

ASP03: ASP.NET Database Development Using EF 4.1 and Code-First

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
 - Microsoft MVP
 - Programmer Manager 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*

Slides & Samples Download

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

ASP.NET DB Develop using EF 4.1 & Code-First

- **Entity Framework 4.1**
- Using DbContext
- Code First Development
- MVC Scaffolding

Entity Framework

- Entity Framework is an object-relational mapping (ORM) framework for the .NET Framework

Entity Framework 4.1

- Two major new features in 4.1 release
 - DbContext API
 - Code First

ASP.NET DB Develop using EF 4.1 & Code-First

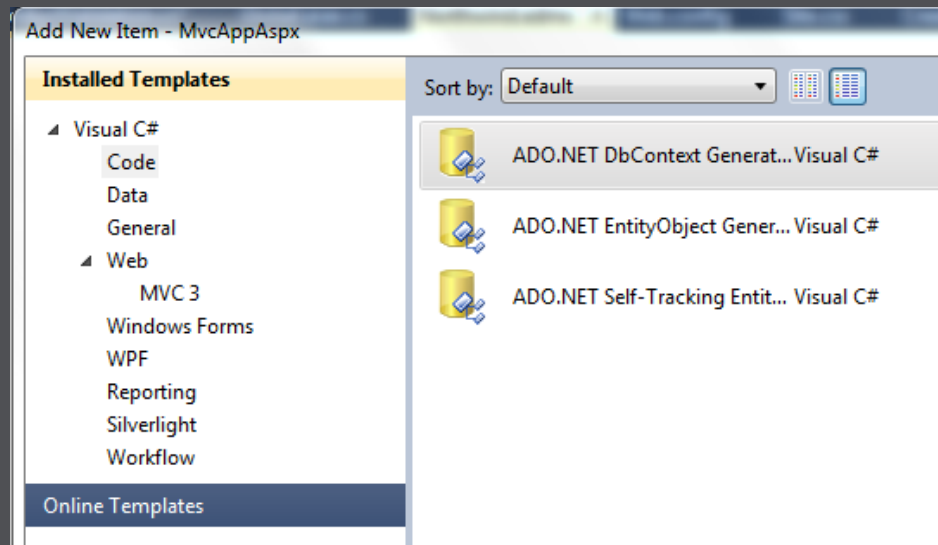
- Entity Framework 4.1
- **Using DbContext**
- Code First Development
- MVC Scaffolding
- EF Power Tools

Using DbContext

- What is DbContext?
 - Simplified abstraction overObjectContext
 - Optimized for common tasks and coding patterns
- DbContext works with Database First, Model First, and Code First

Using DbContext

- Entity Framework usesObjectContext by default
 - You need to override this default
 - Change EF code generation to **DbContext** using **Add Code Generation Item...** pop-up menu



Demo

- Using DbContext...
 - Web Forms app
 - Database First development

Coding AgainstObjectContext vs. DbContext

```
public void Insert(Product productToInsert)
{
    db.Products.AddObject(productToInsert);
    db.SaveChanges();
}

public void Update(Product productToUpdate)
{
    db.Products.Attach(productToUpdate);
    db.ObjectStateManager.ChangeObjectState(
        productToUpdate,
        EntityState.Modified);
    db.SaveChanges();
}
```

```
public void Insert(Product productToInsert)
{
    db.Products.Add(productToInsert);
    db.SaveChanges();
}

public void Update(Product productToUpdate)
{
    db.Products.Attach(productToUpdate);
    db.Entry(productToUpdate).State =
        EntityState.Modified;
    db.SaveChanges();
}
```

ASP.NET DB Develop using EF 4.1 & Code-First

- Entity Framework 4.1
- Using DbContext
- **Code First Development**
- MVC Scaffolding

Code First Development

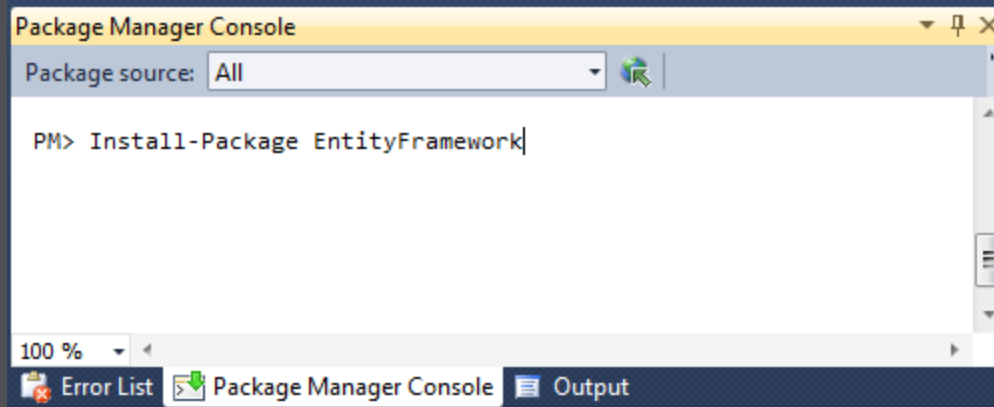
- Supported by EF 4.1
- Create classes
- No database; no ORM model
- EF creates database

Demo

Using Code First

Code First MVC App Demo

- Step 0
 - Install nuget from nuget.org.
- Step 1
 - Install EntityFramework nuget pkg to project



Code First MVC App Demo

- Step 2
 - Create model class(es)

```
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; }
    }
}
```

Code First MVC App Demo

- Step 3
 - Optionally decorate classes & properties with DataAnnotations

```
public class RandomActor
{
    public int RandomActorId { get; set; }
    [Required]
    public string FirstName { get; set; }
    [Required]
    public string LastName { get; set; }
}
```

Code First MVC App Demo

- Step 4
 - Add class that derives from DbContext

```
namespace MvcCodeFirst.Models
{
    public class MvcCodeFirstContext : DbContext
    {
        public DbSet<MvcCodeFirst.Models.RandomAct> RandomActs { get; set; }
        public DbSet<MvcCodeFirst.Models.RandomActor> RandomActors { get; set; }
        ...
    }
}
```

Code First MVC App Demo

- **Step 5**
 - Build app on DbContext class entities
- **Step 6**
 - First time app is run, EF CodeFirst will create database

Code First Model Discovery

- DbContext works out what classes to include in model by looking at DbSet properties
- It then 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

Code First Model Discovery

- You can modify default behavior using data annotations or fluent API
- Supported Annotations
 - Key
 - StringLength
 - MaxLength
 - ConcurrencyCheck
 - Required
 - Timestamp
 - ComplexType
 - Column
 - Table
 - InverseProperty
 - ForeignKey
 - DatabaseGenerated
 - NotMapped

Code First Model Discovery

- Database

- EF, by default, creates DB using SQL Express with name derived from DbContext class
 - E.g, MvcCodeFirst.Models.MvcCodeFirstContext
- You can add connection string to web.config, e.g., to redirect DB to SQL Server instance matching cnx string name to context name

```
<connectionStrings>  
  <add  
    name="MvcCodeFirstContext"  
    providerName="System.Data.SqlClient"  
    connectionString="Server=.;Database=RandomActs;Trusted_Connection=true;"/>  
</connectionStrings>
```

Code First: Regenerating Database

```
namespace MvcCodeFirst.Models
{
    public class MvcCodeFirstContext : DbContext
    {
        public MvcCodeFirstContext()
        {
            Database.SetInitializer(new
                DropCreateDatabaseIfModelChanges<MvcCodeFirstContext>());
        }
    }
}
```

ASP.NET DB Develop using EF 4.1 & Code-First

- Entity Framework 4.1
- Using DbContext
- Code First Development
- **MVC Scaffolding**

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

Demo

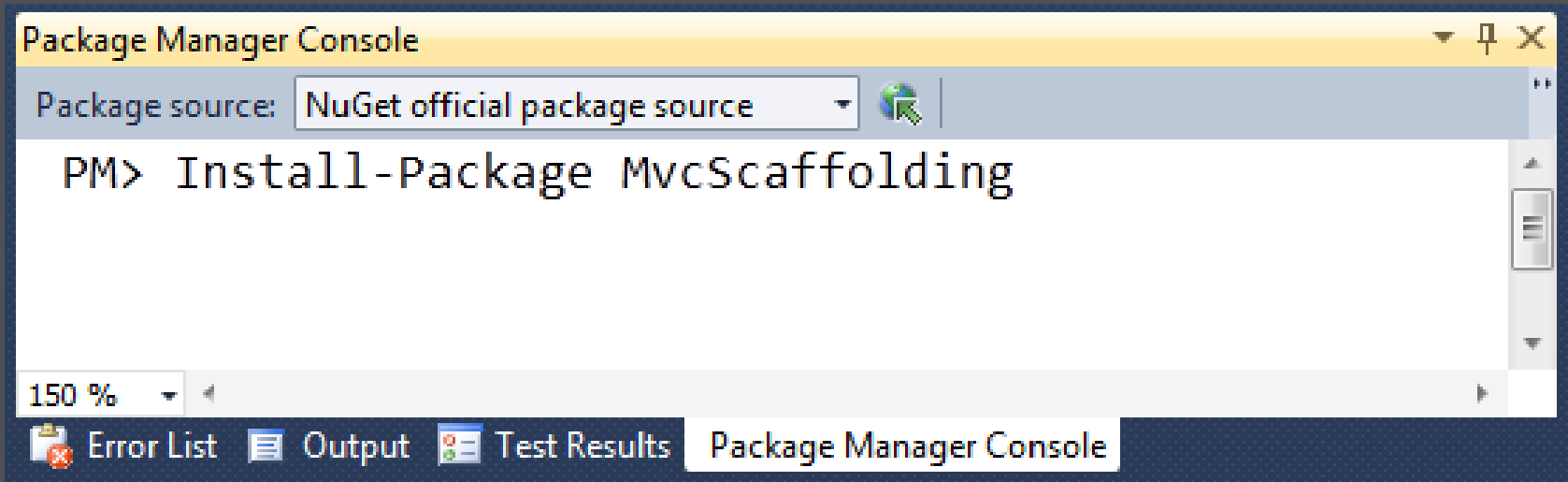
Building a Code First app using
EF Code First, MVC 3, NuGet, and
MVCScaffolding

MVC Scaffolding Demo

- Step 1
 - Install nuget from nuget.org.

MVC Scaffolding Demo

- Step 2
 - Install MvcScaffolding



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 of the console, there is a zoom level of "150 %". The bottom of the image shows the Visual Studio interface with tabs for "Error List", "Output", "Test Results", and "Package Manager Console".

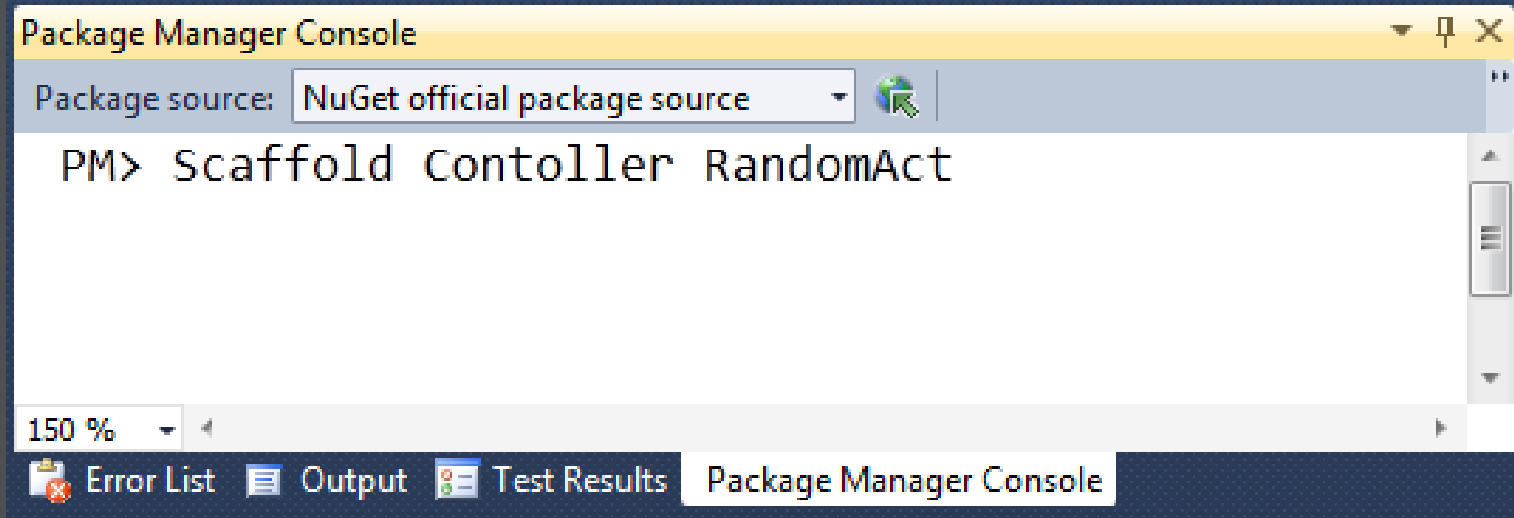
MVC Scaffolding Demo

- Step 3
 - Create model class(es)

```
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; }
    }
}
```

MVC Scaffolding Demo

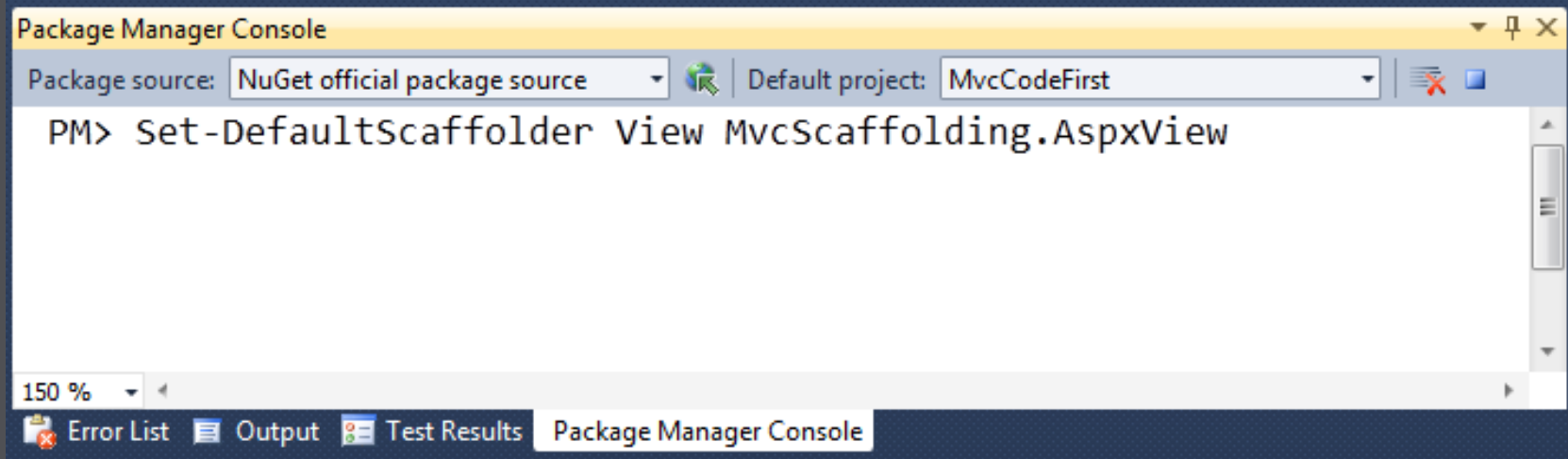
- Step 4
 - Scaffold the controller



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> Scaffold Controller RandomAct". At the bottom of the console, there is a zoom level of "150 %". The taskbar at the bottom of the window shows icons for "Error List", "Output", "Test Results", and "Package Manager Console".

MVC Scaffolding Demo

- Step 5 (Optional)
 - Change default view scaffolder



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

MVC Scaffolding Demo

- **Step 6**
 - Add additional classes
 - Add navigation members to support relationships
- **Step 7**
 - Re-scaffold with **–force** and **–repository** options
 - Need to uncomment **Database.SetInitialize** line in **projectContext.cs** file

MVC Scaffolding Demo

- Step 8
 - Add **DisplayColumn** to class to control which field is used in dropdown

```
DisplayColumn("Name", "Name", false)]  
public class RandomActor  
{  
    ...  
}
```

MVC Scaffolding Demo

- Next Steps
 - Build our your app using the scaffolded app as a starting point
 - For example...
 - Add validation
 - Change styles
 - Etc...

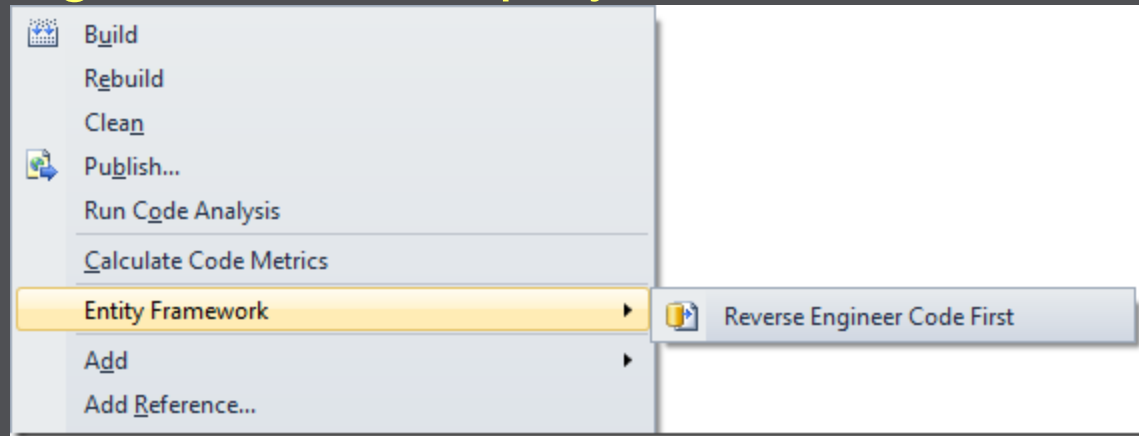
“MVCScaffolding” Nuget Package

- **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

Tip: Download EF Power Tools

- EF Power Tools (currently CTP1)
 - Use to reverse engineer existing database to Code First
 - Then you can use with MVCSc scaffolding
 - Right-click on C# project



Tip: Download EF Power Tools

- EF Power Tools (currently CTP1)
 - Display EDM model for Code First model

Resources

- <http://asp.net/mvc>
 - Starting place for everything on ASP.NET MVC
- <http://weblogs.asp.net/scottgu/>
 - Msft Genius Dev Leader Scott Guthrie's blog
- <http://hanselman.com/blog/>
 - Msft Community Guru Scott Hanselman's blog
- <http://blog.stevensanderson.com/>
 - Msft PM for MVC Scaffolding
- <http://blogs.msdn.com/adonet/>
 - ADO.NET Team blog on EF
- <http://msdn.microsoft.com/en-us/magazine/gg232765.aspx>
 - Nice short article by Julie Lerman explaining EF 4.1 objects and methods and how they contrast with EF 4 objects and methods

Slides & Samples Download

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

Your Feedback is Important

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

Thank you!