



# Evolution of an ORM

A view into the more esoteric parts of .NET

Walden Leverich (WaldenL@techsoftinc.com)  
Matt Whelan (MattW@techsoftinc.com)

Tech Software (www.techsoftinc.com)

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Overview



- What's an ORM
- Why write our own?
- Evolution in 5 easy steps
- Attributes
- Reflection
- Dynamic compilation and dynamic IL
- Plugins
- Other ORM related stuff

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Object Relational Mapping



- Data is stored in databases, relational databases
- Modern programming is done w/objects
- Impedance mismatch
- Mapping code is repetitive
- Best practices not typically followed
- Core component, should be best programmers
  - But it's boring code!
- NHibernate, SubSonic, Telerik's OpenAccess...

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Why write our own?



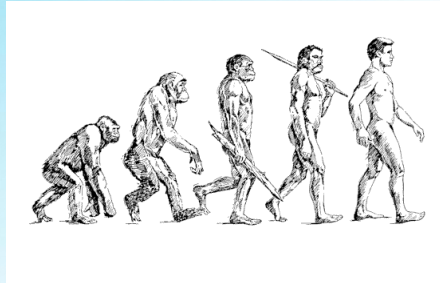
- We were drunk?
- DB is the root of all systems – get it right!
- Started our 5 years ago – it's evolved
- Don't like XML!
- Not intended to be everything to everyone
  - Firm believer in 80/20 rule
- If it's ours we can do whatever we want with it
- Custom solution to custom problem
- Session driven connection strings
  - Different database based on who is logged into system

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Evolution of an ORM



- Inline SQL
- AppDataHelper
- Object-based hand coded
- Reflection based ORM
- IL generation based ORM
- The future?



© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Inline SQL



- .NET Programming 101
- Connection, Command, DataReader, etc.
- SQL Code directly in UI layer
- Tightly bound to db implementation
- No SQL error checking until runtime
  - Slect, updat, delete form, etc.
- Not all bad
  - Full power of dynamic SQL at your fingertips
  - Complete flexibility to do whatever you want
  - Most familiar to programmers from legacy environments

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## We've all done it!



```
SqlConnection cn = new SqlConnection(
    "connection-string");

SqlCommand cm = cn.CreateCommand();
cm.CommandText = "Select ID, Name from customer
    where status = 'A'";

SqlDataReader dr = cm.ExecuteReader();

while (dr.Read())
{
    DoSomething(dr["ID"], dr["Name"]);
}

dr.Close();
cn.Close();
```

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

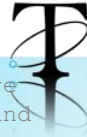
## AppDataHelper



- This code is rather redundant
- Isolate connect query process results
- Start to implement best practices
- Microsoft's Patterns and Practices group
- Static methods that take SQL return rows

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## This *is* an improvement!



```
string sql = "select top 1 * from xFormInstanceStage where
FormInstanceGUID = @Instance and FormStageGUID = @Stage and
IsDeleted='N' Order By EnteredAt desc";
```

```
DataSet ds = IRBData.ExecuteDataset(
    CommandType.Text,
    sql,
    new SqlParameter("@Instance", FormInstance),
    new SqlParameter("@Stage", FormStage)
);

if (1 != ds.Tables[0].Rows.Count)
    throw new ApplicationException(
        string.Format("Retrieved {0} rows for instance stage
using instance {1} and stage {2}",
            ds.Tables[0].Rows.Count, FormInstance, FormStage)
    );

DataRow dr = ds.Tables[0].Rows[0];
```

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Object Based Hand Coded



- UI layer no longer contains SQL (much)
- Business layer maps between objects and DB
- All data access hand coded in business layer
  - May use AppDataHelper concept
- Setting a property may directly update DB
  - Or there may be a .Save() method
- UI isolated from DB changes



LoanCreditCard.cs

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Land Ho! True ORM spotted

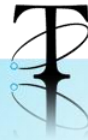


- Called “BusinessObjects”
  - Should have been DBObjects?
- No more (ok, very little) SQL in business objects
- Core code that handles DB IO
  - Able to apply best practices and performance optimizations
- Base class for all business objects
  - Running argument between base-type ORMs and POCO/Interface ORMs
- Use Attributes to supply DB metadata to runtime
- Rough parallel between objects and tables

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>



## Attributes and Metadata



- Metadata is data about data
- Native metadata, user metadata
- Compilers write metadata into IL
- Attributes record user metadata about
  - Fields, Methods, Classes, Properties, Etc.
- C# attributes in [], VB attributes in <>
- [Serializable], [ComVisible], [MarshalAs], etc.

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Attribute is Just Another Class



- Standard says you end with the word Attribute
- Don't need to specify Attribute in usage
- [Serializable] == [SerializableAttribute]
- Define your own inherit from System.Attribute
- Attributes have attributes!
  - AttributeUsageAttribute



Note.cs



DatabaseFieldAttribute.cs



DatabaseTableAttribute.cs

© 2009 Tech Software Inc.  
http://www.techsoftinc.com

## Cool System Attributes



- Obsolete
  - Marks member, or entire class as no longer usable with option of making use a compile warning or error
  - Runtime usages won't blow up; provide indication of replacement member/class
- ComponentModel.EditorBrowsable
  - Hides member from the object browser
  - Nice for obsolete members!
- DebuggerDisplay
  - Controls what property(ies) VS shows in debug window
  - [DebuggerDisplay("{FieldName} - {DBType}")]

© 2009 Tech Software Inc.  
http://www.techsoftinc.com

## Reflection – Not just in the mirror!



- How you “read” metadata at runtime
- Also called introspection
- Core component of .NET runtime
- System.Reflection namespace
- Slow! (relatively speaking)
- xxxInfo classes
  - FieldInfo, MethodInfo, ClassInfo, etc.
- Concept of protected/private is out the window

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Reflector!



- You *must have* reflector!
- Originally by Lutz Roeder, purchased by RedGate
- Friendly version of ILDASM with so much more
- Plugins to do all sorts of cool things w/.DLLs
  - Show me IL in c#, generate c# to emit this IL



Reflector.exe

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## FieldInfo Class



- Contains Info about a Field, duh!
- Type of field, where declared, if it's a literal, if it's private, etc.
- Methods to get it's value, set it's value, get it's attributes, etc.
- Retrieved from the Type object returned by `typeof()` or `GetType()`
- `type.GetFieldInfo(<FieldName>, Flags)`

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Setting fields with reflection



```
Customer c = new Customer;
c.Name = "George";

FieldInfo fi = typeof(Customer)
    .GetField("Name");
fi.SetValue(c, "George");

Customer c = new Customer;
c.PrivateName = "Linda"; //Compiler boom! Can't set privates

FieldInfo fi = typeof(Customer)
    .GetField("PrivateName", BindingFlags.NonPublic);
fi.SetValue(c, "Linda");
```

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Properties in BusinessObjects



- Properties are simple, yes?

```
private string _name;
public string Name
{
    set { _name = value; }
    get { return _name; }
}
```

- Our setter isn't that simple though.
  - We need to indicate that this object has changed
  - We only want to actually update the field if the value really changed
  - Support of auto-trim for strings
  - Length checking at set time
  - We could code all this in each setter – but why?

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## FieldInfo and SetValue to Rescue



- Wrote our own SetField() function
  - Takes two parms (field and value) and does a bunch of work
- Our properties are simple

```
public string Description {
    get { EnsureLoaded(); return _Description; }
    set { SetField("_Description", value); }
}
```

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Generate SQL



- Attributes tell us the column names for a field, their types, and key-ness
- ChangeList tells us what fields have changed
- We can generate the appropriate SQL for crud operations
- Performance optimizations
  - Sequential Access and SingleRow command behaviors
- OUTPUT clause on insert statement
  - Read back data from insert, issue w/triggers; must use table variable to get correct results

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Reflection works, but it's slow



- Relative statement
- Some caching options (cache field info, type info)
- Can't cache away actual SetValue call
- But we can replace it!
- Could write c# on fly and call csc (asp.net model)
- Could supply IL and in-memory assembly
- We chose to create Dynamic Methods

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Dynamic Method IL Generation



- Reflection class adds methods to type at runtime
- DynamicMethod constructor
  - NameOfNewMethod, ReturnType, array of parameter types, and owning-type (where to put this method)
- Get an ILGenerator object from the DynamicMethod object
- ILGenerator has an emit method that writes IL opcodes
- Once you've emitted IL you can create a delegate for this new method you just created

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Plugins



- What would you plug in?
- OID type
  - Combination of Guid and int slammed into 20 bytes
  - First 16 is guid, last 4 identify type of object
- TypeTranslator handles mapping of OID type to actual implementation
- Translator is any class you want, that implements the ITypeTranslator interface
- Why here?
  - OIDs are cool, but plugin loader uses... reflection.

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Plugin Loader



- Walks over list of DLLs in app directory
- Loads each DLL via reflection and checks each type to see if it implements the interface
- Those that do are added to the “plugin list”



TypeTranslator.cs



PluginLoader.cs

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Debug logging and the JIT



- Big believers in log4net
- Logging can be slow
  - Parts of code accessed millions of times
  - SetField() used *often*
- Log levels in log4net, Error, Warn, Info, Debug
  - First check is if logging is turned on, if not, bail
  - Still a function call, stack allocation, etc.

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Calls to log4net



```
static readonly log4net.ILog log =
    log4net.LogManager.GetLogger(
        System.Reflection.MethodBase
            .GetCurrentMethod().DeclaringType);
...
Log.debug("Some debug log data");
```



© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## More about OIDs



- Used as PK to our tables
- 20-byte binary blob
- Can load from DB knowing only PK
  - Don't know the type (customer/order/item/whatever)
- 0xF0AB3EED242DA7489F6AE044C44CEDA601000000
- Generic owner for row (Notes)
  - Notes owned by events, customers, orders, users, contacts, etc.

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## Query by Example



- Replaces simple select statements
- New up an *example* objects and ask the ORM to return a list of all objects in the db that look like the example

```
Activity example = Activity.NewActivity();
example.SetAsExample();
example.BilledDate = null;
example.UserID = User.UserID;
List<Activity> = Activity.GetByExample(example);
```

© 2009 Tech Software Inc.  
http://www.techsoftinc.com

## Query By SQL



- Allow business objects to use handcrafted SQL to query the DB
- This is the 80/20 rule.
  - No reason to define a new query language

```
public static List<Activity> GetUnbilledForUser(
    ActivityUser User) {
    const string sql = "select * from activity where
        UserID=@UserID and BilledDate is null";

    SqlParameter[] p = new SqlParameter[1];
    p[0] = new SqlParameter("@UserID", SqlDbType.Binary, 20);
    p[0].Value = (byte[])User.UserID;

    return GetBySql<Activity>(sql, p, null);
}
```

© 2009 Tech Software Inc.  
http://www.techsoftinc.com

## Some Issues We Have



- Joins
  - We support them, but we return an object graph
  - Not a SQL problem, a representational problem
    - We have a customer and a zipcode object, but not a customer-zipcode object.
  - Can walk up one-many tree, but not down
    - Lazy loading child lists
- Caching
  - Used extensively w/in core code, but no facility for caching data
  - Needs to support distributed cache like memcached, but business objects aware so API doesn't change

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>

## More issues



- No metadata correlation between properties and fields
  - Requires us to specify field name in call to SetField
  - Need to know private field names when doing some join statements
- No support for partial ghosting
  - Either the object is a ghost, or it's completely loaded

© 2009 Tech Software Inc.  
<http://www.techsoftinc.com>