

Sitecore CMS 6.4-6.5

Integrating an ASP.NET MVC Web Application in Sitecore CMS

A configuration guide for developers

Table of Contents

Chapter 1	Introduction	2
Chapter 2	Creating an ASP.NET MVC 2 Web Application	3
2.1	Prerequisites.....	4
2.2	Creating an ASP.NET MVC 2 Web Application Project	5
2.2.1	Moving the Project Files	6
2.2.2	Completing the Application Configuration	7
Chapter 3	Sitecore Configuration with ASP.NET MVC 2	9
3.1	Web.config Configuration	10
3.1.1	Configuring the Web.config File for .NET Framework 4.0.....	10
3.1.2	Configuring the Web.config File for .NET Framework 3.5.....	12
3.2	Configuring Internet Information Services (IIS)	14
3.2.1	.NET Framework version of the Application Pool.....	14
3.2.2	Managed Pipeline Mode of the Application Pool.....	14
Chapter 4	Creating an ASP.NET MVC 3 Web Application	15
4.1	Prerequisites.....	16
4.2	Creating an ASP.NET MVC 3 Web Application Project	17
4.2.1	Moving the Project Files	18
4.2.2	Completing the Application Configuration	19
Chapter 5	Sitecore Configuration with ASP.NET MVC 3	21
5.1	Web.config Configuration	22
5.1.1	Configuring Web.config for .NET Framework 4.0	22
5.2	Configuring Internet Information Services (IIS)	25
5.2.1	.NET Framework version of the Application Pool.....	25
5.2.2	Managed Pipeline Mode of the Application Pool.....	25

Chapter 1

Introduction

This document describes how to configure the environment where a Sitecore website and an ASP.NET MVC web application run under the same hostname.

For more information about the ASP.NET MVC framework, visit the <http://www.asp.net/mvc/> website.

Chapter 2

Creating an ASP.NET MVC 2 Web Application

This chapter describes the process of creating an ASP.NET MVC 2 Web Application that can run in parallel with a Sitecore website.

2.1 Prerequisites

The instructions in this chapter assume that you have the following components installed on your computer:

- .NET Framework 3.5 or .NET Framework 4.0.
- In the appropriate application pool, the **Managed Pipeline Mode** must be set to **Integrated**.
- Sitecore CMS 6.4.0 or later.
- Microsoft Visual Studio 2008 or 2010.
- Microsoft ASP.NET MVC 2 Framework.

By default the ASP.NET MVC 2 Framework is installed with Microsoft Visual Studio 2010. If you use Microsoft Visual Studio 2008, you must download and install ASP.NET MVC 2 Framework separately. The framework is available at the following link:

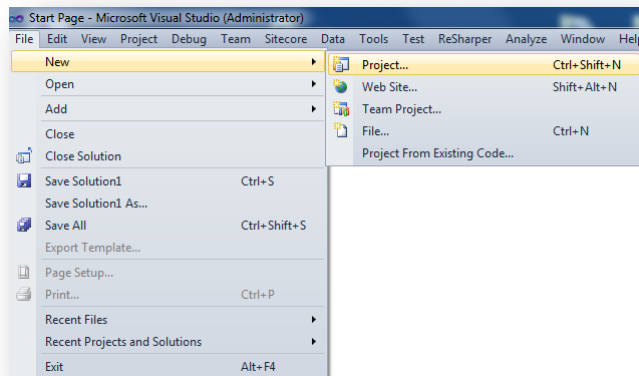
<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=c9ba1fe1-3ba8-439a-9e21-def90a8615a9&displaylang=en>

2.2 Creating an ASP.NET MVC 2 Web Application Project

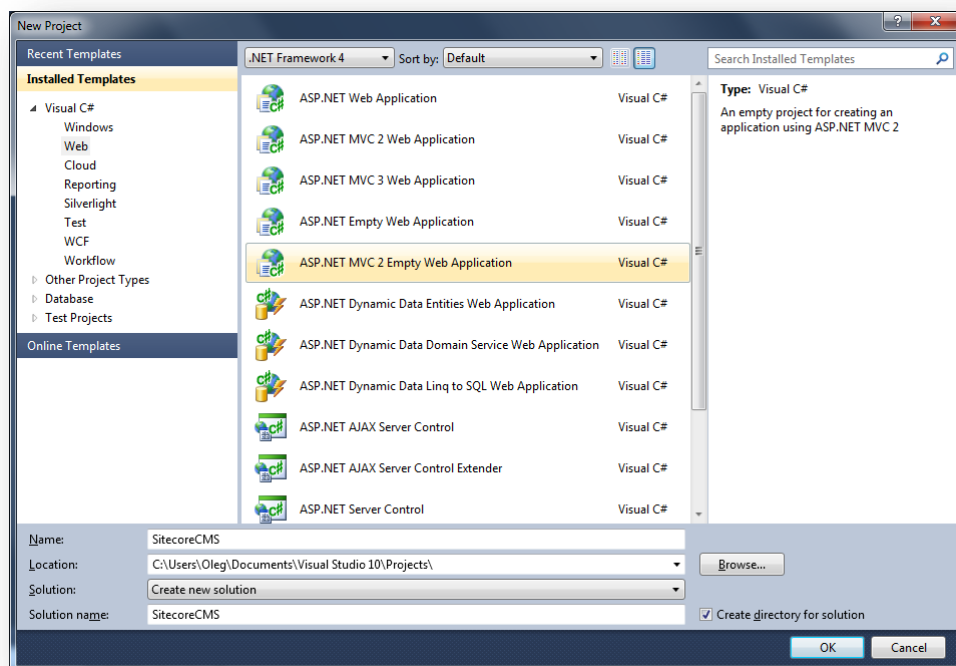
You need to create a web application project to generate the files you need for your Sitecore project. When you create the project it actually generates more files than you need. In later sections you move the required files to the appropriate Sitecore folder and delete the extra files.

To create the web application project:

1. Open Visual Studio 2008 or 2010.
2. Click **File, New, Project**:



3. Select the **ASP.NET MVC 2 Empty Web Application** template and provide a name for the new project:



Important

In Visual Studio 2010 you can create ASP.NET MVC 2 Web Application under .NET Framework 3.5 and 4.0.

4. Click **OK**.

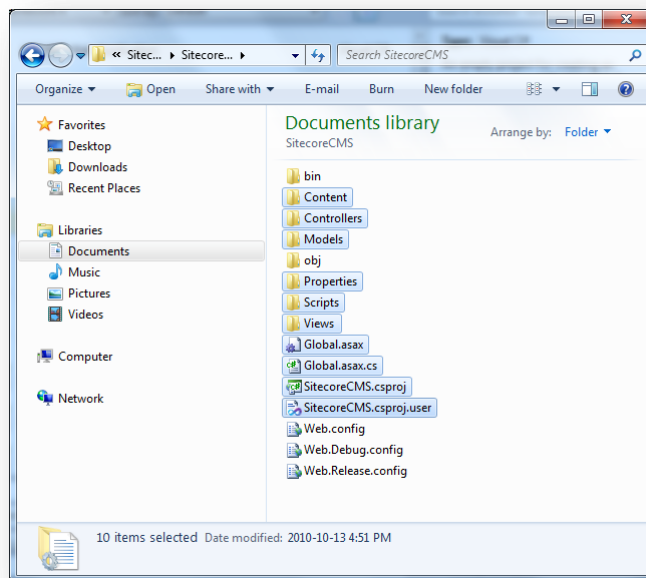
Visual Studio creates the related files in the folder with the name you specified. Some files are not required, we will remove them later.

5. Close the solution or exit Visual Studio.

2.2.1 Moving the Project Files

Since you do not need all the files that Visual Studio generates, you move only the files you need to the Sitecore web root.

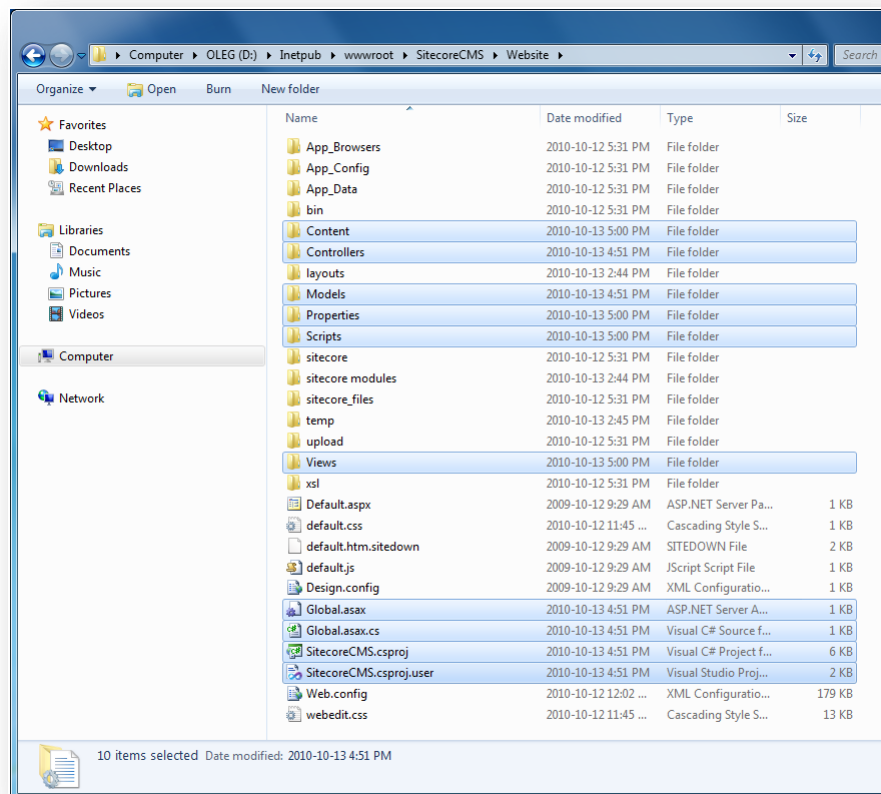
1. In Windows Explorer, locate the project folder that was generated by Visual Studio (as shown below).



2. Move or copy the following files and folders to the Sitecore web root:

- o Content folder
- o Controllers folder
- o Models folder
- o Properties folder
- o Scripts folder
- o Views folder
- o Global.asax file
- o Global.asax.cs file
- o *.csproj file

- o *.csproj.user file



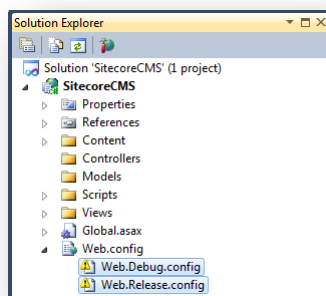
Note

The default `Global.asax` file must be replaced. If you have customized your `Global.asax` file, you must merge it.

2.2.2 Completing the Application Configuration

To complete the configuration:

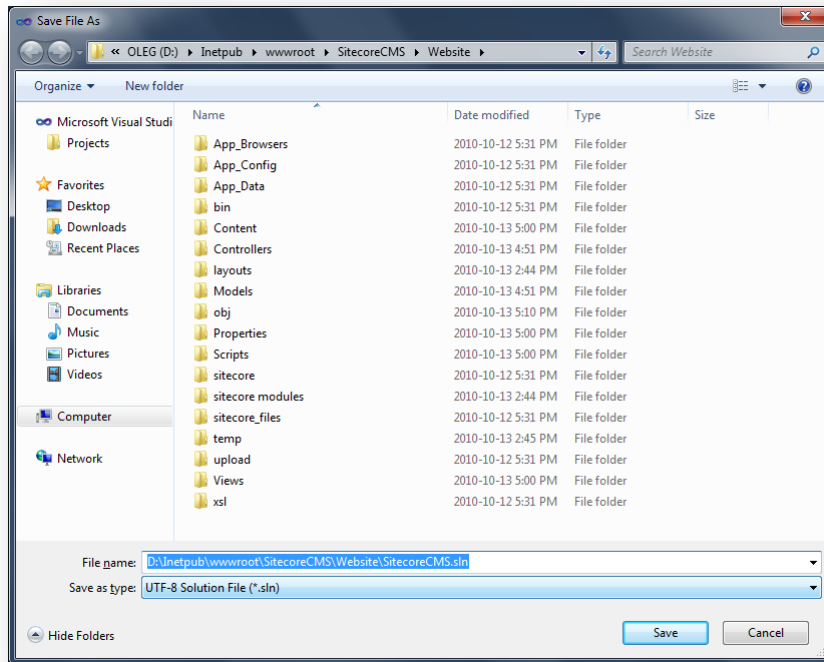
1. In the Sitecore web root, double-click the `.csproj` file to open the project in Visual Studio.



Note

If you are using the Visual Studio 2010, expand the `web.config` node in Solution Explorer. Delete the `Web.Debug.config` and `Web.Release.config` definitions from the project.

2. In Visual Studio, on the **Standard Tool** bar, click **Save All**. Visual Studio opens the **Save File As** dialog box. Specify the name of your solution and click **Save**.



Chapter 3

Sitecore Configuration with ASP.NET MVC 2

This chapter describes the Sitecore CMS configuration required for the ASP.NET MVC 2 Web Application to run in parallel with the Sitecore website.

3.1 Web.config Configuration

Visual Studio 2010 lets you use the .NET Framework version 4.0 or version 3.5 for creating the ASP.NET MVC web applications, while Visual Studio 2008 only supports version 3.5.

In general terms, Sitecore CMS is a mixed ASP.NET Web Form. It is necessary to make the following changes in the `Web.config` file to use the mixed ASP.NET Web Form and an ASP.NET MVC project in parallel.

3.1.1 Configuring the Web.config File for .NET Framework 4.0

This section contains the list of changes that you must make in the `Web.config` file:

1. Remove from the `<configuration><configSections>` section:

```
<sectionGroup name="system.web.extensions"
type="System.Web.Configuration.SystemWebExtensionsSectionGroup, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
</sectionGroup>

<sectionGroup name="scripting" type="System.Web.Configuration.ScriptingSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
</sectionGroup>

<sectionGroup name="webServices"
type="System.Web.Configuration.ScriptingWebServicesSectionGroup, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
</sectionGroup>
```

2. Modify the `<configuration><system.webServer><modules>` section:

```
<modules runAllManagedModulesForAllRequests="true">
```

3. Remove from the `<configuration><system.webServer><modules>` section:

```
<remove name="ScriptModule"/>

<add name="ScriptModule" preCondition="managedHandler"
type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
```

4. Remove from the `<configuration><system.webServer><handlers>` section:

```
<remove name="WebServiceHandlerFactory-Integrated" />
<remove name="ScriptHandlerFactory" />
<remove name="ScriptHandlerFactoryAppServices" />
<remove name="ScriptResource" />

<add name="ScriptHandlerFactory" verb="*" path="*.asmx" preCondition="integratedMode"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<add name="ScriptHandlerFactoryAppServices" verb="*" path="* AppService.axd"
preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<add name="ScriptResource" verb="GET,HEAD" path="ScriptResource.axd"
preCondition="integratedMode" type="System.Web.Handlers.ScriptResourceHandler,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
```

5. Add to the `<configuration><system.web><pages>` section:

```
<namespaces>
<add namespace="System.Web.Mvc"/>
<add namespace="System.Web.Mvc.Ajax"/>
<add namespace="System.Web.Mvc.Html" />
<add namespace="System.Web.Routing"/>
<add namespace="System.Linq"/>
<add namespace="System.Collections.Generic"/>
</namespaces>
```

6. Remove from the `<configuration><system.web><httpModules>` section:

```
<add name="ScriptModule" type="System.Web.Handlers.ScriptModule, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
```

7. Remove from the <configuration><system.web><httpHandlers> section:

```
<remove verb="*" path="*.asmx" />

<add verb="*" path="*.asmx" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<add verb="*" path="*_AppService.axd" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<add verb="GET,HEAD" path="ScriptResource.axd" validate="false"
type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
```

8. Change in the <configuration>< system.web> section the following lines:

```
<compilation defaultLanguage="c#" debug="false">
  <assemblies>
    <add assembly="System.Core, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
    <add assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Xml.Linq, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
    <add assembly="System.Data.DataSetExtensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
    <add assembly="System.Data.Linq, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
  </assemblies>
</compilation>
```

To the following new ones:

```
<compilation defaultLanguage="c#" debug="false" targetFramework="4.0">
  <assemblies>
    <add assembly="System.Web.Abstractions, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Web.Routing, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Web.Mvc, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Data.Linq, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
  </assemblies>
</compilation>
```

9. In the <configuration><sitecore><settings><setting name="IgnoreUrlPrefixes" > setting add the root value of your MVC router map that is used as the URL, so that Sitecore CMS ignores MVC requests. For example, if you have the following page that uses ASP.NET MVC <http://www.SitecoreCMS.com/Data>, you must add the /Data value to the setting:

```
<setting name="IgnoreUrlPrefixes" value="/Data|/sitecore/default.aspx .../>
```

10. Remove from the <configuration> section:

```
<system.codedom>
  <compilers>
    <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CSharp.CSharpCodeProvider, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" warningLevel="4">
      <providerOption name="CompilerVersion" value="v3.5"/>
      <providerOption name="WarnAsError" value="false"/>
    </compiler>
  </compilers>
</system.codedom>

<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1" appliesTo="v2.0.50727">
    <dependentAssembly>
```

```

        <assemblyIdentity name="System.Web.Extensions" publicKeyToken="31bf3856ad364e35"/>
        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="3.5.0.0"/>
      </dependentAssembly>
    </dependentAssembly>
    <assemblyIdentity name="System.Web.Extensions.Design"
publicKeyToken="31bf3856ad364e35"/>
    <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="3.5.0.0"/>
  </dependentAssembly>
</assemblyBinding>
</runtime>

```

3.1.2 Configuring the Web.config File for .NET Framework 3.5

This section contains the list of changes that you must make in the Web.config file:

1. Modify the <configuration><system.webServer><modules> section:

```
<modules runAllManagedModulesForAllRequests="true">
```

2. Add to the <configuration><system.webServer><modules> section:

```

<remove name="UrlRoutingModule" />

<add name="UrlRoutingModule" type="System.Web.Routing.UrlRoutingModule,
System.Web.Routing, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

```

3. Add to the <configuration><system.webServer><handlers> section:

```

<add name="UrlRoutingModule" preCondition="integratedMode" verb="*"
path="UrlRoutingModule.ashx" type="System.Web.HttpForbiddenHandler, System.Web, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

```

4. Add to the <configuration><system.web><pages> section:

```

<namespaces>
  <add namespace="System.Web.Mvc"/>
  <add namespace="System.Web.Mvc.Ajax"/>
  <add namespace="System.Web.Mvc.Html"/>
  <add namespace="System.Web.Routing"/>
  <add namespace="System.Linq"/>
  <add namespace="System.Collections.Generic"/>
</namespaces>

```

5. Add to the <configuration><system.web><httpModules> section:

```

<add name="UrlRoutingModule" type="System.Web.Routing.UrlRoutingModule,
System.Web.Routing, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

```

6. Add to the the <configuration>< system.web> section:

```

<compilation defaultLanguage="c#" debug="false">
  <assemblies>
    <add assembly="System.Web.Abstractions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Web.Routing, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Web.Mvc, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Xml.Linq, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
  </assemblies>
</compilation>

```

7. In the <configuration><sitecore><settings><setting name="IgnoreUrlPrefixes" > setting add the root value of your MVC router map that is used as the URL, so that Sitecore CMS ignores MVC requests. For example, if you have the following page that uses ASP.NET MVC <http://www.SitecoreCMS.com/Data>, you must add the /Data value to the setting:

```
<setting name="IgnoreUrlPrefixes" value="/Data|/sitecore/default.aspx .../>
```

8. Remove from the <configuration> section:

```
<system.codedom>
```

```
<compilers>
  <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CSharp.CSharpCodeProvider,System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" warningLevel="4">
    <providerOption name="CompilerVersion" value="v3.5"/>
    <providerOption name="WarnAsError" value="false"/>
  </compiler>
</compilers>
</system.codedom>

<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1" appliesTo="v2.0.50727">
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Extensions" publicKeyToken="31bf3856ad364e35"/>
      <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="3.5.0.0"/>
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Extensions.Design"
publicKeyToken="31bf3856ad364e35"/>
      <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="3.5.0.0"/>
    </dependentAssembly>
  </assemblyBinding>
</runtime>
```

3.2 Configuring Internet Information Services (IIS)

You must change the configuration of IIS to make Sitecore work with ASP.NET MVC 2.

3.2.1 .NET Framework version of the Application Pool

If the ASP.NET MVC Web Application uses .NET Framework 4.0 as a target framework, you must switch the Sitecore CMS application pool to .NET Framework 4.0.

If the ASP.NET MVC Web Application uses .NET Framework 3.5 as a target framework, you must switch the Sitecore CMS application pool to .NET Framework 2.0.

3.2.2 Managed Pipeline Mode of the Application Pool

We recommend that you switch the **Managed Pipeline Mode** to **Integrated**.

An ASP.NET MVC Web Application based on .NET Framework 4.0 or 3.5 does not require any special configuration when running in the Integrated pipeline mode under IIS 7.0 or higher.

If you use the Classic pipeline mode under IIS 7.5 or earlier, you must perform special configuration to use an ASP.NET MVC Web Application. To read on how to make this configuration, refer to the Microsoft documentation: <http://www.asp.net/mvc/tutorials/using-asp-net-mvc-with-different-versions-of-iis-cs>.

Chapter 4

Creating an ASP.NET MVC 3 Web Application

This chapter describes the process of creating an ASP.NET MVC 3 Web Application that can run in parallel with a Sitecore website.

4.1 Prerequisites

The instructions in this chapter assume that you have the following components installed on your computer:

- .NET Framework 4.0.
- In the appropriate application pool, the **Managed Pipeline Mode** must be set to **Integrated**.
- Sitecore CMS 6.4.0 or later.
- Microsoft Visual Studio 2010.
- Microsoft ASP.NET MVC 3 Framework.

By default the ASP.NET MVC 3 Framework is not installed with Microsoft Visual Studio 2010. You must download and install ASP.NET MVC 3 Framework separately. The framework is available at the following link:

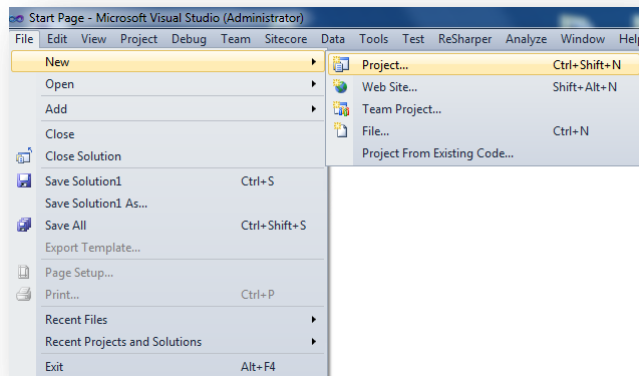
<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=d2928bc1-f48c-4e95-a064-2a455a22c8f6>

4.2 Creating an ASP.NET MVC 3 Web Application Project

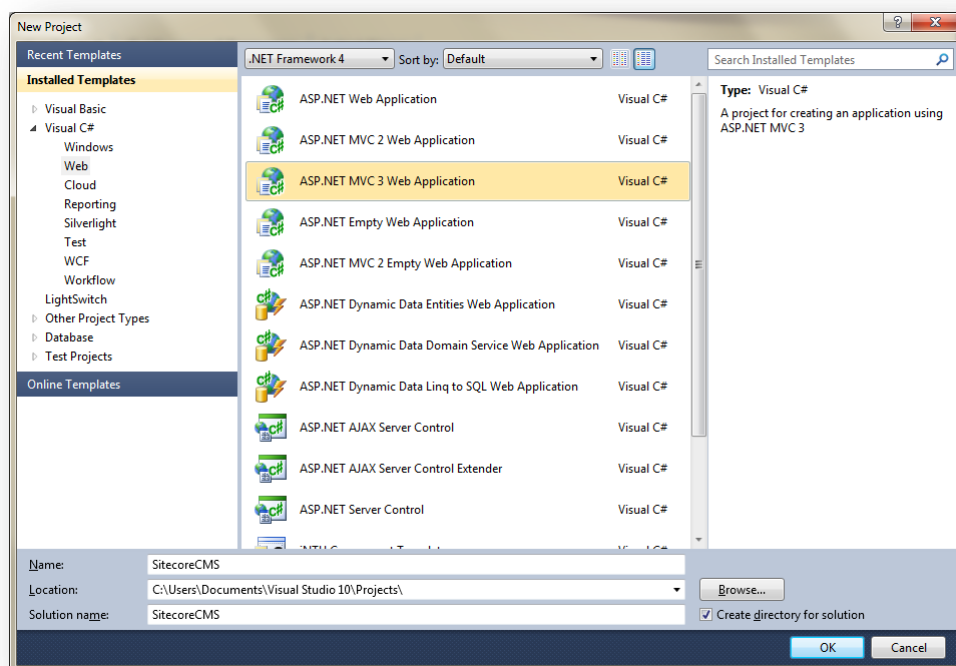
You need to create a web application project to generate the files you need for your Sitecore project. When you create the project it actually generates more files than you need. In later sections you move the required files to the appropriate Sitecore folder and delete the extra files.

To create the web application project:

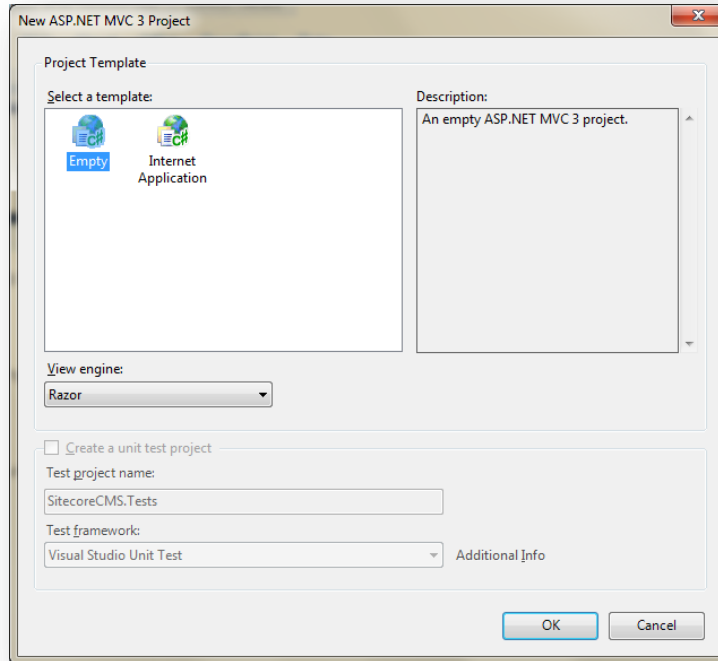
1. Open Visual Studio 2010.
2. Click **File, New, Project**:



3. Select the **ASP.NET MVC 3 Web Application** template and enter a name for the new project:



4. Click **OK**. Project template dialog box appears.
5. Select **Empty** project template and select View engine: **ASPX** or **Razor**.

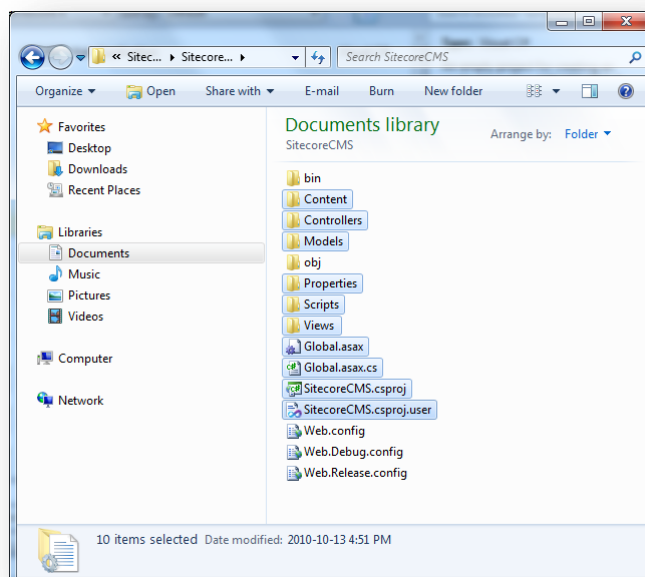


6. Click **OK**. Visual Studio creates the related files in the folder with the name you specified. Some files are not required, you will remove them later.
7. Close the solution or exit Visual Studio.

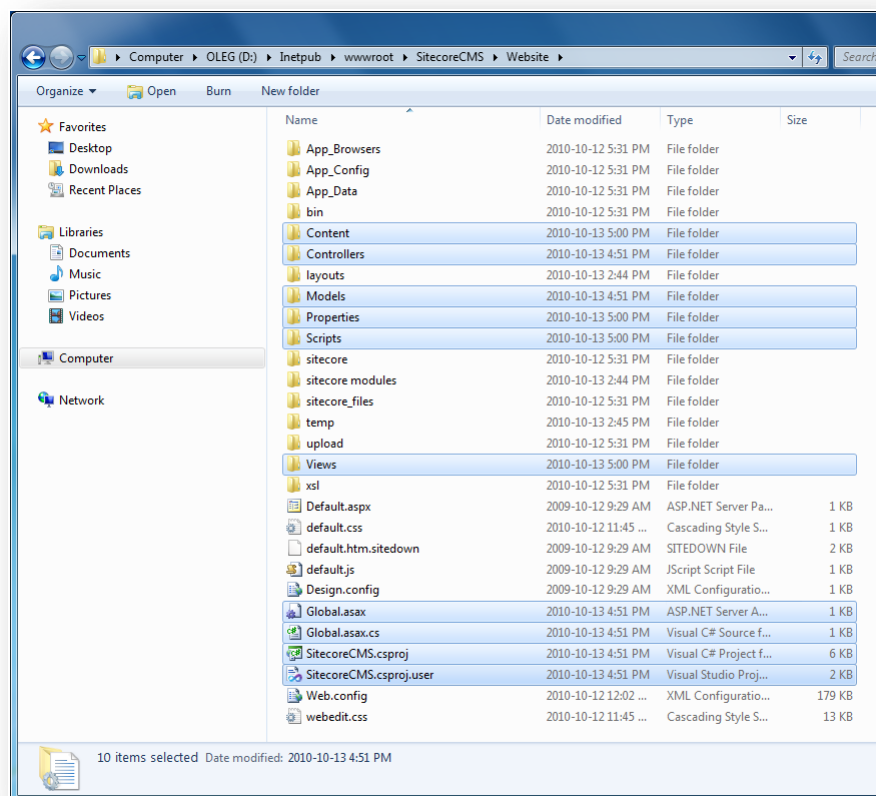
4.2.1 Moving the Project Files

Since you do not need all the files that Visual Studio generates, you move only the files you need to the Sitecore web root.

1. In Windows Explorer, locate the project folder that was generated by Visual Studio (as shown below).



2. Move or copy the following files and folders to the Sitecore web root:
 - Content folder
 - Controllers folder
 - Models folder
 - Properties folder
 - Scripts folder
 - Views folder
 - Global.asax file
 - Global.asax.cs file
 - *.csproj file
 - *.csproj.user file

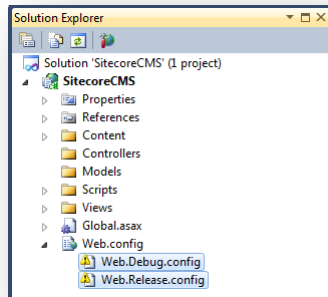
**Note**

The default Global.asax file must be replaced. If you have customized your Global.asax file, you must merge it.

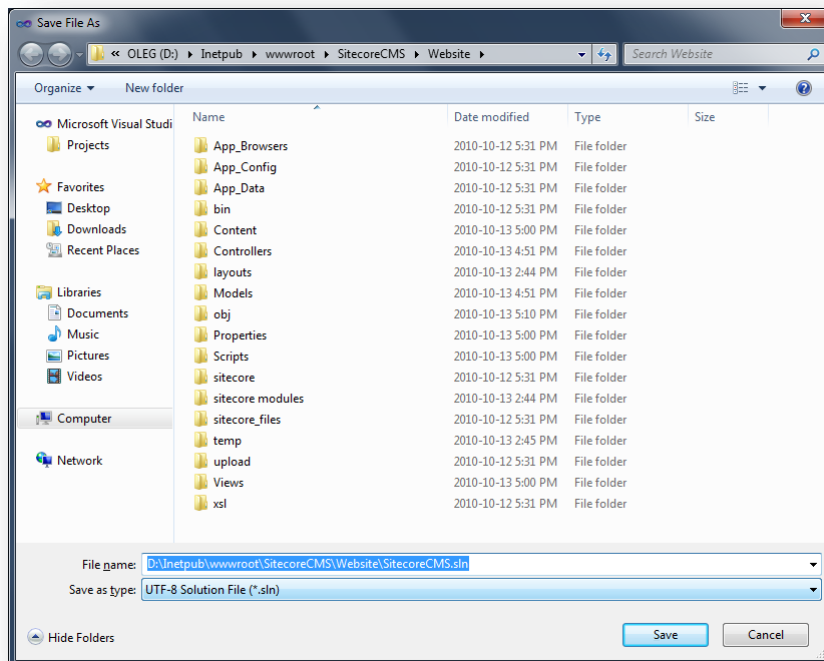
4.2.2 Completing the Application Configuration

To complete the configuration:

1. In the Sitecore web root, double-click the .csproj file to open the project in Visual Studio.
2. In Visual Studio expand the web.config node in Solution Explorer. Delete the Web.Debug.config and Web.Release.config definitions from the project.



3. In Visual Studio, on the **Standard Tool** bar, click **Save All**. Visual Studio opens the **Save File As** dialog box. Enter the name of your solution and click **Save**.



Chapter 5

Sitecore Configuration with ASP.NET MVC 3

This chapter describes the Sitecore CMS configuration required for the ASP.NET MVC 3 Web Application to run in parallel with the Sitecore website.

5.1 Web.config Configuration

Visual Studio 2010 lets you use the .NET Framework version 4.0 only for creating the ASP.NET MVC 3 web applications.

In general terms, Sitecore CMS is a mixed ASP.NET Web Form. It is necessary to make the following changes in the `Web.config` file to use the mixed ASP.NET Web Form and an ASP.NET MVC project in parallel.

5.1.1 Configuring Web.config for .NET Framework 4.0

This section contains the list of changes that you must make in the `Web.config` file:

1. In the `<configuration><configSections>` section, remove the following strings:

```
<sectionGroup name="system.web.extensions"
type="System.Web.Configuration.SystemWebExtensionsSectionGroup, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
</sectionGroup>

<sectionGroup name="scripting" type="System.Web.Configuration.ScriptingSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
</sectionGroup>

<sectionGroup name="webServices"
type="System.Web.Configuration.ScriptingWebServicesSectionGroup, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
</sectionGroup>
```

2. In the `<configuration><system.webServer><modules>` section, modify the following string:

```
<modules runAllManagedModulesForAllRequests="true">
```

3. In the `<configuration><system.webServer><modules>` section, remove the following strings:

```
<remove name="ScriptModule"/>

<add name="ScriptModule" preCondition="managedHandler"
type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
```

4. In the `<configuration><system.webServer><handlers>` section, remove the following strings:

```
<remove name="WebServiceHandlerFactory-Integrated" />
<remove name="ScriptHandlerFactory" />
<remove name="ScriptHandlerFactoryAppServices" />
<remove name="ScriptResource" />

<add name="ScriptHandlerFactory" verb="*" path="*.asmx" preCondition="integratedMode"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<add name="ScriptHandlerFactoryAppServices" verb="*" path="* AppService.axd"
preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<add name="ScriptResource" verb="GET,HEAD" path="ScriptResource.axd"
preCondition="integratedMode" type="System.Web.Handlers.ScriptResourceHandler,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
```

5. In the `<configuration><system.web><pages>` section, add the following strings:

```
<namespaces>
<add namespace="System.Web.Helpers" />
<add namespace="System.Web.Mvc"/>
<add namespace="System.Web.Mvc.Ajax"/>
<add namespace="System.Web.Mvc.Html" />
<add namespace="System.Web.Routing"/>
<add namespace="System.Web.WebPages"/>
</namespaces>
```

6. In the `<configuration><system.web><httpModules>` section, remove the following strings:

```
<add name="ScriptModule" type="System.Web.Handlers.ScriptModule, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
```

7. In the <configuration><system.web><httpHandlers> section, remove the following strings:

```
<remove verb="*" path="*.asmx" />

<add verb="*" path="*.asmx" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<add verb="*" path="*_AppService.axd" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<add verb="GET,HEAD" path="ScriptResource.axd" validate="false"
type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
```

8. In the <configuration>< system.web> section, replace the following strings:

```
<compilation defaultLanguage="c#" debug="false">
  <assemblies>
    <add assembly="System.Core, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
    <add assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Xml.Linq, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
    <add assembly="System.Data.DataSetExtensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
    <add assembly="System.Data.Linq, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
  </assemblies>
</compilation>
```

With the following new ones:

```
<compilation defaultLanguage="c#" debug="false" targetFramework="4.0">
  <assemblies>
    <add assembly="System.Web.Abstractions, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Web.Helpers, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
    <add assembly="System.Web.Routing, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Web.Mvc, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add assembly="System.Web.WebPages, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
  </assemblies>
</compilation>
```

9. In the <configuration><sitecore><settings><setting name="IgnoreUrlPrefixes" > setting, add the root value of your MVC router map that is used as the URL, so that Sitecore CMS ignores MVC requests. For example, if you have the following page that uses ASP.NET MVC <http://www.SitecoreCMS.com/Data>, you must add the /Data value to the following setting:

```
<setting name="IgnoreUrlPrefixes" value="/Data|/sitecore/default.aspx .../>
```

10. In the <configuration> section, remove the following strings:

```
<system.codedom>
  <compilers>
    <compiler language="c#;cs;sharp" extension=".cs"
type="Microsoft.CSharp.CSharpCodeProvider, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" warningLevel="4">
      <providerOption name="CompilerVersion" value="v3.5"/>
      <providerOption name="WarnAsError" value="false"/>
    </compiler>
  </compilers>
</system.codedom>

<runtime>
```



```
<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1" appliesTo="v2.0.50727">
  <dependentAssembly>
    <assemblyIdentity name="System.Web.Extensions" publicKeyToken="31bf3856ad364e35"/>
    <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="3.5.0.0"/>
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="System.Web.Extensions.Design"
publicKeyToken="31bf3856ad364e35"/>
    <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="3.5.0.0"/>
  </dependentAssembly>
</assemblyBinding>
</runtime>
```

5.2 Configuring Internet Information Services (IIS)

You must change the configuration of IIS to make Sitecore work with ASP.NET MVC.

5.2.1 .NET Framework version of the Application Pool

If the ASP.NET MVC 3 Web Application uses .NET Framework 4.0 as a target framework, you must switch the Sitecore CMS application pool to .NET Framework 4.0.

5.2.2 Managed Pipeline Mode of the Application Pool

We recommend that you switch the **Managed Pipeline Mode** to **Integrated**.

An ASP.NET MVC Web Application based on .NET Framework 4.0 does not require any special configuration when running in the Integrated pipeline mode under IIS 7.0 or higher.

If you use the Classic pipeline mode under IIS 7.5 or earlier, you must perform special configuration to use an ASP.NET MVC Web Application. For more information about this configuration, refer to the Microsoft documentation: <http://www.asp.net/mvc/tutorials/using-asp-net-mvc-with-different-versions-of-iis-cs>.