**[ADO.NET入门教程（一） 初识ADO.NET](http://www.cnblogs.com/liuhaorain/archive/2012/02/06/2340409.html)**

**目录**

* [什么是ADO.NET?](http://www.cnblogs.com/liuhaorain/archive/2012/02/06/2340409.html#title_1)
* [理清ADO.NET与ADO的关系](http://www.cnblogs.com/liuhaorain/archive/2012/02/06/2340409.html#title_2)
* [认识ADO.NET最核心的组成部分](http://www.cnblogs.com/liuhaorain/archive/2012/02/06/2340409.html#title_3)
* [ADO.NET扩展](http://www.cnblogs.com/liuhaorain/archive/2012/02/06/2340409.html#title_4)

**1. 什么是ADO.NET?**

      简单的讲，**ADO.NET是一组允许.NET开发人员使用标准的，结构化的，甚至无连接的方式与数据交互的技术。**对于ADO.NET来说，可以处理数据源是多样的。可以是应用程序唯一使用的创建在内存中数据，也可以是与应用程序分离，存储在存储区域的数据（如文本文件、XML、关系数据库等）。

      具体来说，ADO.NET 对 Microsoft SQL Server 和 XML 等数据源以及通过 OLE DB 和 XML 公开的数据源提供一致的访问。数据共享使用者应用程序可以使用 ADO.NET 来连接到这些数据源，并检索、处理和更新所包含的数据。

      作为.NET框架的重要组成部分，ADO.NET 类封装在 System.Data.dll 中，并且与 System.Xml.dll 中的 XML 类集成。当编译使用 System.Data 命名空间的代码时，需要引用System.Data.dll 和 System.Xml.dll。

**2. 理清ADO.NET与ADO的关系**

      作为一个普通的缩略词，"ADO.NET”并只不是"ADO”的简单升级版本。严格的讲，**ADO.NET和ADO是两种截然不同的数据访问方式**。

      ADO的全称是Activex Data Objects，它是早期（.NET还未实施）开发人员用来访问数据的组件。随着.NET的发展，ADO.NET顺其自然地以其显著的优越性逐步取代ADO。从技术层面讲，ADO使用OLE DB接口并基于微软的COM技术，而ADO.NET拥有自己的ADO.NET接口并且基于微软的.NET体系架构。

      虽然大多数基于 .NET 的新应用程序将使用 ADO.NET 来编写，但 .NET 程序员仍然可以通过 .NET COM 互操作性服务来使用 ADO。

**3. 认识ADO.NET最核心的组成部分**

      System.Data命名空间提供了不同的ADO.NET类，它们既分工明确，又相互协作地提供表格数据的访问服务。该类库包含两组重要的类：一组负责处理软件内部的实际数据（DataSet），一组负责与外部数据系统通信（Data Provider）。具体架构如下图所示：

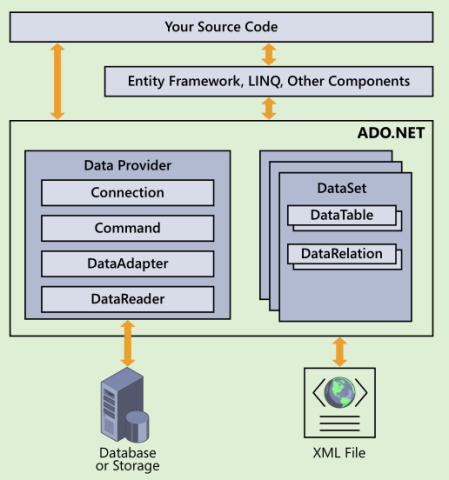


                        图1.3.1  ADO.NET核心组件

**DataSet 是 ADO.NET 的非连接（断开）结构的核心组件。**DataSet 的设计目的很明确：为了实现独立于任何数据源的数据访问。因此，ADO.NET结构可以用于多种不同的数据源，用于 XML 数据，或用于管理应用程序本地的数据。DataSet 包含一个或多个 DataTable 对象的集合，这些对象由数据行和数据列以及主键、外键、约束和有关 DataTable 对象中数据的关系信息组成。

      ADO.NET 结构的另一个核心元素是 .NET 数据提供程序（Data Provider）。具体包括：

* Connection 对象提供与数据源的连接。
* Command对象使您能够访问用于返回数据、修改数据、运行存储过程以及发送或检索参数信息的数据库命令。
* DataReader 对象从数据源中提供快速的，只读的数据流。
* DataAdapter 对象提供连接 DataSet 对象和数据源的桥梁。DataAdapter 使用 Command 对象在数据源中执行 SQL 命令，以便将数据加载到 DataSet 中，并使对 DataSet 中数据的更改与数据源保持一致。

**4. ADO.NET扩展**

      提供一致的数据访问，是使用ADO.NET的一个关键的优势。但是对于开发人员来说，更大的优势是通过ADO.NET将管理的数据作为对象来说处理。 表中的每个字段都是强类型成员，与.NET 通用类型系统（Common Type System）完全兼容。个别的字段甚至可以作为局部变量来使用。数据行或者其他的数据集对象是标准的.NET 集合（Collections），可以用标准的迭代方法处理。

      Entity Framework和LINQ是微软为了提高ADO.NET核心功能而建立的两个新的工具。需要注意的是，它们**并不是ADO.NET的基本组成部分**。

      Entity Framework 利用了抽象化数据结构的方式，将每个数据库对象都转换成应用程序对象 (entity)，而数据字段都转换为属性 (property)，关系则转换为结合属性 (association)，让数据库的 E/R 模型完全的转成对象模型。而在抽象化的结构之下，则是高度集成与对应结构的概念层、对应层和储存层，以及支持 Entity Framework 的数据提供者 (provider)，让数据访问的工作得以顺利与完整的进行。

      LINQ允许编写C#或者Visual Basic代码以查询数据库相同的方式操作内存数据。LINQ是一个通用的数据工具，可以让你非常容易地融合不同数据源的数据，并得到单一的数据结果集。