Using MySql 4.1 and 5.0 with Connector/Net

Reggie Burnett
Software Developer
reggie@mysql.com

MySQL Users Conference, 2005
April 18-21, 2005
Agenda

• Thank you and I’m sorry!!
• History and Goals
• Use of the new features from the connector
• Discussion of common support issues
• Roadmap for future versions
History and Goals

• History
  – Original ByteFX provider
  – Acquired in April 2004

• Goals
  – No one should want or need to use a different .NET connector!!
.NET is the best!!

• Superior runtime
• Superior language support
• Superior tool support
• Largest developer community
• Superior database provider!!
So What’s New?

• 4.1
  – Prepared statements
  – Better internationalization
  – Subselects

• 5.0
  – Stored procedures (yeah baby!!)
  – Views
Prepared Statements

• What is a prepared statement?
  – Server preprocessed statement

• Why use a prepared statement?
  – More efficient
  – Faster!!
Using Prepared Statements

• Call the MySqlCommand.Prepare() method
  – Setting CommandText resets this
• Process results of the command normally
  – Totally transparent
  – Emulated on pre-4.1 environments
They Are For Parameters

- Assign your parameters to the command before the prepare!
- If your parameter has a specific size, it is important that you set it before the prepare.
They Are Faster!

- PerfSample shows 20-30% throughput improvement
  - Test performed was a simple insert of 50,000 rows
  - Each row contained 3 integer values
  - Test was performed with client and server on the same computer
You Can’t Prepare Everything

- SHOW commands can’t be prepared
- Neither can CALL commands
  - Stored procedures
  - Stored functions
Example

MySqlCommand cmd = new MySqlCommand("INSERT INTO test VALUES (?p1, ?p2)", conn);

cmd.Parameters.Add("?p1", MySqlDbType.Int32);

cmd.Parameters.Add("?p2", MySqlDbType.Int16);

cmd.Prepare();

cmd.Parameters[0].Value = 2;

cmd.Parameters[1].Value = 4;

cmd.ExecuteNonQuery();
Internationalization

- Specify a character set down to the field level
- Specify character sets for submission of queries and returning results
- 4.0 only had a single character set for an entire database!
- Use parameters to aid with localization
Keys to Remember

• All strings in .NET are Unicode
• Use the charset=xxx option
• Make sure you have the proper language packs available
• Double check your default character sets!
• .NET != ODBC
Working with Queries

- GetString and GetValue will return string values using the server’s charset.
- Queries are sent to MySQL using only one charset.
- Multilanguage systems should use “charset=utf8”
Stored Procedures

• They are here and Connector/Net supports them!!
• In, out, and inout parameters
• Multiple resultsets
Creating a Stored Procedure

- Use normal CREATE PROCEDURE statement with ExecuteNonQuery
- You may have to set a delimiter
  – If you are not allowing batch statements
- Server Explorer includes a stored procedure editor
Example

```csharp
string sql = "CREATE PROCEDURE spTest(in p1 int, out p2 int) BEGIN 
    SET p2=p1+33; SELECT * FROM mysql.db; 
    SELECT * FROM mysql.user; END";
MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.ExecuteNonQuery();
```
Executing a Stored Procedure

• Use any of the ExecuteXXX methods
• Use MySqlParameter objects
  – Don’t forget the ? syntax
• When using ExecuteReader, the values of OUT and INOUT parameters will not be known until the reader is closed
  – Extra processing is required
Example

```csharp
MySqlCommand cmd = new MySqlCommand("spTest", conn);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add("?p1", 66);
    MySqlDataReader reader = cmd.ExecuteReader();
    reader.Read();
    reader.NextResults();
    reader.Read();
    reader.Close();
    Int v = cmd.Parameters[1].Value; // should be 99
```
Views

- Views behave exactly like tables
- Metadata retrieved by GetSchemaTable will reflect the different table sources
- Server Explorer will include a tool for editing views
Where To Get Support

- **Online Forums**
  - http://forums.mysql.com

- **Mailing Lists**
  - http://lists.mysql.com
  - http://discuss.develop.com

- **Newsgroups**
  - comp.database.*
My Connection Won’t Close

- Pooling is enabled by default!
- Disable with “pooling=no”
- Close all connections
- Connections will end when app ends
Understanding Pooling

- Tied to a connection string
- Can be optimized
  - MinPoolSize
  - MaxPoolSize
- Pool cleaning happens on each open
- 1.1 will add pool flushing.
MySqlDateTime

• Why is it necessary?
  – MySql supports date values that .NET cannot handle
    • 0000-00-00 is an example
  – Rounding doesn’t always work
  – Some databases use a zero date value
What Should You Do?

- Don’t use illegal date values in your database!!!!!
- If needed, “allow zero datetime=yes” will enable support
- GetValue will return MySqlDateTime or DateTime
- MySqlDateTime can cast to DateTime
Road To Version 1.1

- SSL support
- Embedded/compact support
- Integration into Visual Studio & Whidbey
- Async query support
- Much more!
- Summer ‘05
Road to 2.0

• Type specific objects (geometry, GIS, blob)
• Clustering & replication support
• Support for our new XA work
• Improve our integrations
• Winter ’05/Spring ‘06
Thank You!