**[ADO.NET入门教程（二）了解.NET数据提供程序](http://www.cnblogs.com/liuhaorain/archive/2012/02/11/2346312.html)**

**1. 什么是.NET数据提供程序？**

      .NET Framework数据提供程序**用于连接数据库、执行命令和检索结果。**这些结果将被直接处理，放置在 DataSet 中以便根据需要向用户公开、与多个源中的数据组合，或在层之间进行远程处理。.NET Framework 数据提供程序是轻量的，它在数据源和代码之间创建最小的分层，并在不降低功能性的情况下提高性能。

      下表列出了 .NET Framework 中所包含的数据提供程序。

|  |  |
| --- | --- |
| **.NET数据提供程序** | **说明** |
| 用于 SQL Server 的数据提供程序 | 提供对 Microsoft SQL Server 7.0 或更高版本中数据的访问。使用 **System.Data.SqlClient** 命名空间。 |
| 用于 OLE DB 的数据提供程序 | 提供对使用 OLE DB 公开的数据源中数据的访问。使用 **System.Data.OleDb** 命名空间。 |
| 用于 ODBC 的数据提供程序 | 提供对使用 ODBC 公开的数据源中数据的访问。使用 **System.Data.Odbc** 命名空间。 |
| 用于 Oracle 的数据提供程序 | 适用于 Oracle 数据源。用于 Oracle 的 .NET Framework 数据提供程序支持 Oracle 客户端软件 8.1.7 和更高版本，并使用 **System.Data.OracleClient** 命名空间。 |
| EntityClient 提供程序 | 提供对实体数据模型 (EDM) 应用程序的数据访问。使用 **System.Data.EntityClient** 命名空间。 |

**2. .NET数据提供程序的核心对象**

      在上一篇文章中，我们知道Connection对象、Command对象、DataReader对象以及DataAdapter对象构成了.NET数据提供程序的骨架。这四个对象非常重要，在后续的文章中，我将详细的讲解。如果需要了解这些对象的作用，可以参考上一篇文章[《你必须知道的ADO.NET（一） 初识ADO.NET》](http://www.cnblogs.com/liuhaorain/archive/2012/02/06/2340409.html)。

**3. 其他重要的对象**

      如果说上述四大对象构成了.NET数据提供程序的骨架，那么下面我要说的这些对象可以说是.NET数据提供程序的血肉了。这些对象虽然没有四大核心对象那么的光鲜耀眼，但却也是“八仙过海，各显神通”。

**3.1 Parameter对象**

      说对Parameter对象，也许大部分人会说：“哦，原来是它啊”。尽管大部分人已经很熟悉了，我还是要在这里唠叨几句。我需要强调是，这一系列的文章主要写给对ADO.NET还不熟悉，或者刚入门的读者，旨在讲解ADO.NET最最基础却又非常重要的内容。

      简单的讲，**Parameter对象定义了命令和存储过程的输入、输出和返回值参数。**哦！看起来，好像并不是那么强大，那么Parameter对象到底有什么本领呢？

先看这样一段代码，也许很多人曾经都写过，包括我自己：

strSQL = "SELECT \* FROM users WHERE (name = '" + userName + "') and (pw = '"+ passWord +"');"

只要懂一点SQL语法的童鞋都知道，这不就是一个简单的登陆查询验证代码吗？好，先别急，你知道的，说不定别人还不知道。

试想一下，如果用户（一般是那些技术高超，自称为“黑客”的高级用户）填入

userName = "' OR '**1**'='**1**";

与

passWord = "' OR '**1**'='**1**";

接下来，将见证奇迹的时刻：该用户竟然成功登陆网站了。哇！看起来这一切似乎多么的魔幻和神奇，其实我们稍作分析发现这也不过是一些雕虫小计。

我们将userName和passWord变量带入strSQL变量后，将得到这样的一条SQL语句：

strSQL = "SELECT \* FROM users WHERE (name = '' OR '1'='1') and (pw = '' OR '1'='1');"

也就是实际上运行的SQL命令会变成下面这样的

strSQL = "SELECT \* FROM users;"

到这一步，我想也不需要我多说了吧，你懂的！上面的情况，用专业术语来说就是一个简单的SQL注入（SQL injection）。记得上政治课的时候，我印象最深的一句话是，“万物都是矛盾统一的”。这句话经典而又真实，以至于时刻在我的脑海里浮现。有SQL注入的出现，因此就有参数化查询（Parameterized Query ）的出现。

**参数化查询是指在设计与数据库连结并存取资料时，在需要填入数值或资料的地方，使用参数 (Parameter) 来给值**，这个方法目前已被视为最有效可预防SQL注入(SQL Injection) 的攻击手法的防御方式。在使用参数化查询的情况下，数据库服务器不会将参数的内容视为SQL指令的一部份来处理，而是在数据库完成 SQL 指令的编译后，才套用参数执行，因此就算参数中含有具破坏性的指令，也不会被数据库所执行。说了这么多，无非是想说明Parameter对象的重要性。黄婆卖瓜，也得自卖自夸一下吧！Parameter对象有两个非常重要的属性：DBType和Value。DBType用来设置或获取参数的类型，Value则用来设置或获取参数的值。

      好了，现在我们用Parameter对象来改写简单的登陆验证代码：

1 strSQL = "SELECT \* FROM users WHERE Name = @Name and Password = @Password";  
2 SqlParamter[] paras = new SqlParamter[]{//参数数组  
3 new SqlParamter("@Name",SqlDBType.Varchar,**50**)  
4 new SqlParamter("@Password",SqlDBType.Varchar,**50**)};  
5 paras[0].value = userName;//绑定用户名  
6 paras[1].value = password;//绑定用户密码

**3.2 两大得力助手：ConnectionStringBuilder和CommandBuilder**

* ConnectionStringBuilder：它提供一种用于创建和管理由 Connection 对象使用的连接字符串的内容的简单方法。 所有 ConnectionStringBuilder 对象的基类均为 DbConnectionStringBuilder 类。
* CommandBuilder ：它自动生成 DataAdapter 的命令属性或从存储过程中派生参数信息，并填充 Command 对象的 Parameters 集合。 所有 CommandBuilder 对象的基类均为 DbCommandBuilder 类。

**4. 理解.NET数据提供程序**

**4.1 用于 SQL Server 的 .NET Framework 数据提供程序 (SqlClient)**

      用于 SQL Server 的 .NET Framework 数据提供程序 (SqlClient) **使用自己的协议与 SQL Server 进行通信。** 它是轻量的且性能良好，因为它进行了优化，可直接访问 SQL Server，而无需添加 OLE DB 或开放式数据库连接 (ODBC) 层。 下图4.1.1将用于 SQL Server 的 .NET Framework 数据提供程序与用于 OLE DB 的 .NET Framework 数据提供程序进行对比。 用于 OLE DB 的 .NET Framework 数据提供程序通过 OLE DB 服务组件（它提供连接池和事务服务）和用于数据源的 OLE DB 访问接口与 OLE DB 数据源进行通信。

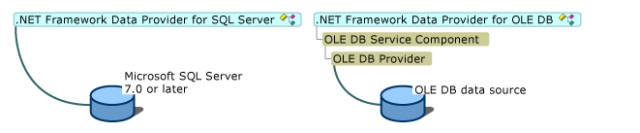


       图4.1.1  SQL Server 与 OLE DB .NET Framework 数据提供程序进行对比

      若要使用用于 SQL Server 的 .NET Framework 数据提供程序，您**必须具有对 SQL Server 7.0 或更高版本的访问权限。** 用于 SQL Server 类的 .NET Framework 数据提供程序位于 System.Data.SqlClient 命名空间中。 对于早期版本的 SQL Server，请将用于 OLE DB 的 .NET Framework 数据提供程序与 SQL Server OLE DB 访问接口 System.Data.OleDb 一起使用。

      用于 SQL Server 的 .NET Framework 数据提供程序支持本地事务和分布式事务。 对于分布式事务，默认情况下，用于 SQL Server 的 .NET Framework 数据提供程序会自动登记在事务中，并自动从 Windows 组件服务或 System.Transactions 获取事务详细信息。

      如果你使用SQL Server数据提供程序需要引入：

using System.Data.SqlClient;

**4.2 用于 OLE DB 的 .NET Framework 数据提供程序**

      用于 OLE DB 的 .NET Framework 数据提供程序 (OleDb) **通过 COM 互操作使用本机 OLE DB 来启用数据访问。** 用于 OLE DB 的 .NET Framework 数据提供程序支持本地事务和分布式事务。 对于分布式事务，默认情况下，用于 OLE DB 的 .NET Framework 数据提供程序会自动登记在事务中，并自动从 Windows 2000 组件服务获取事务详细信息。用于 OLE DB 类的 .NET Framework 数据提供程序位于 System.Data.OleDb 命名空间中。

      如果你使用OLE DB数据提供程序需要引入：

using System.Data.OleDb;

**4.3 用于 ODBC 的 .NET Framework 数据提供程序**

      用于 ODBC 的 .NET Framework 数据提供程序 (Odbc) **使用本机 ODBC 驱动程序管理器 (DM) 来启用数据访问。** ODBC 数据提供程序支持本地事务和分布式事务两者。 对于分布式事务，默认情况下，ODBC 数据提供程序会自动登记在事务中，并自动从 Windows 2000 组件服务获取事务详细信息。用于 ODBC 类的 .NET Framework 数据提供程序位于 System.Data.Odbc 命名空间中。

     如果你使用ODBC数据提供程序需要引入：

using System.Data.Odbc;

**4.4 用于 Oracle 的 .NET Framework 数据提供程序**

       用于 Oracle 的 .NET Framework 数据提供程序 (OracleClient) **通过 Oracle 客户端连接软件启用对 Oracle 数据源的数据访问。** 该数据提供程序支持 Oracle 客户端软件 8.1.7 版或更高版本。 该数据提供程序支持本地事务和分布式事务两者。

**用于 Oracle 的 .NET Framework 数据提供程序要求系统上安装有 Oracle 客户端软件（8.1.7 版或更高版本），才能连接到 Oracle 数据源。**

      用于 Oracle 类的 .NET Framework 数据提供程序位于 System.Data.OracleClient 命名空间中，并包含在 System.Data.OracleClient.dll 程序集中。 当编译使用该数据提供程序的应用程序时，必须同时引用 System.Data.dll 和 System.Data.OracleClient.dll。

      如果你使用Oracle数据提供程序需要引入：

1 using System.Data;  
2 using System.Data.OracleClient;

**5. 选择合适的 .NET 数据提供程序**

      应用程序或者数据源不同，我们就需要选择不同的.NET数据提供程序。在此，微软官方已经给了我们很好的建议，如下表：

|  |  |
| --- | --- |
| **提供程序** | **说明** |
| 用于 SQL Server 的数据提供程序 | 建议用于使用 Microsoft SQL Server 7.0 或更高版本的中间层应用程序。  建议用于使用 Microsoft 数据库引擎 (MSDE) 或 SQL Server 7.0 或更高版本的单层应用程序。  建议将用于 SQL Server 的 OLE DB 访问接口 (SQLOLEDB) 与用于 OLE DB 的 .NET Framework 数据提供程序一起使用。  对于 SQL Server 6.5 和更新版本，您必须将用于 SQL Server 的 OLE DB 访问接口与用于 OLE DB 的 .NET Framework 数据提供程序一起使用。 |
| 用于 OLE DB 的数据提供程序 | 建议用于使用 SQL Server 6.5 或早期版本的中间层应用程序。  对于 SQL Server 7.0 或更高版本，建议使用用于 SQL Server 的 .NET Framework 数据提供程序。  还建议用于使用 Microsoft Access 数据库的单层应用程序。 不建议将 Access 数据库用于中间层应用程序。 |
| 用于 ODBC 的数据提供程序 | 建议用于使用 ODBC 数据源的中间层应用程序和单层应用程序。 |
| 用于 Oracle 的数据提供程序 | 建议用于使用 Oracle 数据源的中间层应用程序和单层应用程序。 |