

Developing and deploying MVC /HTML SDK application to IIS

DigitalOfficePro Inc.

Introduction

This tutorial explain how to create and deploy a MVC4 web application that using command line exe for PowerPoint to HTML5 conversion.

Tested machines:

We tested the HTML5 SDK web application in the following combinations.

Win 8 x64, PowerPoint 2013 x64, IIS 8.0

Win 8.1 x64, PowerPoint 2013 x64, IIS 8.5

Win Server 2008 x64, PowerPoint 2013 x64, IIS 7.0

Requirements

Development machine:

PowerPoint 2007 or later.

Visual Studio 2012.

Install WebDeploy_amd64 for Publishing the App.

Install HTML5Point SDK latest version (4.0.51 later)

.Net 4 or .Net 4.5

Target Machine:

.Net 4 or .Net 4.5

IIS 7 or IIS 8

PowerPoint 2007 or later.

Install HTML5Point SDK latest version (4.0.51 or later)

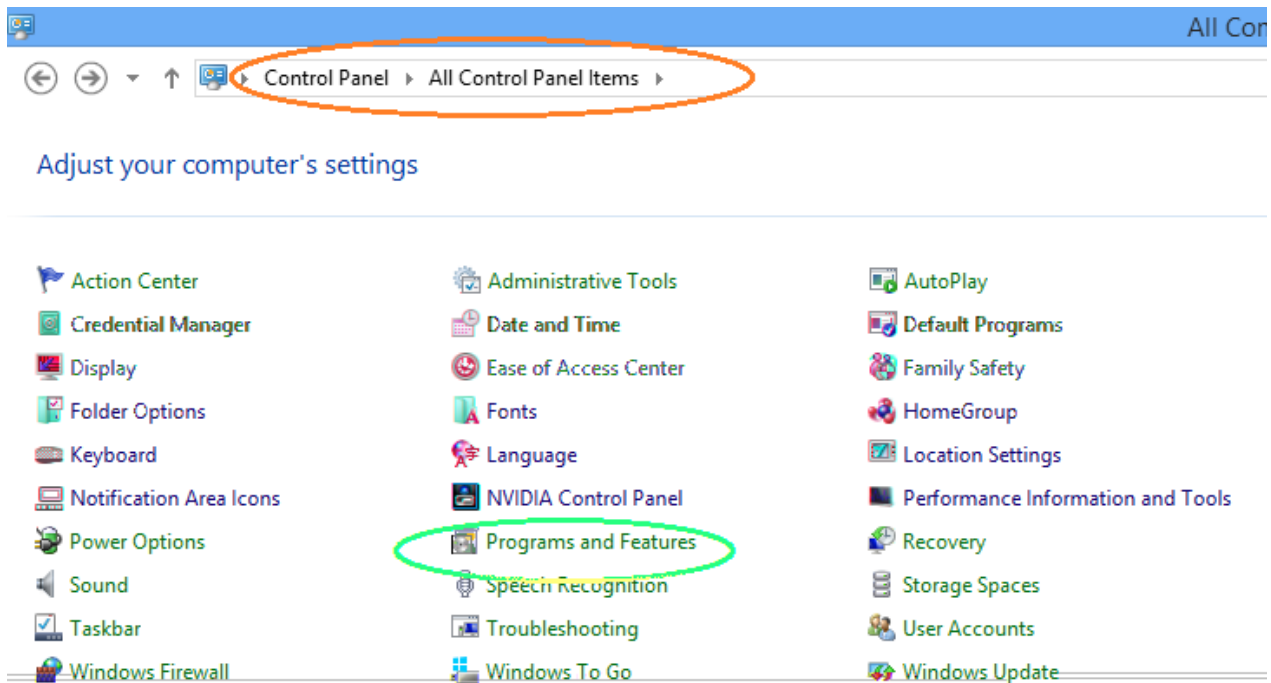
Install WebDeploy_amd64 for Importing the App.

Create command line exe using HTML Point SDK

Install latest HTML5Point SDK. And then create a **command line exe** for PowerPoint to HTML5 conversion.

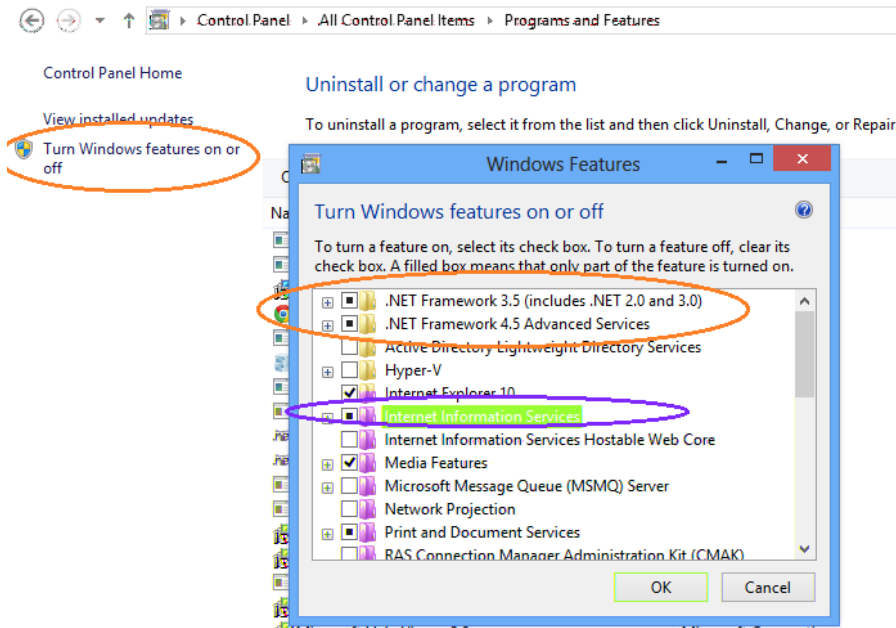
Install and configure IIS

Open Control panel ->Click 'Programs and Features'

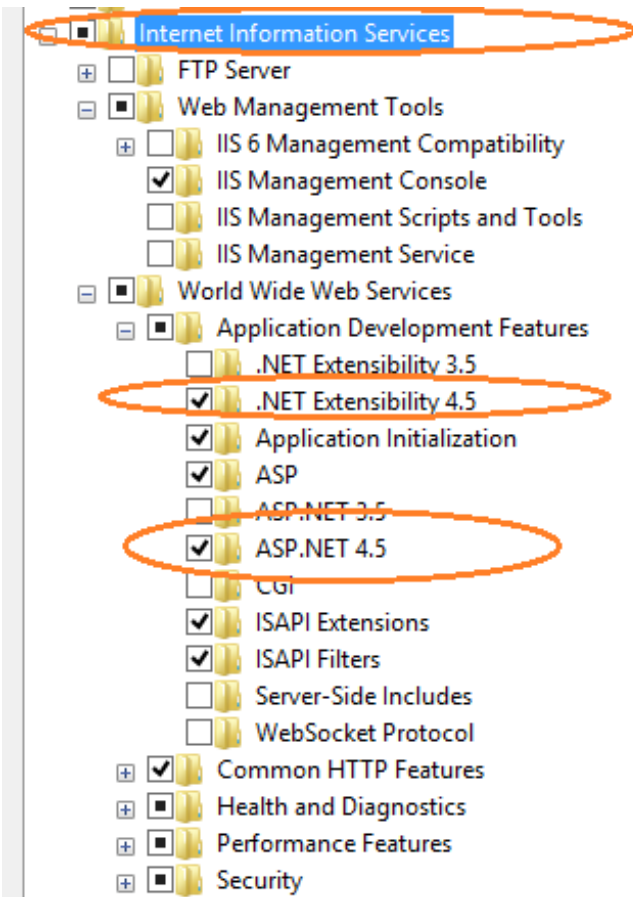


Click 'Turn Windows features on or off-> In Windows features dialog check on .Net Framework 4.5 advanced service. If you want to develop a 3.5 app then check on .Net 3.5 Framework support also.

Check on Internet information Service.



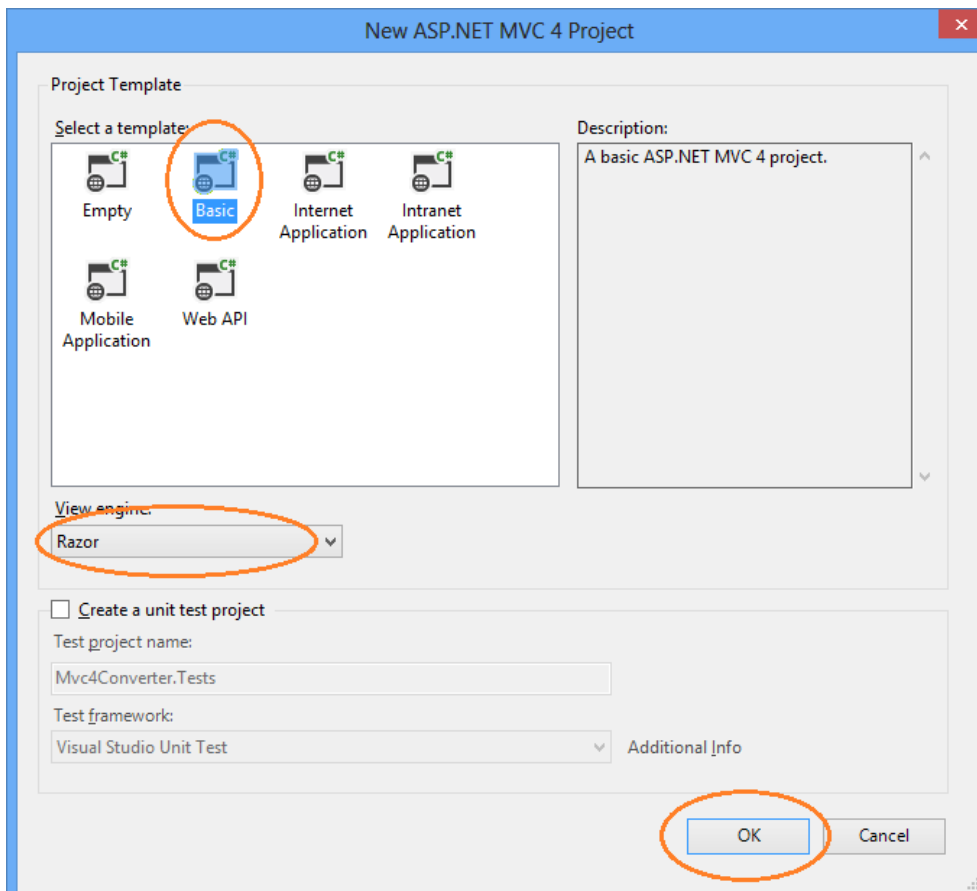
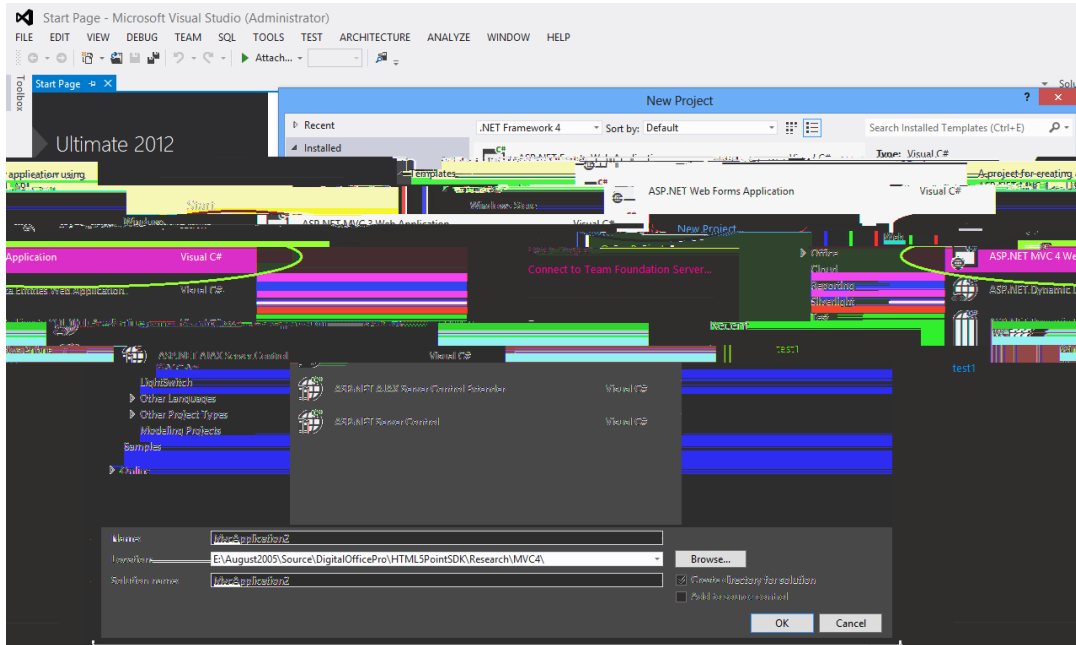
Make sure the following marked features are checked on. If you want to develop app in 3.5 then check on ASP .Net 3.5 also. Then click OK.



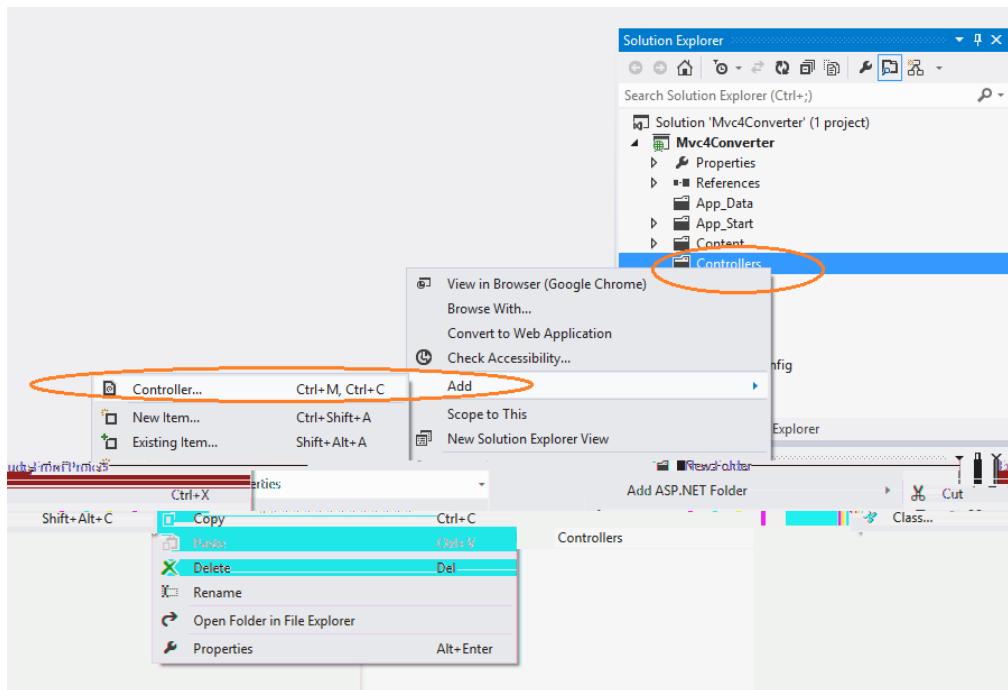
Create and publish MVC web application.

Install WebDeploy_amd64_en-US for publishing the web application. Sample project (HTML5PointSDK\Example\MVC4\Mvc4Converter) Create a new MVC 4 project.

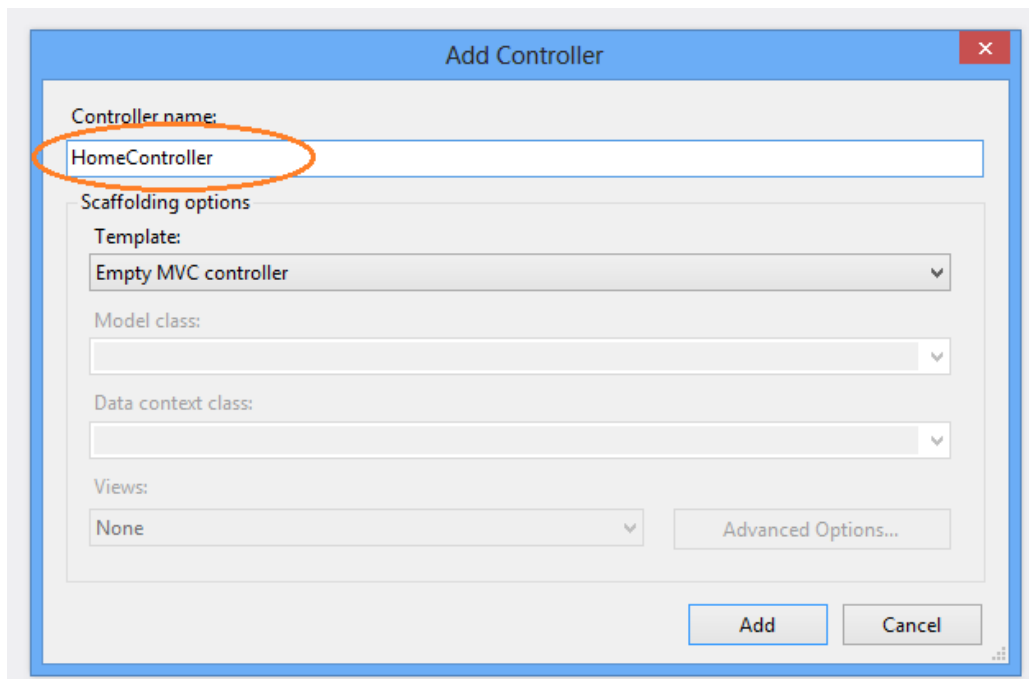
Open Visual Studio 012->New Project-> ASP .NET MVC4 Web Application



Add a new controller.

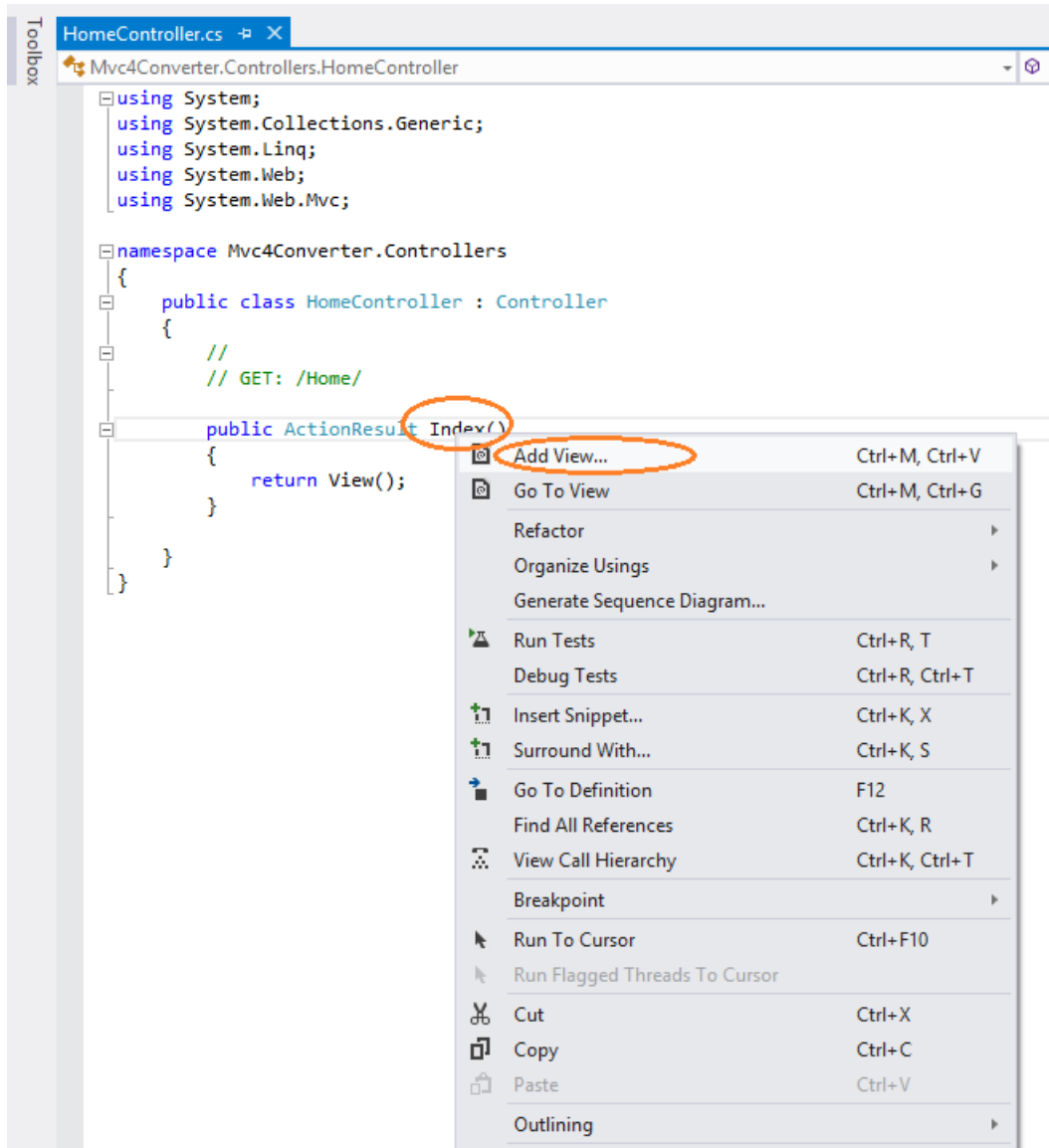


Set controller name to HomeController-> Then click Add button.



Add view for Index page.

Right click on Index function->Click Add View



Add the following code into Index view. This will allow you to upload a presentation file.

```
@{
    ViewBag.Title = "Publish Presentation";
}

<h2>Upload PowerPoint presentation:</h2><p></p>
@using (Html.BeginForm("FileUpload", "Home", FormMethod.Post, new { enctype =
"multipart/form-data" }))
{
    @Html.ValidationSummary();
    <ol>
    <li class="lifile">
        <input type="file" id="fileToUpload" name="file"/>
        <span class="field-validation-error" id="spanfile"></span>
    </li>
    </ol>
    <input type="submit" id="btnSubmit" value="Upload"/>
}
```

```
}
```

Add the following functions to HomeController for saving uploaded file and then convert it.

```
[HttpPost]
public ActionResult FileUpload(HttpPostedFileBase file)
{
    if (ModelState.IsValid)
    {
        if (file == null)
        {
            ModelState.AddModelError("File", "Please Upload Your file");
        }
        elseif (file.ContentLength > 0)
        {
            int MaxContentLength = 1024 * 1024 * 3; //3 MB
            string[] AllowedFileExtensions = newstring[] { ".pptx", ".ppt" };

            if
            (!AllowedFileExtensions.Contains(file.FileName.Substring(file.FileName.LastIndexOf('.'))))
            )
            {
                ModelState.AddModelError("File", "Please file of type: " +
                string.Join(", ", AllowedFileExtensions));
            }

            elseif (file.ContentLength > MaxContentLength)
            {
                ModelState.AddModelError("File", "Your file is too large, maximum
                allowed size is: " + MaxContentLength + " MB");
            }
            else
            {
                //TO:DO
                var fileName = Path.GetFileName(file.FileName);
                var path = Path.Combine(Server.MapPath("~/Content/Upload"), fileName);
                file.SaveAs(path);
                ModelState.Clear();

                PublishPresentation(path,
                Path.Combine(Server.MapPath("~/Content/Upload/out"), "out.html"));

                ViewBag.Message = "File published successfully";
                ViewBag.OutputPath = "../Content/Upload/out/out.html";
                //ViewBag.OutputPath = Path.Combine(Server.MapPath("~/bin"), "PptAutomate.exe");
            }
        }
    }
    return View();
} //public ActionResult FileUpload(HttpPostedFileBase file)

bool PublishPresentation(string srcpath, string outputPath)
{

```



```

/*PresentationConverter pc = new PresentationConverter();

    pc.OpenPresentation(srcpath);

    pc.Convert(outPath);

    pc.ClosePresentation();*/

ProcessStartInfo psi = new ProcessStartInfo();
    psi.FileName = Path.Combine(Server.MapPath("~/bin"), "ppt2html5.exe");
    psi.Arguments = " /i=" + srcpath + " /o=" + outPath ;//srcpath + " " +
outPath;//

//psi.UseShellExecute = true;
Process proc = new Process();
    proc.StartInfo = psi;
    proc.Start();

    proc.WaitForExit();
//MessageBox.Show("Success");
return true;
}

```

Add view for FileUpload: Right click on FileUpload function and select Add View command.

Add following code snippet into FileUpload view file(FileUpload.cshtml)

```

@{
    ViewBag.Title = "FileUploaded";
}

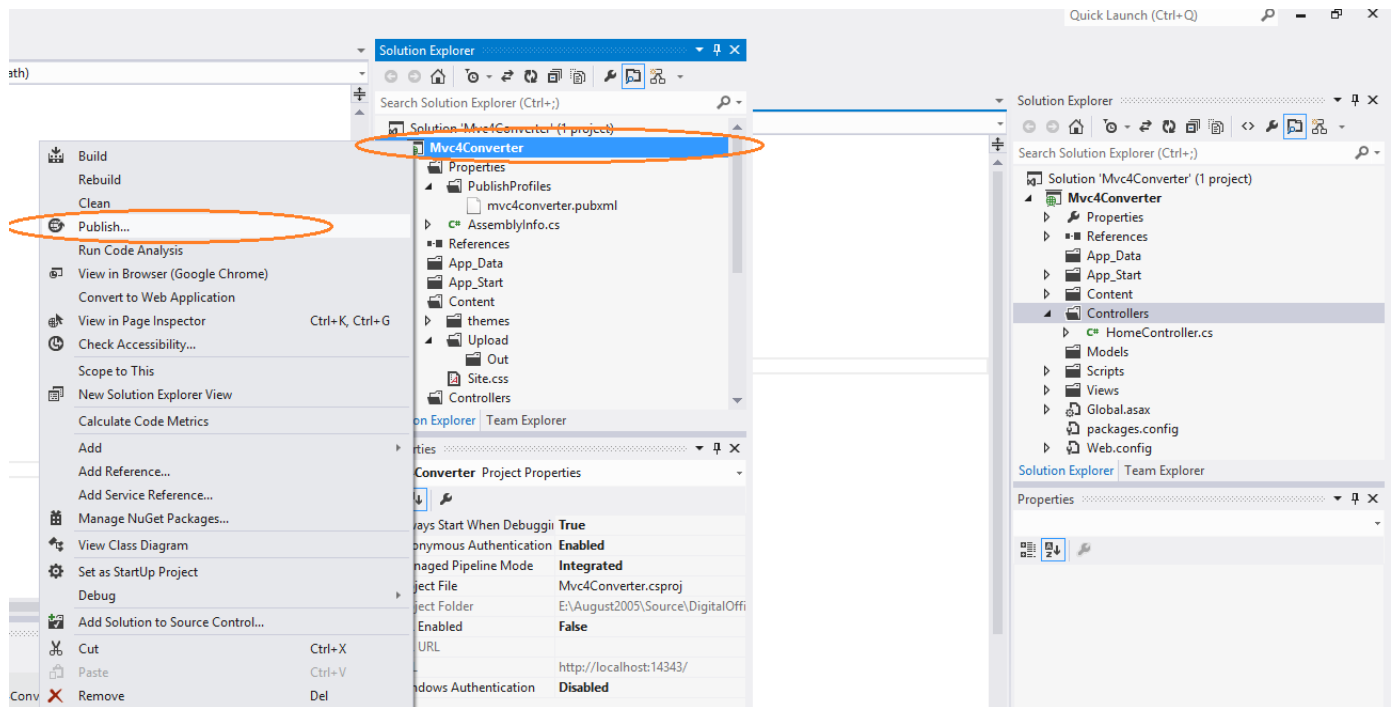
<h2>@ViewBag.Message</h2><br/>
<h1>@ViewBag.OutputPath</h1>
<a href="@ViewBag.OutputPath">Click here to view the HTML5</a>

```

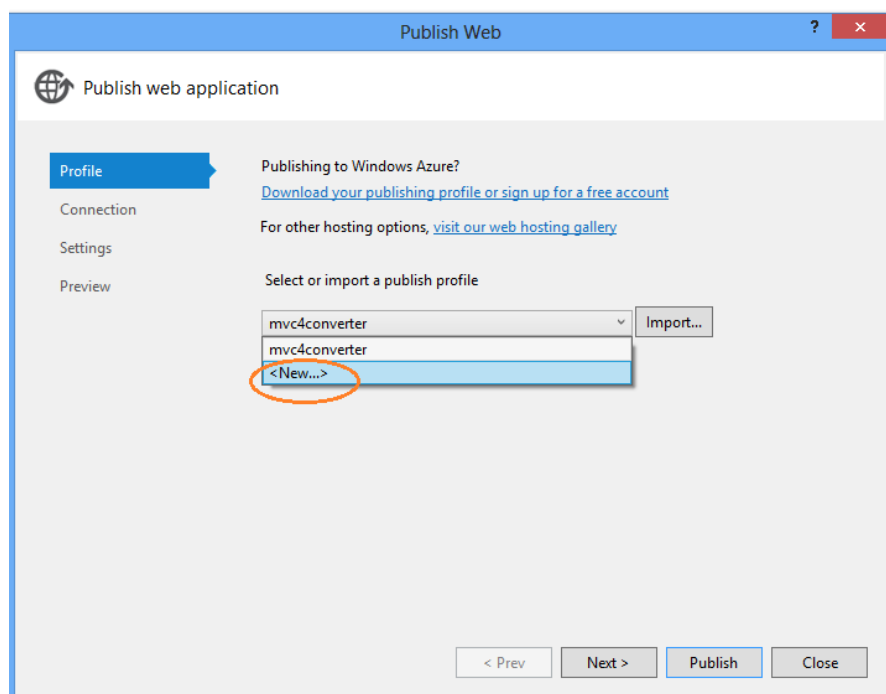
Build the application and then press F5 to run/debug.

Publish the application:

Right click on project in solution explorer->Select Publish command

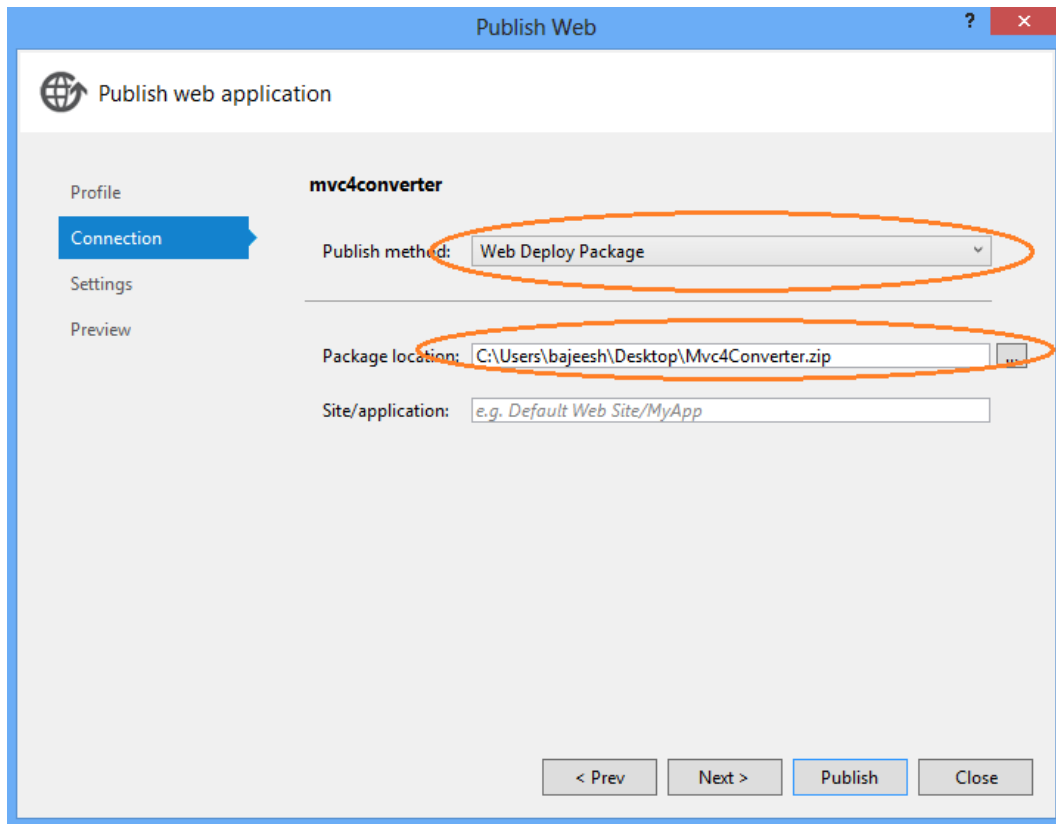


Click New and add profile name.



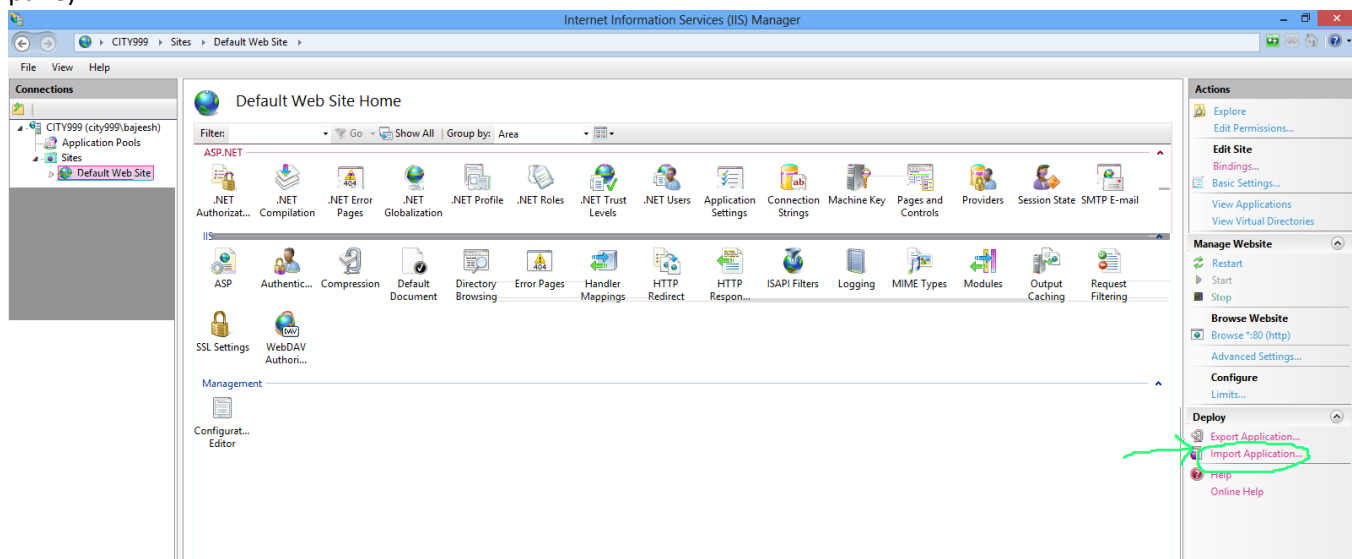
Select publish method to Web Deploy Package

Select publish location then click publish.



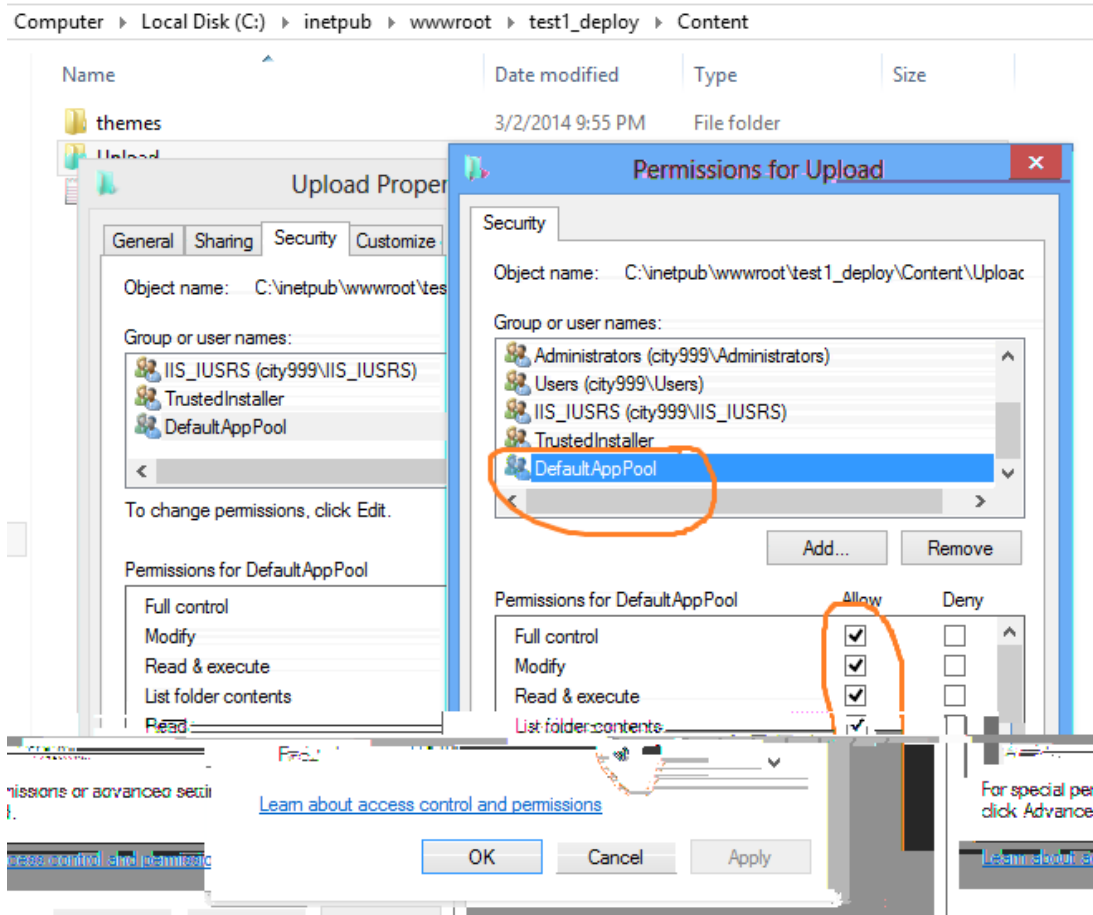
Deploy MVC4 application on IIS

1. Import MVC4 package into IIS. (Select 'Default Web Site' and click 'Import Application' in Action pane)

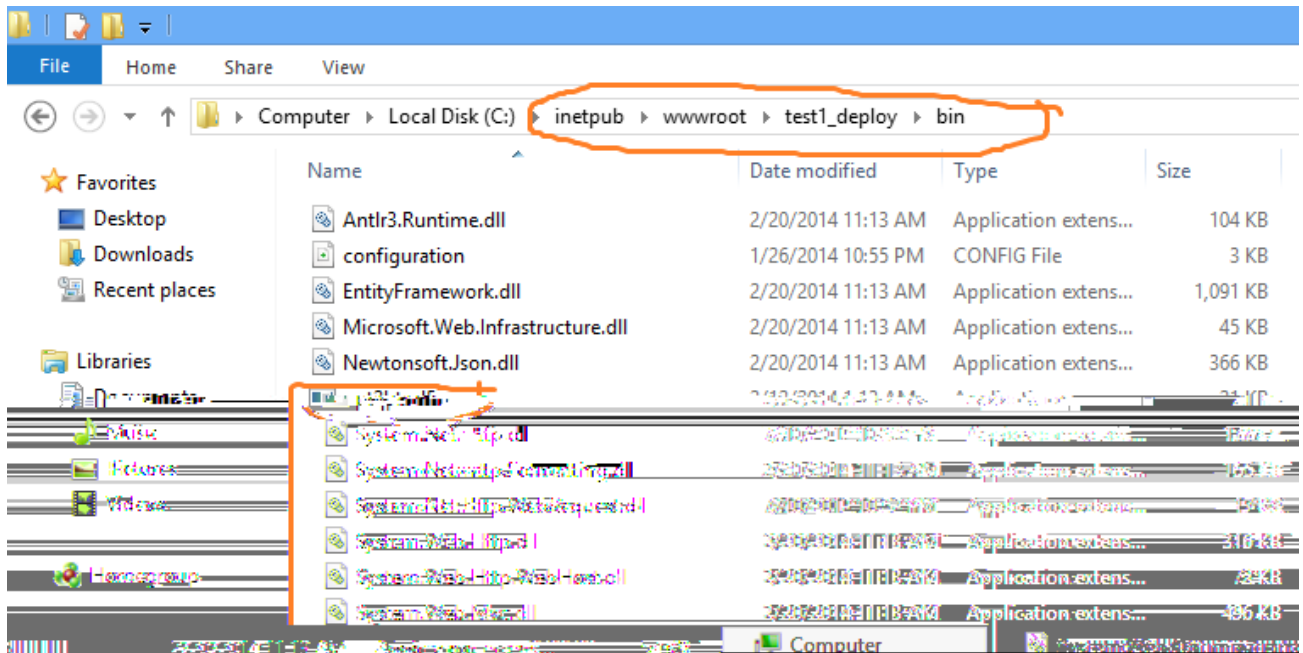


2. Create a folder named 'Upload' under imported path (eg: C:\inetpub\wwwroot\test1_deploy\Content\Upload) where presentations are kept for conversion.
3. Create 'out' folder under 'Upload' folder where we are publishing the presentation.
4. Set full permission for 'Upload' folder for DefaultAppPool user.

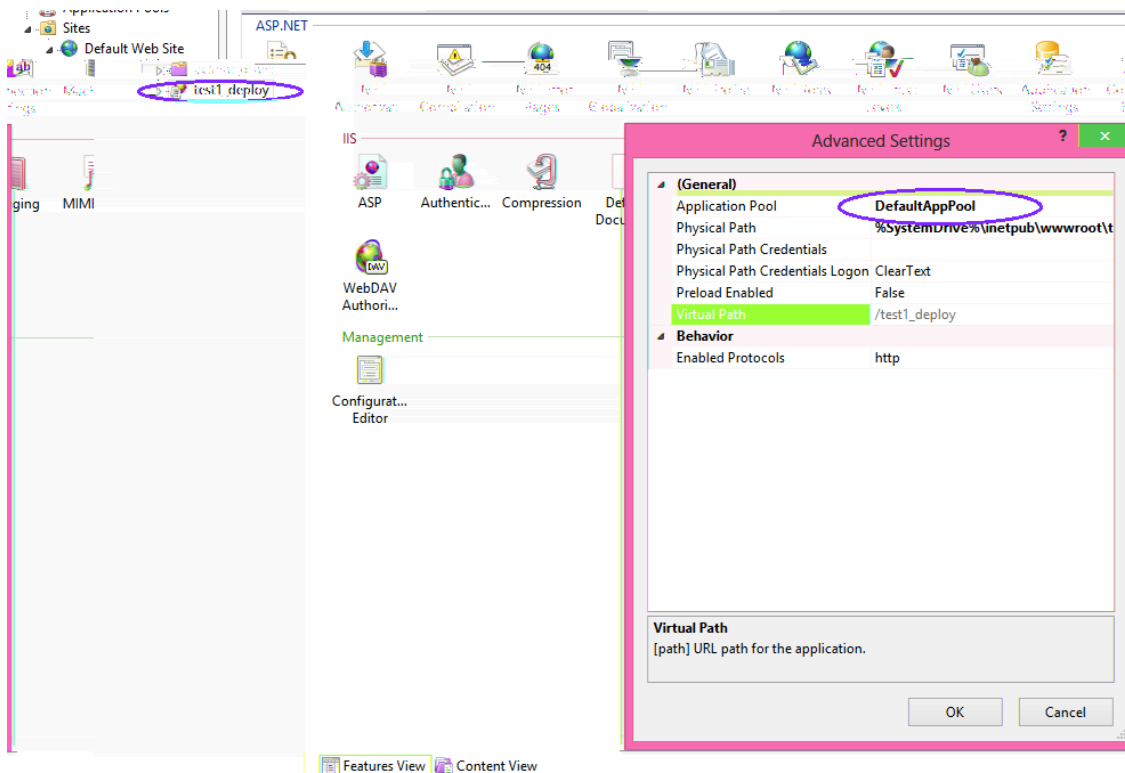
- Right click 'Upload' folder->Select security tab->Click Edit button
- In the Permission dialog->Select 'DefaultAppPool' in 'Group or User name'
- Allow full control for 'DefaultAppPool'



5. If we are using a command line exe for ppt to HTML5 conversion then copy the exe and all dependent assemblies into bin folder.

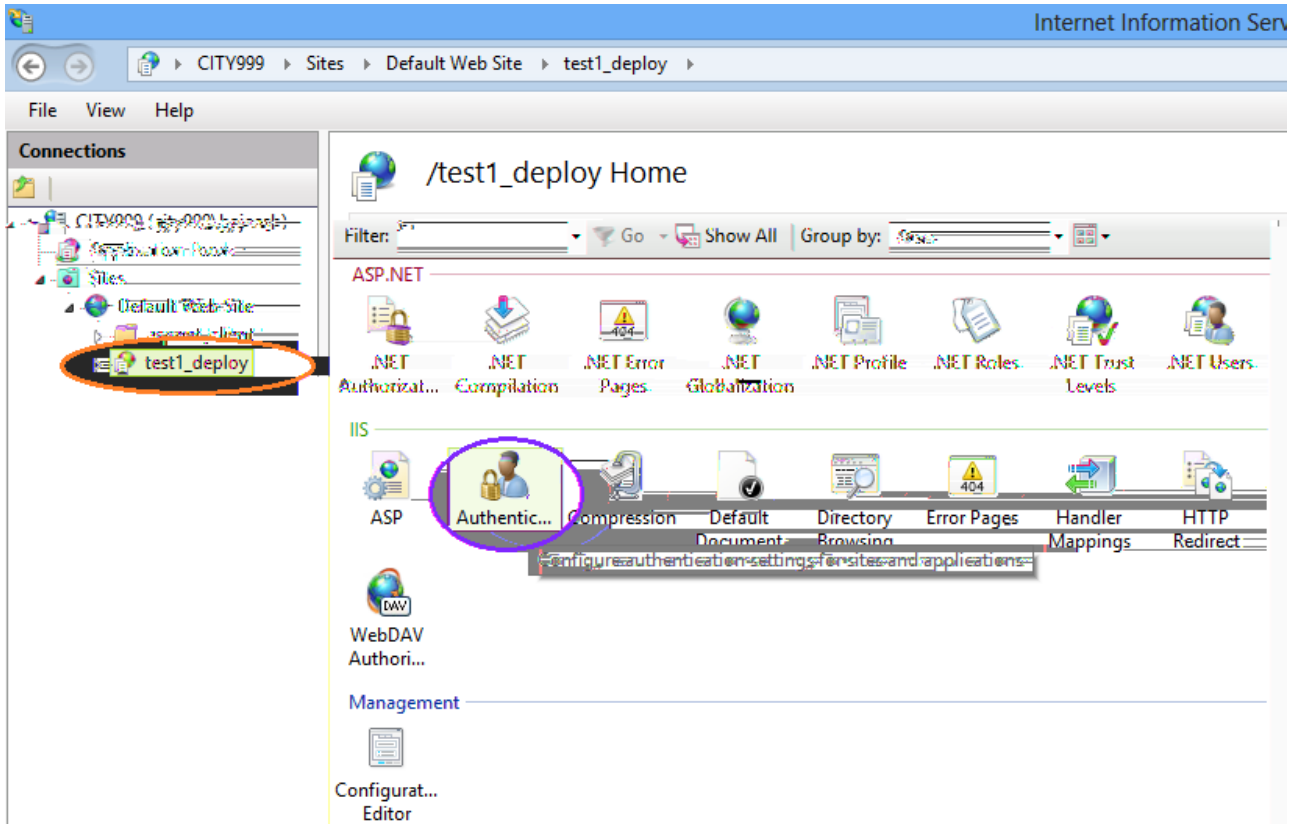


6. Make sure whether 'DefaultAppPool' is selected for MVC4 app.

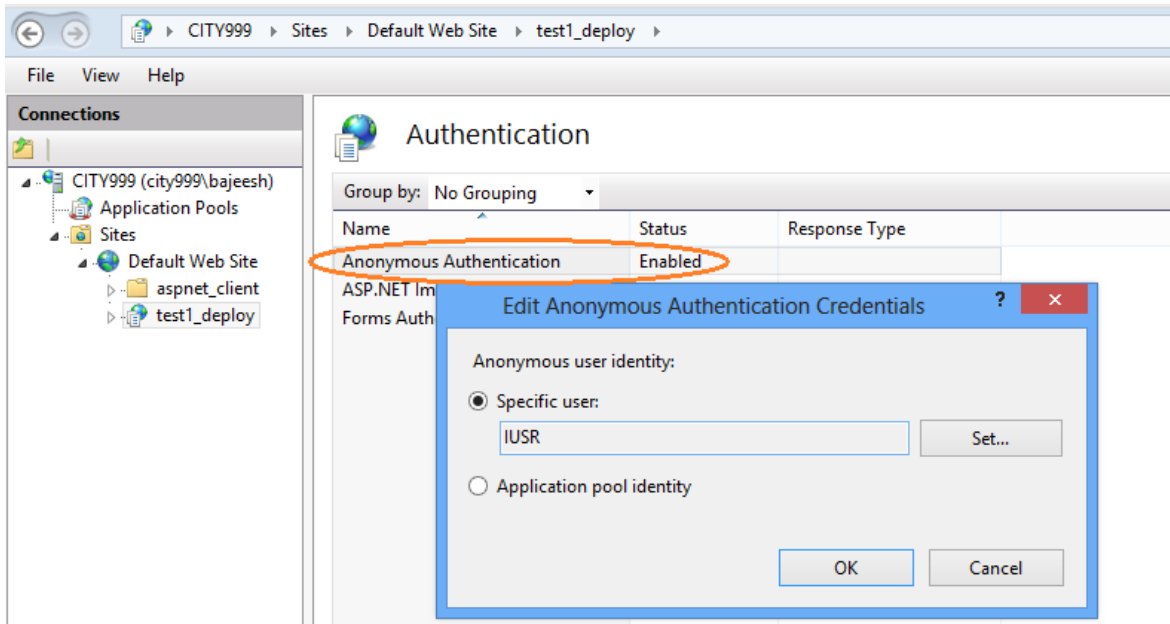


7. Configure Authentication settings (Not necessary).

- Select your new app in connections pane ->Double click Authentication icon in Featured View.



- Right click AnonymousAuthentication and select Edit command.



- Set specific user by clicking Set button.
- Set current user and password here, and then click OK. (Remember this user information, need to use the same later)

8. Change DefaultAppPool identity.

- Select Application Pool-> Select DefaultAppPool -> Click 'Advanced settings' command in Action pane->Select identity and click edit button->Select Custom account

Internet Information Services (IIS) Manager

CITY999 > Application Pools

File View Help

Connections

- CITY999 (city999\bajeesh)
 - Application Pools
 - Sites
 - Default Web Site
 - aspnet_client
 - test1_deploy

Application Pools

This page lets you view and manage the list of applications.

Filter: [Go]

Name	Status	.NET Framework
.NET v4.5	Started	v4.0
.NET v4.0 Classic	Started	v4.0
DefaultAppPool	Started	v4.0

Advanced Settings

Start Mode: OnDemand

CPU

- Limit (1/1000 of %): 0
- Limit Action: NoAction
- Limit Interval (minutes): 5
- Processor Affinity Enabled: False
- Processor Affinity Mask: 4294967295
- Processor Affinity Mask (64-bit): 4294967295

Process Model

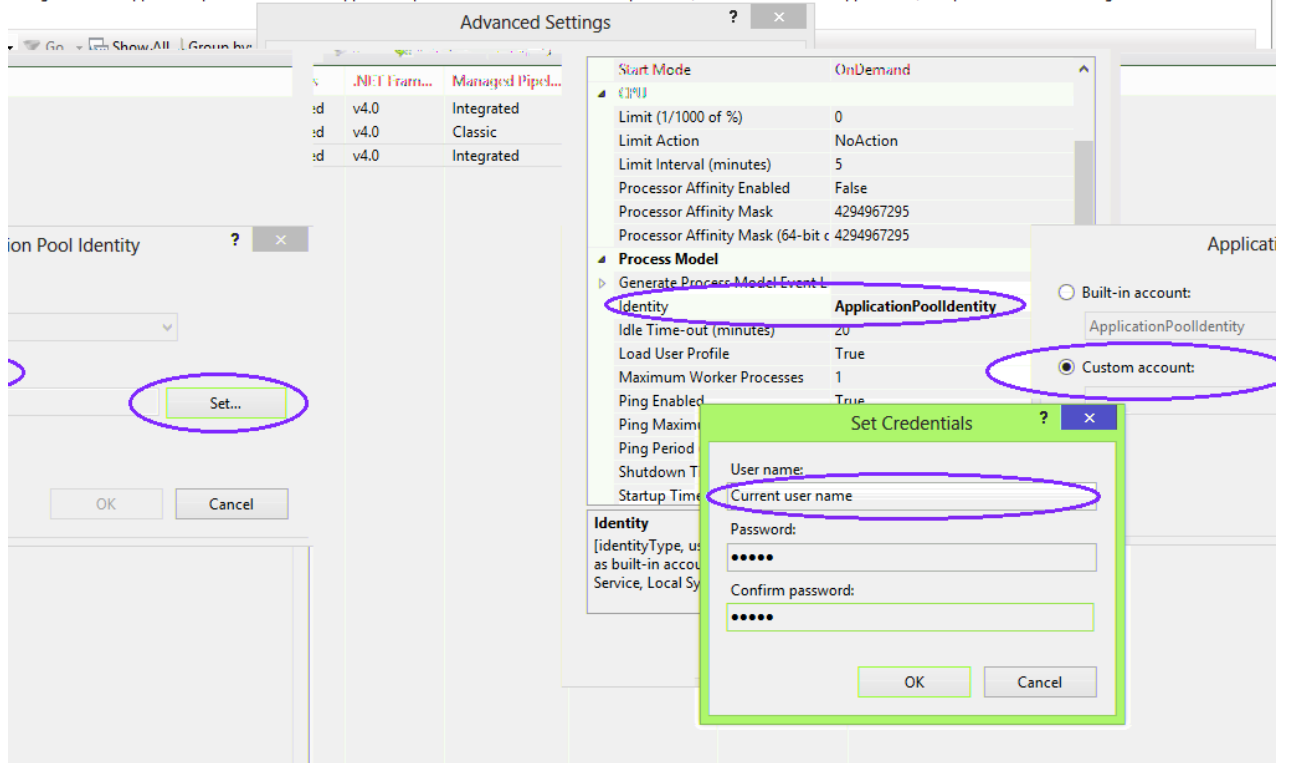
- Generate Process Model Event Log: [X]
- Identity**: ApplicationPoolIdentity
- Idle Time-out (minutes): 20
- Load User Profile: True
- Maximum Worker Processes: 1
- Ping Enabled: True
- Ping Maximum Response Time (seconds): 90
- Ping Period (seconds): 30
- Shutdown Time Limit (seconds): 90
- Startup Time Limit (seconds): 90

Identity

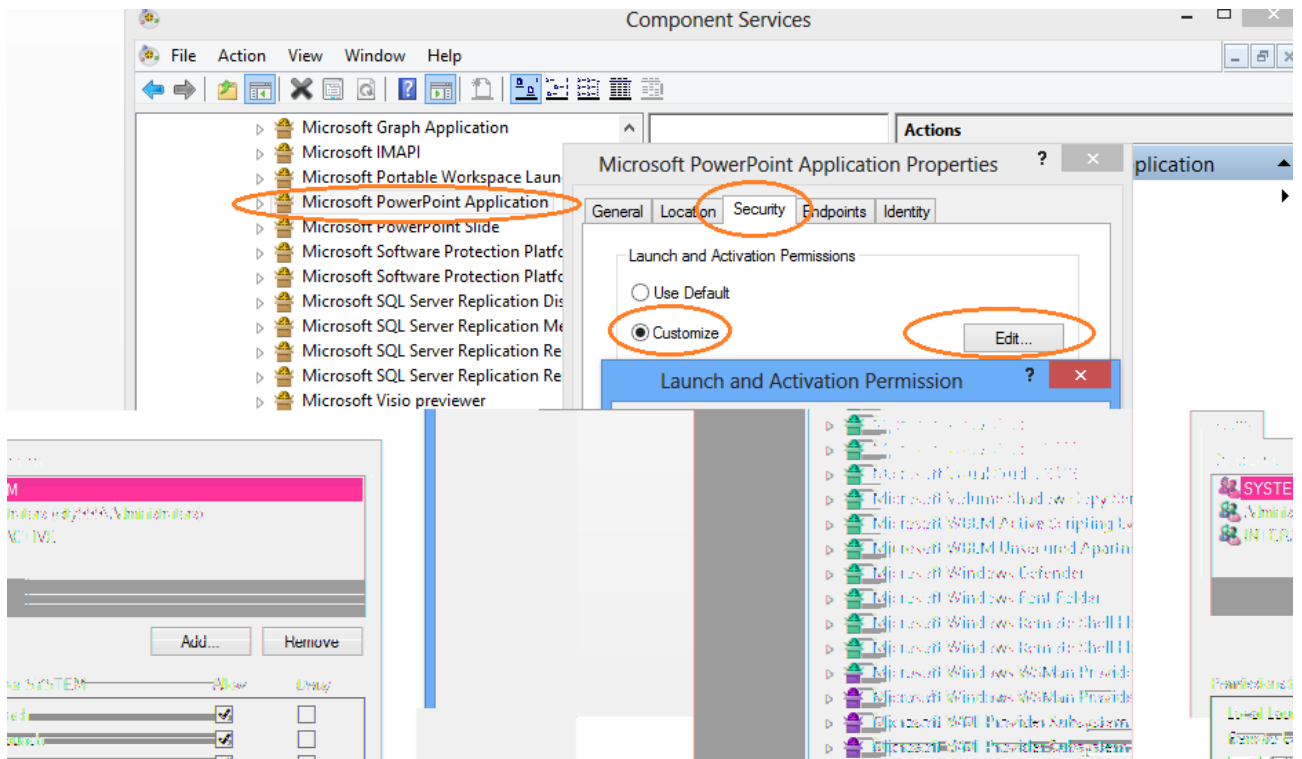
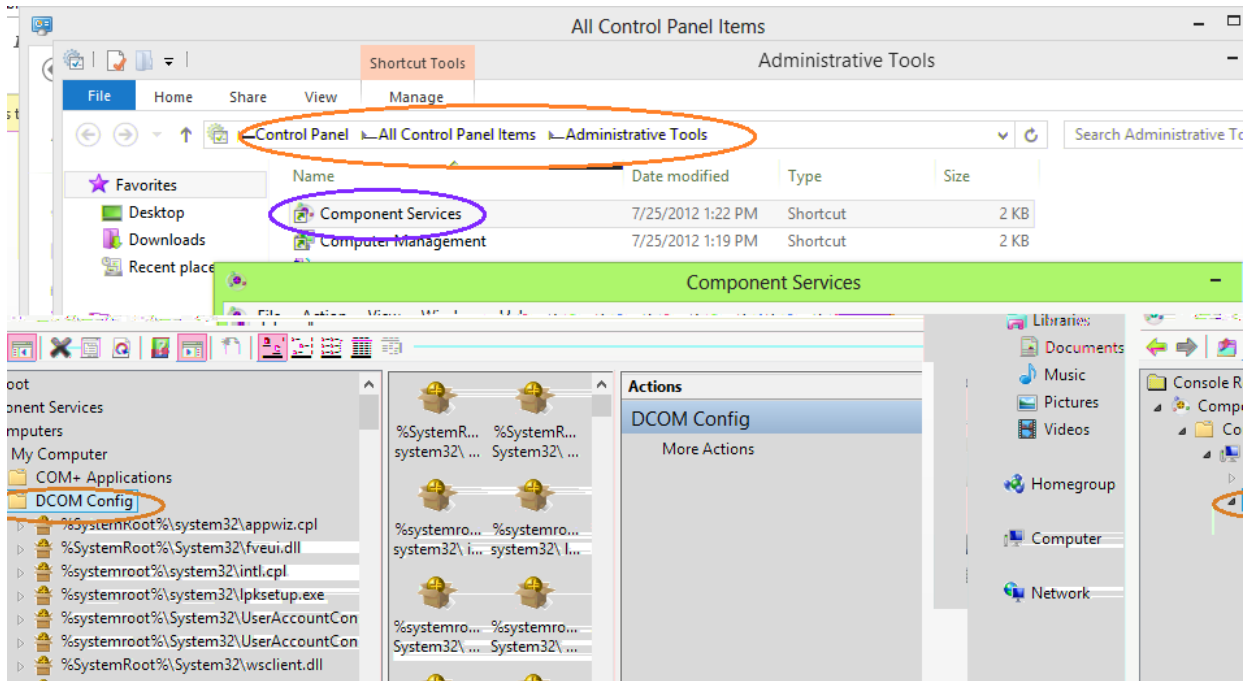
[identityType, username, password] Configures the application pool to run as built-in account, i.e. Application Pool Identity (recommended), Network Service, Local System, Local Service, or as a specific user identity.

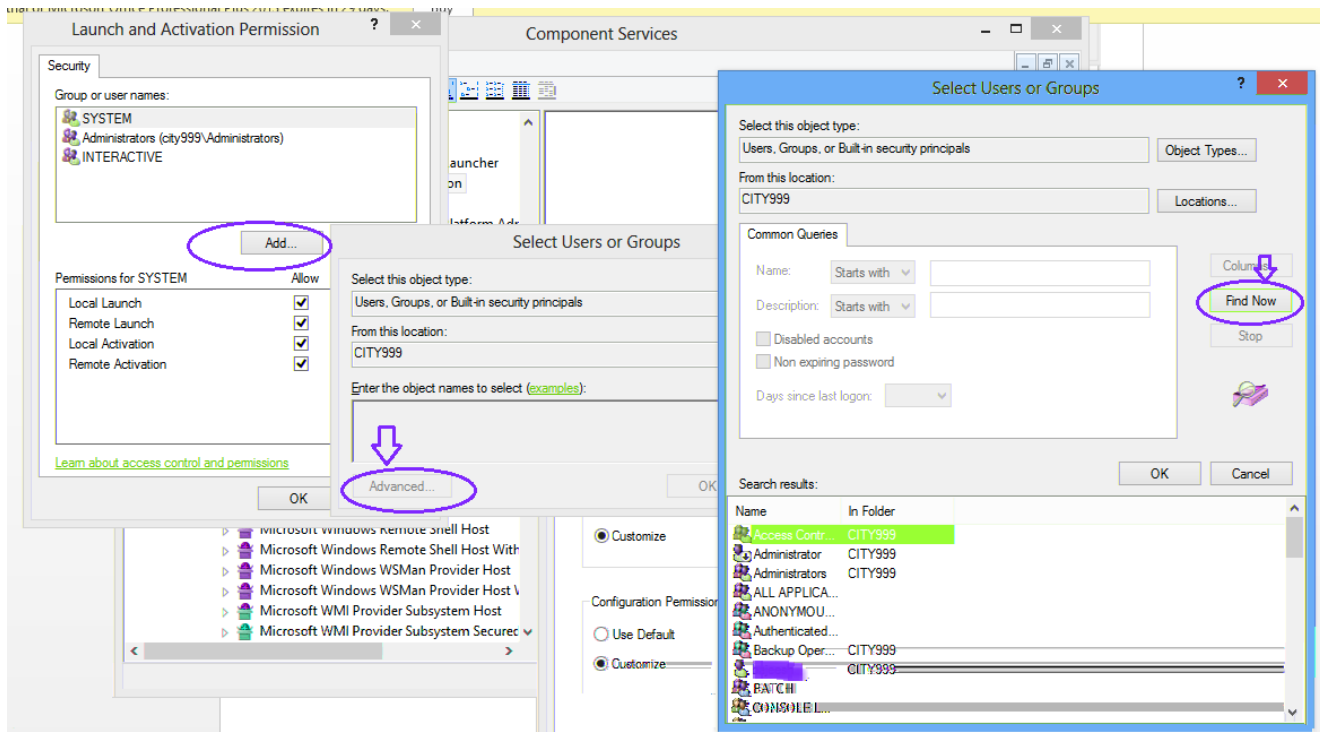
OK Cancel

manage the list of application pools on the server. Application pools are associated with worker processes, contain one or more applications, and provide isolation among different



9. Set local activation permission for current user for PowerPoint (**Not necessary**).
 - Open Component Service under Administrative tools. Then open DCOM Config folder->Select 'Microsoft PowerPoint Application' component.
 - Right click and select Properties command->Select Security tab-> Select 'Custom' options and click Edit button-> Click Add button->Then click Advanced button-> Click 'Find Now' button in Select Users & Group dialog->Select current user (as in the previous step) and click OK->Check on 'Local Activation' for added user under 'Launch and Activation Permission' dialog.





10. In order to run converter exe create Desktop under the following folders.

- C:\Windows\SysWOW64\config\systemprofile\Desktop
- C:\Windows\System32\config\systemprofile\Desktop