [MySQL.Web.Properties](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string UsernameNotFound { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UsernameNotFound As String

Visual C++

public:
static property String^ UsernameNotFound {
    String^ get ();
}

See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to User not in role..

Namespace:  MySql.Web.Properties
Syntax

C#

public static string UserNotInRole { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property UserNotInRole As String

Visual C++

public:
static property String^ UserNotInRole {
    String^ get ();
}
See Also

Resources Class
MySQL.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Looks up a localized string similar to The validate password operation was canceled.

**Namespace:** MySql.Web.Properties
**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public static string ValidatePasswordCanceled { get; }

Visual Basic (Declaration)

Public Shared ReadOnly Property ValidatePasswordCanceled As String

Visual C++

public:
static property String^ ValidatePasswordCanceled {
    String^ get ();
}


See Also

Resources Class
MySql.Web.Properties Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MySQLMembershipProvider</strong></td>
<td>Manages storage of membership information for an ASP.NET application in a MySQL database.</td>
</tr>
<tr>
<td><strong>MySQLRoleProvider</strong></td>
<td>Manages storage of role membership information for an ASP.NET application in a MySQL database.</td>
</tr>
</tbody>
</table>

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL MembershipProvider Class

Manages storage of membership information for an ASP.NET application in a MySQL database.

**Namespace:**  [MySql.Web.Security](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public sealed class MySQLMembershipProvider : MembershipProvider
```

Visual Basic (Declaration)

```vbnet
Public NotInheritable Class MySQLMembershipProvider
    Inherits MembershipProvider
```

Visual C++

```cpp
public ref class MySQLMembershipProvider sealed : public MembershipProvider
```
Remarks

This class is used by the Membership and MembershipUser classes to provide membership services for ASP.NET applications using a MySQL database.
<configuration>
  <connectionStrings>
    <add name="LocalMySqlService" connectionString="server=localhost;user id=myuser;password=mypass;database=test" />
  </connectionStrings>
  <system.web>
    <authentication mode="Forms">
      <forms loginUrl="login.aspx" name=".ASPXFORMSAUTH" />
    </authentication>
    <authorization>
      <deny users="?" />
    </authorization>
    <membership defaultProvider="MySQLProvider" userIsOnlineTimeWindow="false" />
    <providers>
      <add name="MySQLProvider" type="MySql.Web.Security.MySQLMembershipProvider" connectionStringName="LocalMySqlService" applicationName="MyApplication" enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestionAndAnswer="true" requiresUniqueEmail="false" passwordFormat="Hashed" maxInvalidPasswordAttempts="5" passwordAttemptWindow="10" />
    </providers>
  </system.web>
</configuration>
Inheritance Hierarchy

System..:::Object
System.Configuration.Provider..:::ProviderBase
System.Web.Security..:::MembershipProvider
  MySql.Web.Security..:::MySQLMembershipProvider
See Also

MySQLMembershipProvider Members

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLMembershipProvider** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQLMembershipProvider</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✪ <strong>ChangePassword</strong></td>
<td>Changes the password.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider..::.ChangePassword</code>, <code>String</code>.)</td>
</tr>
<tr>
<td>✪ <strong>ChangePasswordQuestionAndAnswer</strong></td>
<td>Changes the password question and answer.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider..::.ChangePasswordQuestionAndAnswer</code>, <code>String</code>, <code>String</code>.)</td>
</tr>
<tr>
<td>✪ <strong>CreateUser</strong></td>
<td>Adds a new membership user to the data source.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider..::.CreateUser</code>, <code>String</code>, <code>String</code>, <code>Boolean</code>, <code>Object</code>, <code>MembershipCreateStatus</code>.)</td>
</tr>
<tr>
<td>✪ <strong>DecryptPassword</strong></td>
<td>Decrypts an encrypted password.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MembershipProvider</code>.)</td>
</tr>
<tr>
<td>✪ <strong>DeleteUser</strong></td>
<td>Removes a user from the membership data source.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider..::.DeleteUser</code>, <code>Boolean</code>.)</td>
</tr>
<tr>
<td>✪ <strong>EncryptPassword</strong></td>
<td>Encrypts a password.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MembershipProvider</code>.)</td>
</tr>
<tr>
<td>✪ <strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>✪ <strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>✪ <strong>FindUsersByEmail</strong></td>
<td>Gets a collection of membership users where the e-mail address contains the specified e-mail address to match.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider..::.FindUsersByEmail</code>, <code>Int32</code>, <code>Int32%</code>.)</td>
</tr>
<tr>
<td>✪ <strong>FindUsersByName</strong></td>
<td>Gets a collection of membership users where the user name contains the specified user name to match.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider..::.FindUsersByName</code>, <code>Int32</code>, <code>Int32%</code>.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>GetAllUsers</strong></td>
<td>Gets a collection of all the users in the data source. (Overrides <code>MembershipProvider..::.GetAllUsers</code>).</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular object. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>GetNumberOfUsersOnline</strong></td>
<td>Gets the number of users currently accessing the application. (Overrides <code>MembershipProvider..::.GetNumberOfUsersOnline</code>).</td>
</tr>
<tr>
<td><strong>GetPassword</strong></td>
<td>Gets the password for the specified user. (Overrides <code>MembershipProvider..::.GetPassword</code>).</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>GetUser</strong></td>
<td>Gets the user name associated with the specified user name. (Overrides <code>MembershipProvider..::.GetUser</code>).</td>
</tr>
<tr>
<td><strong>GetUserNameByEmail</strong></td>
<td>Gets the user name associated with the specified email address. (Overrides <code>MembershipProvider..::.GetUserNameByEmail</code>).</td>
</tr>
<tr>
<td><strong>Initialize</strong></td>
<td>Initializes the MySQL membership provider with the property values specified in the ASP.NET application's configuration file. This method is not intended to be used directly. (Overrides <code>ProviderBase..::.Initialize(String,NameValueCollection)</code>).</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current object. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>OnValidatingPassword</strong></td>
<td>Raises the <code>ValidatingPassword</code> event if an event handler has been defined. (Inherited from <code>MembershipProvider</code>).</td>
</tr>
<tr>
<td><strong>ResetPassword</strong></td>
<td>Resets a user's password to a new, automatically generated password. (Overrides <code>MembershipProvider..::.ResetPassword(String,String)</code>).</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current object. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>UnlockUser</strong></td>
<td>Unlocks the user. (Overrides <code>MembershipProvider..::.UnlockUser(String)</code>).</td>
</tr>
<tr>
<td><strong>UpdateUser</strong></td>
<td>Updates information about a user in the data source. (Overrides <code>MembershipProvider..::.UpdateUser(MembershipUser)</code>).</td>
</tr>
<tr>
<td><strong>ValidateUser</strong></td>
<td>Verifies that the specified user name and password exist in the data source. (Overrides <code>MembershipProvider..::.ValidateUser(String,String)</code>).</td>
</tr>
</tbody>
</table>
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationName</td>
<td>The name of the application using the MySQL membership provider. (Overrides MembershipProvider::</td>
</tr>
<tr>
<td>Description</td>
<td>Gets a brief, friendly description suitable for display in administrative tools or other user interfaces. (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>EnablePasswordReset</td>
<td>Indicates whether the membership provider is configured to allow users to reset their passwords. (Overrides MembershipProvider::</td>
</tr>
<tr>
<td>EnablePasswordRetrieval</td>
<td>Indicates whether the membership provider is configured to allow users to retrieve their passwords. (Overrides MembershipProvider::</td>
</tr>
<tr>
<td>MaxInvalidPasswordAttempts</td>
<td>Gets the number of invalid password or password-answer attempts allowed before the membership user is locked out. (Overrides MembershipProvider::</td>
</tr>
<tr>
<td>MinRequiredNonAlphanumericCharacters</td>
<td>Gets the minimum number of special characters that must be present in a valid password. (Overrides MembershipProvider::</td>
</tr>
<tr>
<td>MinRequiredPasswordLength</td>
<td>Gets the minimum length required for a password. (Overrides MembershipProvider::</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the friendly name used to refer to the provider during configuration. (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>PasswordAttemptWindow</td>
<td>Gets the number of minutes in which password or password-answer attempts are allowed before the membership user is locked out. (Overrides MembershipProvider::</td>
</tr>
<tr>
<td>PasswordFormat</td>
<td>Gets a value indicating the format for storing passwords in the membership data store. (Overrides MembershipProvider::</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>PasswordStrengthRegularExpression</td>
<td>Gets the regular expression used to evaluate a password. (Overrides MembershipProvider::PasswordStrengthRegularExpression)</td>
</tr>
<tr>
<td>RequiresQuestionAndAnswer</td>
<td>Gets a value indicating whether the membership provider is configured to require the user to answer a password question for password reset and retrieval. (Overrides MembershipProvider::RequiresQuestionAndAnswer)</td>
</tr>
<tr>
<td>RequiresUniqueEmail</td>
<td>Gets a value indicating whether the membership provider is configured to require a unique e-mail address for each user name. (Overrides MembershipProvider::RequiresUniqueEmail)</td>
</tr>
<tr>
<td>WriteExceptionsToEventLog</td>
<td>Gets or sets a value indicating whether exceptions are written to the event log.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ValidatingPassword</td>
<td>Occurs when a user is created, a password is changed, or a password is reset. (Inherited from MembershipProvider.)</td>
</tr>
</tbody>
</table>
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Syntax

C#

public MySQLMembershipProvider()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySQLMembershipProvider()
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLMembershipProvider** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChangePassword</td>
<td>Changes the password. (Overrides MembershipProvider..:::ChangePassword(String, String, String))</td>
</tr>
<tr>
<td>ChangePasswordQuestionAndAnswer</td>
<td>Changes the password question and answer. (Overrides MembershipProvider..:::ChangePasswordQuestionAndAnswer(String, String, String))</td>
</tr>
<tr>
<td>CreateUser</td>
<td>Adds a new membership user to the data source. (Overrides MembershipProvider..:::CreateUser(String, String, Boolean, Object, MembershipCreateStatus))</td>
</tr>
<tr>
<td>DecryptPassword</td>
<td>Decrypts an encrypted password. (Inherited from MembershipProvider.)</td>
</tr>
<tr>
<td>DeleteUser</td>
<td>Removes a user from the membership data source. (Overrides MembershipProvider..:::DeleteUser(String, Boolean))</td>
</tr>
<tr>
<td>EncryptPassword</td>
<td>Encrypts a password. (Inherited from MembershipProvider.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object equals the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is collected. (Inherited from Object.)</td>
</tr>
<tr>
<td>FindUsersByEmail</td>
<td>Gets a collection of membership users where the e-mail address contains the specified e-mail address to match. (Overrides MembershipProvider..:::FindUsersByEmailAddress(String, Int32, Int32%))</td>
</tr>
<tr>
<td>FindUsersByName</td>
<td>Gets a collection of membership users where the user name contains the specified user name to match. (Overrides MembershipProvider..:::FindUsersByName(String, Int32, Int32%))</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>GetAllUsers</strong></td>
<td>Gets a collection of all the users in the data source.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider...::GetAllUsers(Int32,)</code>).</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>GetNumberOfUsersOnline</strong></td>
<td>Gets the number of users currently accessing the application.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider...::GetNumberOfUsersOnline</code>).</td>
</tr>
<tr>
<td><strong>GetPassword</strong></td>
<td>Gets the password for the specified user.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider...::GetPassword(String,)</code>).</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>GetUser</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetUserNameByEmail</strong></td>
<td>Gets the user name associated with the specified email address.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider...::GetUserNameByEmail(String,)</code>).</td>
</tr>
<tr>
<td><strong>Initialize</strong></td>
<td>Initializes the MySQL membership provider with the property values</td>
</tr>
<tr>
<td></td>
<td>specified in the ASP.NET application's configuration file.</td>
</tr>
<tr>
<td></td>
<td>This method is not intended to be used directly.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>ProviderBase...::Initialize(String,)</code>).</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>OnValidatingPassword</strong></td>
<td>Raises the <code>ValidatingPassword</code> event if an event handler has been defined.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>MembershipProvider</code>).</td>
</tr>
<tr>
<td><strong>ResetPassword</strong></td>
<td>Resets a user's password to a new, automatically generated password.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider...::ResetPassword(String,)</code>).</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>UnlockUser</strong></td>
<td>Unlocks the user.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider...::UnlockUser(String,)</code>).</td>
</tr>
<tr>
<td><strong>UpdateUser</strong></td>
<td>Updates information about a user in the data source.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider...::UpdateUser(MembershipUser,)</code>).</td>
</tr>
<tr>
<td><strong>ValidateUser</strong></td>
<td>Verifies that the specified user name and password exist in the data source.</td>
</tr>
<tr>
<td></td>
<td>(Overrides <code>MembershipProvider...::ValidateUser(String,)</code>).</td>
</tr>
</tbody>
</table>
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Changes the password.

Syntax

C#

```csharp
public override bool ChangePassword(
    string username,
    string oldPassword,
    string newPassword
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function ChangePassword ( _
    username As String, _
    oldPassword As String, _
    newPassword As String _
) As Boolean
```

Visual C++

```cpp
public:
virtual bool ChangePassword( 
    String^ username, 
    String^ oldPassword, 
    String^ newPassword 
) override
```

Parameters

username
  Type: System::String
  The username.

oldPassword
  Type: System::String
  The old password.

newPassword
  Type: System::String
The new password.

**Return Value**

true if the password was updated successfully, false if the supplied old password is invalid, the user is locked out, or the user does not exist in the database.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Changes the password question and answer.

**Namespace:** [MySQL.Web.Security](https://www.mysql.com/productsconnectornet)

**Assembly:** [MySQL.Web](https://www.mysql.com/productsconnectornet) (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override bool ChangePasswordQuestionAndAnswer(
    string username,
    string password,
    string newPasswordQuestion,
    string newPasswordAnswer
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function ChangePasswordQuestionAndAnswer (_
    username As String, _
    password As String, _
    newPasswordQuestion As String, _
    newPasswordAnswer As String _
) As Boolean
```

Visual C++

```cpp
public:
virtual bool ChangePasswordQuestionAndAnswer(
    String^ username,
    String^ password,
    String^ newPasswordQuestion,
    String^ newPasswordAnswer
) override
```

Parameters

username
Type: `System::String`
The username.

password
Type: `System::String`
The password.
newPwdQuestion
  Type: System::String
  The new password question.

newPwdAnswer
  Type: System::String
  The new password answer.

Return Value

true if the update was successful; otherwise, false. A value of false is also
returned if the password is incorrect, the user is locked out, or the user does not
exist in the database.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Adds a new membership user to the data source.

**Namespace:**  [MySQL.Web.Security](#)  
**Assembly:**  [MySql.Web](#) (in MySql.Web.dll)  
**Version:**  6.2.2.0
Syntax

C#

public override MembershipUser CreateUser(
    string username,
    string password,
    string email,
    string passwordQuestion,
    string passwordAnswer,
    bool isApproved,
    Object providerUserKey,
    out MembershipCreateStatus status
)

Visual Basic (Declaration)

Public Overrides Function CreateUser (_
    username As String, _
    password As String, _
    email As String, _
    passwordQuestion As String, _
    passwordAnswer As String, _
    isApproved As Boolean, _
    providerUserKey As Object, _
    <OutAttribute> ByRef status As MembershipCreateStatus _
) As MembershipUser

Visual C++

public:
    virtual MembershipUser^ CreateUser(
        String^ username,
        String^ password,
        String^ email,
        String^ passwordQuestion,
        String^ passwordAnswer,
        bool isApproved,
        Object^ providerUserKey,
        [OutAttribute] MembershipCreateStatus% status
    ) override
**Parameters**

username
Type: System::String
The user name for the new user.

password
Type: System::String
The password for the new user.

demail
Type: System::String
The e-mail address for the new user.

passwordQuestion
Type: System::String
The password question for the new user.

passwordAnswer
Type: System::String
The password answer for the new user.

isApproved
Type: System::Boolean
Whether or not the new user is approved to be validated.

providerUserKey
Type: System::Object
The unique identifier from the membership data source for the user.

status
Type: System.Web.Security::MembershipCreateStatus
A MembershipCreateStatus enumeration value indicating whether the user was created successfully.

**Return Value**

A MembershipUser object populated with the information for the newly created user.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLMembershipProvider.DeletesUser Method

Removes a user from the membership data source.

## Syntax

### C#

```csharp
public override bool DeleteUser(
    string username,
    bool deleteAllRelatedData
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function DeleteUser ( _
    username As String, _
    deleteAllRelatedData As Boolean _
) As Boolean
```

### Visual C++

```cpp
public:
    virtual bool DeleteUser(
        String^ username,
        bool deleteAllRelatedData
    ) override
```

## Parameters

**username**
- Type: `System::::String`
- The name of the user to delete.

**deleteAllRelatedData**
- Type: `System::::Boolean`
- true to delete data related to the user from the database; false to leave data related to the user in the database.

## Return Value

true if the user was successfully deleted; otherwise, false.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a collection of membership users where the e-mail address contains the specified e-mail address to match.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public override MembershipUserCollection FindUsersByEmail(
    string emailToMatch,
    int pageIndex,
    int pageSize,
    out int totalRecords
)

Visual Basic (Declaration)

Public Overrides Function FindUsersByEmail ( _
    emailToMatch As String, _
    pageIndex As Integer, _
    pageSize As Integer, _
    <OutAttribute> ByRef totalRecords As Integer _
) As MembershipUserCollection

Visual C++

public:
    virtual MembershipUserCollection^ FindUsersByEmail(
        String^ emailToMatch,
        int pageIndex,
        int pageSize,
        [OutAttribute] int% totalRecords
    ) override

Parameters

emailToMatch
Type: System::String
The e-mail address to search for.

pageIndex
Type: System::Int32
The index of the page of results to return. pageIndex is zero-based.
pageSize
Type: System..::.Int32
The size of the page of results to return.

totalRecords
Type: System..::.Int32
The total number of matched users.

Return Value

A MembershipUserCollection collection that contains a page of pageSize MembershipUser objects beginning at the page specified by pageIndex.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Membership Provider::FindUsersByName Method

Gets a collection of membership users where the user name contains the specified user name to match.

**Namespace:** MySQL.Web.Security

**Assembly:** MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override MembershipUserCollection FindUsersByName(
    string usernameToMatch,
    int pageIndex,
    int pageSize,
    out int totalRecords
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function FindUsersByName ( _
    usernameToMatch As String, _
    pageIndex As Integer, _
    pageSize As Integer, _
    <OutAttribute> ByRef totalRecords As Integer _
) As MembershipUserCollection
```

### Visual C++

```cpp
public:
virtual MembershipUserCollection^ FindUsersByName(
    String^ usernameToMatch,
    int pageIndex,
    int pageSize,
    [OutAttribute] int% totalRecords
) override
```

## Parameters

**usernameToMatch**
Type: `System::String`
The user name to search for.

**pageIndex**
Type: `System::Int32`
The index of the page of results to return. pageIndex is zero-based.
pageSize
  Type: System::Int32
  The size of the page of results to return.

totalRecords
  Type: System::Int32 %
  The total number of matched users.

Return Value

A MembershipUserCollection collection that contains a page of pageSize MembershipUser objects beginning at the page specified by pageIndex.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Membership Provider PROVIDER...:::GetAllUsers Method
MySQLMembershipProvider Class  See Also  Send Feedback

Gets a collection of all the users in the data source in pages of data.

**Namespace:** [MySql.Web.Security](#)

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public override MembershipUserCollection GetAllUsers(
    int pageIndex,
    int pageSize,
    out int totalRecords
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function GetAllUsers ( _
    pageIndex As Integer, _
    pageSize As Integer, _
    <OutAttribute> ByRef totalRecords As Integer _
) As MembershipUserCollection
```

**Visual C++**

```cpp
public:
    virtual MembershipUserCollection^ GetAllUsers(
        int pageIndex,
        int pageSize,
        [OutAttribute] int% totalRecords
    ) override
```

### Parameters

**pageIndex**
- **Type:** `System::Int32`
- The index of the page of results to return. `pageIndex` is zero-based.

**pageSize**
- **Type:** `System::Int32`
- The size of the page of results to return.

**totalRecords**
- **Type:** `System::Int32 %`
- The size of the page of results to return.
The total number of matched users.

**Return Value**

A [MembershipUserCollection](#) collection that contains a page of pageSize[MembershipUser](#) objects beginning at the page specified by pageIndex.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLMembershipProvider...

Gets the number of users currently accessing the application.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override int GetNumberOfUsersOnline()

Visual Basic (Declaration)

Public Overrides Function GetNumberOfUsersOnline As Integer

Visual C++

public:
virtual int GetNumberOfUsersOnline() override

Return Value

The number of users currently accessing the application.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Membership Provider

GetPassword Method

MySQLMembershipProvider Class

See Also

Send Feedback

Gets the password for the specified user name from the data source.


Syntax

C#

public override string GetPassword(
    string username,
    string answer
)

Visual Basic (Declaration)

Public Overrides Function GetPassword ( _
    username As String, _
    answer As String _
) As String

Visual C++

public:
    virtual String^ GetPassword(
        String^ username,
        String^ answer
    ) override

Parameters

username
    Type: System::String
    The user to retrieve the password for.

answer
    Type: System::String
    The password answer for the user.

Return Value

The password for the specified user name.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Membership Provider...::: Get User Method

MySQLMembershipProvider Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GetUser(Object, Boolean)</strong></td>
<td>Gets user information from the data source based on the unique identifier for the membership user. Provides an option to update the last-activity date/time stamp for the user. (Overrides MembershipProviderFLICT.GetUser(Object, Boolean).)</td>
</tr>
<tr>
<td><strong>GetUser(String, Boolean)</strong></td>
<td>Gets information from the data source for a user. Provides an option to update the last-activity date/time stamp for the user. (Overrides MembershipProviderFLICT.GetUser(String, Boolean).)</td>
</tr>
</tbody>
</table>
See Also

MySQLMembershipProvider Class
MySQLMembershipProvider Members
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets user information from the data source based on the unique identifier for the membership user. Provides an option to update the last-activity date/time stamp for the user.

**Assembly:** MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public override MembershipUser GetUser(
    Object providerUserKey,
    bool userIsOnline
)

Visual Basic (Declaration)

Public Overrides Function GetUser (_
    providerUserKey As Object, _
    userIsOnline As Boolean _
) As MembershipUser

Visual C++

public:
    virtual MembershipUser^ GetUser(
        Object^ providerUserKey,
        bool userIsOnline
    ) override

Parameters

providerUserKey
    Type: System::Object
    The unique identifier for the membership user to get information for.

userIsOnline
    Type: System::Boolean
    true to update the last-activity date/time stamp for the user; false to return user information without updating the last-activity date/time stamp for the user.

Return Value
A MembershipUser object populated with the specified user's information from the data source.
See Also

MySQLMembershipProvider Class
GetUser Overload

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Membership Provider...

**GetUser Method (String, Boolean)**

**MySQLMembershipProvider Class**  See Also  Send Feedback

GETS INFORMATION FROM THE DATA SOURCE FOR A USER. PROVIDES AN OPTION TO UPDATE THE LAST-ACTIVITY DATE/TIME STAMP FOR THE USER.

**Namespace:** MySql.Web.Security

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#
public override MembershipUser GetUser(  
    string username,  
    bool userIsOnline  
)

Visual Basic (Declaration)
Public Overrides Function GetUser (  
    username As String,  
    userIsOnline As Boolean  
) As MembershipUser

Visual C++
public:  
virtual MembershipUser^ GetUser(  
    String^ username,  
    bool userIsOnline  
) override

Parameters

username  
Type: System::String  
The name of the user to get information for.

userIsOnline  
Type: System::Boolean  
true to update the last-activity date/time stamp for the user; false to return user information without updating the last-activity date/time stamp for the user.

Return Value
A MembershipUser object populated with the specified user's information from the data source.
See Also

MySQLMembershipProvider Class
GetUser Overload

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLMembershipProvider.GetUserByNameByEmailAddress Method

Gets the user name associated with the specified e-mail address.

**Syntax**

**C#**

```csharp
public override string GetUserByNameByEmail(
    string email
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function GetUserByNameByEmail ( _
    email As String _
) As String
```

**Visual C++**

```cpp
public:
virtual String^ GetUserByNameByEmail(
    String^ email
) override
```

**Parameters**

`email`

Type: `System::::String`

The e-mail address to search for.

**Return Value**

The user name associated with the specified e-mail address. If no match is found, return null.
See Also

MySQLMembershipProvider Class
MySQL_web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Initializes the MySQL membership provider with the property values specified in the ASP.NET application's configuration file. This method is not intended to be used directly from your code.

**Namespace:** MySql.Web.Security

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override void Initialize(
    string name,
    NameValueCollection config
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub Initialize (_
    name As String, _
    config As NameValueCollection _
)
```

Visual C++

```cpp
public:
    virtual void Initialize(
        _In_ String^ name,
        _In_ NameValueCollection^ config
    ) override
```

Parameters

name
Type: `System::String`
The name of the `MySQLMembershipProvider` instance to initialize.

cfg
Type: `System.Collections.Specialized::NameValueCollection`
A collection of the name/value pairs representing the provider-specific attributes specified in the configuration for this provider.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System..::.ArgumentNullException</code></td>
<td>config is a null reference.</td>
</tr>
<tr>
<td><code>System..::.InvalidOperationException</code></td>
<td>An attempt is made to call <code>Initialize(String, NameValueCollection)</code> on a provider after the provider has already been initialized.</td>
</tr>
<tr>
<td><code>System.Configuration.Provider..::.ProviderException</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Membership Provider Reset Password Method

Reset a user's password to a new, automatically generated password.

**Namespace:**  [MySQL.Web.Security](https://www.mysql.com)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override string ResetPassword(
    string username,
    string answer
)

Visual Basic (Declaration)

Public Overrides Function ResetPassword ( _
    username As String, _
    answer As String _
) As String

Visual C++

public:
virtual String^ ResetPassword(
    String^ username,
    String^ answer
) override

Parameters

username
    Type: System::String
    The user to reset the password for.

answer
    Type: System::String
    The password answer for the specified user.

Return Value

The new password for the specified user.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Unlocks the user.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

class CustomClass
{
    public override bool UnlockUser(string username)
    {
        // Implementation
    }
}

Visual Basic (Declaration)

Public Overrides Function UnlockUser( _
    ByVal username As String _
) As Boolean

Visual C++

public:
    virtual bool UnlockUser(_
        String^ username _
    ) override

Parameters

username

Type: System::String
The username.

Return Value

true if the membership user was successfully unlocked; otherwise, false. A value of false is also returned if the user does not exist in the database.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLMembershipProvider::UpdateUser Method

Updates information about a user in the data source.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public override void UpdateUser(
    MembershipUser user
)

Visual Basic (Declaration)

Public Overrides Sub UpdateUser ( _
    user As MembershipUser _
)

Visual C++

public:
virtual void UpdateUser(
    MembershipUser^ user
) override

Parameters

user

Type: System.Web.Security..:::MembershipUser
A MembershipUser object that represents the user to update and the updated information for the user.
See Also

MySQLMembershipProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Verifies that the specified user name and password exist in the data source.

**Namespace:**  [MySQL.Web.Security](#)  
**Assembly:**  [MySQL.Web](#) (in MySQL.Web.dll) Version: 6.2.2.0
**Syntax**

**C#**

```csharp
public override bool ValidateUser(
    string username,
    string password
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function ValidateUser ( _
    username As String, _
    password As String _
) As Boolean
```

**Visual C++**

```cpp
public:
    virtual bool ValidateUser(
        String^ username,
        String^ password
    ) override
```

**Parameters**

**username**

Type: `System::String`

The name of the user to validate.

**password**

Type: `System::String`

The password for the specified user.

**Return Value**

true if the specified username and password are valid; otherwise, false.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLMembershipProvider` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ApplicationName</strong></td>
<td>The name of the application using the MySQL membership provider. (Overrides <code>MembershipProvider...::</code> MembershipProvider...)</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Gets a brief, friendly description suitable for display in administrative tools or other user interfaces. (Inherited from <code>ProviderBase...</code>)</td>
</tr>
<tr>
<td><strong>EnablePasswordReset</strong></td>
<td>Indicates whether the membership provider is configured to allow users to reset their passwords. (Overrides <code>MembershipProvider...::</code> MembershipProvider...)</td>
</tr>
<tr>
<td><strong>EnablePasswordRetrieval</strong></td>
<td>Indicates whether the membership provider is configured to allow users to retrieve their passwords. (Overrides <code>MembershipProvider...::</code> MembershipProvider...)</td>
</tr>
<tr>
<td><strong>MaxInvalidPasswordAttempts</strong></td>
<td>Gets the number of invalid password or password-answer attempts allowed before the membership user is locked out. (Overrides <code>MembershipProvider...::</code> MembershipProvider...)</td>
</tr>
<tr>
<td><strong>MinRequiredNonAlphanumericCharacters</strong></td>
<td>Gets the minimum number of special characters that must be present in a valid password. (Overrides <code>MembershipProvider...::</code> MembershipProvider...)</td>
</tr>
<tr>
<td><strong>MinRequiredPasswordLength</strong></td>
<td>Gets the minimum length required for a password. (Overrides <code>MembershipProvider...::</code> MembershipProvider...)</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Gets the friendly name used to refer to the provider during configuration. (Inherited from <code>ProviderBase...</code>)</td>
</tr>
<tr>
<td><strong>PasswordAttemptWindow</strong></td>
<td>Gets the number of minutes in which password or password-answer attempts are allowed before the membership user is locked out. (Overrides <code>MembershipProvider...::</code> MembershipProvider...)</td>
</tr>
<tr>
<td><strong>PasswordFormat</strong></td>
<td>Gets a value indicating the format for storing passwords in the membership data store. (Overrides <code>MembershipProvider...::</code> MembershipProvider...)</td>
</tr>
</tbody>
</table>
PasswordStrengthRegularExpression

Gets the regular expression used to evaluate a password. (Overrides MembershipProvider..::PasswordStrengthRegularExpression)

RequiresQuestionAndAnswer

Gets a value indicating whether the membership provider is configured to require the user to answer a password question for password reset and retrieval. (Overrides MembershipProvider..::RequiresQuestionAndAnswer)

RequiresUniqueEmail

Gets a value indicating whether the membership provider is configured to require a unique e-mail address for each user. (Overrides MembershipProvider..::RequiresUniqueEmail)

WriteExceptionsToEventLog

Gets or sets a value indicating whether exceptions are written to the event log.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The name of the application using the MySQL membership provider.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override string ApplicationName { get; set; }

Visual Basic (Declaration)

Public Overrides Property ApplicationName As String

Visual C++

public:
virtual property String^ ApplicationName {
    String^ get () override;
    void set (String^ value) override;
}

Field Value

The name of the application using the MySQL membership provider. The default is the application virtual path.
Remarks

The ApplicationName is used by the MySqlMembershipProvider to separate membership information for multiple applications. Using different application names, applications can use the same membership database. Likewise, multiple applications can make use of the same membership data by simply using the same application name. Caution should be taken with multiple applications as the ApplicationName property is not thread safe during writes.
Examples

The following example shows the membership element being used in an applications web.config file. The application name setting is being used.

C#

```csharp
<membership defaultProvider="MySQLMembershipProvider">
    <providers>
        <add name="MySqlMembershipProvider"
            connectionStringName="LocalMySqlServer"
            enablePasswordRetrieval="true"
            enablePasswordReset="false"
            requiresQuestionAndAnswer="true"
            requiresUniqueEmail="false"
            passwordFormat="Encrypted"
            maxInvalidPasswordAttempts="3"
            passwordAttemptWindow="20"
            minRequiredNonAlphanumericCharacters="1"
            minRequiredPasswordLength="11"
            applicationName="MyApplication" />
    </providers>
</membership>
```
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLMembershipProvider::EnablePasswordReset Property

Indicates whether the membership provider is configured to allow users to reset their passwords.

Syntax

C#

public override bool EnablePasswordReset { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property EnablePasswordReset As Boolean

Visual C++

public:
virtual property bool EnablePasswordReset {
    bool get () override;
}

Field Value

ture if the membership provider supports password reset; otherwise, false. The default is true.
Remarks

Allows the user to replace their password with a new, randomly generated password. This can be especially handy when using hashed passwords since hashed passwords cannot be retrieved.
Examples

The following example shows the membership element being used in an applications web.config file.

C#

```csharp
<membership defaultProvider="MySQLMembershipProvider">
    <providers>
        <add name="MySqlMembershipProvider"
             connectionStringName="LocalMySqlServer"
             enablePasswordRetrieval="true"
             enablePasswordReset="false"
             requiresQuestionAndAnswer="true"
             requiresUniqueEmail="false"
             passwordFormat="Encrypted"
             maxInvalidPasswordAttempts="3"
             passwordAttemptWindow="20"
             minRequiredNonAlphanumericCharacters="1"
             minRequiredPasswordLength="11"
             applicationName="MyApplication" />
    </providers>
</membership>
```
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Indicates whether the membership provider is configured to allow users to retrieve their passwords.

**Namespace:**  MySQL.Web.Security  
**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override bool EnablePasswordRetrieval { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property EnablePasswordRetrieval As Boolean

Visual C++

public:
virtual property bool EnablePasswordRetrieval {
    bool get () override;
}

Field Value

true if the membership provider is configured to support password retrieval; otherwise, false. The default is false.
Remarks

If the system is configured to use hashed passwords, then retrieval is not possible. If the user attempts to initialize the provider with hashed passwords and enable password retrieval set to true then a ProviderException is thrown.
Examples

The following example shows the membership element being used in an applications web.config file.

C#

```csharp
<membership defaultProvider="MySQLMembershipProvider">
    <providers>
        <add name="MySqlMembershipProvider"
             connectionStringName="LocalMySqlServer"
             enablePasswordRetrieval="true"
             enablePasswordReset="false"
             requiresQuestionAndAnswer="true"
             requiresUniqueEmail="false"
             passwordFormat="Encrypted"
             maxInvalidPasswordAttempts="3"
             passwordAttemptWindow="20"
             minRequiredNonAlphanumericCharacters="1"
             minRequiredPasswordLength="11"
             applicationName="MyApplication" />
    </providers>
</membership>
```
See Also

MySQLMembershipProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLMembershipProvider.MaxInvalidPasswordAttempts Property

MySQLMembershipProvider Class  Example  See Also  Send Feedback

Gets the number of invalid password or password-answer attempts allowed before the membership user is locked out.

Syntax

C#

```csharp
public override int MaxInvalidPasswordAttempts { get; }
```

Visual Basic (Declaration)

```vbnet
Public Overrides ReadOnly Property MaxInvalidPasswordAttempts As Int
```

Visual C++

```cpp
public:
virtual property int MaxInvalidPasswordAttempts {
    int get () override;
}
```

Field Value

The number of invalid password or password-answer attempts allowed before the membership user is locked out.
Examples

The following example shows the membership element being used in an applications web.config file.

C#<membership defaultProvider="MySQLMembershipProvider">
  <providers>
    <add name="MySqlMembershipProvider" type="MySql.Web.Security.MySQLMembershipProvider" connectionStringName="LocalMySqlServer"
      enablePasswordRetrieval="true"
      enablePasswordReset="false"
      requiresQuestionAndAnswer="true"
      requiresUniqueEmail="false"
      passwordFormat="Encrypted"
      maxInvalidPasswordAttempts="3"
      passwordAttemptWindow="20"
      minRequiredNonAlphanumericCharacters="1"
      minRequiredPasswordLength="11"
      applicationName="MyApplication" />
  </providers>
</membership>
See Also

MySQLMembershipProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MinRequiredNonAlphanumericCharacters Property

MySQLMembershipProvider Class  Example  See Also  Send Feedback

Gets the minimum number of special characters that must be present in a valid password.

Syntax

C#

public override int MinRequiredNonAlphanumericCharacters { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property MinRequiredNonAlphanumericCharact

Visual C++

public:
virtual property int MinRequiredNonAlphanumericCharacters {
    int get () override;
}

Field Value

The minimum number of special characters that must be present in a valid password.
Examples

The following example shows the membership element being used in an applications web.config file.

C#

```csharp
<membership defaultProvider="MySQLMembershipProvider">
  <providers>
    <add name="MySqlMembershipProvider" type="MySql.Web.Security.MySQLMembershipProvider" connectionStringsName="LocalMySqlServer" enablePasswordRetrieval="true" enablePasswordReset="false" requiresQuestionAndAnswer="true" requiresUniqueEmail="false" passwordFormat="Encrypted" maxInvalidPasswordAttempts="3" passwordAttemptWindow="20" minRequiredNonAlphanumericCharacters="1" minRequiredPasswordLength="11" applicationName="MyApplication" />
  </providers>
</membership>
```
See Also

MySQLMembershipProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the minimum length required for a password.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  [MySql.Web](#) (in MySql.Web.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public override int MinRequiredPasswordLength { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides ReadOnly Property MinRequiredPasswordLength As Integer
```

**Visual C++**

```c++
public:
virtual property int MinRequiredPasswordLength {
    int get () override;
}
```

**Field Value**

The minimum length required for a password.
Examples

The following example shows the membership element being used in an applications web.config file.

```csharp
<membership defaultProvider="MySQLMembershipProvider">
    <providers>
        <add name="MySqlMembershipProvider" type="MySql.Web.Security.MySQLMembershipProvider"
        connectionStringName="LocalMySqlServer"
        enablePasswordRetrieval="true"
        enablePasswordReset="false"
        requiresQuestionAndAnswer="true"
        requiresUniqueEmail="false"
        passwordFormat="Encrypted"
        maxInvalidPasswordAttempts="3"
        passwordAttemptWindow="20"
        minRequiredNonAlphanumericCharacters="1"
        minRequiredPasswordLength="11"
        applicationName="MyApplication" />
    </providers>
</membership>
```
See Also

MySQLMembershipProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Membership Provider...:::PasswordAttemptWindow Property
MySQLMembershipProvider Class  Example  See Also  Send Feedback

Gets the number of minutes in which a maximum number of invalid password or password-answer attempts are allowed before the membership user is locked out.

**Syntax**

**C#**

```csharp
public override int PasswordAttemptWindow { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides ReadOnly Property PasswordAttemptWindow As Integer
```

**Visual C++**

```cpp
public:
virtual property int PasswordAttemptWindow {
    int get () override;
}
```

**Field Value**

The number of minutes in which a maximum number of invalid password or password-answer attempts are allowed before the membership user is locked out.
Examples

The following example shows the membership element being used in an applications web.config file.

C#

```csharp
<membership defaultProvider="MySQLMembershipProvider">
  <providers>
    <add name="MySqlMembershipProvider" type="MySql.Web.Security.MySQLMembershipProvider" connectionStringName="LocalMySqlServer"
        enablePasswordRetrieval="true"
        enablePasswordReset="false"
        requiresQuestionAndAnswer="true"
        requiresUniqueEmail="false"
        passwordFormat="Encrypted"
        maxInvalidPasswordAttempts="3"
        passwordAttemptWindow="20"
        minRequiredNonAlphanumericCharacters="1"
        minRequiredPasswordLength="11"
        applicationName="MyApplication" />
  </providers>
</membership>
```
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a value indicating the format for storing passwords in the membership data store.

**Namespace:**  [MySql.Web.Security](https://example.com)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

**C#**

```csharp
public override MembershipPasswordFormat PasswordFormat { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides ReadOnly Property PasswordFormat As MembershipPasswordFormat
```

**Visual C++**

```csharp
public:
virtual property MembershipPasswordFormat PasswordFormat {
    MembershipPasswordFormat get () override;
}
```

**Field Value**

One of the `MembershipPasswordFormat` values indicating the format for storing passwords in the data store.
Examples

The following example shows the membership element being used in an applications web.config file.

C#

```csharp
<membership defaultProvider="MySQLMembershipProvider">
  <providers>
    <add name="MySqlMembershipProvider"
         connectionStringName="LocalMySqlServer"
         enablePasswordRetrieval="true"
         enablePasswordReset="false"
         requiresQuestionAndAnswer="true"
         requiresUniqueEmail="false"
         passwordFormat="Encrypted"
         maxInvalidPasswordAttempts="3"
         passwordAttemptWindow="20"
         minRequiredNonAlphanumericCharacters="1"
         minRequiredPasswordLength="11"
         applicationName="MyApplication" />
  </providers>
</membership>
```
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets the regular expression used to evaluate a password.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  [MySQL.Web](#) (in [MySQL.Web.dll](#)) Version: 6.2.2.0
Syntax

**C#**

```csharp
public override string PasswordStrengthRegularExpression { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides ReadOnly Property PasswordStrengthRegularExpression
```

**Visual C++**

```cpp
public:
virtual property String^ PasswordStrengthRegularExpression { 
    String^ get () override;
}
```

**Field Value**

A regular expression used to evaluate a password.
Examples

The following example shows the membership element being used in an applications web.config file. In this example, the regular expression specifies that the password must meet the following criteria:

Is at least seven characters. Contains at least one digit. Contains at least one special (non-alphanumeric) character.

C#

```csharp
<membership defaultProvider="MySQLMembershipProvider">
    <providers>
        <add name="MySqlMembershipProvider" type="MySql.Web.Security.MySQLMembershipProvider" connectionStringName="LocalMySqlServer" enablePasswordRetrieval="true" enablePasswordReset="false" requiresQuestionAndAnswer="true" requiresUniqueEmail="false" passwordFormat="Encrypted" maxInvalidPasswordAttempts="3" passwordAttemptWindow="20" minRequiredNonAlphanumericCharacters="1" minRequiredPasswordLength="11" applicationName="MyApplication" />
    </providers>
</membership>
```
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a value indicating whether the membership provider is configured to require the user to answer a password question for password reset and retrieval.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  [MySQL.Web](in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public override bool RequiresQuestionAndAnswer { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property RequiresQuestionAndAnswer As Bool]

Visual C++

public:
virtual property bool RequiresQuestionAndAnswer {
    bool get () override;
}

Field Value

ture if a password answer is required for password reset and retrieval; otherwise, false. The default is false.
Examples

The following example shows the membership element being used in an applications web.config file.

C#<membership defaultProvider="MySQLMembershipProvider">
  <providers>
    <add name="MySqlMembershipProvider"
connectionStringName="LocalMySqlServer"
enablePasswordRetrieval="true"
enablePasswordReset="false"
requiresQuestionAndAnswer="true"
requiresUniqueEmail="false"
passwordFormat="Encrypted"
maxInvalidPasswordAttempts="3"
passwordAttemptWindow="20"
minRequiredNonAlphanumericCharacters="1"
minRequiredPasswordLength="11"
applicationName="MyApplication" />
  </providers>
</membership>
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLMembershipProvider...:: RequiresUniqueEmail Property

MySQLMembershipProvider Class  Example  See Also  Send Feedback

Gets a value indicating whether the membership provider is configured to require a unique e-mail address for each user name.

Syntax

C#

public override bool RequiresUniqueEmail { get; }

Visual Basic (Declaration)

Public Overrides ReadOnly Property RequiresUniqueEmail As Boolean

Visual C++

public:
virtual property bool RequiresUniqueEmail {
    bool get () override;
}

Field Value

true if the membership provider requires a unique e-mail address; otherwise, false. The default is true.
Examples

The following example shows the membership element being used in an applications web.config file.

```
<membership defaultProvider="MySQLMembershipProvider">
  <providers>
    <add name="MySqlMembershipProvider"
         connectionStringName="LocalMySqlServer"
         enablePasswordRetrieval="true"
         enablePasswordReset="false"
         requiresQuestionAndAnswer="true"
         requiresUniqueEmail="false"
         passwordFormat="Encrypted"
         maxInvalidPasswordAttempts="3"
         passwordAttemptWindow="20"
         minRequiredNonAlphanumericCharacters="1"
         minRequiredPasswordLength="11"
         applicationName="MyApplication" />
  </providers>
</membership>
```
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySQLMembershipProvider...:::WriteExceptionsToEventLog Property
MySQLMembershipProvider Class  See Also  Send Feedback

Gets or sets a value indicating whether exceptions are written to the event log.

**Namespace:**  MySqlCommand.Web.Security
**Assembly:**  MySqlConnection (in MySqlConnection.dll) Version: 6.2.2.0
Syntax

C#

public bool WriteExceptionsToEventLog { get; set; }

Visual Basic (Declaration)

Public Property WriteExceptionsToEventLog As Boolean

Visual C++

public:
property bool WriteExceptionsToEventLog {
    bool get ();
    void set (bool value);
}

Field Value

true if exceptions should be written to the log; otherwise, false.
See Also

MySQLMembershipProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLMembershipProvider` type exposes the following members.
## Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ValidatingPassword</strong></td>
<td>Occurs when a user is created, a password is changed, or a password is reset. (Inherited from MembershipProvider.)</td>
</tr>
</tbody>
</table>
See Also

MySQLMembershipProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLRoleProvider Class

Manages storage of role membership information for an ASP.NET application in a MySQL database.

**Namespace:**  [MySQL.Web.Security](#)  
**Assembly:**  MySQL.Web (in MySQL.Web.dll)  
**Version:**  6.2.2.0
Syntax

C#

public sealed class MySQLRoleProvider : RoleProvider

Visual Basic (Declaration)

Public NotInheritable Class MySQLRoleProvider
    Inherits RoleProvider

Visual C++

public ref class MySQLRoleProvider sealed : public RoleProvider
Inheritance Hierarchy

System:::Object
System.Configuration.Provider:::ProviderBase
System.Web.Security:::RoleProvider
See Also

MySQLRoleProvider Members
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLRoleProvider** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQLRoleProvider</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddUsersToRoles</td>
<td>Adds the users to roles. (Overrides <code>RoleProvider.AddUsersToRoles([[], array&lt;String&gt;[]]).</code>)</td>
</tr>
<tr>
<td>CreateRole</td>
<td>Creates the role. (Overrides <code>RoleProvider.CreateRole(String).</code>)</td>
</tr>
<tr>
<td>DeleteRole</td>
<td>Deletes the role. (Overrides <code>RoleProvider.DeleteRole(String, Boolean)</code>)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>FindUsersInRole</td>
<td>Finds the users in role. (Overrides <code>RoleProvider.FindUsersInRole(String)</code>. )</td>
</tr>
<tr>
<td>GetAllRoles</td>
<td>Gets a list of all the roles for the configured applicationName. (Overrides <code>RoleProvider.GetAllRoles()</code>. )</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetRolesForUser</td>
<td>Gets a list of the roles that a specified user is in for the configured applicationName. (Overrides <code>RoleProvider.GetRolesForUser(String)</code>. )</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetUsersInRole</td>
<td>Gets the users in role. (Overrides <code>RoleProvider.GetUsersInRole(String)</code>. )</td>
</tr>
<tr>
<td>Initialize</td>
<td>Initializes the provider. (Overrides <code>ProviderBase.Initialize(String, NameValueCollection).</code>)</td>
</tr>
<tr>
<td></td>
<td>Determines whether [is user in role] [the specified user]</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsUserInRole</td>
<td>(Overrides RoleProvider..:::IsUserInRole(String, String))</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>RemoveUsersFromRoles</td>
<td>Removes the users from roles. (Overrides RoleProvider..:::RemoveUsersFromRoles(array&lt;String&gt;[], [])).</td>
</tr>
<tr>
<td>RoleExists</td>
<td>Roles the exists. (Overrides RoleProvider..:::RoleExists(String).)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationName</td>
<td>Gets or sets the name of the application to store and retrieve role information for. (Overrides RoleProvider..:::ApplicationName.)</td>
</tr>
<tr>
<td>Description</td>
<td>Gets a brief, friendly description suitable for display in administrative tools or other user interfaces (UIs). (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the friendly name used to refer to the provider during configuration. (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>WriteExceptionsToEventLog</td>
<td>Gets or sets a value indicating whether [write exceptions to event log].</td>
</tr>
</tbody>
</table>
See Also

MySQLRoleProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Syntax

C#

public MySQLRoleProvider()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MySQLRoleProvider()
See Also

MySQLRoleProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySQLRoleProvider` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddUsersToRoles</td>
<td>Adds the users to roles. ( Overrides <code>RoleProvider...AddUsersToRoles(array&lt;[][], array&lt;String&gt;[][])</code>. )</td>
</tr>
<tr>
<td>CreateRole</td>
<td>Creates the role. ( Overrides <code>RoleProvider...CreateRole(String)</code>. )</td>
</tr>
<tr>
<td>DeleteRole</td>
<td>Deletes the role. ( Overrides <code>RoleProvider...DeleteRole(String, Boolean)</code>. )</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. ( Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> is reclaimed by garbage collection. ( Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>FindUsersInRole</td>
<td>Finds the users in role. ( Overrides <code>RoleProvider...FindUsersInRole(String)</code>. )</td>
</tr>
<tr>
<td>GetAllRoles</td>
<td>Gets a list of all the roles for the configured applicationName. ( Overrides <code>RoleProvider...GetAllRoles()()</code>. )</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. ( Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetRolesForUser</td>
<td>Gets a list of the roles that a specified user is in for the configured applicationName. ( Overrides <code>RoleProvider...GetRolesForUser(String)</code>. )</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. ( Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetUsersInRole</td>
<td>Gets the users in role. ( Overrides <code>RoleProvider...GetUsersInRole(String)</code>. )</td>
</tr>
<tr>
<td>Initialize</td>
<td>Initializes the provider. ( Overrides <code>ProviderBase...Initialize(String, NameValueCollection)</code>.)</td>
</tr>
<tr>
<td></td>
<td>Determines whether [is user in role] [the specified user].------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>IsUserInRole</strong></td>
<td>(Overrides RoleProvider...::IsUserInRole(String, String)) Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>(Inherited from Object.) Removes the users from roles. (Overrides RoleProvider...::RemoveUsersFromRoles(array&lt;String&gt;[], array&lt;String&gt;[])[])</td>
</tr>
<tr>
<td><strong>RemoveUsersFromRoles</strong></td>
<td>(Overrides RoleProvider...::RemoveUsersFromRoles(array&lt;String&gt;[], array&lt;String&gt;[])[]) Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>RoleExists</strong></td>
<td>Roles the exists. (Overrides RoleProvider...::RoleExists(String).)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLRoleProvider..::.AddUsersToRoles Method
MySQLRoleProvider Class  See Also  Send Feedback

Adds the users to roles.

Syntax

C#

public override void AddUsersToRoles(
    string[] usernames,
    string[] rolenames
)

Visual Basic (Declaration)

Public Overrides Sub AddUsersToRoles ( _
    usernames As String(), _
    rolenames As String() _
)

Visual C++

public:
    virtual void AddUsersToRoles(
        array<String^>^ usernames,
        array<String^>^ rolenames
    ) override

Parameters

usernames
    Type: array< System::::String >[](0]
    The usernames.

rolenames
    Type: array< System::::String >[](0]
    The rolenames.
See Also

MySQLRoleProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Creates the role.

**Namespace:** MySql.Web.Security

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override void CreateRole(
    string rolename
)

Visual Basic (Declaration)

Public Overrides Sub CreateRole ( _
    rolename As String _
)

Visual C++

public:
virtual void CreateRole(
    String^ rolename
) override

Parameters

rolename
    Type: System::String
    The rolename.
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
DeleteRole Method

MySQLRoleProvider Class  See Also  Send Feedback

Deletes the role.

Syntax

C#

```csharp
public override bool DeleteRole(
    string rolename,
    bool throwOnPopulatedRole
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function DeleteRole ( _
    rolename As String, _
    throwOnPopulatedRole As Boolean _
) As Boolean
```

Visual C++

```cpp
public:
virtual bool DeleteRole(
    String^ rolename,
    bool throwOnPopulatedRole
) override
```

Parameters

rolename
Type: `System::::String`
The rolename.

throwOnPopulatedRole
Type: `System::::Boolean`
if set to true [throw on populated role].

Return Value

true if the role was successfully deleted; otherwise, false.
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Finds the users in role.

**Namespace:**  [ MySql.Web.Security](#)  
**Assembly:**  [Mysql.Web](in Mysql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override string[] FindUsersInRole(
    string rolename,
    string usernameToMatch
)

Visual Basic (Declaration)

Public Overrides Function FindUsersInRole ( _
    rolename As String, _
    usernameToMatch As String _
) As String()

Visual C++

public:
virtual array<String^>^ FindUsersInRole(
    String^ rolename,
    String^ usernameToMatch
) override

Parameters

rolename
    Type: System::String
    The rolename.

usernameToMatch
    Type: System::String
    The username to match.

Return Value

A string array containing the names of all the users where the user name matches usernameToMatch and the user is a member of the specified role.
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a list of all the roles for the configured applicationName.

**Namespace:**  [MySQL.Web.Security](#)  
**Assembly:**  MySQL.Web (in MySQL.Web.dll)  
**Version:**  6.2.2.0
## Syntax

### C#

```csharp
public override string[] GetAllRoles()
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function GetAllRoles As String()
```

### Visual C++

```cpp
public:
virtual array<String^>^ GetAllRoles() override
```

## Return Value

A string array containing the names of all the roles stored in the data source for the configured applicationName.
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets a list of the roles that a specified user is in for the configured applicationName.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  [MySQL.Web](#) (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

public override string[] GetRolesForUser(
    string username
)

Visual Basic (Declaration)

Public Overrides Function GetRolesForUser ( _
    username As String _
) As String()

Visual C++

public:
virtual array<string^>^ GetRolesForUser(
    string^ username
) override

Parameters

username

Type: System::String
The user to return a list of roles for.

Return Value

A string array containing the names of all the roles that the specified user is in
for the configured applicationName.
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Role Provider

**GetUsersInRole Method**

MySQLRoleProvider Class  See Also  Send Feedback

Gets the users in role.

**Namespace:**  MySql.Web.Security

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override string[] GetUsersInRole(
    string rolename
)

Visual Basic (Declaration)

Public Overrides Function GetUsersInRole ( _
    rolename As String _
) As String()

Visual C++

public:
virtual array<String^>^ GetUsersInRole(
    String^ rolename
) override

Parameters

rolename
    Type: System::String
    The rolename.

Return Value

A string array containing the names of all the users who are members of the specified role.
See Also

MySQLRoleProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQLRoleProvider Class  See Also  Send Feedback

Initializes the provider.

Syntax

C#

public override void Initialize(
    string name,
    NameValueCollection config
)

Visual Basic (Declaration)

Public Overrides Sub Initialize ( _
    name As String, _
    config As NameValueCollection _
)

Visual C++

public:
virtual void Initialize(
    String^ name,
    NameValueCollection^ config
) override

Parameters

name
Type: System::String
The friendly name of the provider.

config
Type: System.Collections.Specialized::NameValueCollection
A collection of the name/value pairs representing the provider-specific attributes specified in the configuration for this provider.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System..::.ArgumentNullException</strong></td>
<td>The name of the provider is null.</td>
</tr>
<tr>
<td><strong>System..::.ArgumentOutOfRangeException</strong></td>
<td>The name of the provider has a length of zero.</td>
</tr>
<tr>
<td><strong>System..::.InvalidOperationException</strong></td>
<td>An attempt is made to call Initialize(String, NameValueCollection) on a provider after the provider has already been initialized.</td>
</tr>
</tbody>
</table>
See Also

MySQLRoleProvider Class
MySQL.Web.Security Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Determines whether [is user in role] [the specified username].

**Namespace:** [MySQL.Web.Security](#)  
**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override bool IsUserInRole(
    string username,
    string rolename
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Function IsUserInRole ( _
    username As String, _
    rolename As String _
) As Boolean
```

### Visual C++

```cpp
public:
virtual bool IsUserInRole(
    String^ username,
    String^ rolename
) override
```

## Parameters

- **username**
  - Type: `System::String`
  - The username.

- **rolename**
  - Type: `System::String`
  - The rolename.

## Return Value

true if [is user in role] [the specified username]; otherwise, false.
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Removes the users from roles.

**Namespace:**  [MySQL.Web.Security](#)  
**Assembly:**  [MySql.Web](#) (in MySql.Web.dll)  
**Version:** 6.2.2.0
### Syntax

#### C#

public override void RemoveUsersFromRoles(
    string[] usernames,
    string[] rolenames
)

#### Visual Basic (Declaration)

Public Overrides Sub RemoveUsersFromRoles ( _
    usernames As String(), _
    rolenames As String() _
)

#### Visual C++

public:
    virtual void RemoveUsersFromRoles(
        array<String>^ usernames,
        array<String>^ rolenames
    ) override

### Parameters

**usernames**
- Type: array< System::::String >[](0)
- The usernames.

**rolenames**
- Type: array< System::::String >[](0)
- The rolenames.
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Roles the exists.

**Namespace:**  [MySQL.Web.Security](https://www.mysql.com)

**Assembly:**  [MySQL.Web](https://www.mysql.com) (in [MySQL.Web.dll](https://www.mysql.com)) Version: 6.2.2.0
Syntax

C#

```csharp
public override bool RoleExists(string rolename)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function RoleExists ( _
    rolename As String _
) As Boolean
```

Visual C++

```cpp
public:
    virtual bool RoleExists(
        String^ rolename
    ) override
```

Parameters

rolename
  Type: System::String
  The rolename.

Return Value

true if the role name already exists in the database; otherwise, false.
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The **MySQLRoleProvider** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationName</td>
<td>Gets or sets the name of the application to store and retrieve role information for. (Overrides RoleProvider..::..ApplicationName.)</td>
</tr>
<tr>
<td>Description</td>
<td>Gets a brief, friendly description suitable for display in administrative tools or other user interfaces (UIs). (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the friendly name used to refer to the provider during configuration. (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>WriteExceptionsToEventLog</td>
<td>Gets or sets a value indicating whether [write exceptions to event log].</td>
</tr>
</tbody>
</table>
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Gets or sets the name of the application to store and retrieve role information for.

**Namespace:**  [MySQL.Web.Security](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override string ApplicationName { get; set; }

Visual Basic (Declaration)

Public Overrides Property ApplicationName As String

Visual C++

public:
    virtual property String^ ApplicationName {
        String^ get () override;
        void set (String^ value) override;
    }

Field Value

The name of the application to store and retrieve role information for.
Examples

C#

```csharp
<RoleManager defaultProvider="MySqlProvider" enabled="true">
    <providers>
        <add
            name="MySqlProvider"
            connectionStringName="LocalMySQLServices"
            writeExceptionsToEventLog="false"
            applicationName="MyApplication" />
    </providers>
</RoleManager>
```
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Role Provider WriteExceptionsToEventLog Property

**Namespace:**  [MySQL.Web.Security](#)  
**Assembly:**  [MySql.Web](#)  (in MySql.Web.dll)  
**Version:**  6.2.2.0

Gets or sets a value indicating whether [write exceptions to event log].
Syntax

C#

public bool WriteExceptionsToEventLog { get; set; }

Visual Basic (Declaration)

Public Property WriteExceptionsToEventLog As Boolean

Visual C++

public:
property bool WriteExceptionsToEventLog {

    bool get ();
    void set (bool value);
}

Field Value

true if exceptions should be written to the event log; otherwise, false.
<roleManager defaultProvider="MySqlProvider"
  enabled="true">
  <providers>
    <add
      name="MySqlProvider"
      connectionStringName="LocalMySqlServices"
      writeExceptionsToEventLog="false"
      applicationName="MyApplication" />
  </providers>
</roleManager>
See Also

MySQLRoleProvider Class

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Send Feedback
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlSessionStateStore</td>
<td>This class allows ASP.NET applications to store and manage session state information in a MySQL database. Expired session data is periodically deleted from the database.</td>
</tr>
</tbody>
</table>

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySQLSessionStateStore Class

This class allows ASP.NET applications to store and manage session state information in a MySQL database. Expired session data is periodically deleted from the database.

Namespace: MySql.Web.SessionState
Syntax

C#

public class MySqlSessionStateStore : SessionStateStoreProviderBase

Visual Basic (Declaration)

Public Class MySqlSessionStateStore
    Inherits SessionStateStoreProviderBase

Visual C++

public ref class MySqlSessionStateStore : public SessionStateStoreProviderBase
Inheritance Hierarchy

System...:::Object
   System.Configuration.Provider...:::ProviderBase
      System.Web.SessionState...:::SessionStateStoreProviderBase
         MySql.Web.SessionState...:::MySqlSessionStateStore
See Also

**MySqlSessionStateStore Members**
**MySql.Web.SessionState Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlSessionStateStore` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySqlSessionStateStore</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CreateNewStoreData</strong></td>
<td>This method creates a new SessionStateStoreData object for the current request. (Overides <code>SessionStateStoreProviderBase::.CreateNewStoreData(HttpContext, Int32)</code>).</td>
</tr>
<tr>
<td><strong>CreateUninitializedItem</strong></td>
<td>This method adds a new session state item to the database. (Overides <code>SessionStateStoreProviderBase::.CreateUninitializedItem(HttpContext, String, Int32)</code>).</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>This method releases all the resources for this instance. (Overides <code>SessionStateStoreProviderBase::.Dispose(HttpContext)</code>).</td>
</tr>
<tr>
<td><strong>EndRequest</strong></td>
<td>This method allows the MySqlSessionStateStore provider to perform any cleanup that may be required for the current request. (Overides <code>SessionStateStoreProviderBase::.EndRequest(HttpContext)</code>).</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an <code>Object</code> to attempt to free resources and the <code>Object</code> is reclaimed by garbage collection. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>GetItem</strong></td>
<td>This method returns a read-only session item from the database. (Overides <code>SessionStateStoreProviderBase::.GetItem(HttpContext, String, Boolean%, TimeSpan%, Object%, SessionStateActions%)</code>).</td>
</tr>
<tr>
<td><strong>GetItemExclusive</strong></td>
<td>This method locks a session item and returns it to the caller. (Overides <code>SessionStateStoreProviderBase::.GetItemExclusive(HttpContext, String, Boolean%, TimeSpan%, Object%, SessionStateActions%)</code>).</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>Initialize</strong></td>
<td>Initializes the provider with the property values specified in the ASP.NET application configuration file. (Overides <code>ProviderBase::.Initialize(String, NameValueCollection)</code>).</td>
</tr>
<tr>
<td><strong>InitializeRequest</strong></td>
<td>This method performs any per-request initialization required by the provider. (Overides <code>SessionStateStoreProviderBase::.InitializeRequest(HttpContext)</code>).</td>
</tr>
</tbody>
</table>
**MemberwiseClone**

Creates a shallow copy of the current `Object`.
(Inherited from `Object`.)

This method forcibly releases the lock on a session item in the database if multiple attempts to retrieve the session item fail.
(Overides `SessionStateStoreProviderBase`.)

**ReleaseItemExclusive**

This method forcibly releases the lock on a session item in the database, if multiple attempts to retrieve the session item fail.
(Overides `SessionStateStoreProviderBase`.)

**RemoveItem**

This method removes the specified session item from the database.
(Overides `SessionStateStoreProviderBase`.)

**ResetItemTimeout**

This method resets the expiration date and timeout for a session item in the database.
(Overides `SessionStateStoreProviderBase`.)

**SetAndReleaseItemExclusive**

This method updates the session time information of a session item, and releases the lock.
(Overides `SessionStateStoreProviderBase`.)

**SetItemExpireCallback**

This method returns a false value to indicate that callbacks for expired sessions are not supported.
(Overides `SessionStateStoreProviderBase`.)

**ToString**

Returns a `String` that represents the current `Object`.
(Inherited from `Object`.)
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationName</td>
<td>Gets a brief, friendly description suitable for display in administrative tools or other user interfaces (UIs). (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>Description</td>
<td>Gets the friendly name used to refer to the provider during configuration. (Inherited from ProviderBase.)</td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>WriteExceptionsToEventLog</td>
<td>Indicates whether to write exceptions to event log</td>
</tr>
</tbody>
</table>
See Also

MySQLSessionStateStore Class
MySQL.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
C#  Visual Basic
Visual C++
MySQL Connector/Net

**MySQLSessionStateStore Constructor**

[MySQLSessionStateStore Class](#)  See Also  Send Feedback

**Namespace:**  [MySQL.Web.SessionState](#)
**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#
public MySqlSessionStateStore() 

Visual Basic (Declaration)
Public Sub New

Visual C++
public:
MySqlSessionStateStore();
See Also

MySQLSessionStateStore Class
MySQL.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlSessionStateStore` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CreateNewStoreData</strong></td>
<td>This method creates a new SessionStateStoreData object. (Overrides SessionStateStoreProviderBase.CreateNewStoreData(HttpContext, Int32))</td>
</tr>
<tr>
<td><strong>CreateUninitializedItem</strong></td>
<td>This method adds a new session state item to the database. (Overrides SessionStateStoreProviderBase.CreateUninitializedItem(HttpContext, String, Int32).)</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>This method releases all the resources for this instance. (Overrides SessionStateStoreProviderBase.Dispose(HttpContext))</td>
</tr>
<tr>
<td><strong>EndRequest</strong></td>
<td>This method allows the MySqlSessionStateStore instance to perform any cleanup required for the current request. (Overrides SessionStateStoreProviderBase.EndRequest(HttpContext))</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetItem</strong></td>
<td>This method returns a read-only session item from the database. (Overrides SessionStateStoreProviderBase.GetItem(HttpContext, String, Boolean, TimeSpan, Object, SessionStateActions).)</td>
</tr>
<tr>
<td><strong>GetItemExclusive</strong></td>
<td>This method locks a session item and returns it. (Overrides SessionStateStoreProviderBase.GetItemExclusive(HttpContext, String, Boolean, TimeSpan, Object, SessionStateActions).)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Initialize</strong></td>
<td>Initializes the provider with the property values specified in the ASP.NET configuration file. (Overrides ProviderBase.Initialize(String, NameValueCollection).)</td>
</tr>
<tr>
<td><strong>InitializeRequest</strong></td>
<td>This method performs any per-request initialization the provider requires. (Overrides SessionStateStoreProviderBase.InitializeRequest(HttpContext).)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td><strong>ReleaseItemExclusive</strong></td>
<td>This method forcibly releases the lock on a session to retrieve the session item fail.</td>
</tr>
<tr>
<td><strong>RemoveItem</strong></td>
<td>This method removes the specified session item</td>
</tr>
<tr>
<td><strong>ResetItemTimeout</strong></td>
<td>This method resets the expiration date and timeout for a session item in the database.</td>
</tr>
<tr>
<td><strong>SetAndReleaseItemExclusive</strong></td>
<td>This method updates the session time information in the database with the specified session item, and releases the lock.</td>
</tr>
<tr>
<td><strong>SetItemExpireCallback</strong></td>
<td>This method returns a false value to indicate that callbacks for expired sessions are not supported.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object.</td>
</tr>
</tbody>
</table>
See Also

`MySqlSessionStateStore Class`
`MySql.Web.SessionState Namespace`

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method creates a new SessionStateStoreData object for the current request.

**Namespace:**  [MySQL.Web.SessionState](#)

**Assembly:**  [MySQL.Web](#) (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override SessionStateStoreData CreateNewStoreData(
    HttpContext context,
    int timeout
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Function CreateNewStoreData ( _
    context As HttpContext, _
    timeout As Integer _
) As SessionStateStoreData
```

Visual C++

```cpp
public:
virtual SessionStateStoreData^ CreateNewStoreData(  
    HttpContext^ context,
    int timeout
) override
```

Parameters

context
  Type: `System.Web:::HttpContext`
  The HttpContext object for the current request.

timeout
  Type: `System:::Int32`
  The timeout value (in minutes) for the SessionStateStoreData object that is created.
See Also

MySqlSessionStateStore Class
MySql.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net

MySqlConnection

See Also

Send Feedback

This method adds a new session state item to the database.

Namespace:  MySql.Web.SessionState
Syntax

C#

```csharp
public override void CreateUninitializedItem(
    HttpContext context,
    string id,
    int timeout
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub CreateUninitializedItem ( _
    context As HttpContext, _
    id As String, _
    timeout As Integer _
)
```

Visual C++

```cpp
public:
virtual void CreateUninitializedItem(
    HttpContext^ context,
    String^ id,
    int timeout
) override
```

Parameters

context
Type: `System.Web::System.Web::HttpContext`
The HttpContext object for the current request.

id
Type: `System::System::String`
The session ID for the current request.

timeout
Type: `System::System::Int32`
The timeout value for the current request.
See Also

**MySqlSessionStateStore Class**  
**MySql.Web.SessionState Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method releases all the resources for this instance.

**Namespace:**  [MySQL.Web.SessionState](#)
**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override void Dispose()

Visual Basic (Declaration)

Public Overrides Sub Dispose

Visual C++

public:
virtual void Dispose() override
See Also

MySqlSessionStateStore Class
MySql.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method allows the MySqlSessionStateStore object to perform any cleanup that may be required for the current request.

**Namespace:**  [MySql.Web.SessionState](#)

**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
## Syntax

**C#**

```csharp
public override void EndRequest(
    HttpContext context
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub EndRequest (_,
    context As HttpContext _
)
```

**Visual C++**

```cpp
public:
  virtual void EndRequest(
    HttpContext^ context
) override
```

## Parameters

**context**

- **Type:** System.Web::::HttpContext
- The HttpContext object for the current request
See Also

MySQLSessionStateStore Class
MySQL.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method returns a read-only session item from the database.

**Namespace:** MySql.Web.SessionState  
**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override SessionStateStoreData GetItem(
    HttpContext context,
    string id,
    out bool locked,
    out TimeSpan lockAge,
    out Object lockId,
    out SessionStateActions actions
)

Visual Basic (Declaration)

Public Overrides Function GetItem ( _
    context As HttpContext, _
    id As String, _
    <OutAttribute> ByRef locked As Boolean, _
    <OutAttribute> ByRef lockAge As TimeSpan, _
    <OutAttribute> ByRef lockId As Object, _
    <OutAttribute> ByRef actions As SessionStateActions _
) As SessionStateStoreData

Visual C++

public:
    virtual SessionStateStoreData^ GetItem(
        HttpContext^ context,
        String^ id,
        [OutAttribute] bool% locked,
        [OutAttribute] TimeSpan% lockAge,
        [OutAttribute] Object% lockId,
        [OutAttribute] SessionStateActions% actions
    ) override

Parameters

c context
    Type: System.Web.HttpContext

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>System..:::String</td>
</tr>
<tr>
<td>locked</td>
<td>System..:::Boolean</td>
</tr>
<tr>
<td>lockAge</td>
<td>System..:::TimeSpan</td>
</tr>
<tr>
<td>lockId</td>
<td>System..:::Object</td>
</tr>
<tr>
<td>actions</td>
<td>System.Web.SessionState..:::SessionStateActions</td>
</tr>
</tbody>
</table>
See Also

MySqlSessionStateStore Class
MySql.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method locks a session item and returns it from the database

**Namespace:** [MySQL.Web.SessionState](https://www.mysql.com)

**Assembly:** [MySQL.Web](https://www.mysql.com) (in [MySQL.Web.dll](https://www.mysql.com)) Version: 6.2.2.0
### Syntax

**C#**

```csharp
public override SessionStateStoreData GetItemExclusive(
    HttpContext context,
    string id,
    out bool locked,
    out TimeSpan lockAge,
    out Object lockId,
    out SessionStateActions actions)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function GetItemExclusive ( _
    context As HttpContext, _
    id As String, _
    <OutAttribute> ByRef locked As Boolean, _
    <OutAttribute> ByRef lockAge As TimeSpan, _
    <OutAttribute> ByRef lockId As Object, _
    <OutAttribute> ByRef actions As SessionStateActions _
) As SessionStateStoreData
```

**Visual C++**

```cpp
public:
    virtual SessionStateStoreData^ GetItemExclusive( _
    HttpContext^ context,
    String^ id,
    [OutAttribute] bool% locked,
    [OutAttribute] TimeSpan% lockAge,
    [OutAttribute] Object^% lockId,
    [OutAttribute] SessionStateActions% actions
) override
```

### Parameters

**context**

Type: `System.Web::HttpContext`

The HttpContext object for the current request
id
Type: System..:::String
The session ID for the current request

locked
Type: System..:::Boolean %
true if the session item is locked in the database; otherwise, it is false.

lockAge
Type: System..:::TimeSpan %
TimeSpan object that indicates the amount of time the session item has been locked in the database.

lockId
Type: System..:::Object %
A lock identifier object.

actions
Type: System.Web.SessionState..:::SessionStateActions %
A SessionStateActions enumeration value that indicates whether or not the session is uninitialized and cookieless.

**Return Value**
See Also

**MySQLSessionStateStore Class**  
**MySQL.Web.SessionState Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySQL Connector/Net
MySQLSessionStateStore..::..Initialize Method
MySQLSessionStateStore Class  See Also  Send Feedback

Initializes the provider with the property values specified in the ASP.NET application configuration file

Namespace:  MySql.Web.SessionState
## Syntax

### C#

```csharp
public override void Initialize(
    string name,
    NameValueCollection config
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Sub Initialize (_
    name As String, _
    config As NameValueCollection _
)
```

### Visual C++

```cpp
public:
virtual void Initialize(
    String^ name,
    NameValueCollection^ config
) override
```

## Parameters

**name**

Type: `System::::String`

The name of the provider instance to initialize.

**config**

Type: `System.Collections.Specialized::::NameValueCollection`

Object that contains the names and values of configuration options for the provider.
See Also

MySQLSessionStateStore Class
MySql.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method performs any per-request initializations that the MySqlSessionStateStore provider requires.

**Namespace:**  MySQL.Web.SessionState
**Assembly:**  MySQL.Web (in MySQL.Web.dll) Version: 6.2.2.0
Syntax

C#

```csharp
public override void InitializeRequest(
    HttpContext context
)
```

Visual Basic (Declaration)

```vbnet
Public Overrides Sub InitializeRequest ( _
    context As HttpContext _
)
```

Visual C++

```cpp
public:
virtual void InitializeRequest(
    HttpContext^ context
) override
```

Parameters

context
  Type: System.Web::::HttpContext
See Also

**MySqlSessionStateStore Class**  
**MySql.Web.SessionState Namespace**

Send comments on this topic to [support@mysql.com](mailto:support@mysql.com)

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method forcibly releases the lock on a session item in the database, if multiple attempts to retrieve the session item fail.

**Namespace:**  MySql.Web.SessionState  
**Assembly:**  MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
**Syntax**

C#

```csharp
public override void ReleaseItemExclusive(
    HttpContext context,
    string id,
    Object lockId
)
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Sub ReleaseItemExclusive ( _
    context As HttpContext, _
    id As String, _
    lockId As Object _
)
```

**Visual C++**

```c++
public:
    virtual void ReleaseItemExclusive(
        HttpContext^ context,
        String^ id,
        Object^ lockId
    ) override
```

**Parameters**

**context**

Type: `System.Web::HttpContext`

The HttpContext object for the current request.

**id**

Type: `System::String`

The session ID for the current request.

**lockId**

Type: `System::Object`
The lock identifier for the current request.
See Also

MySQLSessionStateStore Class
MySQL.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method removes the specified session item from the database.

**Namespace:** MySql.Web.SessionState

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override void RemoveItem(
    HttpContext context,
    string id,
    Object lockId,
    SessionStateStoreData item
)

Visual Basic (Declaration)

Public Overrides Sub RemoveItem ( _
    context As HttpContext, _
    id As String, _
    lockId As Object, _
    item As SessionStateStoreData _
)

Visual C++

public:
virtual void RemoveItem(
    HttpContext^ context,
    String^ id,
    Object^ lockId,
    SessionStateStoreData^ item
) override

Parameters

context
Type: System.Web:::HttpContext
The HttpContext object for the current request

id
Type: System:::String
The session ID for the current request
lockId
Type: **System::Object**
The lock identifier for the current request.

item
Type: **System.Web.SessionState::SessionStateStoreData**
The session item to remove from the database.
See Also

MySQLSessionStateStore Class
MySQL.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method resets the expiration date and timeout for a session item in the database.

**Namespace:**  [MySQL.Web.SessionState](#)

**Assembly:**  [MySQL.Web (in MySQL.Web.dll)](#) Version: 6.2.2.0
## Syntax

### C#

```csharp
public override void ResetItemTimeout(
    HttpContext context,
    string id
)
```

### Visual Basic (Declaration)

```vbnet
Public Overrides Sub ResetItemTimeout ( _
    context As HttpContext, _
    id As String _
)
```

### Visual C++

```cpp
public:
virtual void ResetItemTimeout(
    HttpContext^ context,
    String^ id
) override
```

## Parameters

**context**

- **Type:** `System.Web::::HttpContext`
- The HttpContext object for the current request

**id**

- **Type:** `System::::String`
- The session ID for the current request
See Also

MySqlSessionStateStore Class
MySql.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method updates the session time information in the database with the specified session item, and releases the lock.

**Namespace:** MySql.Web.SessionState

**Assembly:** MySql.Web (in MySql.Web.dll) Version: 6.2.2.0
Syntax

C#

public override void SetAndReleaseItemExclusive(
    HttpContext context,
    string id,
    SessionStateStoreData item,
    Object lockId,
    bool newItem
)

Visual Basic (Declaration)

Public Overrides Sub SetAndReleaseItemExclusive (_context As HttpContext, _
id As String, _
item As SessionStateStoreData, _
lockId As Object, _
newItem As Boolean _
)

Visual C++

public:
virtual void SetAndReleaseItemExclusive(
    HttpContext^ context,
    String^ id,
    SessionStateStoreData^ item,
    Object^ lockId,
    bool newItem
) override

Parameters

context
Type: System.Web::HttpContext
The HttpContext object for the current request

id
Type: `System.String`
The session ID for the current request

item
Type: `System.Web.SessionState.SessionStateStoreData`
The session item containing new values to update the session item in the database with.

lockId
Type: `System.Object`
The lock identifier for the current request.

newItem
Type: `System.Boolean`
A Boolean value that indicates whether or not the session item is new in the database. A false value indicates an existing item.
See Also

MySQLSessionStateStore Class
MySQL.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
This method returns a false value to indicate that callbacks for expired sessions are not supported.

Namespace:  MySql.Web.SessionState
Syntax

**C#**

public override bool SetItemExpireCallback(  
    SessionStateItemExpireCallback expireCallback  
)

**Visual Basic (Declaration)**

Public Overrides Function SetItemExpireCallback ( _  
    expireCallback As SessionStateItemExpireCallback _  
) As Boolean

**Visual C++**

public:  
    virtual bool SetItemExpireCallback(  
    SessionStateItemExpireCallback^ expireCallback  
) override

**Parameters**

expireCallback
    Type: System.Web.SessionState..::.SessionStateItemExpireCallback

**Return Value**

false
See Also

MySQLSessionStateStore Class
MySql.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
The `MySqlSessionStateStore` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationName</td>
<td>Gets a brief, friendly description suitable for display in administrative tools or other user interfaces (UIs). (Inherited from <a href="#">ProviderBase</a>.)</td>
</tr>
<tr>
<td>Description</td>
<td>Gets the friendly name used to refer to the provider during configuration. (Inherited from <a href="#">ProviderBase</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Indicates whether to write exceptions to event log</td>
</tr>
</tbody>
</table>
See Also

**MySqlSessionStateStore Class**

**MySql.Web.SessionState Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
Namespace: MySql.Web.SessionState
## Syntax

### C#

```csharp
public string ApplicationName { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property ApplicationName As String
```

### Visual C++

```cpp
public:
property String^ ApplicationName {
    String^ get ();
    void set (String^ value);
}
```
See Also

MySqlSessionStateStore Class
MySql.Web.SessionState Namespace

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.
MySqlSessionStateStore..::.WriteExceptionsToEventLog Property

MySqlSessionStateStore Class  See Also  Send Feedback

Indicates whether to write exceptions to event log

Namespace:  MySql.Web.SessionState
Syntax

C#

public bool WriteExceptionsToEventLog { get; set; }

Visual Basic (Declaration)

Public Property WriteExceptionsToEventLog As Boolean

Visual C++

public:
property bool WriteExceptionsToEventLog {
    bool get ();
    void set (bool value);
}
See Also

**MySqlSessionStateStore Class**
**MySql.Web.SessionState Namespace**

Send comments on this topic to support@mysql.com

© 2004-2008 MySQL AB, 2008-2009 Sun Microsystems, Inc. All rights reserved.