

Making ADO.NET Easy to Use



Paul D. Sheriff

BUSINESS SOLUTIONS ARCHITECT, FAIRWAY TECHNOLOGIES, INC.

www.fairwaytech.com psheriff@fairwaytech.com



Module Goals



Introduce ADO.NET Wrapper

Simplify ADO.NET coding

Show wrapper class samples

- Retrieval of data
- Data modification
- Multiple result sets
- Transactions

ADO.NET Wrapper Classes



ADO.NET Wrapper Classes

**Wrapper around
ADO.NET**

**A lot less code to
write**

**Performance is
great**

**No need to learn
other tools or
configuration**

**Easy to work with
any .NET provider**



ADO.NET Wrapper Classes

**Dynamic SQL or
stored procedures**

**Create List<T>
with a single line
of code**

**Can use [Column]
attribute**

**Validation using
Data Annotations
and/or custom code**

**Transaction
support**



Blog Posts for More Information

**Simplify ADO.NET Using
Generic Methods and
Reflection**

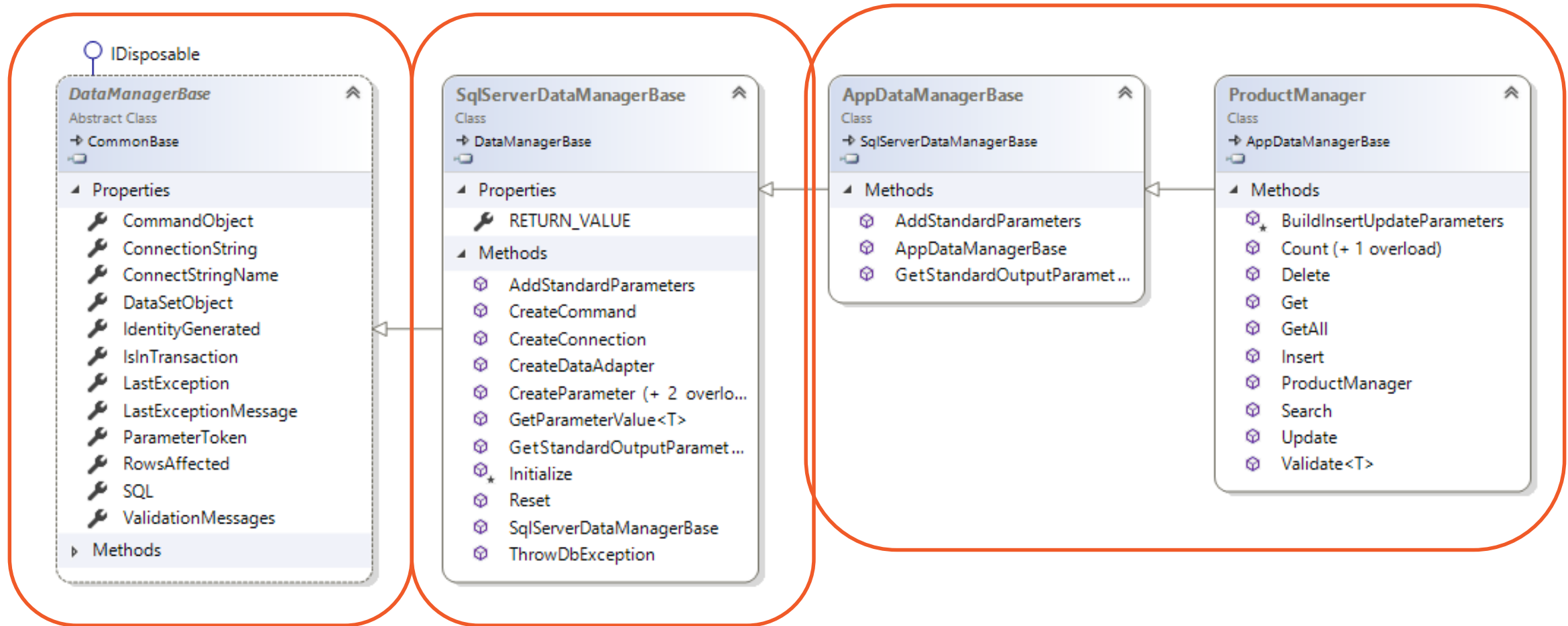
<https://bit.ly/2MrIBRI>

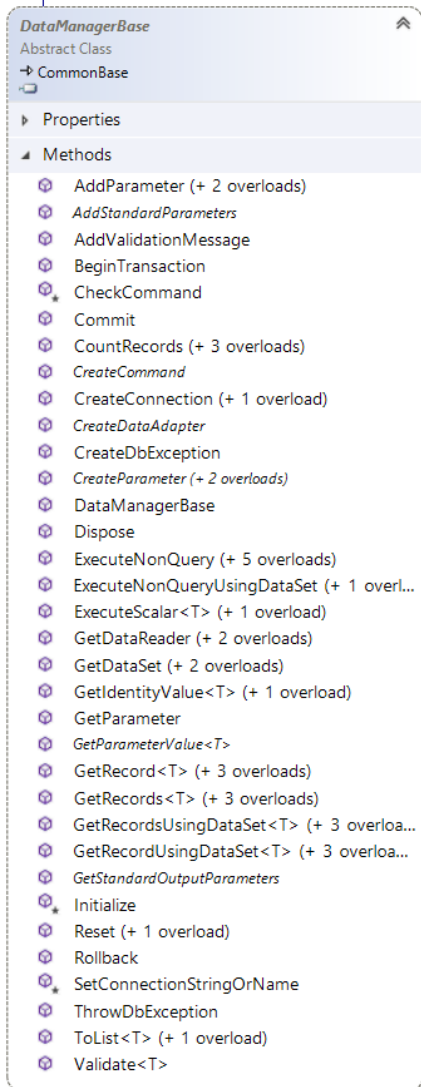
**ADO.NET Data Wrapper
Classes**

<https://bit.ly/2MTC7sh>



Classes in ADO.NET Wrapper

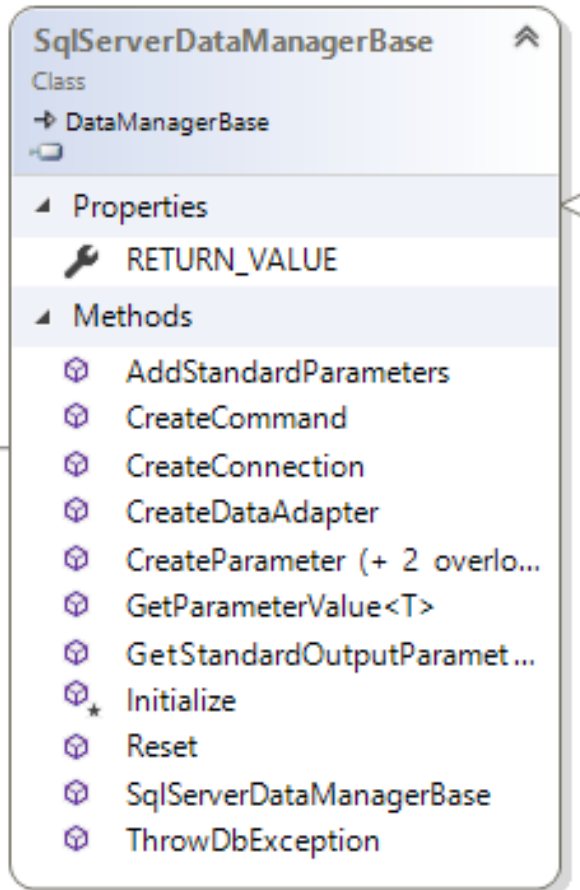




DataManagerBase class

- Wraps up ADO.NET calls
- Uses ADO.NET Interfaces
- Disposable
- Methods to...
 - Get Record(s)
 - Execute action statements
 - Execute scalar functions
 - Validation using Data Annotations
 - Return last IDENTITY value





SqlServerDataManagerBase class

- Inherits `DataManagerBase`
- To interact with SQL Server
- Uses ADO.NET Sql* classes
- Overrides methods in `DataManagerBase`
 - `CreateCommand`
 - `CreateConnection`
 - `CreateDataAdapter`
 - `CreateParameter`

Retrieval of Data



Get All Rows

```
public partial class ProductManager : AppDataManagerBase
{
    public List<Product> GetAll()
    {
        // Submit SQL to get rows from a table
        return GetRecords<Product>("SELECT * FROM SalesLT.Product");
    }
}
```



Get A Single Row

```
public virtual Product Get(int productId)
{
    SQL = "SELECT * FROM SalesLT.Product WHERE ProductID = @ProductID";

    var parameters = new List<IDbDataParameter> {
        CreateParameter("ProductID", (object)productId, false)
    };

    return GetRecord<Product>(SQL, parameters.ToArray());
}
```



Demo



Retrieve data using dynamic SQL



Demo



Using parameters

Search for products based on criteria



Demo



Retrieve data using stored procedure



Demo



Multiple result sets



Data Modification



Demo



Insert data



Transactions



Demo



Transactions



Summary



Wrapper class = less code to write

Very fast data access method

Create your entity classes

Create your manager classes

Create base classes for...

- Oracle, Progress, SQL Lite, etc.

Course Summary



ADO.NET is easy to use

Only open connection when needed

Dispose of objects when done

Use a data reader for best performance

Create custom exception class

Use wrapper class to simplify



I hope you enjoyed
this course!



Paul D. Sheriff

Business Solutions Architect, Fairway Technologies, Inc.

www.fairwaytech.com

psheriff@fairwaytech.com

