

# .NET Connection Pooling

Applications using Foundations.NET rely on ADO.NET for accessing the database, thus all connections pass through the ADO.NET which uses an optimization technique called "connection pooling", which minimizes the cost of repeatedly opening and closing connections. Connection pooling is handled differently for the .NET Framework data providers. **Details on how the pooling actually works (eg: how the pool grows/shrinks, how and when are connections released, etc...) is best detailed on Microsoft's .NET documentation.** Some links/resources are listed below.

Relevant .NET resources regarding Connection Pooling:

- ADO.NET Connection Pooling <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/connection-pooling>
- SQL Server Connection Pooling: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql-server-connection-pooling>
- Foundations .NET API documentation regarding pooling configurations: <http://docs.morphis-tech.com/foundations/dotnet/api/Foundations.Core.AppDataLayer.Configuration.PoolingSettings.html>

Since one of the features provided by Foundations, is database provider independence (meaning that your application could be prepared to different providers, without any code changes), Foundations itself is responsible for initializing the connection string. This is relevant, because any connection pool configurations needs to be performed through the connection string. This means, that Foundations also exposes a way of configuring the connection pool. Details about all the pooling settings available can be inspected in the link above to the Foundations .NET API documentation. Here's a configuration example which initializes the pooling with a single connection and instructs the pool to grow by 5 connections whenever needed:

```
<appDataLayerSettings
dbFactory="Foundations.Core.AppDataLayer.Providers.ODP.Managed.ODPDataBase
Factory, Foundations.Core.AppDataLayer.Providers.ODP.Managed">
  <connectionSettings ... />
  <poolingSettings minSize="1" incrSize="5" maxSize="100" />
</appDataLayerSettings>
```

To clarify usage of this pool, we've used performance counters to create a couple examples/scenarios. These were obtained using Oracle's ODP.NET Managed provider, but equally apply to other providers, although the performance counters may be different.

In these scenarios, opening new sessions/connections was achieved by opening a new browser session and logging. **Different browsers (ex: IE, Edge, Chrome and Firefox) were used** to guarantee that the browsers weren't sharing cookies and asp.net sessions.

## Scenario 1 - Sessions consumed from pool

In this scenario, the default pooling settings were used.



## Scenario 2 - Sessions consumed from pool causing the pool to grow

In this scenario, the pooling was configured to have a minimum of 1 connection and grow 5 connections at a time (as shown in the configuration example above).

