

AMV12: Lightning Development with MVC Scaffolding

Paul Litwin

Fred Hutchinson Cancer Research Center /
Deep Training

paul@deeptesting.com

twitter.com/plitwin



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...**

- *Agile ASP.NET Unleashed (writing...)*
- *AppDev SQL Server 2005 & 2008 Reporting Services Courses*
- *ASP.NET for Developers*
- *Access Cookbook*
- *Access 2002 Desktop/Enterprise Dev Handbook*



OpenSpaces at DevConnections

- Free-form “sessions”
- Come to present, discuss, or listen
- Anyone can lead a session

- Tonight from 7:45-9:45
- Free Beer & Pizza
- Sponsored by Microsoft Community

Slides & Samples Download

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

Lightning Development with MVC Scaffolding

- **Scaffolding with MVC Scaffolding**
- Scaffolding Controllers and Views
- Controlling the Scaffolding
- Working with Relationships
- Scaffolding Other Stuff
- Customizing T4 Templates
- Scaffolding an Existing Database

MVC Scaffolding vs. MVC Scaffolding

- **What is Scaffolding?**
 - A program or tool that attempts to intelligently build a piece of an app or an entire application
- **What is MVC Scaffolding?**
 - VS & ASP.NET MVC offer basic scaffolding of controllers & views “out of the box”
- **What is “*MVC Scaffolding*” ?**
 - A nuget package from Microsoft that provides advanced Code First-based scaffolding

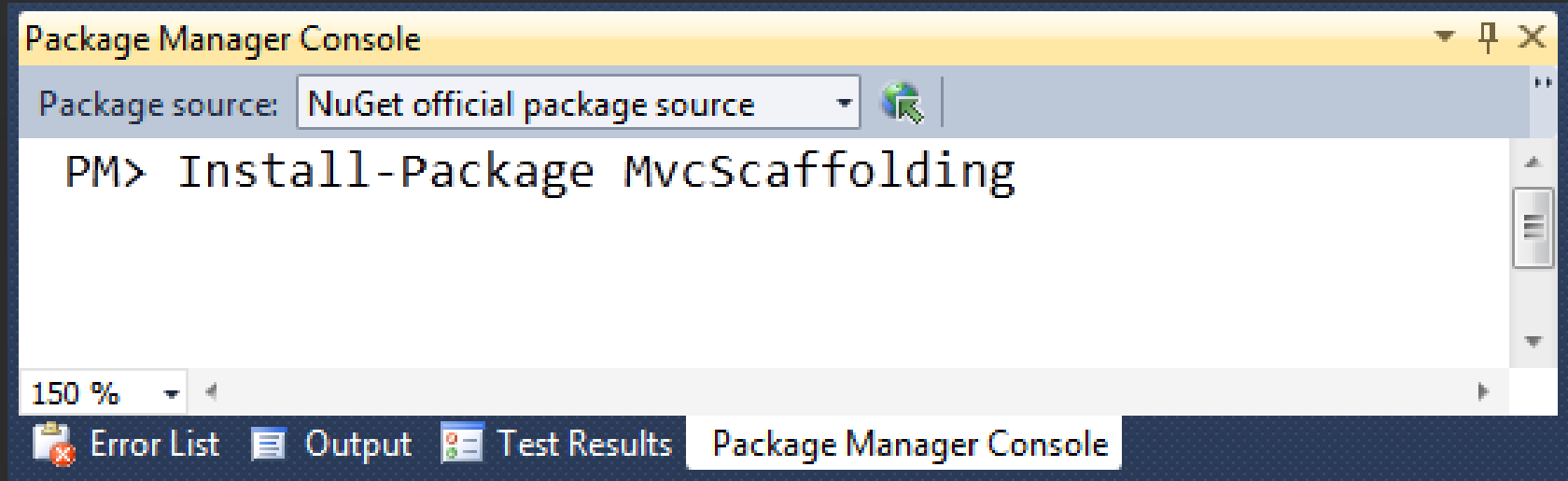
“MVCScaffolding” Nuget Package

- **Features**

- Can create controller and views
- Supports ASPX and Razor views
- Supports both VB and C# code
- Supports entity relationships
- Will optionally create repository classes, ViewModel classes, unit test stubs
- Supports custom scaffolders

MVC Scaffolding

- Install MvcScaffolding from nuget



The screenshot shows the Package Manager Console window in Visual Studio. The title bar reads "Package Manager Console". Below the title bar, there is a dropdown menu for "Package source:" set to "NuGet official package source". The main area of the console contains the command "PM> Install-Package MvcScaffolding". At the bottom, there is a taskbar with icons for "Error List", "Output", "Test Results", and "Package Manager Console". The zoom level is set to "150 %".

```
Package Manager Console
Package source: NuGet official package source
PM> Install-Package MvcScaffolding
150 %
Error List Output Test Results Package Manager Console
```

Lightning Development with MVC Scaffolding

- Scaffolding with MVC Scaffolding
- **Scaffolding Controllers and Views**
- Controlling the Scaffolding
- Working with Relationships
- Scaffolding Other Stuff
- Customizing T4 Templates
- Scaffolding an Existing Database

Scaffolding Controllers and Views

- MVC Scaffolding works with EF Code First
- Before using it, you need to create your model classes

```
namespace MvcCodeFirst.Models
{
    public class RandomAct
    {
        public int RandomActId { get; set; }
        public string Title { get; set; }
        public string Description { get; set; }
        public string Location { get; set; }
        public string Address { get; set; }
        public string City { get; set; }
        public string State { get; set; }
        public DateTime StartTime { get; set; }
        public DateTime EndTime { get; set; }
    }
}
```

Scaffolding Controllers and Views

- Scaffolding controllers & standard CRUD views

```
Scaffold Controller RandomAct
```

How It Works

1. MVC Scaffolding uses your model classes & EF Code First conventions to determine what to do
2. At design time, it scaffolds app elements: model context, controllers, views, etc.
3. Then when you run the app for first time, EF Code First kicks in to turn those classes into database entities

Lightning Development with MVC Scaffolding

- Scaffolding with MVC Scaffolding
- Scaffolding Controllers and Views
- **Controlling the Scaffolding**
- Working with Relationships
- Scaffolding Other Stuff
- Customizing T4 Templates
- Scaffolding an Existing Database

MVC Scaffolding Works with EF Code First Data Annotations

- MVC Scaffolding uses default Code First conventions to find primary keys, foreign keys, etc
 - PKs end in Id
 - PKs of type int/long/short set to identity
- You can modify default behavior using annotations
 - Key
 - StringLength
 - MaxLength
 - ConcurrencyCheck
 - Required
 - Timestamp
 - ComplexType
 - Column
 - Table
 - InverseProperty
 - ForeignKey
 - DatabaseGenerated
 - NotMapped

Using Data Annotations

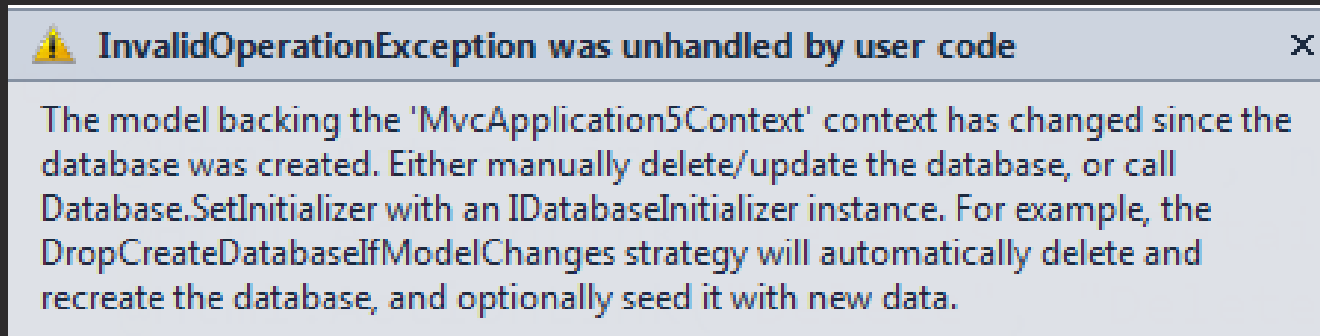
```
public class RandomAct
{
    public int RandomActId { get; set; }
    [Required]
    [StringLength(256)]
    public string Title { get; set; }
    [StringLength(2048)]
    public string Description { get; set; }
    public string Location { get; set; }
    public string Address { get; set; }
    public string City { get; set; }
    public string State { get; set; }
    public DateTime StartTime { get; set; }
    public DateTime EndTime { get; set; }
}
```

Force Option

- Every scaffolder has –force option to allow for over-writing existing files

Modifying Model

- You can modify model and scaffold and re-scaffold to your heart's content
- However, as soon as you run the app based on modified model
 - You get an exception!



Modifying the Model Requires Regenerating Database

- You need to read comment in *appContext.cs* and copy code into *Application_Start()* method of *Global.asax.cs*

```
protected void Application_Start()  
{  
    AreaRegistration.RegisterAllAreas();  
    RegisterGlobalFilters(GlobalFilters.Filters);  
    RegisterRoutes(RouteTable.Routes);  
  
    System.Data.Entity.Database.SetInitializer(new  
        System.Data.Entity.DropCreateDatabaseIfModelChanges  
        <MvcApplication5.Models.MvcApplication5Context>());  
}
```

Controlling the Scaffolding

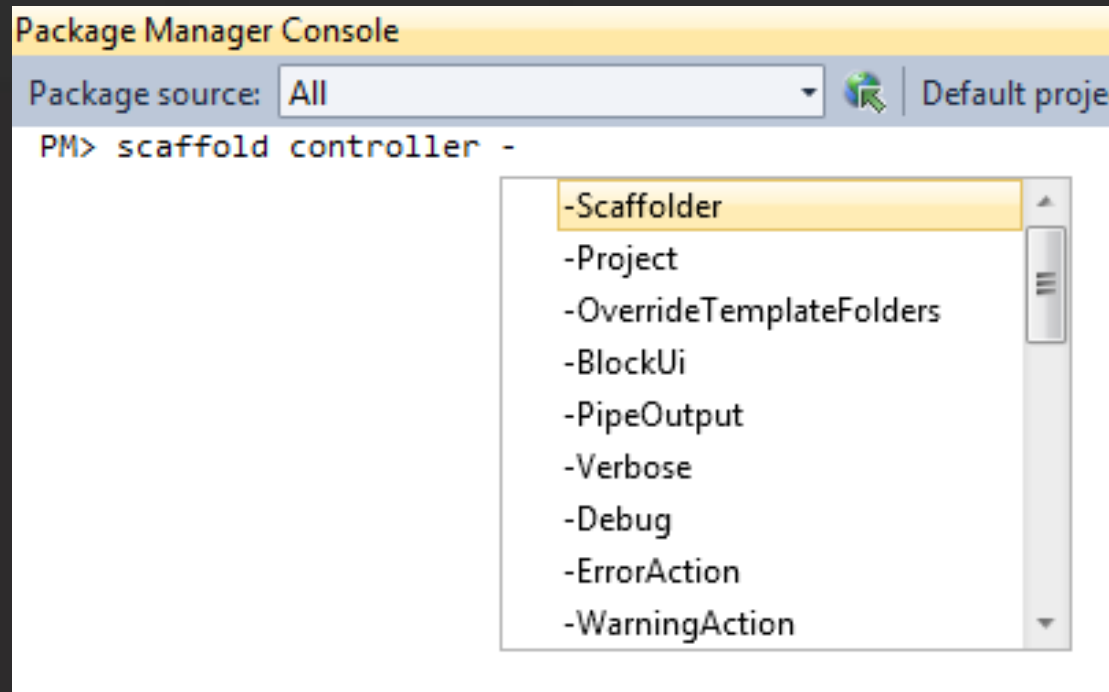
- Controller scaffolding options

Scaffold Controller

- ControllerName \$ControllerName
- ModelType \$ModelType
- Project \$Project
- CodeLanguage \$CodeLanguage
- DbContextType \$DbContextType
- Area \$Area
- ViewScaffolder \$ViewScaffolder
- Layout \$Layout
- SectionNames \$SectionNames
- PrimarySectionName \$PrimarySectionName
- ReferenceScriptLibraries:\$ReferenceScriptLibraries
- NoChildItems:\$NoChildItems
- OverrideTemplateFolders \$TemplateFolders
- Force:\$Force
- ForceMode \$ForceMode
- Repository

Controlling the Scaffolding

- Tip: to get help with options, type command into console plus “ - ” and hit tab



Controlling the Scaffolding

- For example

```
Scaffold Controller
```

```
-ControllerName MickeyMouse
```

```
-ModelType RandomAct
```

```
-CodeLanguage vb
```

```
-ViewScaffolder MvcScaffolding.AspxView
```

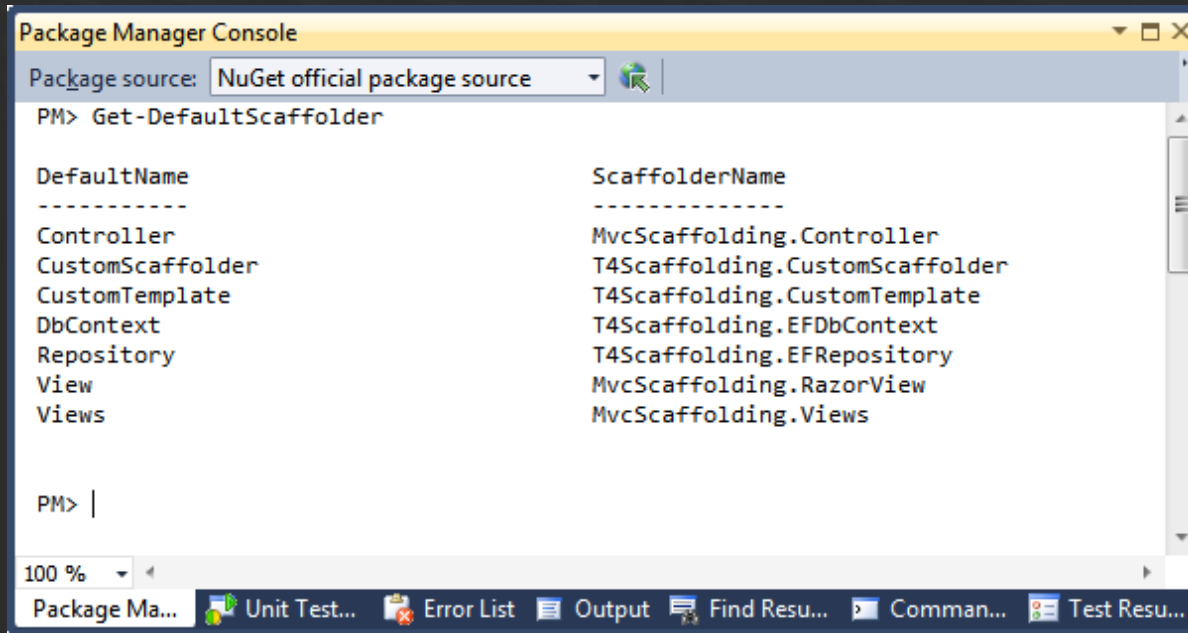
Creating Controller and Repository

- Create controller and repository (and replace existing files)

```
Scaffold Controller controllerName  
-repository -force
```

Controlling the Scaffolding

- Default scaffolders



```
Package Manager Console
Package source: NuGet official package source
PM> Get-DefaultScaffolder

DefaultName          ScaffoldName
-----
Controller           MvcScaffolding.Controller
CustomScaffolder     T4Scaffolding.CustomScaffolder
CustomTemplate       T4Scaffolding.CustomTemplate
DbContext            T4Scaffolding.EFDbContext
Repository           T4Scaffolding.EFRepository
View                 MvcScaffolding.RazorView
Views                MvcScaffolding.Views

PM> |
```

- Changing defaults

```
Set-DefaultScaffolder View MvcScaffolding.AspxView
```

Lightning Development with MVC Scaffolding

- Scaffolding with MVC Scaffolding
- Scaffolding Controllers and Views
- Controlling the Scaffolding
- **Working with Relationships**
- Scaffolding Other Stuff
- Customizing T4 Templates
- Scaffolding an Existing Database

Scaffolding Relationships

- Add navigational members to model classes

```
public class RandomAct
{
    public int RandomActId { get; set; }
    public string Title { get; set; }
    public string Description { get; set; }
    public string Location { get; set; }
    public string Address { get; set; }
    public string City { get; set; }
    public string State { get; set; }
    public DateTime StartTime { get; set; }
    public DateTime EndTime { get; set; }

    public virtual ICollection<RandomActActor> Actors { get; set; }
}
```

- One RandomAct has many Actors

Scaffolding Relationships

- Add navigational members to model classes

```
public class RandomActActor
{
    public int RandomActActorId { get; set; }

    public int RandomActId { get; set; }
    [ForeignKey("RandomActId")]
    public virtual RandomAct Act { get; set; }

    public int RandomActorId { get; set; }
    [ForeignKey("RandomActorId")]
    public virtual RandomActor Actor { get; set; }

    public string Message { get; set; }
}
```

- One RandomActActor row has many Acts and Actors

Scaffolding Relationships

- Annotating model with navigational members...
 - Virtual collection properties create counts on index page
 - Foreign key virtual properties create drop-downs on create/edit pages

Scaffolding Relationships

- Add **DisplayColumn** to class to control which field is used in dropdown

```
DisplayColumn("Name", "Name", false)]
```

```
public class RandomActor  
{  
    ...  
}
```

Lightning Development with MVC Scaffolding

- Scaffolding with MVC Scaffolding
- Scaffolding Controllers and Views
- Controlling the Scaffolding
- Working with Relationships
- **Scaffolding Other Stuff**
- Customizing T4 Templates
- Scaffolding an Existing Database

Scaffolding Other Stuff

- Views
- Actions
- Repositories
- Unit Tests

Scaffolding Views

- Create empty view

```
Scaffold View controllerName viewName
```

- Create templated view

```
Scaffold View controllerName viewName  
-Template Create|Delete|Details|Index|Edit  
-ModelType modelType
```

- Create all views for controller

```
Scaffold Views  
-Controller controllerName  
-ModelType modelType
```

Scaffolding Actions

- Create empty action

```
Scaffold Action controllerName actionName
```

- Create action with ViewModel

```
Scaffold Action controllerName actionName  
-withViewModel
```

- Create action with named ViewModel

```
Scaffold Action controllerName  
actionName  
-ViewModel viewModelName
```

Scaffolding More Actions

- Create action with named ViewModel and post-handling action

```
Scaffold Action controllerName actionName  
  -ViewModel viewModelName  
  -Post
```

Scaffolding Repositories

- Create repository class

```
Scaffold Repository modelType
```

- Create controller and repository (shown earlier)

```
Scaffold Controller controllerName  
-repository
```

Scaffolding Unit Tests

- Assumption
 - You have MS Test unit test project with reference to MVC project & named *mvcProjectName.Test*
- Create Unit Test for Action

```
Scaffold UnitTest controllerName actionName
```

- Create Unit Test for Action w/ ViewModel

```
Scaffold UnitTest controllerName actionName  
-ViewModel viewModelName
```

Scaffolding Action + Unit Test

- Create Action with ViewModel + Unit Test for Action

```
Scaffold MvcScaffolding.ActionWithUnitTest  
controllerName actionName  
-WithViewModel
```

Lightning Development with MVC Scaffolding

- Scaffolding with MVC Scaffolding
- Scaffolding Controllers and Views
- Controlling the Scaffolding
- Working with Relationships
- Scaffolding Other Stuff
- **Customizing T4 Templates**
- **Scaffolding an Existing Database**

Scaffolders and Templates

Scaffolder name	Template name
DbContext	DbContext
DbContext	DbContextEntityMember
Repository	Repository
Action	Action
Action	ActionPost
Action	ViewModel
ActionUnitTest	TestClass
ActionUnitTest	TestMethod
View	Create, _CreateOrEdit, Delete, Details, Edit, Empty, Index
Controller	ControllerWithContext
Controller	ControllerWithRepository

Customizing T4 Template (1 of 2)

1. Determine Scaffold and Template
2. Use the CustomScaffolder to copy a scaffold's T4 template to the CodeTemplates folder of project

```
Scaffold CustomScaffolder scaffoldName  
    customTemplateName
```

Customizing T4 Template (2 of 2)

3. Modify and save T4 templates
4. Modified template used for any scaffolding in the project
5. You can also copy modified templates to packages folder of other projects

Lightning Development with MVC Scaffolding

- Scaffolding with MVC Scaffolding
- Scaffolding Controllers and Views
- Controlling the Scaffolding
- Working with Relationships
- Scaffolding Other Stuff
- Customizing T4 Templates
- **Scaffolding an Existing Database**

MVC Scaffolding in DB First Scenarios

- Steps

1. Use EF **Generate from Database** wizard
2. Add Code Generation Item...
 - **ADO.NET DbContext Generator**
3. Scaffold away!
4. Now change context type reference in controller or repository classes to point to EDMX context (***modelNameEntities***) instead of CodeFirst context (***projectNameContext***)

```
9 namespace BPathMVC.Models
10 {
11     public class ParticipantRepository : IParticipantRepository
12     {
13         //BPathMVCContext context = new BPathMVCContext();
14         BPathEntities context = new BPathEntities();
15     }
```

MVC Scaffolding in DB First Scenarios

- More info in my blog post
 - <http://weblogs.asp.net/paullitwin>
 - Search for “MVC Scaffolding”

USE MVC SCAFFOLDING IN DATABASE FIRST SCENARIOS

Steve Sanderson's [MVC Scaffolding NuGet](#) package is awesome and goes far beyond the very good scaffolding available from the Add Controller dialog in Visual Studio 2010 (after installing the [MVC 3 Tools Update](#)).

But a shortcoming of MVC Scaffolding is that, out of the box, it works only in Code

Resources

- <http://blog.stevensanderson.com/>

Steven Sanderson's blog

As seen on YouTube™

Scaffold your ASP.NET MVC 3 project with the MvcScaffolding package

POSTED ON JANUARY 13, 2011

This post is part of a series about the MvcScaffolding NuGet package:

1. **This post: Scaffold your ASP.NET MVC 3 project with the MvcScaffolding package**
2. Standard usage: Typical use cases and options
3. One-to-Many Relationships
4. Scaffolding Actions and Unit Tests
5. Overriding the T4 templates
6. Creating custom scaffolders
7. Scaffolding custom collections of files

Slides & Samples Download

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

OpenSpaces at DevConnections

- Free-form “sessions”
- Come to present, discuss, or listen
- Anyone can lead a session

- Tonight from 7:45-9:45
- Free Beer & Pizza
- Sponsored by Microsoft Community

Your Feedback is Important

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

Thank you!