



Using MySQL with Visual Studio 2010

**REGGIE BURNETT**

# Agenda

- Review of 6.2
- What's coming in 6.3
- DSP “sneak peek”
- Product Roadmap

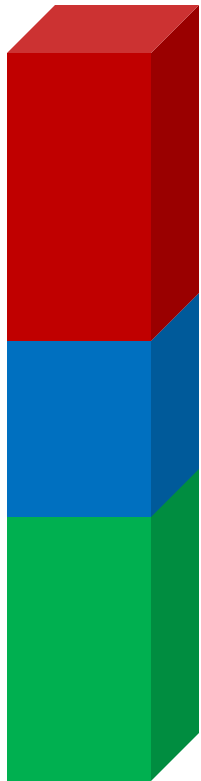
## My Goal

Make the decision between SQL Server and MySQL only a financial one.

## New in 6.2

- Connection pool cleanup timer
- New command timeout implementation
- TableDirect command type
- Support for SSL client certificates
- Completely new logging output

# Connection pool cleanup timer



- Stale connections (idle longer than 3 min) closed by sweeper thread that runs every 3 min. The idle pool never falls below min pool size.
- Idle connections that are not stale ( $< 3$  min) – not touched
- In use connections – not touched until you (or Dispose) close them.

***This is currently not configurable***

# Better CommandTimeout

## Behavior prior to 6.2

- Timing only occurred in ExecuteXX
- Used “wall” time

```
MySqlCommand cmd = new MySqlCommand("SELECT *  
FROM t", c);  
using (MySqlDataReader reader = cmd.ExecuteReader())  
{  
    while (reader.Read())  
    {  
        reader.GetString(1);  
    }  
}
```

This could throw a  
TimeoutException.

This never would!!  
(which means it could  
throw a general  
disconnect exception.

# Better CommandTimeout

## Behavior new to 6.2

- Timing occurs at the network level
- Uses network i/o time. (works with shared mem and named pipes)
- Does not use a worker pool thread to time the query

```
MySqlCommand cmd = new MySqlCommand("SELECT *  
FROM t", c);  
using (MySqlDataReader reader = cmd.ExecuteReader())  
{  
    while (reader.Read())  
    {  
        reader.GetString(1);  
    }  
}
```

This could throw a  
TimeoutException.

And this could too!!

# TableDirect

- Not much to see here
- Basically amounts to a `SELECT *` on the given table
- Not very exciting right?
- Well... what could we do with a single table name that is easy to parse and understand?
  - How about client side table caching?
  - Slow moving tables could be cached.
  - No need to parse the command text
  - Also – what would memcached or velocity add to this picture?
- Not promising we'll do this. Just thinking....



# SSL Client certificates

- Builds on the SSL enhancements found in 6.1
- You can use file-based or store-based certificates
- Connector uses PFX style certificates
  - Must convert from PEM format
  - OpenSSL can do this
- A certificate “thumbprint” is also supported

# New logging output

- Implemented to support Enterprise Monitor 2.2
- Old format had problems
  - Didn't allow you to match log entries with a query
  - Didn't provide an easy way to consume log output
  - Used .NET 1.0 Trace class
- New format
  - Uses TraceSource with “mysql” as the source name
  - All log output is a defined format
  - All log entries are tied to a given thread id/query id
  - Consume log data by overriding TraceEvent in your listener

# New logging output (contd.)

TraceEvent( TraceEventCache eventCache, string source,  
TraceEventType eventType, int id, string format, params Object[] args )

eventCache	Contains process id, thread id, timestamp, callstack, etc
Source	Always “mysql” if the event is from us
eventType	Generic event type – Information, Warning, Error, etc
Id	MySQL event type (see next slide)
Format	A format string that is used for text output
Args	Array of data items that are plugged into the format string. This is your real data!!!

# New logging output (contd.)

Some event and data examples (there are many others)

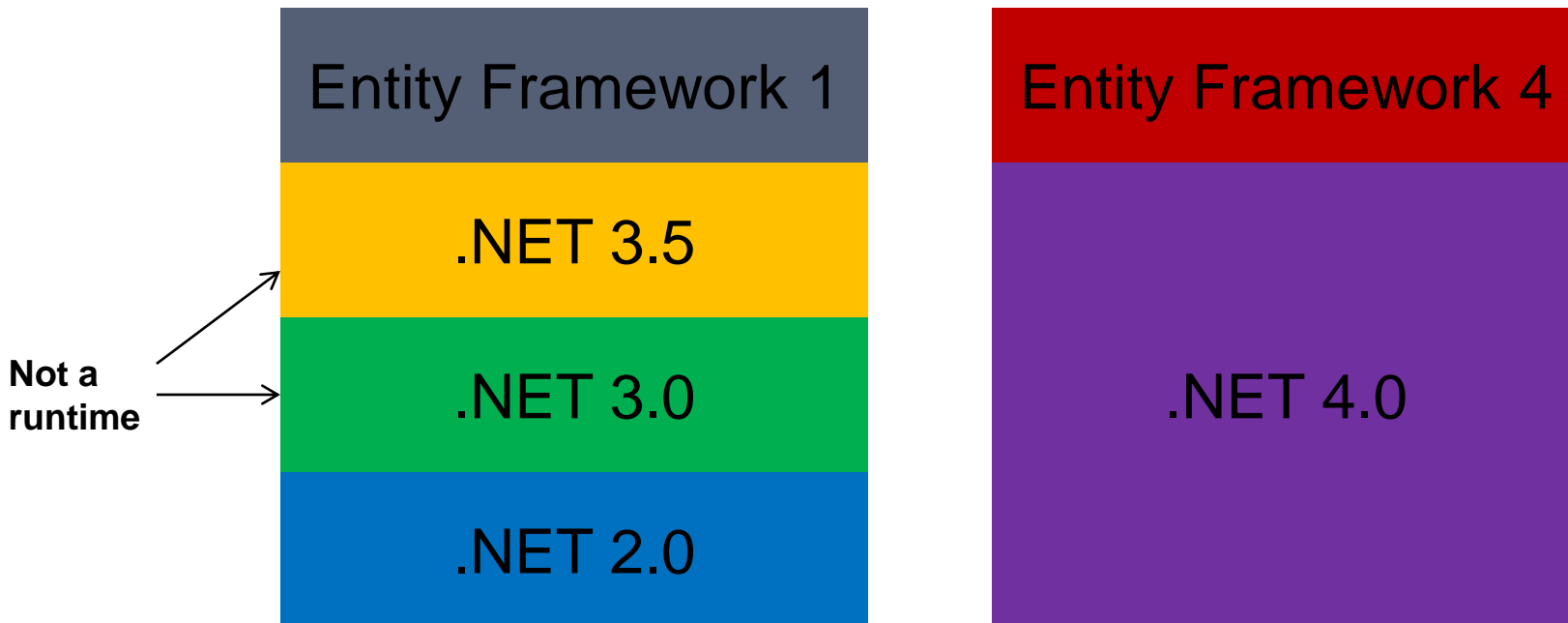
Event	Data
QueryOpened	driver id, thread id, command text
GetResult	driver id, field count, affected rows, inserted id
NextResult	driver id, total rows, skipped rows, row size
PrepareStatment	driver id, command text, statement id

## Coming in 6.3

- Visual Studio 2010 support!!
  - Entity Framework 4 support
  - Design-time, model-first support
  - POCO, code-first support
- New script editor
- SQL Server Mode

# Entity Framework 4

Why the version jump to 4?



Side by side execution is supported

# Entity Framework 4 – Model First



EF 1 →

← EF 4



- Model First is fully extensible allowing you to control every step of your DDL generation

# Entity Framework 4 – Code Only

```
class Person
{
    int id;
    string name;
    int age;
}
```





# Some Things We Are Thinking About

- Replication-aware Connections
- Client-side Table Caching
- Database Server Provider (DSP)
- LinQ Provider
- Stored Procedure Debugging
- Embedded Server Support

# Roadmap

- Connector/Net 6.2.3 available today
- 6.3 Alpha 2 available today
- 6.3 Beta before the end of April
- 6.3 RC mid May
- 6.3 GA end of May/first June
- DSP -- TBD

# How To Reach Me

- reggie.burnett@oracle.com
- [www.twitter.com/reggieburnett](http://www.twitter.com/reggieburnett)
- [www.reggieburnett.com](http://www.reggieburnett.com)