# Handling ADO.NET Exceptions



Paul D. Sheriff
BUSINESS SOLUTIONS ARCHITECT, FAIRWAY TECHNOLOGIES, INC.
www.fairwaytech.com psheriff@fairwaytech.com



#### Module Goals



**View ADO.NET exceptions** 

**Catch specific exceptions** 

**Gather ADO.NET specific information** 



### Exception Handling

```
using (SqlConnection cnn = new SqlConnection("Server=BadServerName")) {
    using (SqlCommand cmd = new SqlCommand("INSERT INTO ...", cnn)) {
      cnn.Open();
      RowsAffected = cmd.ExecuteNonQuery();
catch (Exception ex) {
  ResultText = ex.ToString();
```



Simple exception handling



#### Problems

Losing important info from SQL Server

No access to Command object



### Exception Handling

```
trv {
  // Open connection, submit command, etc.
catch (SqlException ex)
  OtringDuilder sb = new StringBuilder();
  for (int i = 0; i < ex.Errors Count; i++) {</pre>
    sb.AppendLine("Index #.
                                  + 1.103L1111g()),
                                    ex.Errors[i].GetType().FullName);
    sb.AppendLine("Type: "
                                    ex.Errors[i].Message);
    sb.AppendLine("Message: "
    sb.AppendLine("Source: "
                                    ex.Errors[i].Source);
    sb.AppendLine("Number: "
                                    ex.Errors[i].Number.ToString());
    sb.AppendLine("State: "
                                    ex.Errors[i].State.ToString());
    sb.AppendLine("Class: "
                                    ex.Errors[i].Class.ToString());
    sb.AppendLine("Server: "
                                    ex.Errors[i].Server);
    sb.AppendLine("Procedure: "
                                    ex.Errors[i].Procedure);
    sb.AppendLine("LineNumber: "
                                    ex.Errors[i].LineNumber.ToString());
  ResultText = sb.ToString() + Environment.NewLine + ex.ToString();
```



**Catch SqlException** 



## SqlExceptionManager Class

Singleton

Publish(ex, cmd);

Creates a custom exception object



#### SqlServerDataException Class

**SQL** statement

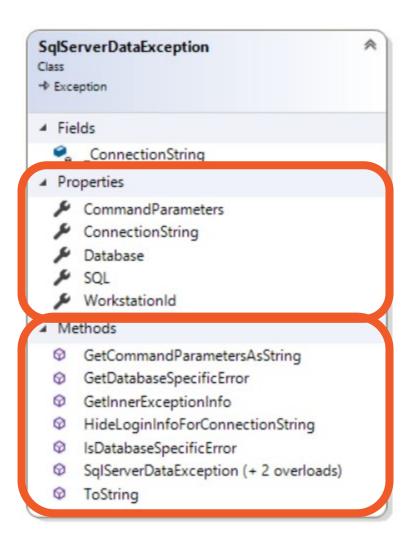
**Command parameters** 

Connection string (minus UserID and Password)

**SQL** Server errors



## SqlServerDataException Class





#### Exception Handling





Gather additional exception information





Walk through of exception code



## Summary



Declare command object outside try
Call exception manager publisher
Gather as much info as you can
Publish the exception

- Using log4net, Microsoft Logging Extensions, etc.





Coming up in the next module...

DataTables and DataViews
Sorting and Filtering Data
Multiple result sets

