

Einführung

MySQL ist ein beliebtes SQL-Datenbankverwaltungssystem und als Open-Source-Software für verschiedene Betriebssysteme verfügbar. Besonders häufig wird MySQL für Webauftritte genutzt. Insbesondere die Tatsache das es sich bei MySQL um ein hochperformantes Datenbanksystem handelt und es für nicht kommerzielle Nutzungen unter die GPL fällt, macht es interessant. Leider stellt Microsoft in seinem .NET Framework keinen ADO.NET Treiber für MySQL zur Verfügung. Dies ist wohl darin begründet das Microsoft bestrebt ist sein eigenes relationales Datenbanksystem namens Microsoft SQL Server zu vermarkten.

In diesem Tutorial werde ich die gängigen Methoden vorstellen mit denen jeder auf einfache Art .NET Applikationen erstellen kann, die in der Lage sind mit einem MySQL Server zu kommunizieren. Es existieren zwei gängige Varianten die sehr häufig zum Einsatz kommen. Die eine Variante nutzt einen ODBC Treiber namens [MyODBC](#). Die ODBC Spezifikation stellt eine prozedurale API für die Nutzung von SQL Queries zum Datenzugriff zur Verfügung. Die andere Variante nutzt einen ADO.NET Treiber der explizit für die MySQL Kommunikation unter .NET entwickelt wurde. Der Treiber ist vollständig in C# geschrieben und der Quellcode ist offen.

Der MySQL Connector/Net implementiert die benötigten ADO.NET Schnittstellen und integriert sich daher problemlos in entsprechende Tools die die ActiveX Data Objects Technologie nutzen. Die Features des MySQL Connector/Net sind:

- Alle MySQL 5.0 Features verfügbar (Stored Procedures, etc.)
- Alle MySQL 4.1 Features verfügbar (Server-seitige preparierte Statements, Unicode, und Shared Memory Zugriff, etc.)
- Support für große Pakete zum Senden und Empfangen von Rows und BLOBs bis zu einer Größe von **2 GigaByte**
- Protokoll Kompression, die es ermöglicht den Datenstrom zwischen Client und Server zu komprimieren
- Unterstützung von Verbindungen über TCP/IP Sockets, Named Pipes oder Shared Memory auf Windows
- Unterstützung von Verbindungen über TCP/IP Sockets oder Unix Sockets
- Unterstützung des Open-Source Frameworks [Mono](#) entwickelt von Novell
- Vollständig verwaltet (managed), macht keinen Gebrauch von der MySQL Client Bibliothek

Verbindung zum MySQL-Server mit C#

Geschrieben von: StarShaper

Samstag, den 08. April 2006 um 11:01 Uhr - Aktualisiert Montag, den 23. November 2009 um 22:32 Uhr

Einstellungen vornehmen

Bevor wir loslegen können müssen wir zunächst den MySQL Connector/NET installieren. Laden Sie sich dafür den aktuellen Treiber in Form des Installers von der offiziellen Seite herunter.



[Download Connector/Net](#)

Anschließend können Sie den Connector installieren.



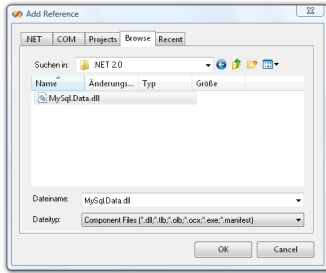
Nachdem Sie den Connector erfolgreich auf Ihrem System installiert haben, können Sie Ihre Visual Studio Entwicklungsumgebung starten. Wir werden in diesem kurzen Tutorial eine kleine Windows Anwendung programmieren die sowohl Daten aus der MySQL Datenbank ausliest als auch Daten in die Datenbank schreibt.

Generieren Sie dazu ein neues C# Projekt und wählen Sie als Projekttyp eine Windows Anwendung. Bevor wir mit der MySQL Datenbank kommunizieren können, müssen wir eine entsprechende Referenz zu MySql.Data hinzufügen. Klicken Sie dazu auf die Option "Add Reference", wählen Sie die Karteikarte "Browse", suchen Sie die MySQL Komponente und fügen Sie sie dem Projekt hinzu. Sie finden die MySql.Data.dll für das .NET Framework 2.0 normalerweise im Verzeichnis "C:\Programme\MySQL\MySQL Connector Net X.X.X\bin\NET 2.0".

Verbindung zum MySQL-Server mit C#

Geschrieben von: StarShaper

Samstag, den 08. April 2006 um 11:01 Uhr - Aktualisiert Montag, den 23. November 2009 um 22:32 Uhr



Eine .NET Applikation die zu einer MySQL Datenbank verbindet

Das in diesem Tutorial vorliegende Programm basiert auf .NET 2.0 und wurde unter Visual Studio 2005 erstellt. Selbstverständlich können Sie den Quellcode auch unter .NET 3.0 oder einer aktuelleren Version nutzen. Beachten Sie aber das Sie den MySQL Connector abhängig von Ihrer Version als Referenz hinzufügen müssen! Neben den genannten Voraussetzungen ist der Zugriff auf einen laufenden MySQL Server erforderlich. Falls Sie diesen nicht haben, können Sie den aktuellen MySQL Datenbank Server samt Client kostenlos von der offiziellen Seite beziehen.

Nach den ganzen Einstellungen etc. können wir mit der Programmierung beginnen und uns direkt dem Code widmen. Die MySQL Klassen und Methoden die für uns relevant sind befinden sich im Namensraum *MySql.Data.MySqlClient*. Das nachfolgende Beispiel zeigt den prinzipiellen Aufbau einer C# Anwendung die MySQL nutzt.

```
[code xml:lang="csharp"]using MySql.Data.MySqlClient; ... string myConnectionString =  
"SERVER=localhost;" + "DATABASE=mydatabase;" +  
"UID=user;" + "PASSWORD=mypassword;"; MySqlConnection connection =  
new MySqlConnection(myConnectionString); MySqlCommand command =  
connection.CreateCommand(); command.CommandText = "SELECT * FROM mytable";  
MySqlDataReader Reader; connection.Open(); Reader = command.ExecuteReader(); while  
(Reader.Read()) { string row = ""; for (int i = 0; i
```

Die Klasse [MySqlConnection](#) repräsentiert ein Objekt welches eine Session zu einer MySQL Server Datenquelle darstellt. Der Konstruktor nimmt einen Connection String als Parameter auf, der die Zugangsdaten, wie beispielsweise den Namen und das Passwort der MySQL Datenbank beinhaltet. Über die Eigenschaft

CommandText

der Klasse MySqlCommand lässt sich das SQL Statement, welches ausgeführt werden soll, setzen. Die Klasse MySqlCommand repräsentiert eine SQL Anweisung (engl. Statement), die gegen die MySQL Datenbank ausgeführt werden kann.

Nachdem wir eine SQL Anweisung definiert haben, mit dem Ziel alle Reihen (engl. Rows) und Felder (engl. Fields) der Tabelle mytable auszulesen, definieren wir ein Objekt mit dem Namen Reader. Es handelt sich um ein Objekt der Klasse *MySqlDataReader*, welche Mittel bereitstellt

Verbindung zum MySQL-Server mit C#

Geschrieben von: StarShaper

Samstag, den 08. April 2006 um 11:01 Uhr - Aktualisiert Montag, den 23. November 2009 um 22:32 Uhr

um Streams von Rows zu lesen.

Nun wird die eigentliche Verbindung zur Datenbank über die MySqlConnection Methode Open() hergestellt. Mithilfe der Methode ExecuteReader() senden wir die SQL Anweisung an die Datenbank. Die Methode gibt ein MySqlDataReader Objekt zurück. Über eine einfache while() Schleife die als Argument eine Methode der Klasse MySqlDataReader nutzt, um von Reihe zu Reihe zu springen, werden die Daten nacheinander ausgelesen. Die Eigenschaft FieldCount gibt die Anzahl der Felder in der Tabellenreihe zurück. In jedem Schleifendurchgang wird die aktuelle Reihe ausgegeben.

Nachfolgend sehen Sie eine Methode zum Einfügen einer Reihe in eine MySQL Datenbank.

```
[code xml:lang="csharp"]public void InsertRow(string myConnectionString) { // If the
connection string is empty, 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); myCommand.Connection =
myConnection; myConnection.Open(); myCommand.ExecuteNonQuery();
myCommand.Connection.Close(); }[/code] AutoSuggestBox mit MySQL und C#
```

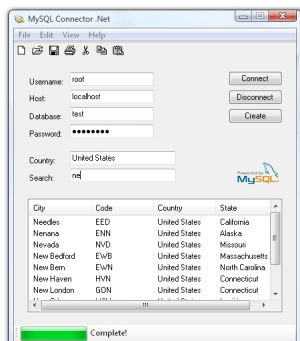
Bestimmt kennen Sie die automatische Vervollständigung von Daten in Textformularen. Moderne Browser speichern beispielsweise Ihre Texteingaben ab, so dass Ihnen bei der nächsten Zeicheneingabe automatisch mögliche Vorschläge unterbreitet werden.

Mit unseren Kenntnissen lässt sich eine derartige Funktion bequem realisieren. Im nachfolgenden werden wir eine MySQL Tabellenstruktur anlegen die die Hauptstädte dieser Welt, mitsamt Länderangabe und Länderkürzel repräsentiert. Die notwendigen Daten befinden sich in einer *.sql (Structured Query Language Data) Datei, welche alle Städtedaten enthält. Unser Programm wird in der Lage sein zu einer Datenbank Verbindung aufzunehmen und bei Eingabe werden uns die entsprechenden Vorschläge angezeigt. Diese werden direkt von der MySQL Datenbank abgerufen. Nachfolgend sehen Sie das Programm wie es am Ende aussehen soll:

Verbindung zum MySQL-Server mit C#

Geschrieben von: StarShaper

Samstag, den 08. April 2006 um 11:01 Uhr - Aktualisiert Montag, den 23. November 2009 um 22:32 Uhr



Schluss

Das Tutorial hat Ihnen hoffentlich einen Einblick in die Verwendung von MySQL unter C#/.NET mithilfe des MySQL Connector/Net Treibers gegeben. Sie finden auf der offiziellen [Entwickler-Seite](#) von MySQL nähere Angaben zu den Klassen und Funktionen des MySQL Connectors. Das oben gezeigte vollständige Programm mitsamt den Datensätzen für die Datenbank finden Sie im Anhang dieses Tutorials. So on, Happy Programming!