ASP.NET MVC for Dummies

Paul Litwin
Deep Training &
Fred Hutchinson Cancer Research Center

paul@deeptraining.com
blog.paullitwin.com
twitter.com/plitwin

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Paul Litwin

- **Developer**
  - Focus: ASP.NET, ASP, C#, SQL Server, Reporting Services
  - MCSD
  - Microsoft MVP
  - Programmer Manger with
    Fred Hutchinson Cancer Research Center (Seattle)
- **Co-Founder and Senior Trainer**
  - Deep Training
    - www.deeptraining.com
- **Conference Chair/Speaker**
  - Chair, Microsoft ASP.NET Connections
  - Member INETA Speakers Bureau
- **Author of over a dozen books & courses, including…**
    - *AppDev SQL Server 2005 & 2008 Reporting Services Courses*
    - *ASP.NET for Developers*
    - *Access Cookbook*
    - *Access 2002 Desktop/Enterprise Dev Handbook*

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Updated Slides & Samples Download

- You can download them from:
  - www.deeptraining.com/litwin

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

ASP.NET MVC for Dummies Agenda

- What?
  - What is ASP.NET MVC?
- Why?
  - Why can’t I just use Web Forms?
  - Why use ASP.NET MVC?
- How?
  - How does it work?
  - Show me how to build an app using it

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

ASP.NET MVC

What?

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

ASP.NET MVC

- A new way to build highly-testable apps on top of the ASP.NET Framework
- An alternative to using RAD Web Forms

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

MVC Design Pattern

- Conery, Hanselman, Haack, Guthrie define it as…
  - An architectural pattern used to separate an application into three main aspects
    - **Model**. Set of classes that represent data and business rules for how data can be changed and manipulated
    - **View**. Application's user interface
    - **Controller**. Set of classes that handles communication from user, overall application flow, and application-specific logic

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

ASP.NET MVC

- ASP.NET MVC 1.0
  - Requires ASP.NET 3.5 / Visual Studio 2008 SP1
  - Download Microsoft ASP.NET MVC 1.0 from http://asp.net/mvc
- ASP.NET MVC 2.0
  - Part of Visual Studio 2010
    - Nothing to download if using with VS 2010
    - What we will use this session
  - Can also be used with Visual Studio 2008
    - Download from http://asp.net/mvc

MICROSOFT ASP.NET & SILVERLIGHT conference and expo

ASP.NET MVC

Why?

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Why Separate Concerns?

- Makes your app code more…
  - testable
  - modifable / refactorable
  - reusable
  - maintainable
  - scalable

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

What's Wrong with Web Forms?

- Nothing…but…
  - Web forms don't allow as precise control over "separation of concerns"
  - High level of abstraction
    - Not as "close to the metal" as some devs want
  - Web forms are difficult to test
  - Search engine optimization (SEO) issues
    - ASP.NET 4.0 addresses this somewhat

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

What Web Forms & MVC Have in Common

- Both hosted in Visual Studio
- Both built on ASP.NET
- Both run on IIS
- Both use .aspx pages
  - Both can use Master pages
  - though you can use a different view engine if you'd like in MVC (NHaml, Spark, Brail, NVelocity, etc.)
- Both can use any data access framework (ADO.NET, LINQ, Entity Framework, etc.)

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

ASP.NET MVC

How?

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Basic Tenets of ASP.NET MVC

- Convention over configuration
- Don’t repeat yourself (DRY)
- Pluggability whenever possible
  - We will use in-the-box parts of ASP.NET MVC but almost everything (data access, testing framework, view engine, validation, etc. can be replaced)
- Try to be helpful, but get out of the developer’s way

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Model, View, Controller

- Model
  - Data and business rules
- View
  - Application's UI. Will employ a view for each page that is displayed to users
- Controller
  - Classes that are responsible for application workflow and logic

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Besides the M, the V, and the C…

- Routing
  - Nice URLs are possible because of routing engine
  - Routes are stored in global.asax
- Binding to data
  - Default ModelBinder auto-magically converts form values and route data into data
  - Allows you to strongly type your pages & controller methods
- Scaffolding Views
  - ASP.NET MVC will automatically provide scaffolding to help you create your views
  - Helpful, but almost always will want to modify views it creates
- Html Helpers
  - Methods you can use in views to generate Html elements, usually bound to data

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building an MVC Application

- Remainder of talk will focus on building an end-to-end MVC application that can be used to display and edit product records from the Northwind database

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 1 –MVC App
  - Create ASP.NET MVC Application
  - Go ahead and let VS create test app
  - Remove following starter files
    - Controllers|HomeController.cs
    - All views in Views|Home folder
    - In Test app, Controllers|HomeControllerTest.cs

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

ASP.NET MVC Project Anatomy

- Content
- Controllers
- Models
- Scripts
- Views
  - Shared
- Global.asax

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building Models

- Model is the part of the MVC app that provides data access
  - You can build model using
    - ADO.NET
    - LINQ to SQL
    - Enterprise library
    - Nhibernate
    - …

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 2 – Creating Basic Model
  - We will use LINQ to SQL
  - We will build a model on 3 tables from Northwind
    - Products
    - Categories
    - Suppliers
  - Set UpdateCheck property on Product fields to Never

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 3 – Adding Repositories to Model
  - While we could have controller call LINQ to SQL classes, directly, it's better to provide another layer using a *repository* pattern
  - We will create repository classes for each model object
    - Very similar to creating a business tier in N-tier app design

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

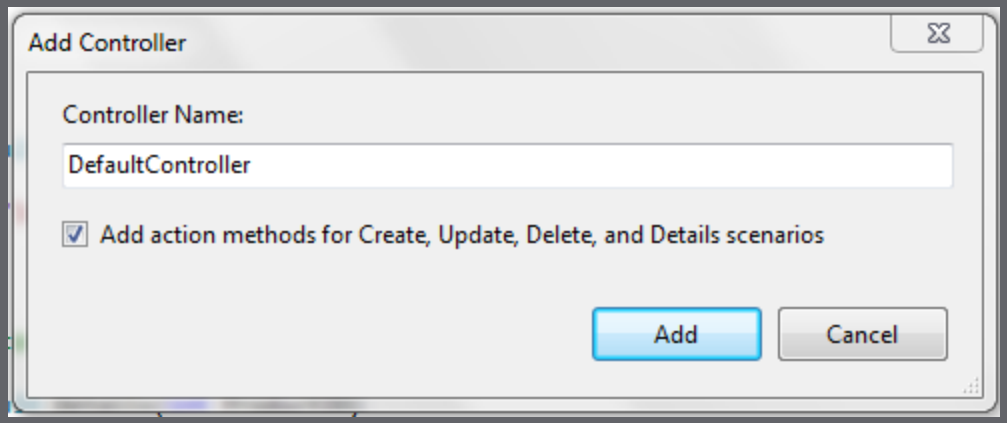
Building an MVC Application

- Demo 4 – Modifying Routing Table
  - Global.asax
    - Default route assumes id is the primary key
    - Need to change to use ProductID for HomeController

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Creating Controllers

- The intelligence of your application
  - Ferries data back & forth to model
  - Methods that eventually call views
  - Add Controller dialog can scaffold CRUD methods for you



MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 5 – Creating Home Controller
  - Create HomeController which will…
    - Be responsible for CRUD operations for Products
    - Replace form collection in post methods for (Create, Edit, Delete) with strongly-typed Product objects

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Constructing Views

- Default view engine is the aspx page
- The UI of the application

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 6 – Creating Views
  - Scaffold our views by right-clicking on each of the HomeController methods and selecting Add View
    - Use strong-typed viewProductsApp.Models.Product
  - Need to modify Index view since generated hyperlinks use id instead of ProductID
  - Also, need to fix up Create and Edit views
    - Remove ProductID from Create
    - Don't allow editing of ProductID in Edit
    - Replace Discontinued textbox with checkbox in Create & Edit

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 7 – Adding Dropdown Support
  - Add ProductFormModel class to HomeController
  - Change Edit/Create methods to pass ProductFormModel to views
  - Change type of Edit and Create views to derive from ProductFormModel class
  - Replace each instance of “model.” with “model.Product.”
  - Add ProductID Html.HiddenFor on Edit page
  - Replace category and supplier Html.TextBoxFor helpers with Html.DropDownListFor helpers

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 8 – Adding Validation
  - Add “Buddy” partial class for Products with data annotations
  - Add `if (!ModelState.IsValid)` to Create and Edit post methods
  - Adding client-side support requires a couple of additional steps
    - See http://haacked.com/archive/2009/11/19/aspnetmvc2-custom-validation.aspx

MICROSOFT ASP.NET & SILVERLIGHT conference and expo

Refactoring App to Support Testing

- Currently, controller and model are tightly coupled; Can't test app without database dependency
- Can be fixed by using dependency injection
  - Extract interface
  - Update controller to support Constructor Injection
  - Create fake repository that also derives from interface

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 9 – Refactoring for Testing
  - Right-click on ProductRepository class and select Refactor|Extract Interface
  - Make interface public
  - Modify HomeController with two constructors
    - Parameterless that creates new ProductRepository
    - Parameterized that allows passing in IProductRepository
  - Add FakeProductRepository class
    - : IProductRepository
    - Right-click on interface and select Implement Interface

MICROSOFT ASP.NET & SILVERLIGHT conference and expo

Constructing Tests

- For each unit test
  - Arrange
    - Arrange test, setting up any pre-requisites
  - Act
    - Act to test feature of application
  - Assert
    - Assert what you expect

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Building an MVC Application

- Demo 10 – Creating & Running Unit Test
  - In Test project, right-click on Controllers, and select Add New Test and select Unit Test
  - Create CreateTestProducts method
  - Create tests using FakeProductRepository
  - To run tests, right-click on HomeControllerTest class name and select Run Tests

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

Summary

- How does MVC compare to Web Forms development?

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

ASP.NET MVC Pluses (Why)

- A new way to build highly-testable apps on top of the ASP.NET Framework
- No post-backs
- Requests map to methods rather than pages
- SEO-friendly URLs
- Supports TDD and Unit Tests
- Encourages use of design patterns and best practices

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

ASP.NET MVC Minuses (Why Not)

- Not as RAD as using Web Forms
- No drag and drop experience
- Doesn’t build on all your experience with ASP.NET server controls
- Encourages use of design patterns and best practices

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

How Do I Decide Which to Use?

- Choose MVC if you…
  - want to embrace design patterns and are willing to put hard work into understanding them
  - are disciplined and want to follow design patterns and best practices
  - are interested in embracing unit tests
  - are interested in embracing TDD
  - are being told to use it by your employer/client

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

How Do I Decide Which to Use?

- Choose Web Forms if you…
  - you like Web Forms
  - your brain hurts when reading about design patterns
  - you don’t like following patterns or best pratices
  - are not interested in embracing unit tests or TDD
  - are being told to use it by your employer/client

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Another Way to Look At It…

- Building an intranet site with lots of data editing
  - Web Forms may be better suited
- Building an Internet site where HTML, performance, and scalability are paramount
  - ASP.NET MVC may be better suited

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo

Something to Consider…

- Once you start on MVC path, you will find you will find yourself getting pulled in deeper and deeper…
  - How do I write good unit tests with good coverage?
  - Do you use a different View engine?
  - Do you use a unit test framework
  - Do you use a mock framework?
  - What about validation framework?
  - What about dependency injection, inversion of control?
  - …

MICROSOFT
ASP.NET&
SILVERLIGHT
conference and expo

What About Me?

- I am intrigued with ASP.NET MVC
- However…
  - My brain does hurt when reading about design patterns
  - I love Web Forms
- OTOH…
  - I like the idea of unit testing, maintainability, and best practices

MICROSOFT ASP.NET & SILVERLIGHT conference and expo

Learning Resources

- Bing/Google for Nerd Dinner MVC
  - Chapter by Scott Guthrie
- ASP.NET MVC Framework Unleashed
  - Walther
- Professional ASP.NET MVC 1.0
  - Conery, Hanselman, Haack, Guthrie
- ASP.NET in Action
  - Palermo, Scheirman, Bogard
- Head First Design Patterns
  - Freeman & Freeman

MICROSOFT ASP.NET & SILVERLIGHT conference and expo

Thank You!

- Please complete evaluation forms
- Contact: paul@deeptraining.com
- Download updated slides & samples from
  - www.deeptraining.com/litwin

MICROSOFT ASP.NET & SILVERLIGHT conference and expo

Your Feedback is Important

Please fill out a session evaluation form
drop it off at the conference registration desk.

Thank you!

MICROSOFT
ASP.NET & SILVERLIGHT
conference and expo