

MySQL et .NET

Microsoft®
CERTIFIED

Application Developer



Microsoft®
CERTIFIED
Professional

Par LEBRUN Thomas 


Date de publication : 1 octobre 2004

MySQL, la base de données la plus connue du monde Open Source, attire les développeurs par bien des égards: ses possibilités, ses performances, son prix...
SGBDR préféré des développeurs PHP, voyons comment les développeurs .NET peuvent également l'utiliser dans leurs projets (professionnels ou personnels).

I - Introduction.....	3
II - Installation du Provider.....	3
III - Les objets MySQL.....	4
IV - Codage.....	4
V - Code Complet.....	5
VI - MySQL via Internet.....	8
VII - Conclusion.....	8
VIII - Téléchargement.....	8

I - Introduction

MySQL n'est pas supporté nativement dans l'environnement .NET. Mais, pour remédier à ce problème, vous pouvez télécharger différents providers qui vous permettront d'intégrer MySQL dans votre IDE.

Je vous recommande le provider de  **MySQL Connector** (anciennement connu sous le nom ByteFx), que j'ai utilisé pour la rédaction de cet article, et que vous pourrez télécharger gratuitement à l'adresse suivante : <http://dev.mysql.com/downloads/connector/net/1.0.html>.

II - Installation du Provider

Après avoir téléchargé et décompressé le fichier ZIP, lancer l'installation (très rapide) en double-cliquant sur le fichier Setup.exe.

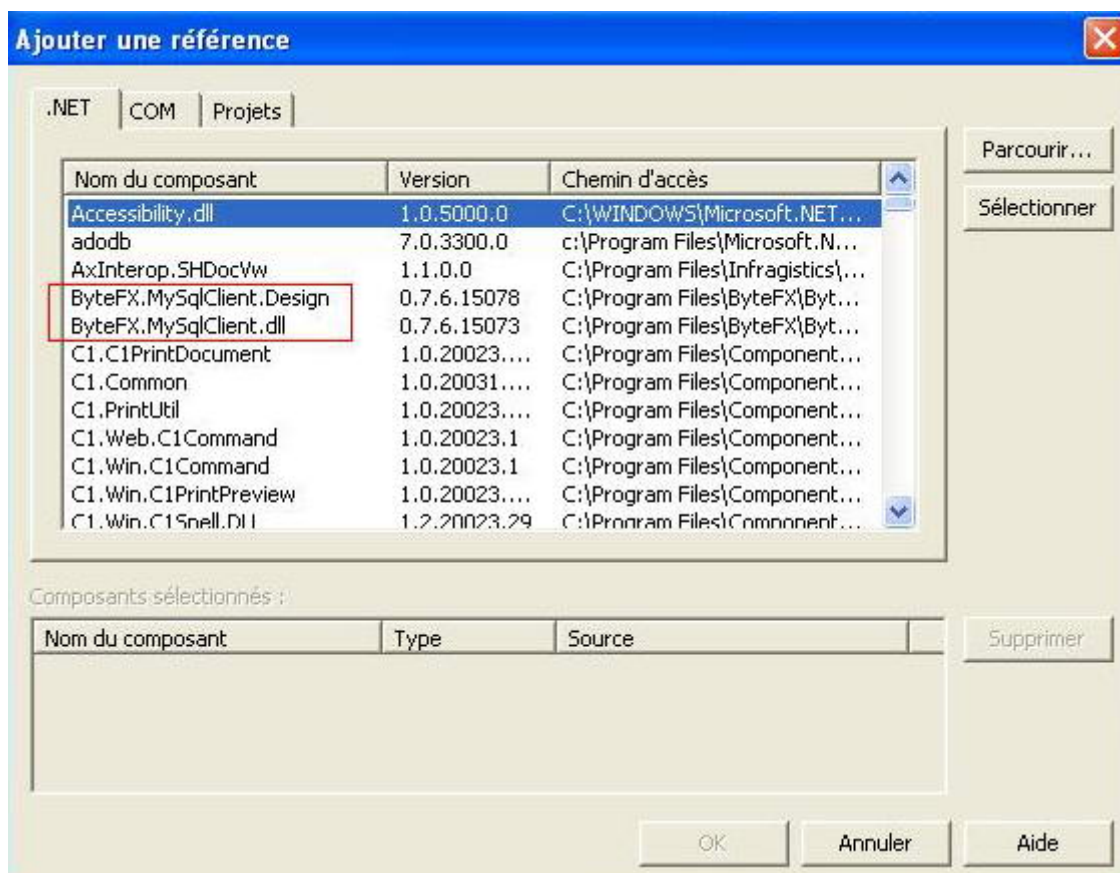
Modifiez, si besoin est, l'emplacement des fichiers à installer pour terminer l'installation.

À priori, rien n'a changé : pas de nouvelle icône sur le bureau, et le menu démarrer ne comporte qu'un nouveau dossier, qui ne contient lui-même que deux raccourcis vers des documentations.

Lancez donc votre environnement de développement (nous partons du principe, pour cet article, que vous travaillez avec Visual Studio 2003) et choisissez de réaliser un nouveau projet (Mode Console ou WinForms, peu importe).

Une fois que tout est ouvert, dans l'explorateur de solution, faites un clic droit sur « **Références** » et choisissez « **Ajouter une référence** ».

Si l'installation du provider s'est bien passée, vous devriez voir deux nouvelles références possibles à ajouter dans vos projets. La capture d'écran ci-jointe vous montrera lesquelles :



Sélectionnez alors **ByteFx.MySqlClient.dll** puis cliquez sur OK pour ajouter la référence à votre projet en cours.

Pensez ensuite à faire le « using » dans votre projet:

Mis en place du using

```
using ByteFX.Data.MySqlClient;
```

III - Les objets MySQL

Si vous avez déjà travaillé avec une base de données, sous .NET, telle que SQL Server, vous serez agréablement surpris de voir que les objets sont les mêmes.

En effet, seul le préfixe des objets change: à la place de **SqlCommand**, nous avons dorénavant **MySqlCommand**.

Voici un tableau récapitulatif des principaux objets :

Objet	Utilité
MySqlCommand	Représente une requête SQL à exécuter
MySqlConnection	Représente une connexion à un serveur MySQL
MySqlDataAdapter	Représente un ensemble de commandes de données utilisées pour remplir un dataset/mettre à jour une base de donnée MySQL
MySqlDbType (Enumération)	Spécifie le type de données d'un champs, d'une propriété, à utiliser avec MySqlParameter
MySqlException	Représente l'exception qui est levée lorsque MySQL retourne une erreur
MySqlParameter	Représente le paramètre pour une commande MySQL

Cette liste, non exhaustive, vous montre les principaux objets que vous pourriez avoir à manipuler avec MySQL.

IV - Codage

Après avoir vu la théorie, voyons un peu le côté pratique: la partie codage !

Le développement d'applications avec MySQL étant très similaire au développement avec SQL Server, je ne vais pas m'attarder trop longtemps sur cette partie: en effet tout ce que je pourrai vous dire ici ne serait qu'un simple copier/coller d'un développement sous SQL Server, donc aucun intérêt....

Je vais donc vous montrer un rapide exemple :

On commence par définir une chaîne de connexion au serveur MySQL :

Chaîne de connexion

```
string ConnectionStr = "Database=BaseDeDonnees;Data Source=localhost;User Id=root;Password=password";
```

Ensuite, on déclare un objet MySqlConnection, auquel on associe cette chaîne de connexion :

Déclaration de la connexion MySQL

```
private MySqlConnection Connection = new MySqlConnection();
Connection.ConnectionString = ConnectionStr;

// Ici, on ouvre la connexion au serveur
Connection.Open();
```

Pour exécuter une simple requête SELECT, rien de plus simple: utilisez l'objet **MySqlDataAdapter** auquel vous associez la requête SELECT à la propriété SelectCommand :

Exécution d'une requête SELECT

```
private MySqlDataAdapter MyAdapter = new MySqlDataAdapter();

// On définit la requête SELECT à exécuter
string MySQLCmd = "SELECT * FROM LaTable";

// On associe cette requête à la propriété SelectCommand du MySqlDataAdapter
MyAdapter.SelectCommand = new MySqlCommand(MySQLCmd, Connection);
```

Si vous avez un DataSet, vous pouvez tout à fait utiliser la méthode **Fill()**, du **MySqlDataAdapter**, pour le remplir :

Remplissage du DataSet

```
private DataSet ds = new DataSet();

// On remplit le DataSet avec le résultat de la commande SELECT
MyAdapter.Fill(ds);
```

Ensuite, libre à vous de travailler avec le **DataSet**, de la même façon que lors de vos autres développements. Pour fermer la connexion au serveur MySQL, rien de plus simple, il vous suffit d'appeler la méthode **Close()**, de l'objet **MySqlConnection** :

Fermeture de la connexion

```
// On ferme la connexion le plus tôt possible
Connection.Close();
```

V - Code Complet

Je vous mets ici une copie du code que j'ai utilisé pour mes tests: vous pourrez tout de même télécharger la solution complète pour Visual Studio 2003 à la fin de cet article.

Code source complet

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using ByteFX.Data.MySqlClient;

namespace TestMySQL
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.TextBox tbhote;
        private System.Windows.Forms.TextBox tblogin;
        private System.Windows.Forms.Button btconnect;
        private System.Windows.Forms.TextBox tbbase;
        private System.Windows.Forms.DataGrid dataGrid1;
        private System.Windows.Forms.TextBox tbpwd;

        private System.ComponentModel.Container components = null;

        private MySqlConnection Connection = new MySqlConnection();
        private MySqlDataAdapter MyAdapter = new MySqlDataAdapter();
        private DataSet ds = new DataSet();

        public Form1()
        {
            //
            // Requis pour la prise en charge du Concepteur Windows Forms
            //
            InitializeComponent();

            //

```

Code source complet

```
// TODO : ajoutez le code du constructeur après l'appel à InitializeComponent
//
}

protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}

#region Code généré par le Concepteur Windows Form
private void InitializeComponent()
{
    this.tbhote = new System.Windows.Forms.TextBox();
    this.tblogin = new System.Windows.Forms.TextBox();
    this.tbpwd = new System.Windows.Forms.TextBox();
    this.btconnect = new System.Windows.Forms.Button();
    this.tbbase = new System.Windows.Forms.TextBox();
    this.dataGrid1 = new System.Windows.Forms.DataGrid();
    ((System.ComponentModel.ISupportInitialize)(this.dataGrid1)).BeginInit();
    this.SuspendLayout();
    //
    // tbhote
    //
    this.tbhote.Location = new System.Drawing.Point(24, 16);
    this.tbhote.Name = "tbhote";
    this.tbhote.TabIndex = 0;
    this.tbhote.Text = "localhost";
    //
    // tblogin
    //
    this.tblogin.Location = new System.Drawing.Point(256, 16);
    this.tblogin.Name = "tblogin";
    this.tblogin.TabIndex = 1;
    this.tblogin.Text = "login";
    //
    // tbpwd
    //
    this.tbpwd.Location = new System.Drawing.Point(376, 16);
    this.tbpwd.Name = "tbpwd";
    this.tbpwd.PasswordChar = '*';
    this.tbpwd.TabIndex = 2;
    this.tbpwd.Text = "password";
    //
    // btconnect
    //
    this.btconnect.Location = new System.Drawing.Point(496, 16);
    this.btconnect.Name = "btconnect";
    this.btconnect.TabIndex = 3;
    this.btconnect.Text = "Connexion";
    this.btconnect.Click += new System.EventHandler(this.btconnect_Click);
    //
    // tbbase
    //
    this.tbbase.Location = new System.Drawing.Point(144, 16);
    this.tbbase.Name = "tbbase";
    this.tbbase.TabIndex = 4;
    this.tbbase.Text = "base";
    //
    // dataGrid1
    //
    this.dataGrid1.DataMember = "";
    this.dataGrid1.HeaderForeColor = System.Drawing.SystemColors.ControlText;
    this.dataGrid1.Location = new System.Drawing.Point(24, 64);
    this.dataGrid1.Name = "dataGrid1";
}
```

Code source complet

```
this.dataGrid1.Size = new System.Drawing.Size(544, 184);
this.dataGrid1.TabIndex = 5;
//
// Form1
//
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(600, 266);
this.Controls.Add(this.dataGrid1);
this.Controls.Add(this.tbbase);
this.Controls.Add(this.btconnect);
this.Controls.Add(this.tbpwd);
this.Controls.Add(this.tblogin);
this.Controls.Add(this.tbhote);
this.Name = "Form1";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "MySQL et .NET";
((System.ComponentModel.ISupportInitialize)(this.dataGrid1)).EndInit();
this.ResumeLayout(false);

}
#endregion

[STAThread]
static void Main()
{
    Application.Run(new Form1());
}

private void btconnect_Click(object sender, System.EventArgs e)
{
    string ConnexionStr = String.Empty;

    if ( (tbhote.Text.Length == 0) || (tbbase.Text.Length == 0) || (tblogin.Text.Length == 0) )
    {
        MessageBox.Show("Erreur, les champs Hôte, Base et Login doivent être remplis", "Erreur",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        // On définit la chaine de connexion
        ConnexionStr = "Database=" + tbbase.Text + ";Data Source=" + tbhote.Text + ";User Id="
        + tblogin.Text + ";Password=" + tbpwd.Text;
    }

    Connection.ConnectionString = ConnexionStr;

    try
    {
        // On ouvre la connexion
        Connection.Open();

        // On définit la requête SELECT à exécuter
        string MySQLCmd = "SELECT * FROM phpbb_users";

        // On associe cette requête à la propriété SelectCommand du MySqlDataAdapter
        MyAdapter.SelectCommand = new MySqlCommand(MySQLCmd, Connection);

        // On remplit le dataset avec le résultat de la commande SELECT
        MyAdapter.Fill(ds);

        // On affiche le datagrid
        dataGrid1.DataSource = ds.Tables[0];
        dataGrid1.Refresh();

        // On ferme la connexion le plus tôt possible
        Connection.Close();
    }
    catch (MySqlException Ex)
    {
        MessageBox.Show("Erreur SQL:\n" + Ex.Message, "Erreur", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
```

Code source complet

```
}  
}  
}  
}
```

VI - MySQL via Internet

Beaucoup d'entre vous, après avoir lu cet article, auront certainement cette idée: « Tiens, pourquoi je n'essayerai pas de joindre la base MySQL de mon site perso ? ».

En effet, rien de tel que d'essayer pour apprendre ;)


Mais pour pouvoir faire ceci, vous devez prendre en considération plusieurs choses :

- Pour joindre l'hôte MySQL de votre serveur, il faut que votre hébergeur ait ouvert le port de MySQL (par défaut 3306) à l'extérieur : sur la plupart des hébergeurs, cela n'est pas fait pour des raisons de sécurité, donc à moins que vous ne possédiez votre propre serveur dédié, il vous sera impossible de réaliser votre idée ;
- Si vous savez que le port est ouvert mais que vous n'arrivez pas à joindre le serveur MySQL, essayez de passer par l'IP du serveur pour voir s'il ne s'agit pas d'un problème de DNS ;
- Joindre un site, par Internet, c'est un peu comme utiliser un Webservice : cela bloque l'affichage. Pour éviter cela, penser à utiliser des Threads.

VII - Conclusion

Comme vous avez certainement pu le remarquer tout au long de cet article, le développement pour MySQL ne diffère pas énormément du développement sous SQL Server : méthodes identiques, propriétés identiques, etc. Voilà qui permet aux jeunes développeurs de pouvoir commencer à développer des applications utilisant une base de données sans qu'ils en aient les possibilités financières.

VIII - Téléchargement

Télécharger un exemple d'application (pour Visual Studio 2003) :  [Sources](#)