# IDisposable Best Practices for C# Developers

**Module 1 Setup Instructions**

In Module 1 we demonstrate the problems you have in .NET apps if you don’t use IDisposable correctly.

## LINQPad Queries

There is a simple LINQPad script which connects to a local SQL Server instance to show how the connection pool can become exhausted if objects are not disposed.

## Sixeyed.Disposable

This is the demo solution for the course. In this module we walk through the code and show the problems it has because IDisposable is not used correctly.

### Pre-requisites

The demo solution is delivered in Visual Studio 2013.

The solution uses NuGet package restore to load packages during the build.

To enable this in Visual Studio, open *Tools…Library Package Manager…Package Manager Settings* and ensure both options (*Allow NuGet to download missing packages* and *Automatically check for missing packages during build in Visual Studio*) are ticked.

The database project has a publish profile you can use if you have a local SQL Server database with a default, unnamed instance – just build the solution and publish using the local profile.

### Running the Solution

Run the console app and it will start listening for files dropped in **c:\books**. In the *Books\A Tale of Two Cities*  folder in the solution you will find the sample .txt files used in the demos.

Depending on your hardware, you should be able to drop *contents.txt* and *excerpt.txt* and they will run correctly, but the full file *pg98.txt* will exhaust the SQL connection pool and the app will error.