

# OleDbDataReader 类

.NET Framework 4

提供从数据源读取数据行的只进流的方法。 此类不能被继承。

## 继承层次结构

System.Object  
System.MarshalByRefObject  
System.Data.Common.DbDataReader  
System.Data.OleDb.OleDbDataReader

命名空间：[System.Data.OleDb](#)  
程序集：[System.Data](#) (在 [System.Data.dll](#) 中)

## 语法

C#

```
public sealed class OleDbDataReader : DbDataReader
```

OleDbDataReader 类型公开以下成员。

## 属性


显示: ☒ 继承 ☒ 保护











	名称	说明
	Depth	获取一个值，用于指示当前行的嵌套深度。（重写 <a href="#">DbDataReader.Depth</a> 。）
	FieldCount	获取当前行中的列数。（重写 <a href="#">DbDataReader.FieldCount</a> 。）
	HasRows	获取一个值，用于指示 OleDbDataReader 是否包含一行或多行。（重写 <a href="#">DbDataReader.HasRows</a> 。）
	IsClosed	指示是否可关闭数据读取器。（重写 <a href="#">DbDataReader.IsClosed</a> 。）


	<a href="#">Item[Int32]</a>	在给定列序号的情况下，获取指定列的以本机格式表示的值。（重写 <a href="#">DbDataReader.Item[Int32]</a> 。）
	<a href="#">Item[String]</a>	在给定列名称的情况下，获取指定列的以本机格式表示的值。（重写 <a href="#">DbDataReader.Item[String]</a> 。）
	<a href="#">RecordsAffected</a>	通过执行 SQL 语句获取更改、插入或删除的行数。（重写 <a href="#">DbDataReader.RecordsAffected</a> 。）
	<a href="#">VisibleFieldCount</a>	获取 OleDbDataReader 中未隐藏的字段的数目。（重写 <a href="#">DbDataReader.VisibleFieldCount</a> 。）


















[页首](#)

# 方法

显示: ☒ 继承 ☒ 保护 

	名称	说明
	<a href="#">Close</a>	关闭 OleDbDataReader 对象。（重写 <a href="#">DbDataReader.Close()</a> 。）
	<a href="#">CreateObjRef</a>	创建一个对象，该对象包含生成用于与远程对象进行通信的代理所需的全部相关信息。（继承自 <a href="#">MarshalByRefObject</a> 。）
	<a href="#">Dispose()</a>	释放 <a href="#">DbDataReader</a> 类的当前实例所使用的所有资源。（继承自 <a href="#">DbDataReader</a> 。）
	<a href="#">Dispose(Boolean)</a>	释放由 <a href="#">DbDataReader</a> 使用的托管资源，还可以选择释放非托管资源。（继承自 <a href="#">DbDataReader</a> 。）
	<a href="#">Equals(Object)</a>	确定指定的 <a href="#">Object</a> 是否等于当前的 <a href="#">Object</a> 。（继承自 <a href="#">Object</a> 。）
	<a href="#">Finalize</a>	允许对象在“垃圾回收”回收之前尝试释放资源并执行其他清理操作。（继承自 <a href="#">Object</a> 。）
	<a href="#">GetBoolean</a>	获取指定列的布尔值形式的值。（重写 <a href="#">DbDataReader.GetBoolean(Int32)</a> 。）
	<a href="#">GetByte</a>	获取指定列的字节形式的值。（重写 <a href="#">DbDataReader.GetByte(Int32)</a> 。）
	<a href="#">GetBytes</a>	从按指定的列偏移量将字节流作为数组从给定的缓冲区偏移量开始读入缓冲区。（重写 <a href="#">DbDataReader.GetBytes(Int32, Int64, Byte[], Int32, Int32)</a> 。）
	<a href="#">GetChar</a>	获取指定列的字符形式的值。（重写

		<a href="#">DbDataReader.GetChar(Int32)</a> 。 )
	<a href="#">GetChars</a>	从指定的列偏移量将字符流作为数组从给定的缓冲区偏移量开始读入缓冲区。 ( 重写 <a href="#">DbDataReader.GetChars(Int32, Int64, Char[], Int32, Int32)</a> 。 )
	<a href="#">GetData</a>	返回请求的列序号的 <a href="#">OleDbDataReader</a> 对象。
	<a href="#">GetDataTypeName</a>	获取源数据类型的名称。 ( 重写 <a href="#">DbDataReader.GetDataTypeName(Int32)</a> 。 )
	<a href="#">GetDateTime</a>	获取指定列的 <a href="#">DateTime</a> 对象形式的值。 ( 重写 <a href="#">DbDataReader.GetDateTime(Int32)</a> 。 )
	<a href="#">GetDbDataReader</a>	返回被请求的列序号的 <a href="#">DbDataReader</a> 对象，可以使用提供程序特定的实现对该对象进行重写。 ( 继承自 <a href="#">DbDataReader</a> 。 )
	<a href="#">GetDecimal</a>	获取指定列的 <a href="#">Decimal</a> 对象形式的值。 ( 重写 <a href="#">DbDataReader.GetDecimal(Int32)</a> 。 )
	<a href="#">GetDouble</a>	获取指定列的双精度浮点数形式的值。 ( 重写 <a href="#">DbDataReader.GetDouble(Int32)</a> 。 )
	<a href="#">GetEnumerator</a>	返回一个 <a href="#">IEnumerator</a> ，可用于循环访问数据读取器中的行。 ( 重写 <a href="#">DbDataReader.GetEnumerator()</a> 。 )
	<a href="#">GetFieldType</a>	获取是对象的数据类型的 <a href="#">Type</a> 。 ( 重写 <a href="#">DbDataReader.GetFieldType(Int32)</a> 。 )
	<a href="#">GetFloat</a>	获取指定列的单精度浮点数形式的值。 ( 重写 <a href="#">DbDataReader.GetFloat(Int32)</a> 。 )
	<a href="#">GetGuid</a>	获取指定列的值作为全局唯一标识符 (GUID)。 ( 重写 <a href="#">DbDataReader.GetGuid(Int32)</a> 。 )
	<a href="#">GetHashCode</a>	用作特定类型的哈希函数。 ( 继承自 <a href="#">Object</a> 。 )
	<a href="#">GetInt16</a>	获取指定列的 16 位有符号整数形式的值。 ( 重写 <a href="#">DbDataReader.GetInt16(Int32)</a> 。 )
	<a href="#">GetInt32</a>	获取指定列的 32 位有符号整数形式的值。 ( 重写 <a href="#">DbDataReader.GetInt32(Int32)</a> 。 )
	<a href="#">GetInt64</a>	获取指定列的 64 位有符号整数形式的值。 ( 重写 <a href="#">DbDataReader.GetInt64(Int32)</a> 。 )
	<a href="#">GetLifetimeService</a>	检索控制此实例的生存期策略的当前生存期服务对象。 ( 继承自 <a href="#">MarshalByRefObject</a> 。 )
	<a href="#">GetName</a>	获取指定列的名称。 ( 重写 <a href="#">DbDataReader.GetName(Int32)</a> 。 )

	GetOrdinal	在给定列名称的情况下获取列序号。（重写 <a href="#">DbDataReader.GetOrdinal(String)</a> 。）
	GetProviderSpecificFieldType	返回指定列的提供程序特定的字段类型。（继承自 <a href="#">DbDataReader</a> 。）
	GetProviderSpecificValue	获取指定列的作为 <a href="#">Object</a> 的实例的值。（继承自 <a href="#">DbDataReader</a> 。）
	GetProviderSpecificValues	获取集合中当前行的所有提供程序特定的特性列。（继承自 <a href="#">DbDataReader</a> 。）
	GetSchemaTable	返回一个 <a href="#">DataTable</a> ，它描述 <a href="#">OleDbDataReader</a> 的列元数据。（重写 <a href="#">DbDataReader.GetSchemaTable()</a> 。）
	GetString	获取指定列的字符串形式的值。（重写 <a href="#">DbDataReader.GetString(Int32)</a> 。）
	GetTimeSpan	获取指定列的 <a href="#">TimeSpan</a> 对象形式的值。
	GetType	获取当前实例的 <a href="#">Type</a> 。（继承自 <a href="#">Object</a> 。）
	GetValue	获取以本机格式表示的指定序号处的列的值。（重写 <a href="#">DbDataReader.GetValue(Int32)</a> 。）
	GetValues	使用当前行的列值来填充对象数组。（重写 <a href="#">DbDataReader.GetValues(Object[])</a> 。）
	InitializeLifetimeService	获取控制此实例的生存期策略的生存期服务对象。（继承自 <a href="#">MarshalByRefObject</a> 。）
	IsDBNull	获取一个值，该值指示列中是否包含不存在的或已丢失的值。（重写 <a href="#">DbDataReader.IsDBNull(Int32)</a> 。）
	MemberwiseClone()	创建当前 <a href="#">Object</a> 的浅表副本。（继承自 <a href="#">Object</a> 。）
	MemberwiseClone(Boolean)	创建当前 <a href="#">MarshalByRefObject</a> 对象的浅表副本。（继承自 <a href="#">MarshalByRefObject</a> 。）
	NextResult	当读取批处理 SQL 语句的结果时，使数据读取器前进到下一个结果。（重写 <a href="#">DbDataReader.NextResult()</a> 。）
	Read	使 <a href="#">OleDbDataReader</a> 前进到下一条记录。（重写 <a href="#">DbDataReader.Read()</a> 。）
	ToString	返回表示当前对象的字符串。（继承自 <a href="#">Object</a> 。）

[页首](#)

## 显式接口实现

	名称	说明
	<code>IDataRecord.GetData</code>	有关此成员的说明，请参见 <a href="#">IDataRecord.GetData</a> 。（继承自 <a href="#">DbDataReader</a> 。）

[页首](#)

## 备注

若要创建 `OleDbDataReader`，必须调用 `OleDbCommand` 对象的 `ExecuteReader` 方法，而不能直接使用构造函数。

在关闭 `OleDbConnection` 之前，请先关闭 `OleDbDataReader` 对象。如果计划重新使用 `OleDbCommand` 对象，那么还必须关闭 `OleDbDataReader` 对象。例如，调用 `Close` 之前不能检索输出参数。

`OleDbDataReader` 的用户可能会看到在读取数据时另一进程或线程对结果集所做的更改。但是，确切的行为与执行时间有关。

当 `OleDbDataReader` 关闭后，只能调用 `IsClosed` 和 `RecordsAffected` 属性。尽管当 `OleDbDataReader` 存在时可以访问 `RecordsAffected` 属性，但是，请在返回 `RecordsAffected` 的值之前始终调用 `Close`，以确保返回精确的值。

## 示例

下面的示例创建一个 `OleDbConnection`、一个 `OleDbCommand` 和一个 `OleDbDataReader`。该示例读取全部数据，并将这些数据写到控制台。最后，该示例先关闭 `OleDbDataReader`，然后关闭 `OleDbConnection`。

C#

```
public static void ReadData(string connectionString, string queryString)
{
    using (OleDbConnection connection = new OleDbConnection(connectionString))
    {
        OleDbCommand command = new OleDbCommand(queryString, connection);

        connection.Open();
        OleDbDataReader reader = command.ExecuteReader();

        while (reader.Read())
        {
            Console.WriteLine(reader[0].ToString());
        }
        reader.Close();
    }
}
```

## 版本信息

.NET Framework

受以下版本支持：4、3.5、3.0、2.0、1.1、1.0

.NET Framework Client Profile

受以下版本支持：4、3.5 SP1

## 平台

Windows 7, Windows Vista SP1 或更高版本, Windows XP SP3, Windows XP SP2 x64 Edition, Windows Server 2008 ( 不支持服务器核心 ), Windows Server 2008 R2 ( 支持 SP1 或更高版本的服务器核心 ), Windows Server 2003 SP2

.NET Framework 并不是对每个平台的所有版本都提供支持。有关支持的版本的列表，请参见[.NET Framework 系统要求](#)。

## 线程安全

此类型的任何公共 **static** ( 在 Visual Basic 中为 **Shared** ) 成员都是线程安全的。但不保证所有实例成员都是线程安全的。

## 请参见

参考

[System.Data.OleDb 命名空间](#)

其他资源

[Working with DataReaders](#)

## 社区附加资源

此页面有用吗？

是

否

