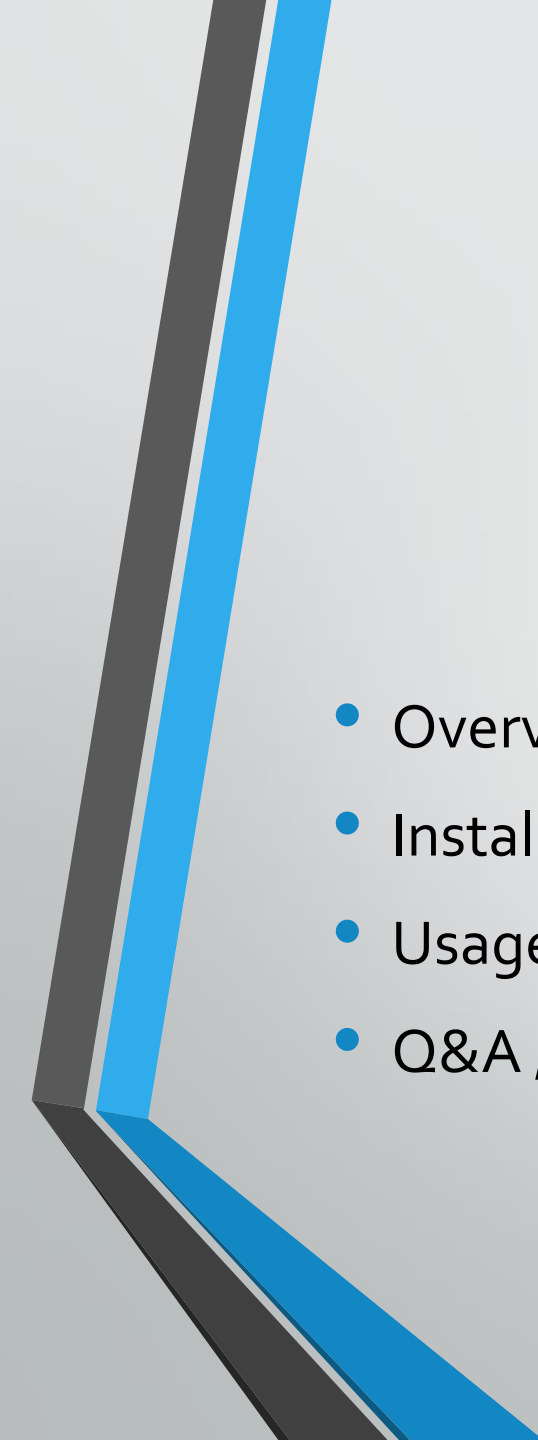




Dapper.NET

A Micro-ORM for the Discerning Developer



Agenda

- Overview of Micro ORMs
- Installation / Setup
- Usage
- Q&A / Wrap-up

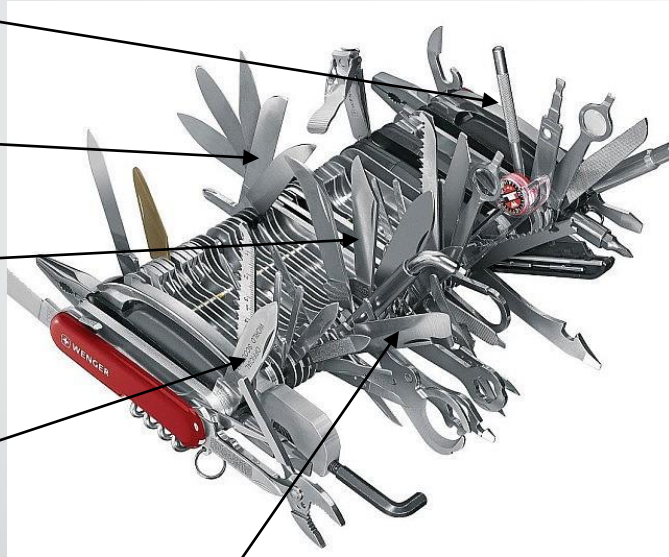
Micro-ORM

- NOT a colloquialism for a Microsoft ORM

Micro-ORM

- NOT a colloquialism for Microsoft's ORM
- Intentionally feature sparse

Traditional ORM (EF/NHibernate)



Caching

Lazy Loading

Validation

DB Migration
& Generation

Object Mapping

Micro-ORM



Object
Mapping

Micro-ORM

- NOT a colloquialism for Microsoft's ORM
- Intentionally feature sparse
- Typically geared toward high performance

Method	Duration	Method	Duration
Hand coded (using a <code>SqlDataReader</code>)	47ms	Linq 2 SQL CompiledQuery	81ms
Dapper <code>ExecuteMapperQuery</code>	49ms	NHibernate HQL	118ms
ServiceStack.OrmLite (QueryById)	50ms	Linq 2 SQL	559ms
PetaPoco	52ms	Entity framework	859ms
BLToolkit	80ms	SubSonic ActiveRecord.SingleOrDefault	3619ms
SubSonic CodingHorror	107ms		
NHibernate SQL	104ms		
Linq 2 SQL <code>ExecuteQuery</code>	181ms		
Entity framework <code>ExecuteStoreQuery</code>	631ms		

Full Disclosure: These are the benchmarks the dapper project produces/publishes. However, they are roughly in line with others (see resources)



Installation and Usage

(Code Demo)



Questions?

Resources

- <https://github.com/StackExchange/dapper-dot-net>
- <https://github.com/FransBouma/RawDataAccessBencher>
- <http://www.toptensoftware.com/petapoco/>