



www.twitter.com/Telerik

www.facebook.com/Telerik

SLICE YOUR DEVELOPMENT TIME WITH ASP.NET MVC AND RAZOR

Philip Japikse (@skimedic)

phil@telerik.com

www.skimedic.com/blog

MVP, MCSD.Net, MCDBA, CSM, CSP

Agile Practices Evangelist, Telerik



PHIL.ABOUT()

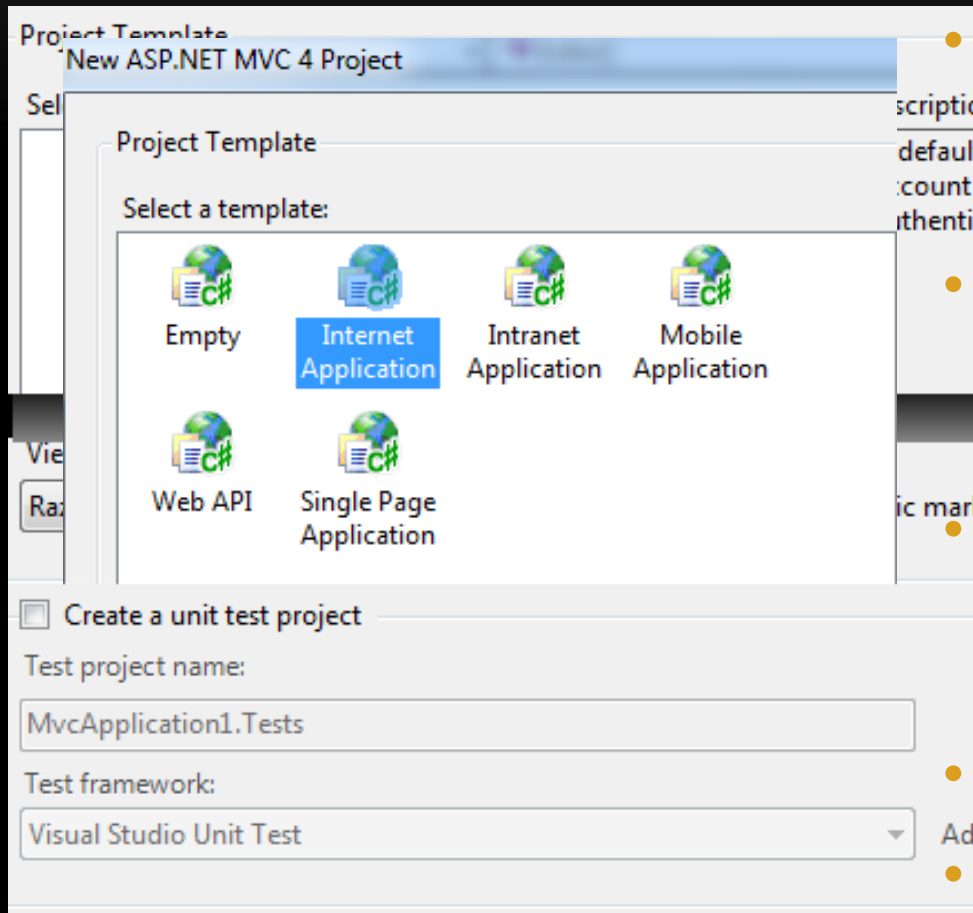
- Agile Practices Evangelist, Telerik, Inc.
- Microsoft MVP, MCSD, MCDBA, CSM, CSP
- Host, Zero To Agile Podcast
 - www.telerik.com/zerotoagile
- Founder, Agile Conferences, Inc.
- President, Cincinnati .NET User's Group
- Columnist, Developer.com

VISUAL STUDIO & MVC

VISUAL STUDIO MVC GOODIES

- Fully functional starter sites
 - Include:
 - ADO.NET Entity Framework
 - Modernizr 1.7
 - jQuery, jQueryUI, jQuery Validation
 - HTML5
- Testing is First Class Citizen
- MVC Specific Code navigation and creation
- Controller and View scaffolding

ASP.NET MVC STARTER PROJECTS



Internet –

- Starter website that uses forms authentication

Intranet –

- Starter website that uses Windows authentication

Mobile (MVC4)

- Optimized for Mobile – forms authentication

WebAPI (MVC4)

Single Page (MVC4)

STARTER SITES

The image shows a screenshot of an ASP.NET MVC starter site template. The main content area has a light blue header with the text "your logo here." and a navigation menu with "Home", "About", and "Contact" links. Below the header, there is a section titled "Home Page. Modify this" with a sub-header "Modify this template to jump-start your ASP.NET MVC application." and a paragraph "To learn more about ASP.NET MVC visit <http://asp.net/mvc> and samples to help you get the most from". A mobile device overlay is positioned in the center, showing a browser address bar with "localhost:2114/" and a "Google" search button. The mobile screen displays the same "Home Page" content, including a "Log in" button and a navigation menu with "Navigation", "About", and "Contact" items. The mobile device also shows a standard Android-style navigation bar at the bottom with back, forward, refresh, and home icons. On the left side of the main page, there is a section titled "We suggest the following:" with three numbered steps: 1. "Configure ASP.NET Membership", 2. "Add NuGet packages and jump start", and 3. "Setup deployment to your web server". At the bottom of the page, there are three large icons: a location pin, a mobile device, and a cloud with an upload arrow.

your logo here.

Register Log in

Home About Contact

Home Page

localhost:2114/ ★ Google

Log in Home Page

Home Page. Modify this

Modify this template to jump-start your ASP.NET MVC application.

To learn more about ASP.NET MVC visit <http://asp.net/mvc> and samples to help you get the most from

We suggest the following:

- 1 Configure ASP.NET Membership**
ASP.NET membership gives you a built local instance of SQL Server, but you c
- 2 Add NuGet packages and jump start**
NuGet makes it easy to install and upc [NuGet Packages](#), and then pick from t
- 3 Setup deployment to your web server**
Use Microsoft Web Deploy to quickly a and choose one of the many options t

Navigation

About

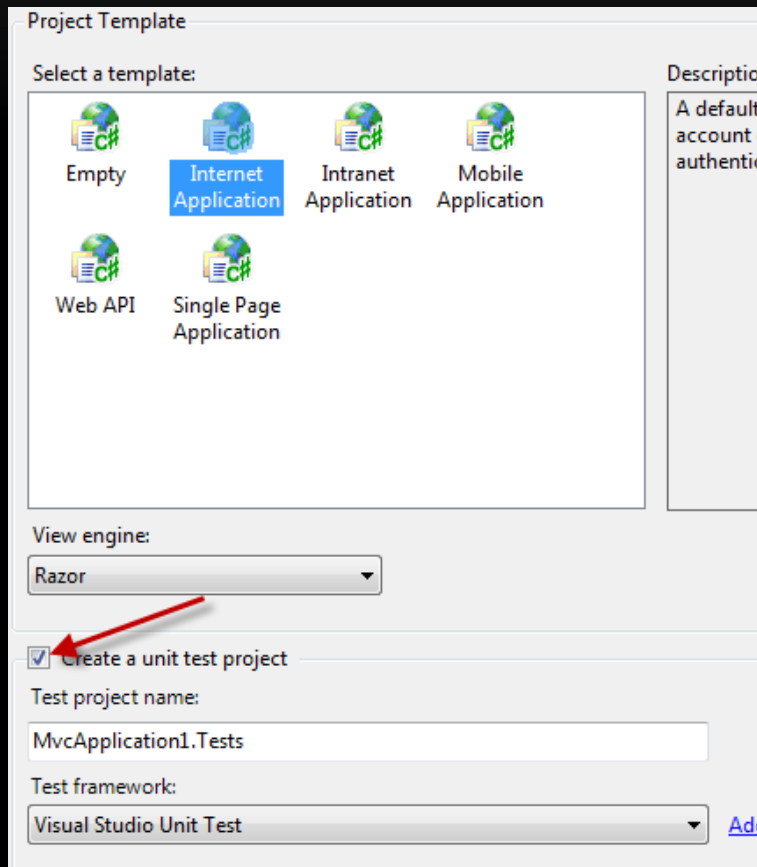
Contact

location pin icon

mobile device icon

cloud upload icon

TESTABILITY IMPROVEMENTS



- MVC Built with testing in mind
- Facilitates TDD/BDD/TED

TEST PROJECT TEMPLATES


- VS – File -> Export Template...
- Copy to:

`%VSInstall%\Common7\IDE\ProjectTemplates\CSharp\Test`

- Hack The Registry
- Run “devenv /installvstemplates”

CREATING A PROJECT TEMPLATE

Export Template Wizard

 **Select Template Options**

Template name:
MVCTestProject

Template description:
Test Project for MVC3 Applications using NUnit, JustMock, MVCContrib

Icon Image:

Preview Image:

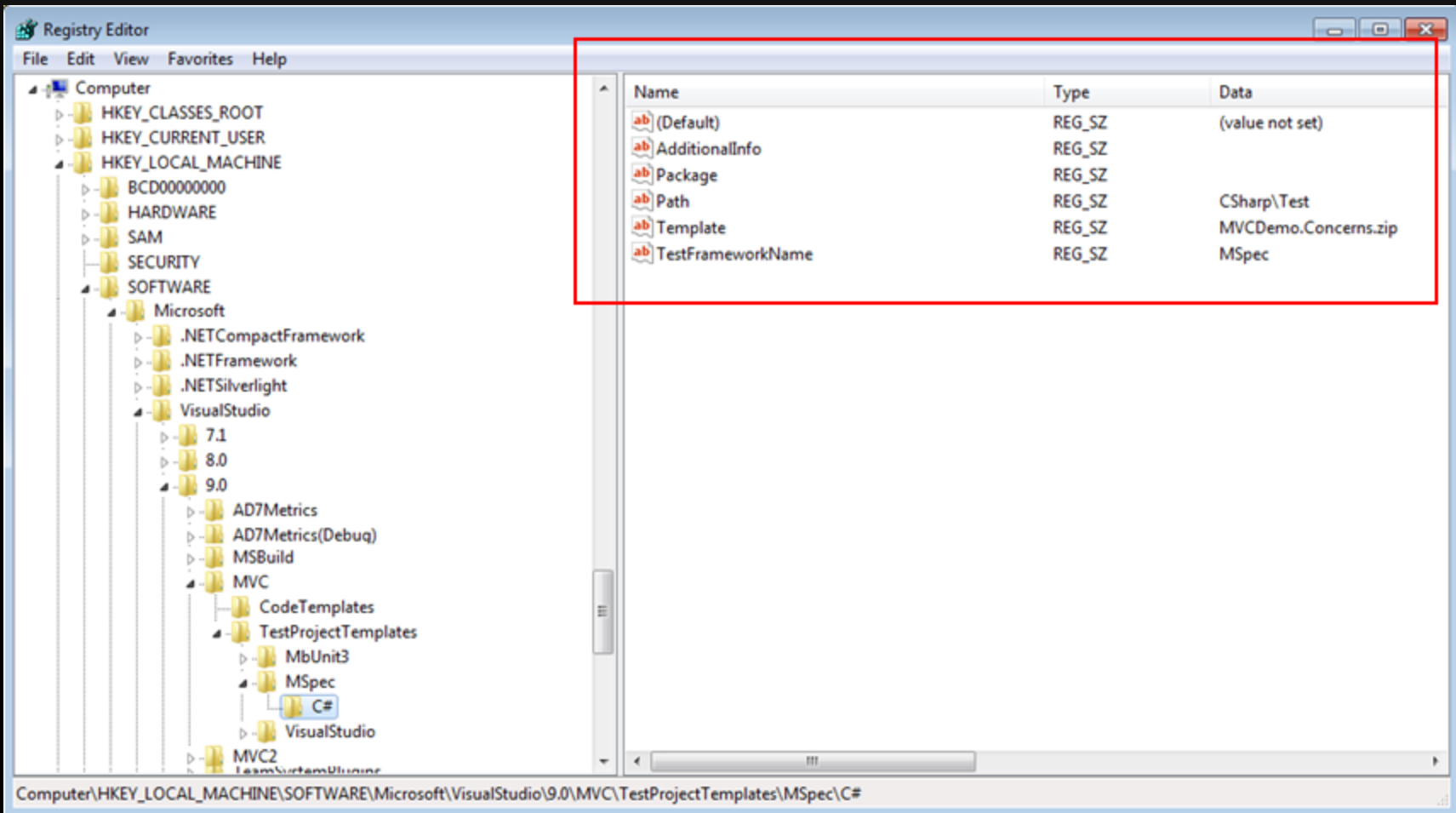
Output location:
C:\Users\Japikse\Documents\Visual Studio 2010\My Exported Templates\MVCTestProject.zip

Automatically import the template into Visual Studio

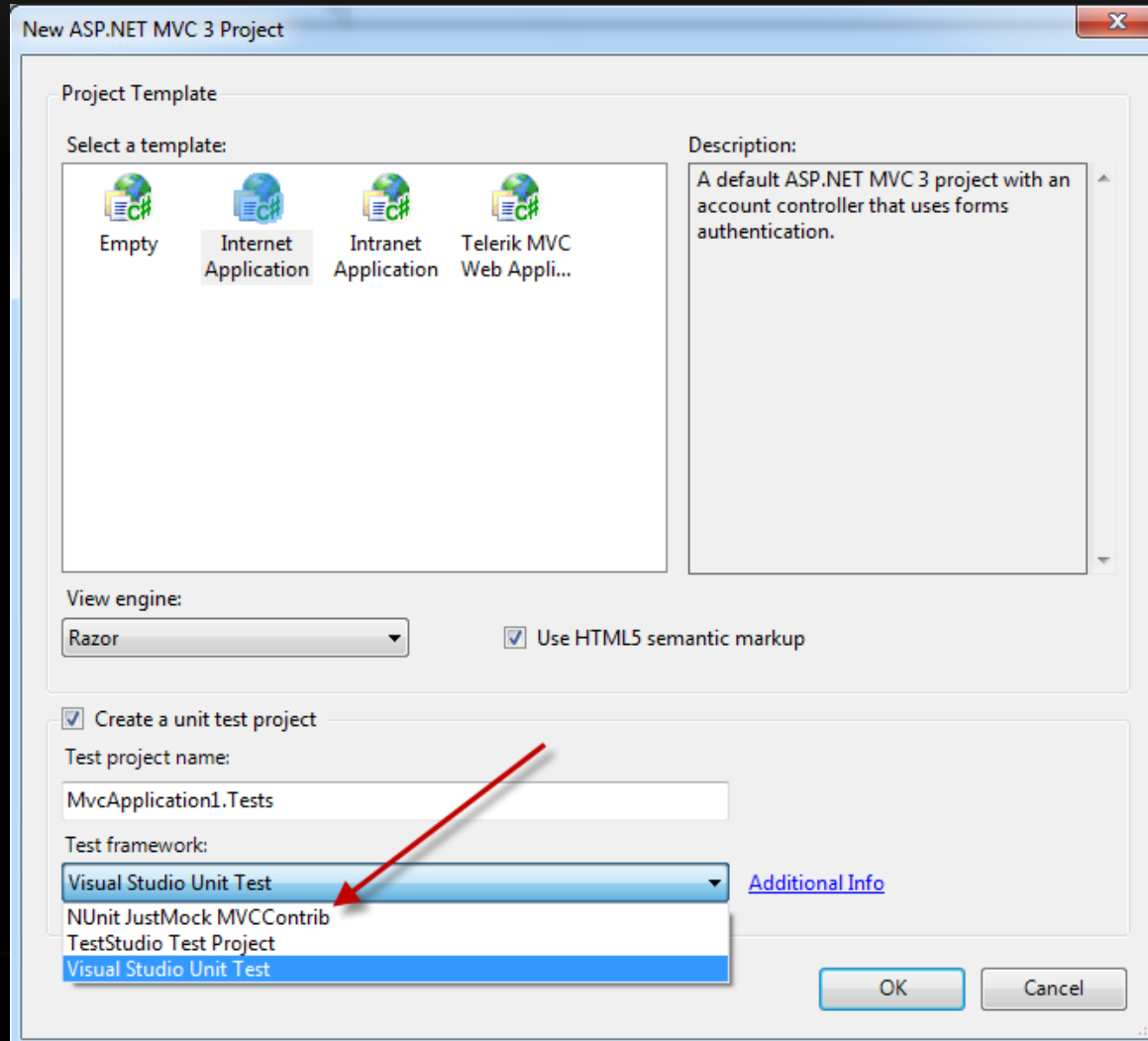
Display an explorer window on the output files folder

< Previous Next > Finish Cancel

TEST PROJECT TEMPLATES REGISTRY HACK



UPDATED TEST PROJECT OPTIONS



ADDING CONTROLLERS (MVC3)

Add Controller

Controller name:
Default1 Controller

Scaffolding options

Template:
Controller with read/write actions and views, using Entity Framework

Model class:
Customer (MVC3Basic.Models)

Data context class:
MVC3Basic.Models.MVC3BasicCont

Views:
Razor (CSHTML)

Advanced Options

- Reference script libraries
- Use a layout or master page:
[Empty text box] ...
(Leave empty if it is set in a Razor _viewstart file)
- ContentPlaceholder ID:
MainContent

OK Cancel

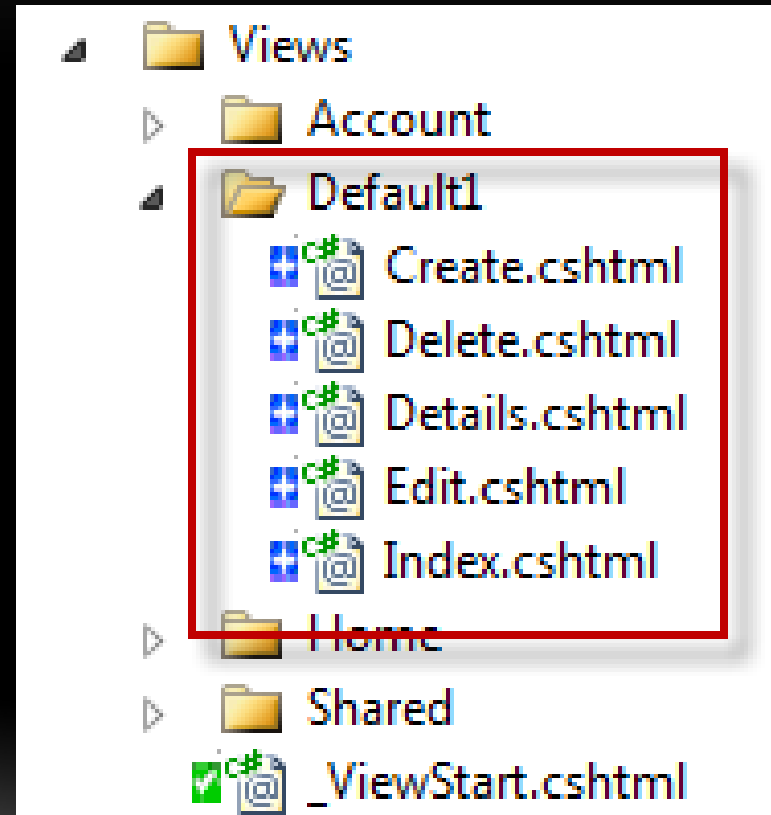
ADDING CONTROLLERS (MVC4)

The screenshot shows the 'Add Controller' dialog box with the following details:

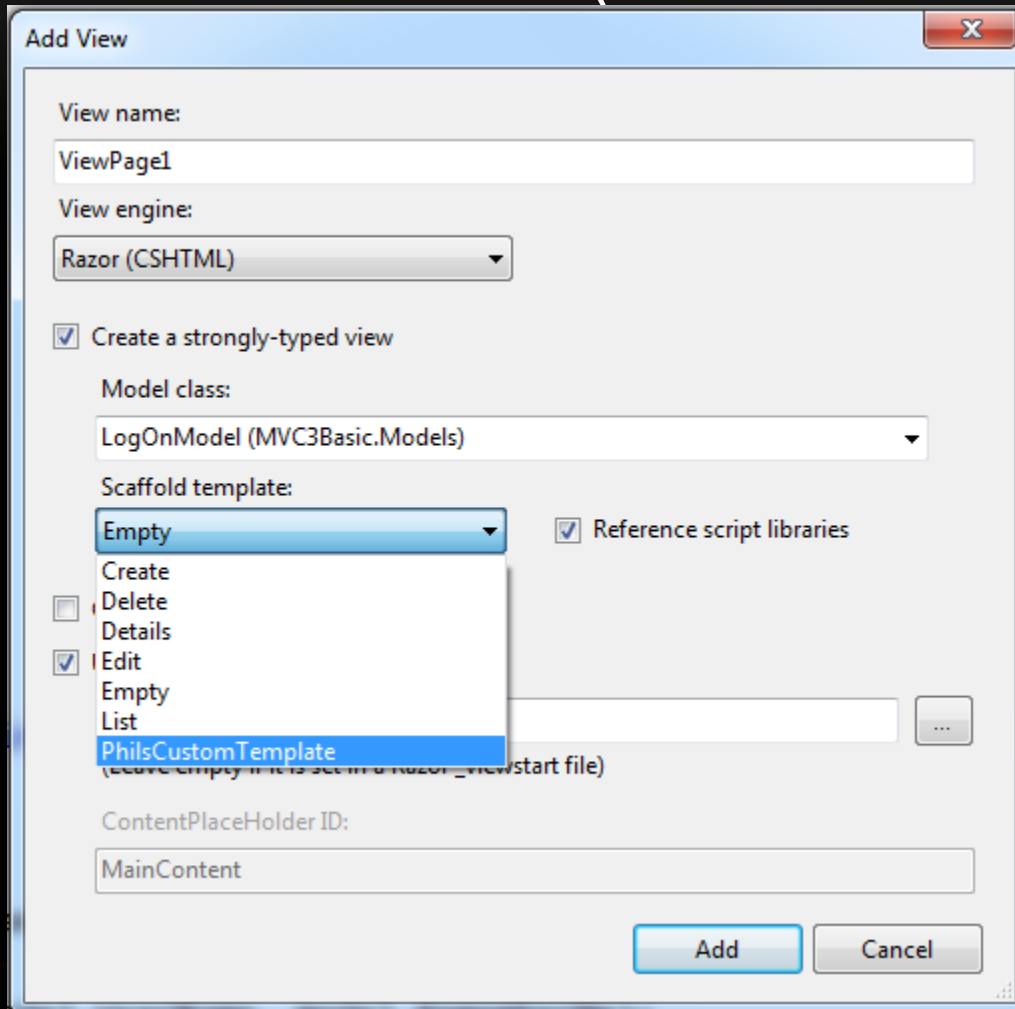
- Controller name:** Default1Controller
- Scaffolding options:**
 - Template:** A dropdown menu is open, showing the following options:
 - Empty controller (selected)
 - Controller with read/write actions and views, using Entity Framework
 - Ajax grid controller with read/write actions and views, using Entity Framework
 - Controller with empty read/write actions
 - Empty API controller
 - API controller with empty read/write actions
 - Single Page Application with read/write actions and views, using Entity Framework
- Views:** None
- Buttons:** Add, Cancel, and an 'Advanced Options...' button.

ADDING READ/WRITE CONTROLLERS

```
public class Default1Controller : Controller
{
    public ActionResult Index() { //Code }
    // GET: /Default1/Details/5
    public ActionResult Details(int id) { //Code }
    // GET: /Default1/Create
    public ActionResult Create() { //Code }
    // POST: /Default1/Create
    [HttpPost]
    public ActionResult Create(Customer customer) { //Code }
    // GET: /Default1/Edit/5
    public ActionResult Edit(int id) { //Code }
    // POST: /Default1/Edit/5
    [HttpPost]
    public ActionResult Edit(Customer customer) { //Code }
    // GET: /Default1/Delete/5
    public ActionResult Delete(int id) { //Code }
    // POST: /Default1/Delete/5
    [HttpPost, ActionName("Delete")]
    public ActionResult DeleteConfirmed(int id) { //Code }
}
```

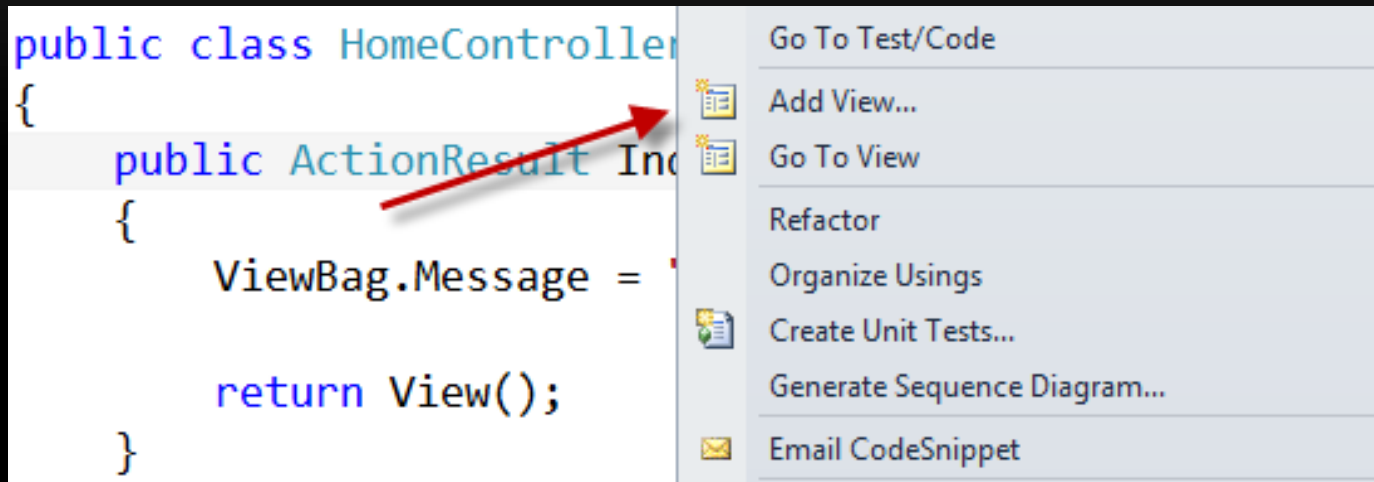


ADDING VIEWS (SOLUTION EXPLORER)



ADDING VIEW (CONTROLLER CODE)

```
public class HomeController
{
    public ActionResult Index()
    {
        ViewBag.Message = "Home Page";
        return View();
    }
}
```



The screenshot shows a code editor with a context menu open over the `Index()` method. A red arrow points to the `Add View...` option in the menu. The menu items are:

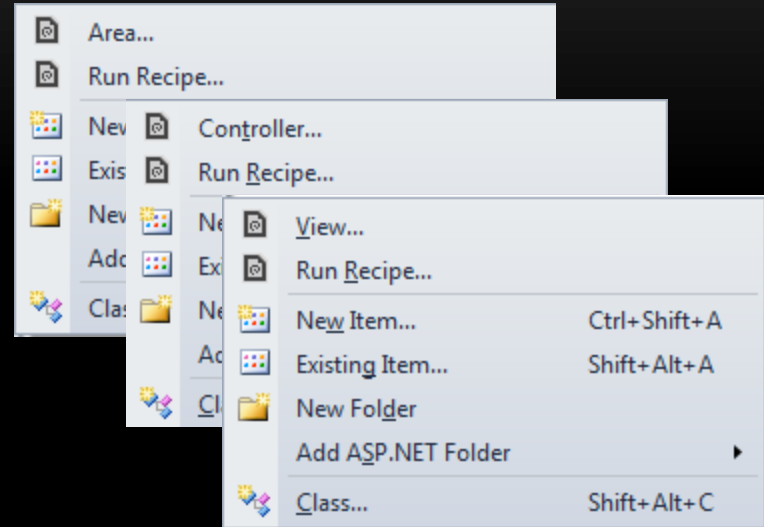
- Go To Test/Code
- Add View...
- Go To View
- Refactor
- Organize Usings
- Create Unit Tests...
- Generate Sequence Diagram...
- Email CodeSnippet

“CREATE” VIEW EXAMPLE

```
@model MVC3Basic.Models.Customer
@{ ViewBag.Title = "Create"; }
<h2>Create</h2>
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")" type="text/javascript"></script>
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Customer</legend>
        <div class="editor-label">@Html.LabelFor(model => model.Name)</div>
        <div class="editor-field">@Html.EditorFor(model => model.Name)
            @Html.ValidationMessageFor(model => model.Name)</div>
        <p><input type="submit" value="Create" /></p>
    </fieldset>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
```

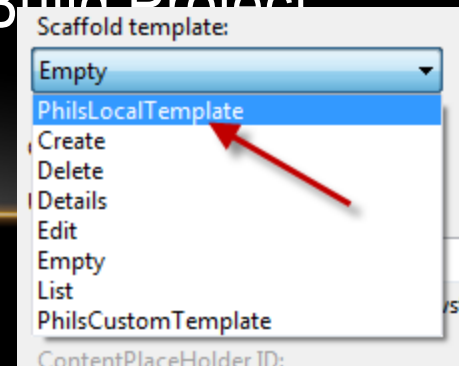
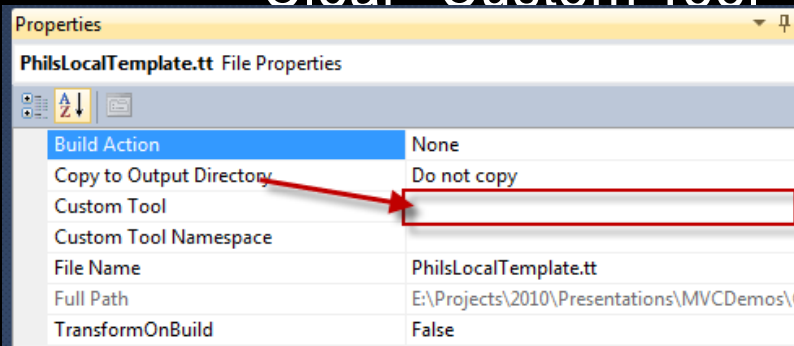
ASP.NET MVC4 RECIPES

- NuGet packages for
 - Area
 - Controller
 - View
- Unique Per Solution
- Shareable via source code control



CUSTOM VIEW TEMPLATES

- System
 - %VSInstall%\Common7\IDE\ItemTemplates\CSharp\Web\MVC 3\CodeTemplates\AddView\CSHTML
- Project
 - Portable through Source Control
 - Create Folder CodeTemplates\AddView\CSHTML
 - Add T4 Template Files
 - Clear “Custom Tool” field => Build Project



ADD MVCSCAFFOLDING FROM NUGET

The screenshot shows the Visual Studio interface with the Package Manager Console and Library Package Manager open. The Package Manager Console is at the top, showing the command `PM> install-package mvc scaffolding` and its output. The Library Package Manager is at the bottom, showing the Package Manager Console icon highlighted.

Tools VisualSVN Architecture Test Analyze Window Help

Package Manager Console

Package source: NuGet official package source Default project: MVC3Basic

```
PM> install-package mvc scaffolding
Attempting to resolve dependency 'T4Scaffolding'.
Attempting to resolve dependency 'EntityFramework (>= 4.1.10311.0)'.
'MvcScaffolding 1.0.6' already installed.
Successfully removed 'EntityFramework 4.1.10331.0' from MVC3Basic.
Successfully added 'EntityFramework 4.1.10715.0' to MVC3Basic.
Successfully added 'T4Scaffolding 1.0.5' to MVC3Basic.
Successfully added 'MvcScaffolding 1.0.6' to MVC3Basic.

PM> |
```

Library Package Manager

- Package Manager Console
- Manage NuGet Packages for Solution...
- Package Visualizer
- Package Manager Settings

CREATE CUSTOM SCAFFOLDER

Package Manager Console

Package source: NuGet official package source



Default project: MVC3Basic

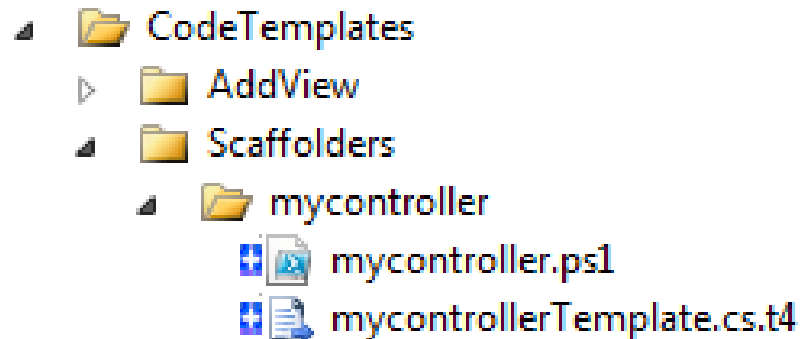


```
PM> scaffold customscaffolder mycontroller
```

```
Added scaffolder script 'CodeTemplates\Scaffolders\mycontroller\mycontroller.ps1'
```

```
Added scaffolder template 'CodeTemplates\Scaffolders\mycontroller\mycontrollerTemplate.cs.t4'
```

```
PM>
```



RAZOR

RAZOR VIEW ENGINE

- Razor views don't extend `System.Web.Page`
 - No more `Page` directive
 - Easier to test
- Cleaner Embedding of Server Side Code
- Supports everything you'd expect from `WebFormsViewEngine`

RAZOR VIEWS

```
return View();
```

```
-----  
@{  
    //Layout =  
    "~/Views/Shared/_BlankLayout.cshtml";  
    ViewBag.Title = "Home Page";  
}  
  
<h2>@ViewBag.Message</h2>  
<p>  
    To learn more about ASP.NET MVC visit  
    <a href="http://asp.net/mvc"  
    title="ASP.NET MVC  
Website">http://asp.net/mvc</a>.  
</p>
```

```
return PartialView()
```

```
-----  
<h2>@ViewBag.Message</h2>  
<p>  
    To learn more about ASP.NET MVC visit  
    <a href="http://asp.net/mvc"  
    title="ASP.NET MVC  
Website">http://asp.net/mvc</a>.  
</p>
```


LESS TYPING, EASIER TO READ

Hello, @world.

```
@for (int i = 0; i <5; i++)
{
    @:Straight Text
    <div>Value:@i</div>
    <text>
    Lines without HTML tag
    </text>
    <br/>
}
```

```
<a
href="/Products/Detail/@p.ProductID">
    @p.ProductName</a>
```

Hello, <%= world %>.

```
<% for (int i = 0; i < 5; i++)
    {
        %>
        Straight Text
        <div>Value:<%=i%></div>
        Lines Without HTML tag
        <br />
    <% } %>
```

```
<
href="/Products/Detail/<%=p.ProductI
D%>"><%= p.ProductName</a>
```

LESS TYPING, EASIER TO READ

```
@{
  //Code Block
  var foo = "Foo";
  var bar = "Bar";
  var htmlString =
"<ul><li>one</li><li>two</li></ul>";
}
@foo<br />
@htmlString<br />
@foo.@bar<br />
@Html.Raw(htmlString)
```

```
<%
  //Code Block
  var foo = "Foo";
  var bar = "Bar";
  var htmlString =
"<ul><li>one</li><li>two</li></u
l>";
%>
<%= foo %><br />
<%= htmlString%><br />
<%=foo %>.<%=bar%><br />
<%=htmlString %>
```

SMART @ HANDLING

@{ var foo = "Foo"; }

@*

Multiline Comments

Hi.

*@

Email Address Handling:

foo@foo.com = foo@foo.com

test@foo = test@foo

test@(foo) = testFoo

FUNCTIONS (NON-UI CODE)

```
@functions {  
    public [static] IList<string> SortList(IList<string> strings) {  
        var list = from s in strings orderby s select s;  
        return list.ToList();  
    }  
}
```

```
@{ var myList = new List<string> {"C", "A", "Z", "F"};  
    var sortedList = SortList(myList); //MyFunctions.SortList(myList)  
}  
@foreach (string s in sortedList) {  
    @s&nbsp;  
}
```

HELPERS (UI CODE)

```
@helper ShowDiscountedPrice(
  Customer cust, Decimal price)
{
  if (cust.IsPreferred) {
    @String.Format("{0:C2}", price * .9)
  } else {
    @String.Format("{0:C2}", price)
  }
}
```

DISPLAY TEMPLATE (BOOLEAN.CSHTML)

```
@Html.DisplayFor(model => model.[boolean])
```

```
@model bool?
```

```
@{  
    if (ViewData.ModelMetadata.IsNullableValueType)  
    {  
        if (!Model.HasValue) { @:Unknown }  
        else if (Model.Value) { @:Yes }  
        else { @:No }  
    } else {  
        if ((bool)Model) { @:Yes }  
        else { @:No }  
    }  
}
```

EDITOR TEMPLATE (BOOLEAN.CSHTML)

```
@Html.EditorFor(model => model.[boolean] )
```

```
@model System.Boolean?
```

```
@{  
    var list = new List<SelectListItem>();  
    if (ViewData.ModelMetadata.IsNullableValueType)  
    { list.Add(new SelectListItem { Text = "", Value = "" }); }  
    list.Add(new SelectListItem { Text = "Yes", Value = "true" });  
    list.Add(new SelectListItem { Text = "No", Value = "false" });  
}  
@(Html.Telerik()  
    .DropDownListFor(m => m)  
    .Name(ViewData.TemplateInfo.GetFullHtmlFieldName(string.Empty))  
    .BindTo(list)  
)
```

EXTENDING @HTML

```
@Html.LabelWithRequiredFor(m => m.UserName)
```

```
public static class LabelHelper
{
    public static MvcHtmlString LabelWithRequiredFor<TModel, TValue>(
        this HtmlHelper<TModel> helper, Expression<Func<TModel, TValue>> expression)
    {
        //Code to create the HTML Label
        //Uses Data Annotations
        return MvcHtmlString.Create(label.ToString());
    }
}
```

User name*

RAZOR DELEGATES

```
@{  
    Func<dynamic, object>  
        b = @<strong>@item</strong>;  
}
```

This will be bold: @b("Foo")

SUMMARY

- Visual Studio Tooling speeds Razor Development
- Razor Design Goals:
 - Compact, Expressive, Fluid
 - Easy to Learn
 - Unit Testable
- Razor Features:
 - No Page Directive
 - Helpers and Functions
 - Powerful Templating
 - Extending @Html
 - Delegates

CONTACT ME

- phil@telerik.com
- www.skimedic.com/blog
- www.twitter.com/skimedic
- www.telerik.com/zerotoagile