**[ADO.NET入门教程（三） 连接字符串，你小觑了吗？](http://www.cnblogs.com/liuhaorain/archive/2012/02/12/2347914.html)**

**摘要**

      ADO.NET强大的优势在于对不同的数据源提供一致的访问。在上一篇文章[《你必须知道的ADO.NET（二）了解.NET数据提供程序》](http://www.cnblogs.com/liuhaorain/archive/2012/02/11/2346312.html)中，我们知道.NET对不同数据源，使用不同的数据提供程序，这使得我们可以更高效的访问相应的数据源。除了.NET数据提供程序外，我还得说说另外一位功臣：连接字符串（Connection Strings），它也为ADO.NET在解决“不同”与“一致”的矛盾中，贡献出了非常重要的力量。

**目录**

* [什么是连接字符串？](http://www.cnblogs.com/liuhaorain/archive/2012/02/12/2347914.html#title_1)
* [理解语法格式](http://www.cnblogs.com/liuhaorain/archive/2012/02/12/2347914.html#title_2)
* [例举几种典型连接字符串](http://www.cnblogs.com/liuhaorain/archive/2012/02/12/2347914.html#title_3)
* [如何构造连接字符串？](http://www.cnblogs.com/liuhaorain/archive/2012/02/12/2347914.html#title_4)
* [在配置文件中存储连接字符串](http://www.cnblogs.com/liuhaorain/archive/2012/02/12/2347914.html#title_5)

**1. 什么是连接字符串？**

      我们已经知道，ADO.NET类库为不同的外部数据源提供了一致的访问。这些数据源可以是本地的数据文件（如excel、txt、access，甚至是SQLite），也可以是远程的数据库服务器（如SQL Server、MySQL、DB2、Oracle等）。数据源似乎琳琅满目，鱼龙混杂。请试想一下，ADO.NET如何能够准确而又高效的访问到不同数据源呢？在上一篇[《你必须知道的ADO.NET（二）了解.NET数据提供程序》](http://www.cnblogs.com/liuhaorain/archive/2012/02/11/2346312.html)中，你或许已经清楚，ADO.NET已经为不同的数据源编写了不同的数据提供程序。但是这个前提是，我们得访问到正确的数据源。否则的话，只会“张冠李戴，驴头不对马嘴”。就好比，我们用SQL Server数据提供程序去处理excel数据源，结果肯定是让人“瞠目结舌”的。英雄总在最需要的时候出现，**连接字符串，就是这样一组被格式化的键值对：它告诉ADO.NET数据源在哪里，需要什么样的数据格式，提供什么样的访问信任级别以及其他任何包括连接的相关信息。**哦！如果你这样想的话，那么以后你使用连接字符串的时候，或许就不会小觑了吧！

**2.理解语法格式**

      其实，连接字符串虽然影响深远，但是其本身的语法却是十分的简单。**连接字符串由一组元素组成，一个元素包含一个键值对，元素之间由“;”分开。**语法如下：

key1=value1;key2=value2;key3=value3...

      典型的元素（键值对）应当包含这些信息：数据源是基于文件的还是基于网络的数据库服务器，是否需要账号密码来访问数据源，超时的限制是多少，以及其他相关的配置信息。我们知道，值（value）是根据键（key）来确定的，那么键（key）如何来确定呢？语法并没有规定键（key）是什么，这需要根据你需要连接的数据源来确定的。在下一节中，我将详细讲解几种常用的连接字符串。

提示：如果需要了解更多连接字符串的格式，可以参考 <http://www.connectionstrings.com/>。

**3.例举几种典型连接字符串**

**3.1 SQL Sever连接字符串**

（1）标准的安全连接

Data Source=myServerAddress;Initial Catalog=myDataBase;User Id=myUsername;Password=myPassword;

说明：

Data Source:需要连接的服务器。需要注意的是，如果使用的时Express版本的SQL Server需要在服务器名后加\SQLEXPRESS。例如，连接本地的SQL Server 2008 Express版本的数据库服务器，可以写成Data Source = (local)\SQLEXPRESS或者.\SQLEXPRESS。

Initial Catalog:默认使用的数据库名称。

User ID:数据库服务器账号。

Password:数据库服务器密码。

或者也可以写成这样：

Server=myServerAddress;Database=myDataBase;User ID=myUsername;Password=myPassword;Trusted\_Connection=False;

（2）可信连接

Data Source=myServerAddress;Initial Catalog=myDataBase;Integrated Security=SSPI;

说明：

Data Source:与上述相同。

Initial Catalog：与上述相同。

Integrate Security:使用存在的windows安全证书访问数据库。

或者也可以写成这样：

Server=myServerAddress;Database=myDataBase;Trusted\_Connection=True;

**3.2 Access连接字符串**

Provider=Microsoft.Jet.OLEDB.**4.0**;Data Source=C:\mydatabase.mdb;User Id=admin;Password=;

**3.3 MySQL连接字符串**

Server=myServerAddress;Database=myDataBase;Uid=myUsername;Pwd=myPassword;

**3.4 DB2连接字符串**

Server=myAddress:myPortNumber;Database=myDataBase;UID=myUsername;PWD=myPassword;

**3.5 Oracle连接字符串**

Data Source=TORCL;User Id=myUsername;Password=myPassword;

**4.如何构造连接字符串？**

      在上面我们知道,连接字符串本质上就是一个字符串，因此我们完全可以用

string connStr = "Data Source=myServerAddress;Initial Catalog=myDataBase;User ID=myUsername;Password=myPassword";

来构造一个连接字符串。实际上，ADO.NET有一个专门的类来处理连接字符串：DbConnectionStringBuilder。**DbConnectionStringBuilder类为强类型连接字符串生成基类**。之所以要有这样一个类，是因为它更加安全与友好。以SQL Server为例，可以这样构建一个连接字符串：

1 SqlClient.SqlConnectionStringBuilder builder =  
2 new SqlClient.SqlConnectionStringBuilder();  
3 builder.DataSource = @"(local)\SQLEXPRESS";  
4 builder.InitialCatalog = "myDataBase";  
5 builder.IntegratedSecurity = true;

**5.在配置文件中存储连接字符串**

      在我们实际开发中，我们一般不会把连接字符串直接写在代码中，而是存储在配置文件里。把连接字符串写死在代码中，不便于维护，每次修改字符串时，还得重新编译代码。以ASP.NET应用程序为例，我们一般把连接字符串写在web.config配置文件的<connectionstrings />节点。例如：

1 <connectionStrings>  
2 <add name="connStr" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=myDataBase;Integrated Security=SSPI" />  
3 </connectionStrings>

      因此，我们只需要在程序中添加相应代码来获取配置文件中的值，比如：

string connStr = ConfigurationManager.ConnectionStrings["connStr"].ToString();

      当然，你可以在配置文件的<appsettings />节点下添加连接字符串，获取字符串值的方法类似。