

# EdgeDoc Developer Guide

## Table of Contents

EdgeDoc Developer Guide .....	1
1. Overview .....	3
2. System Requirements.....	3
3. Quick Start.....	4
3.1 Running Demo Projects in IIS .....	4
3.2 Try Online Demo.....	4
4. Integrate with Visual Studio Project Templates .....	5
4.1 ASP.NET Empty Web Application.....	5
4.2 ASP.NET MVC4 Razor Web Application .....	10
4.3 ASP.NET MVC5 Web Application .....	20
4.4 Setup EdgeDoc Viewer in the IIS Application .....	29
5. Embed EdgeDoc Viewer into Web Page.....	32
6. Document Input & Output .....	35
6.1 Load file into EdgeDoc.....	35
6.2 Save file from EdgeDoc into your server file system or database .....	36
7. PDF Font Management .....	40
7.1 Install Adobe fonts to system .....	40
7.2 Install Adobe fonts in Runtime .....	41
8. Customize EdgeDoc Viewer.....	42
8.1 Re-locate Viewer client side files.....	42
8.2 Re-locate Viewer web service files .....	42
8.3 UI: Viewer width and height settings .....	43
8.4 UI: Viewer startup form view .....	43
8.5 UI: Viewer HTML page title .....	44
8.6 Comments settings.....	44
8.7 Customize toolbar and commands.....	44
8.8 Multiple languages support .....	45
8.9 Cached file data management .....	46
9. Customize, extend Annotations, Comments, Markups .....	47
9.1 Set annotation default properties cookie live time.....	47
9.2 Enable continuous annotations.....	47
9.3 Create a new stamp annotation .....	48
9.4 Get user mouse click coordinates .....	49
10. Enable Watermark in the EdgeDoc .....	50
11. Configure EdgeDoc through Web.config appSettings .....	51
12. Common Exceptions and Errors .....	53
12.1 Could not load file or assembly 'RasterEdge.Imaging.Annotation' .....	53
12.2 Could not load file or assembly 'RasterEdge.Imaging.Font.DLL' .....	54

13	Error logging & Report problems to us .....	55
----	---	----

# 1.Overview

Loading, viewing, annotating, redacting, saving and printing documents and images on web browser can be an easy task with the help of RasterEdgeEdgeDoc ASP.NET Document Viewer for .NET.

It includes a web document viewer DLL library for ASP.NET application that adopts the latest HTML5 technology and owns strong compatibility with most modern web browser environments. This web document viewer library can be easily integrated into your ASP.NET web application.

RasterEdgeEdgeDoc Document Viewer for ASP.NET is a JavaScript based image viewing control that can be created on the client side without additional add-ins and communicates directly with a WebHandler on the service side. With this AJAX Zero-footprint Document Viewer DLL, you can create high quality, customized web document viewer in ASP.NET web project with flexible functionality which can be easily added or removed with a simple JavaScript API.

## 2.System Requirements

1. IIS
2. .net framework 3.5, 4.0 or later version
3. **Visual C++ 2005 Redistributable Package** for .net 3.5  
**Visual C++ 2010 Redistributable Package** for .net 4.0 and later version

## 3.Quick Start

### 3.1 Running Demo Projects in IIS

After you download the latest version of RasterEdge.DocImageSDK, then how to implement the same effect with the online demo? Here, you can do it according to the following steps.

1. In download package, go to EdgeDoc demo project folder “**/DemoProjects/EdgeDoc Demo Project**”

2. Copy corresponding DLL(x86 or x64) in the download package to the “**Bin**” folder of the demo project.

3. After adding relative dlls, configure the Web.config :

    If your IIS server .NET framework is 4.0 or higher, please use the Web.config.

    If your IIS server .NET framework is 3.5, please copy the content in the Web(for .net3.5 or earlier version).config to Web.config.

4. After above steps, you can deploy your demo web application to IIS server.

- 1) Open Internet Information Services Manager;
- 2) Right-click “Sites” and select “Add Web Site”;
- 3) According to your situation, set the “Site name”, and set the “Physical path” to the place where you store EdgeDoc Demo. Last, set the “Port” that monitors the application.
- 4) Click ok to complete deploy your project;
- 5) Click "Application Pools", click "Set Application Pool Defaults..." in the right panel, and set the value "Enable 32-Bit Applications" in accordance with the selected DLL (x86 set true, x64 set false) in the pop-up window.
- 6) Grant user “**IIS\_IUSRS**” full control permission to folder “{Demo project}/**RasterEdge\_Cache**”, where HTML5 Document Viewer will generate cache files.

5. Run the web application

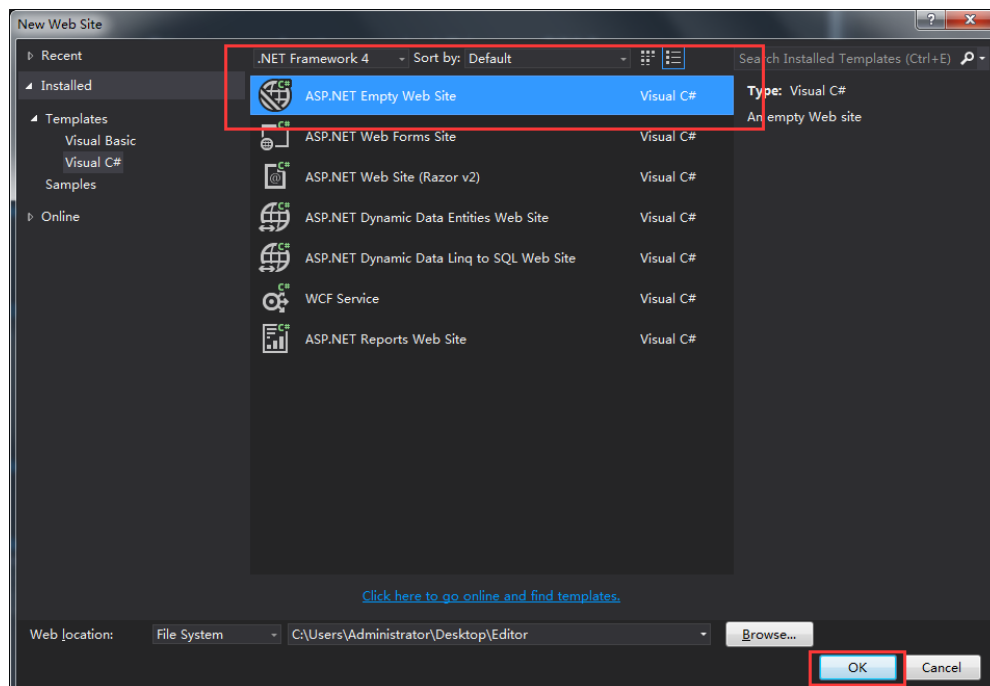
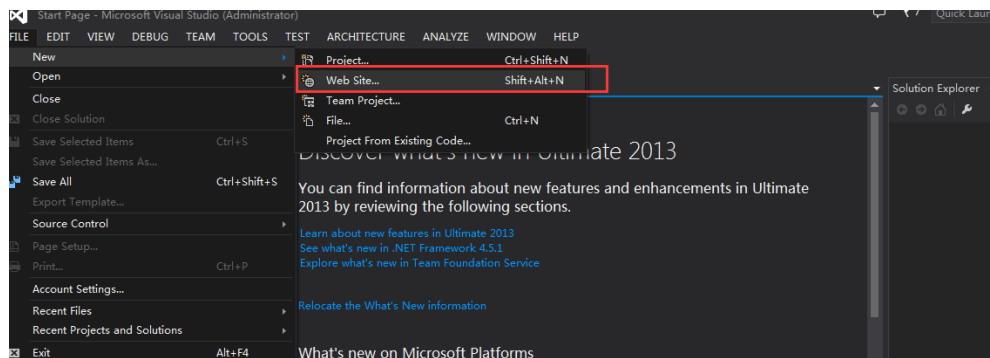
### 3.2 Try Online Demo

Please go to <http://www.rasteredge.com/dotnet-imaging/web-viewer-demo/> to view EdgeDoc online demo.

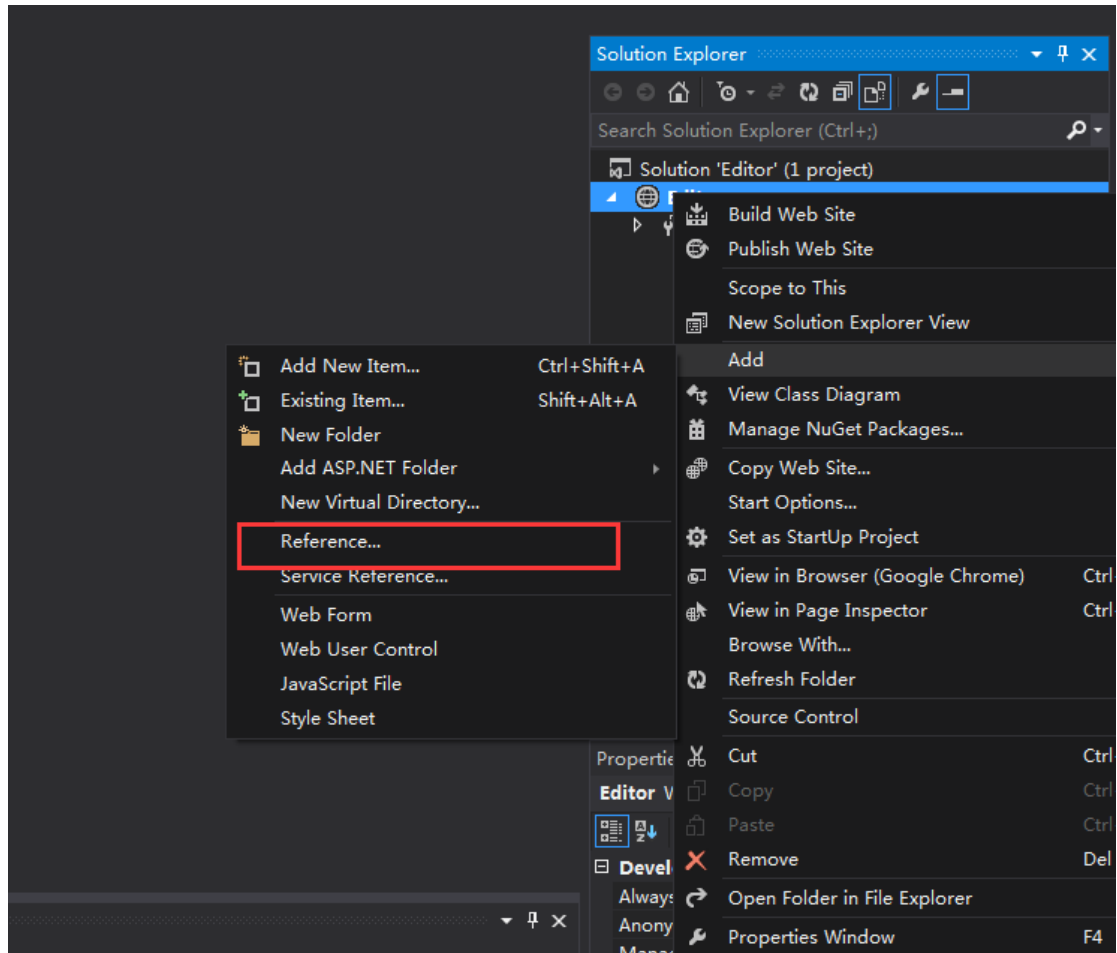
## 4. Integrate with Visual Studio Project Templates

### 4.1 ASP.NET Empty Web Application

1. Create an ASP.NET Empty Web Site, “EdgeDocWebSite”



2. Add RasterEdge.dll reference (add all dll names starting with RasterEdge)

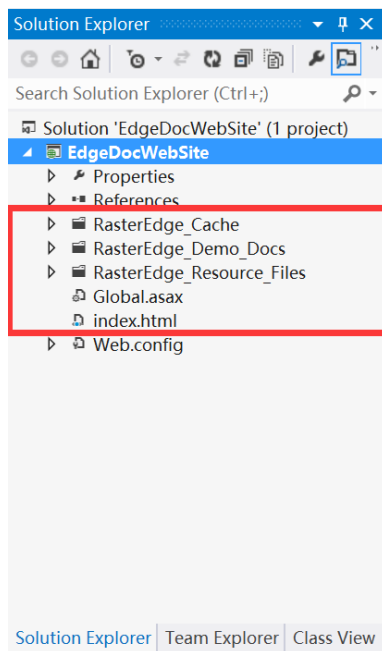


Please go to download package “/Bin”, choose right .net version. Add all **dlls** with starting name “**RasterEdge.**” to your project reference.

**Note: Visual Studio embed web server (IIS Express) is for x86 only.**

**Please use x86 dlls from Bin folder.**

3. Copy contents from **Demo Project** to your web site project.



#### 4. Modify the Web.config

```
<configuration>
<appSettings>
<!-- the cache folder -->
<add key="reCacheFolder" value="RasterEdge_Cache"/>
    <add key="reCustomStampFolder" value="RasterEdge_Resource_Files/images/stamp"/>
<!-- set the server folder if you want to open the file on server-->
<add key="reDefaultFolder" value="RasterEdge_Demo_Docs"/>
<!-- set the default file when open the web first time,if set "" ,this function is disabled.
(reDefaultFolder needed) -->
<add key="reDefaultFilePath" value=""/>
<!-- image quality in PDF document -->
<add key="rePageImageZoom" value="2"/>
<add key="reImageZoom" value="2"/>
<!-- image annotation in PDF document -->
<add key="reAnnotationImageZoom" value="2"/>
<!-- write log -->
<add key="reOutputLogData" value="true"/>
<!-- if true, SDK will automatically convert many shape and path inside PDF page into image-->
<!-- default is false -->
<!--  DEBUG ,INFO ,WARNING,ERROR -->
<add key="reLogLevel" value="DEBUG"/>
<add key="reWDPFileCacheMaxLimit" value="0"/>
<add key="reIntelliDrawShapesToImage" value="true"/>
<!-- if reIntelliDrawShapesToImage is true, -->
<!-- valid values are:      "all", "firefox", "chrome", "ie", "edge", "salari", "others"      if      "all"
included, ignore the rest;      default is "all"      -->
<add key="reIntelliDrawShapesToImageSupportBrowsers" value="all"/>
<!-- set the type of the show page.
Valid values: svg, png
Default: svg
-->
<add key="reDocRenderEngine" value="svg"/>
<add key="reRestfulFolder" value="c:/RasterEdge_Restful"/>
<add key="reI18NFolder" value="/RasterEdge_Resource_Files/i18n"/>
<add key="reServerURL" value=""/><!-- set the domain for remote client-->
</appSettings>

<system.web>
<compilation debug="true" targetFramework="4.0" />
<httpRuntime requestValidationMode="2.0"
executionTimeout="1200"
```



```
maxRequestLength="40960" useFullyQualifiedRedirectUrl="false"/>
</system.web>
<system.webServer>
<validation validateIntegratedModeConfiguration="false"/>
<security>
<requestFilteringallowDoubleEscaping="true">
<requestLimitsmaxAllowedContentLength="41943040"/>
</requestFiltering>
</security>
<staticContent>
    <remove fileExtension=".js"/>
<remove fileExtension=".svg"/>
<remove fileExtension=".woff"/>
<remove fileExtension=".ttc"/>
<remove fileExtension=".xpdf"/>
<remove fileExtension=".fdf"/>
<mimeMapfileExtension=".xpdf" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".fdf" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".woff" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".svg" mimeType="image/svg+xml"/>
<mimeMapfileExtension=".ttc" mimeType="application/x-font-ttc"/>
<mimeMapfileExtension=".js" mimeType="application/x-javascript"/>
</staticContent>
<urlCompressiondoStaticCompression="true" doDynamicCompression="true"/>
</system.webServer>
</configuration>
```

## 5. Change path

In file “Default.aspx”, update JS, CSS file path

```
<scriptsrc="/EdgeDocWebSite/RasterEdge_Resource_Files/javascript/jquery.js"type="text/javascript"></script>
<scriptsrc="/EdgeDocWebSite/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp_Customize.js"type="text/javascript"></script>
<scriptsrc="/EdgeDocWebSite/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp.js"type="text/javascript"></script>
<scriptsrc="/EdgeDocWebSite/RasterEdge_Resource_Files/javascript/RasterEdge.js"type="text/javascript"></script>
<linkrel="stylesheet"href="/EdgeDocWebSite/RasterEdge_Resource_Files/css/process.css"type="text/css"/>
<linkrel="stylesheet"href="/EdgeDocWebSite/RasterEdge_Resource_Files/css/customize.css"type="text/css"/>
<linkrel="stylesheet"href="/EdgeDocWebSite/RasterEdge_Resource_Files/css/jquery-ui.css"type="text/css"/>
```

And update index.html,UserCommandProcessHandler.ashx path

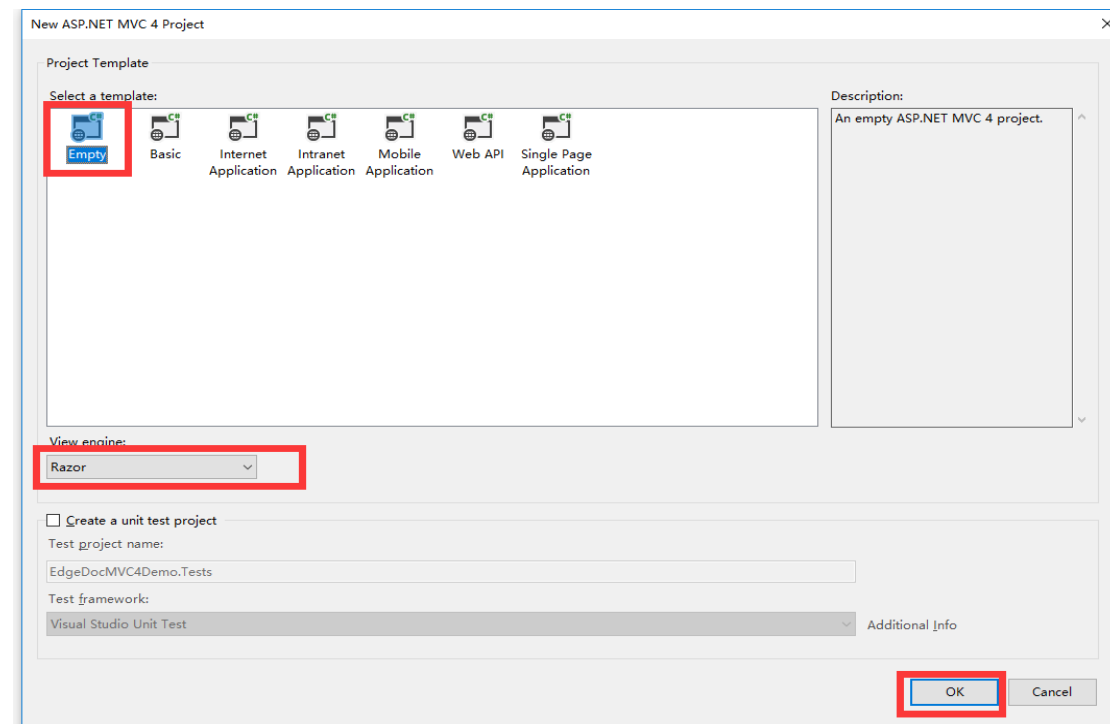
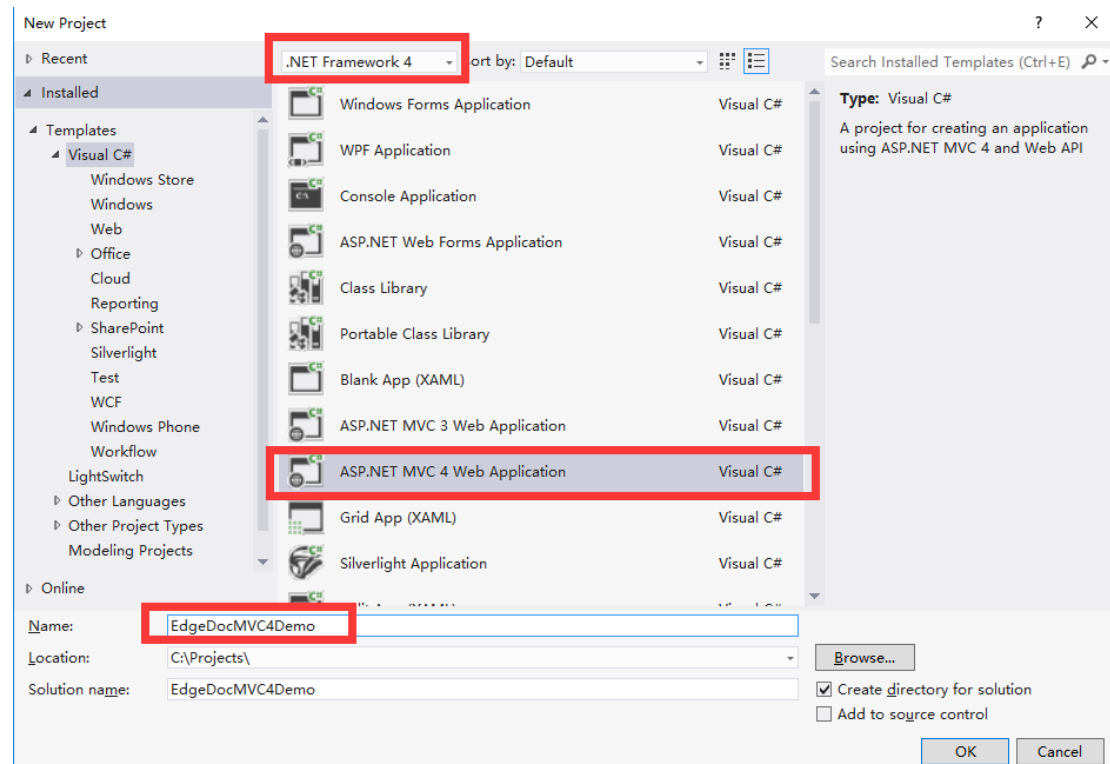
```
var _rootpath = "/EdgeDocWebSite/";
_WDPApp = new WDPOnlineApplication({
    _serverUrl: "RasterEdge_Resource_Files/UserCommandProcessHandler.ashx"
});
```

In var “\_serverUrl”, the value must be started with relative file path, such as “RasterEdge\_Resource\_Files/UserCommandProcessHandler.ashx”. Do not start with “/”.

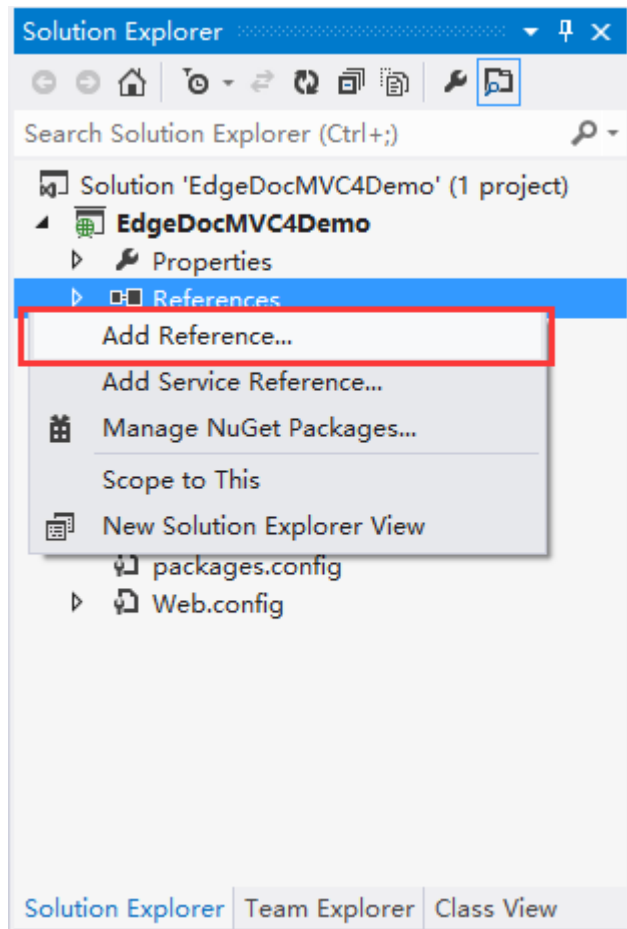
6. It’s done. Now run the website.

## 4.2 ASP.NET MVC4 Razor Web Application

## 1. Create a new MVC4 Project “EdgeDocMVC4Demo”



## 2. Add RasterEdge.dll reference (add all dll names starting with RasterEdge)

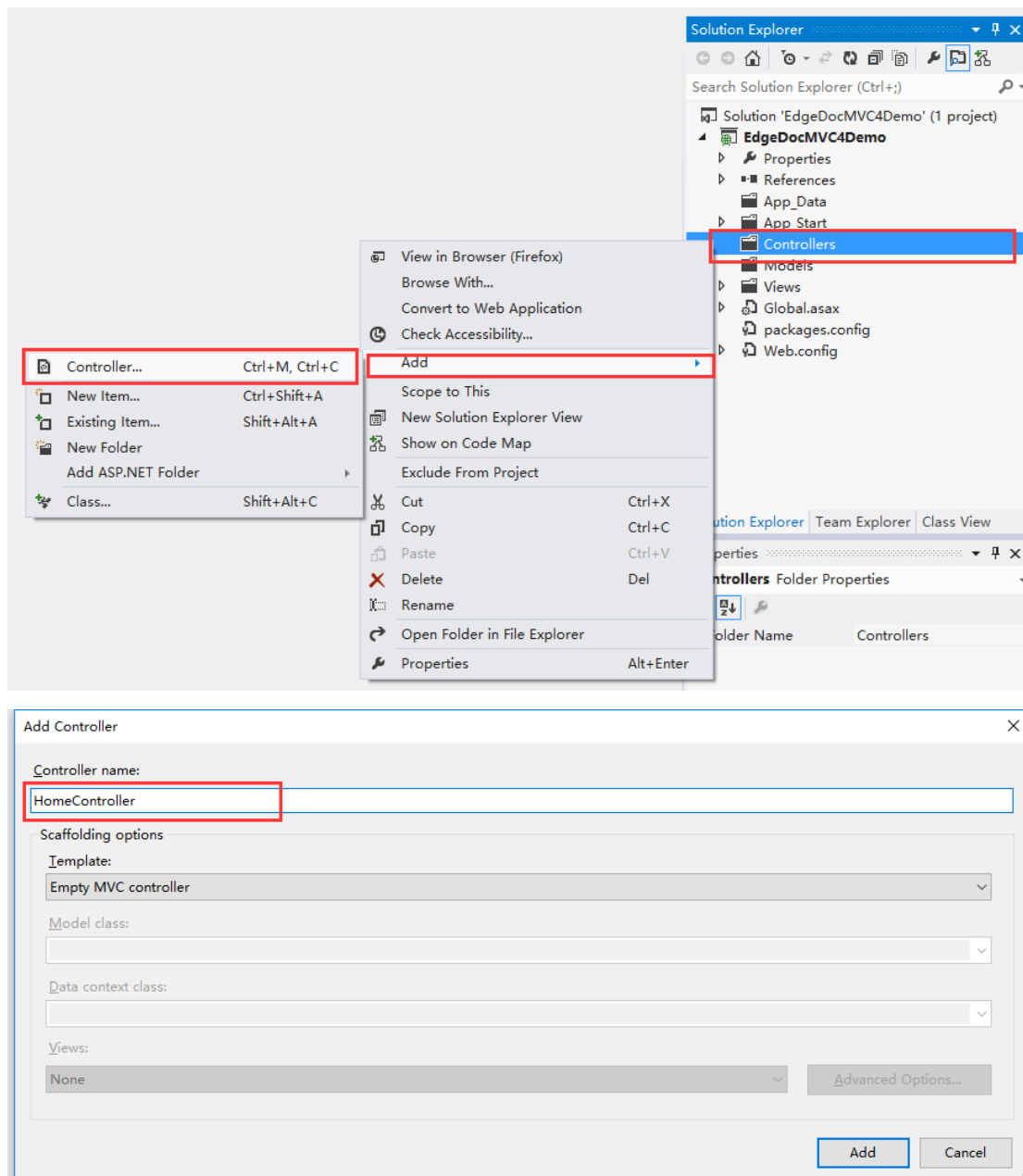


Please go to download package “/Bin”, choose **.net 4.0, x86 dlls**. Add all **dlls** with starting name “**RasterEdge.**” to your project reference.

**Note: Visual Studio embed web server (IIS Express) is for x86 only.**

**Please choose x86 dlls from Bin folder.**

### 3. Create controller

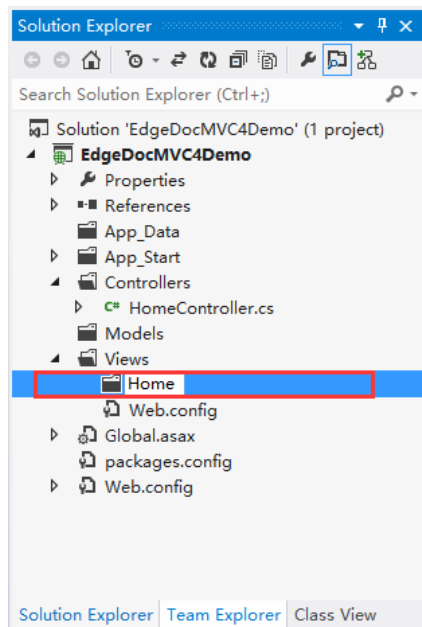
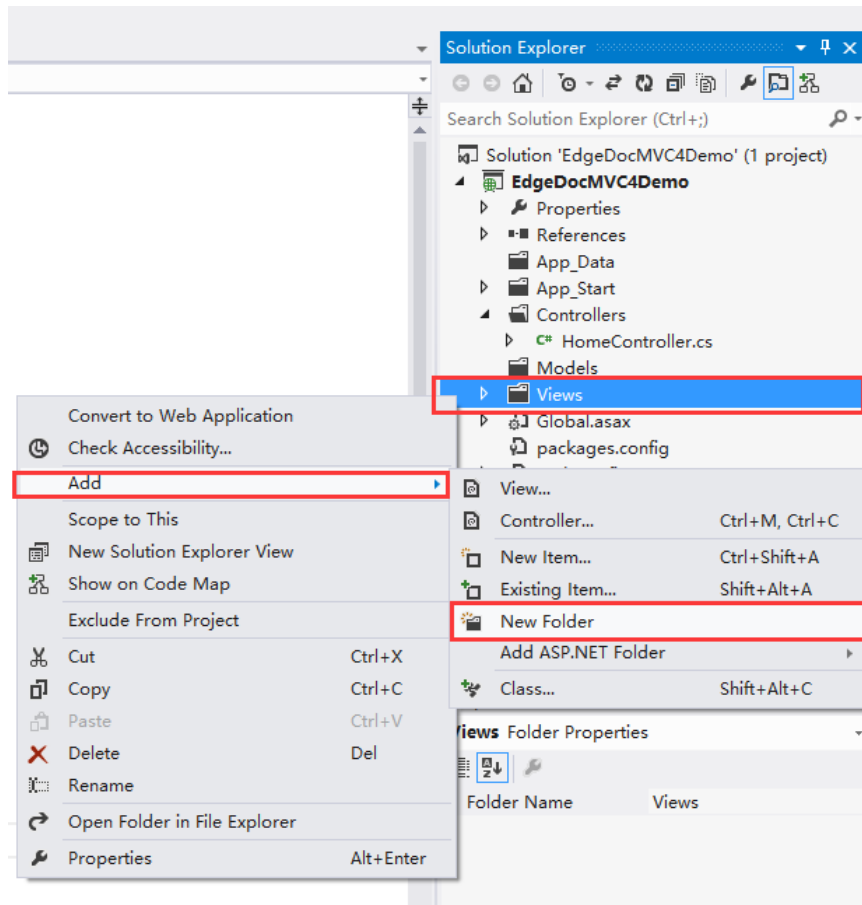


Click **"Add"**, to add a new controller to your project.

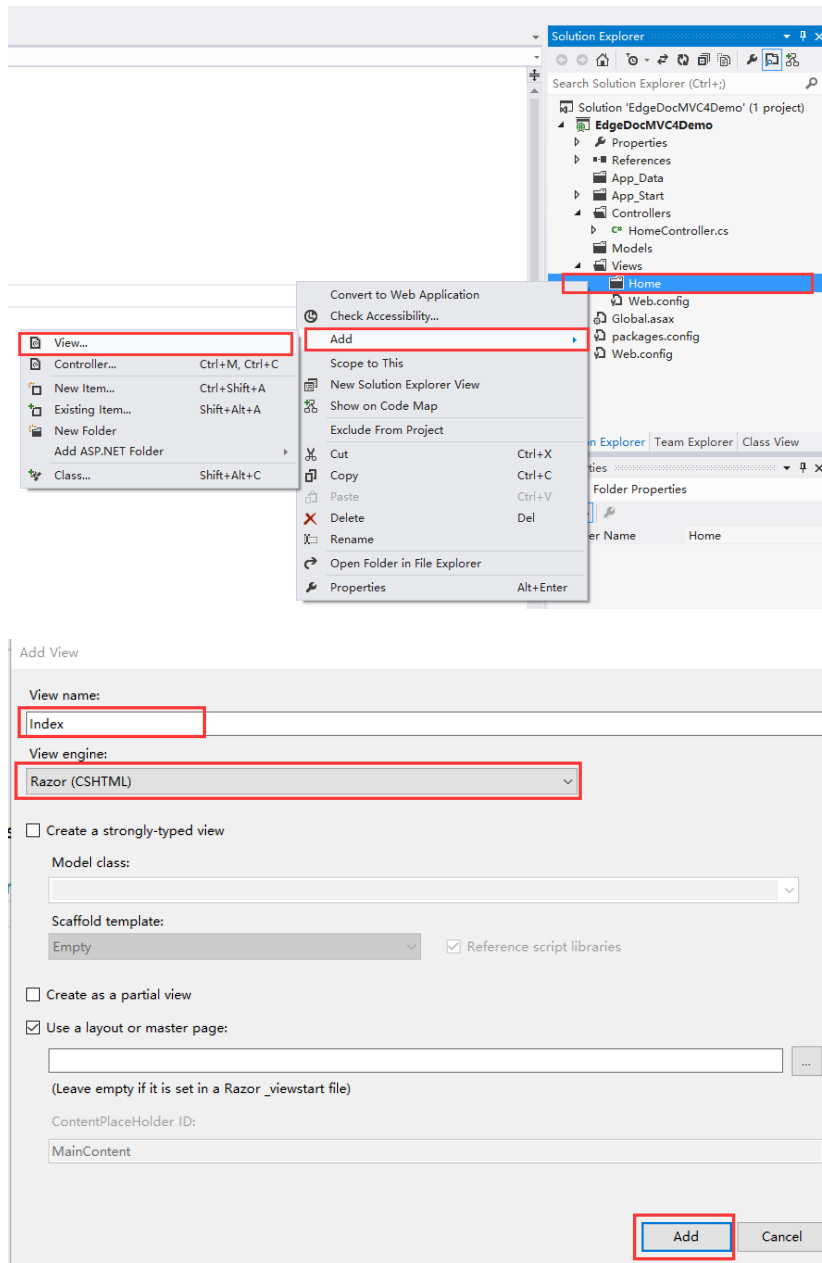
Note: Please use **HomeController** as your new controller name.

#### 4. Create View

Add a new folder “**Home**” to the Views



Add a new View “**Index**” under folder “**Home**”

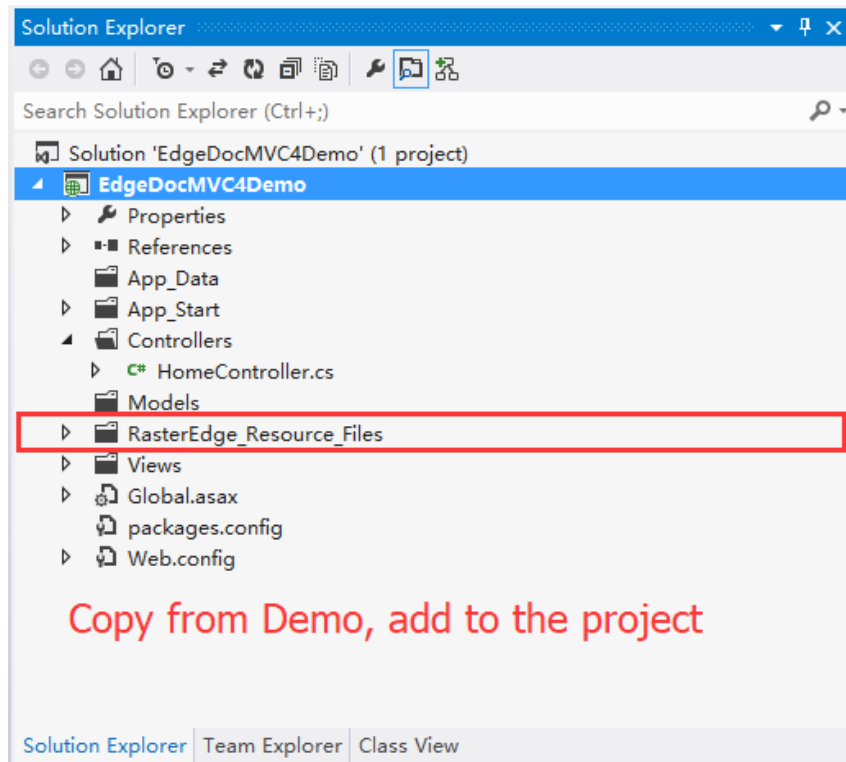


Add the html to **Index.cshtml**.

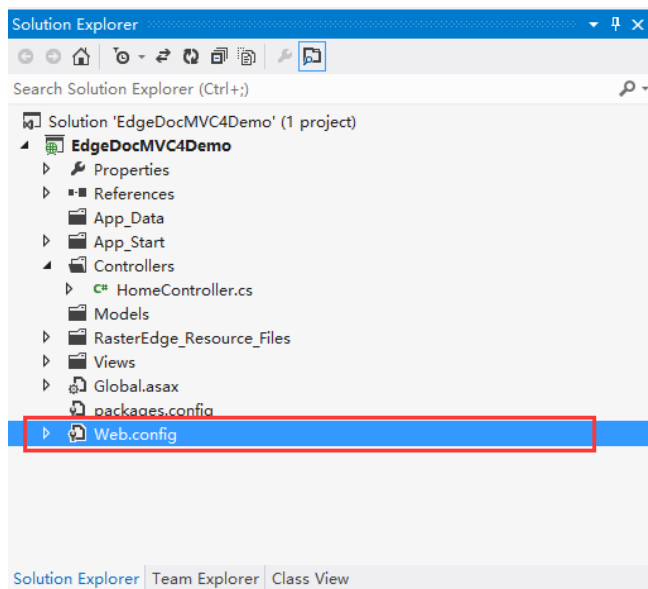
Copy the contents in file “/Developer Guide/EdgeDoc/MVC4 Razor project files/**Index.cshtml**” to your Index html file.

5. Add the folder “/DemoProjects/EdgeDoc Demo

Project/**RasterEdge\_Resource\_Files**”(all contents) to project.



## 6. Set the Web.config



Add the following contents to tag **<appSettings>**:

```
<!-- the cache folder -->
```



```

<add key="reCacheFolder" value="RasterEdge_Cache"/>
    <add key="reCustomStampFolder" value="RasterEdge_Resource_Files/images/stamp"/>
<!-- set the server folder if you want to open the file on server-->
<add key="reDefaultFolder" value="RasterEdge_Demo_Docs"/>
<!-- set the default file when open the web first time,if set "" ,this function is disabled.
(reDefaultFolder needed) -->
<add key="reDefaultFilePath" value=""/>
<!-- image quality in PDF document -->
<add key="rePageImageZoom" value="2"/>
<add key="reImageZoom" value="2"/>
<!-- image annotation in PDF document -->
<add key="reAnnotationImageZoom" value="2"/>
<!-- write log -->
<add key="reOutputLogData" value="true"/>
<!-- if true, SDK will automatically convert many shape and path inside PDF page into image-->
<!-- default is false -->
<!--  DEBUG ,INFO ,WARNING,ERROR -->
<add key="reLogLevel" value="DEBUG"/>
<add key="reWDPFileCacheMaxLimit" value="0"/>
<add key="reIntelliDrawShapesToImage" value="true"/>
<!-- if reIntelliDrawShapesToImage is true, -->
<!-- valid values are:      "all", "firefox", "chrome", "ie", "edge", "salari", "others"    if      "all"
included, ignore the rest;    default is "all"      -->
<add key="reIntelliDrawShapesToImageSupportBrowsers" value="all"/>
<!-- set the type of the show page.
Valid values: svg, png
Default: svg
-->
<add key="reDocRenderEngine" value="svg"/>
<add key="reRestfulFolder" value="c:/RasterEdge_Restful"/>
<add key="reI18NFolder" value="/RasterEdge_Resource_Files/i18n"/>
<add key="reServerURL" value=""/><!-- set the domain for remote client-->

```

Add the following contents to tag **<system.webServer>**

```

<security>
<requestFilteringallowDoubleEscaping="true">

```

```

<requestLimitsmaxAllowedContentLength="41943040"/>
</requestFiltering>
</security>
<staticContent>
    <remove fileExtension=".js"/>
<remove fileExtension=".svg"/>
<remove fileExtension=".woff"/>
<remove fileExtension=".ttc"/>
<remove fileExtension=".xpdf"/>
<remove fileExtension=".fdf"/>
<mimeMapfileExtension=".xpdf" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".fdf" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".woff" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".svg" mimeType="image/svg+xml"/>
<mimeMapfileExtension=".ttc" mimeType="application/x-font-ttc"/>
<mimeMapfileExtension=".js" mimeType="application/x-javascript"/>
</staticContent>
    <urlCompressiondoStaticCompression="true" doDynamicCompression="true"/>

```

Add the following contents to tag **<system.web>**

```

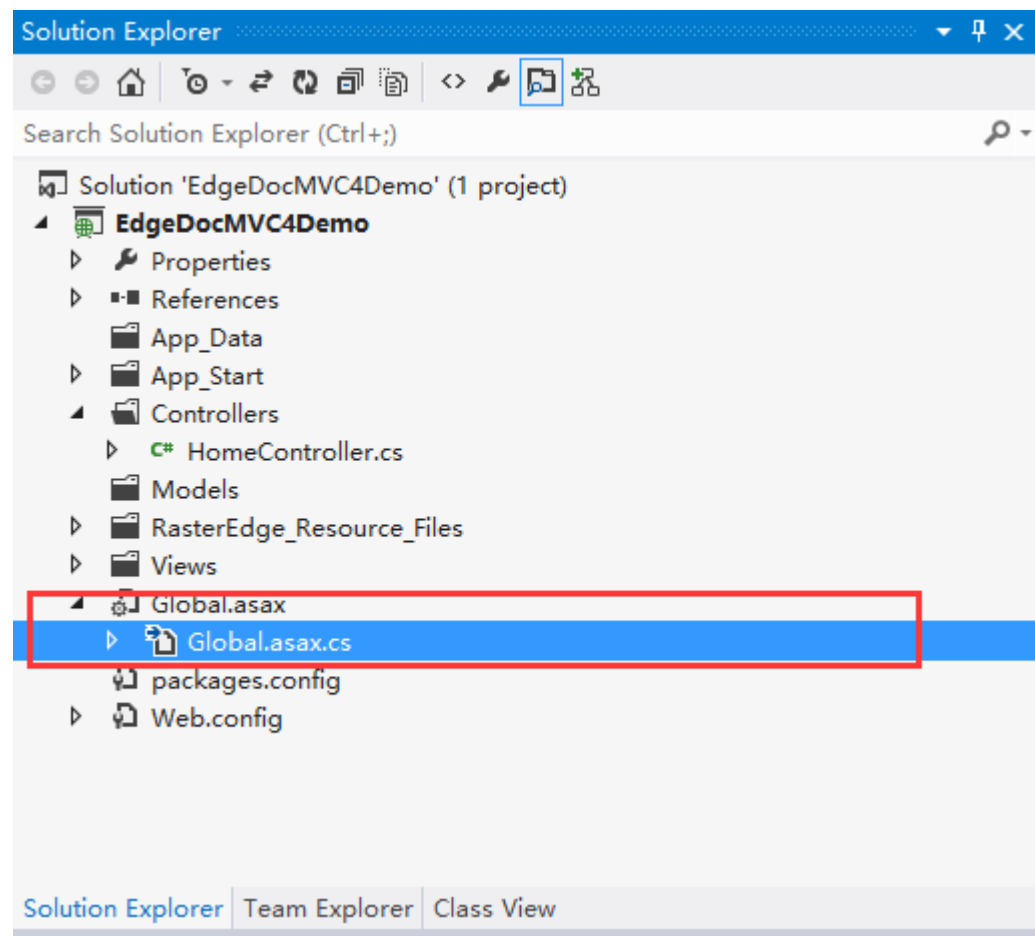
<httpRuntime requestValidationMode="2.0" executionTimeout="1200"
maxRequestLength="40960" useFullyQualifiedRedirectUrl="false"/>

```

## 7. Set the cache folder auto-clear rules

Copy or add the “/Developer Guide/EdgeDoc/MVC4 Razor project files/Global.asax.cs” file to

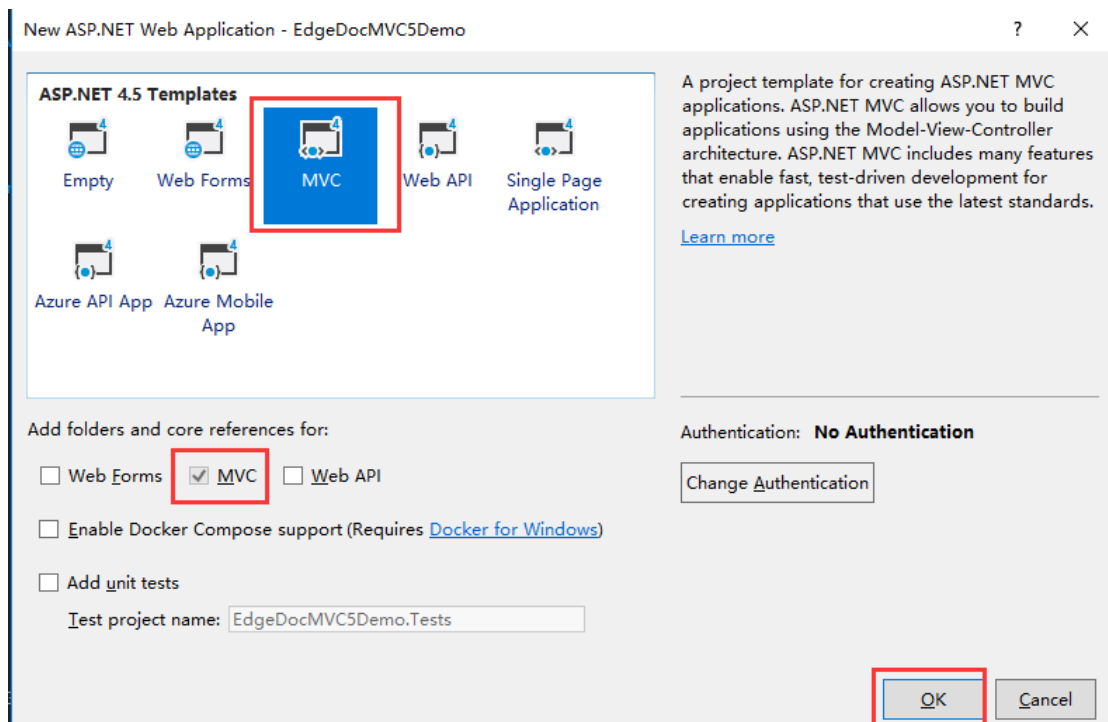
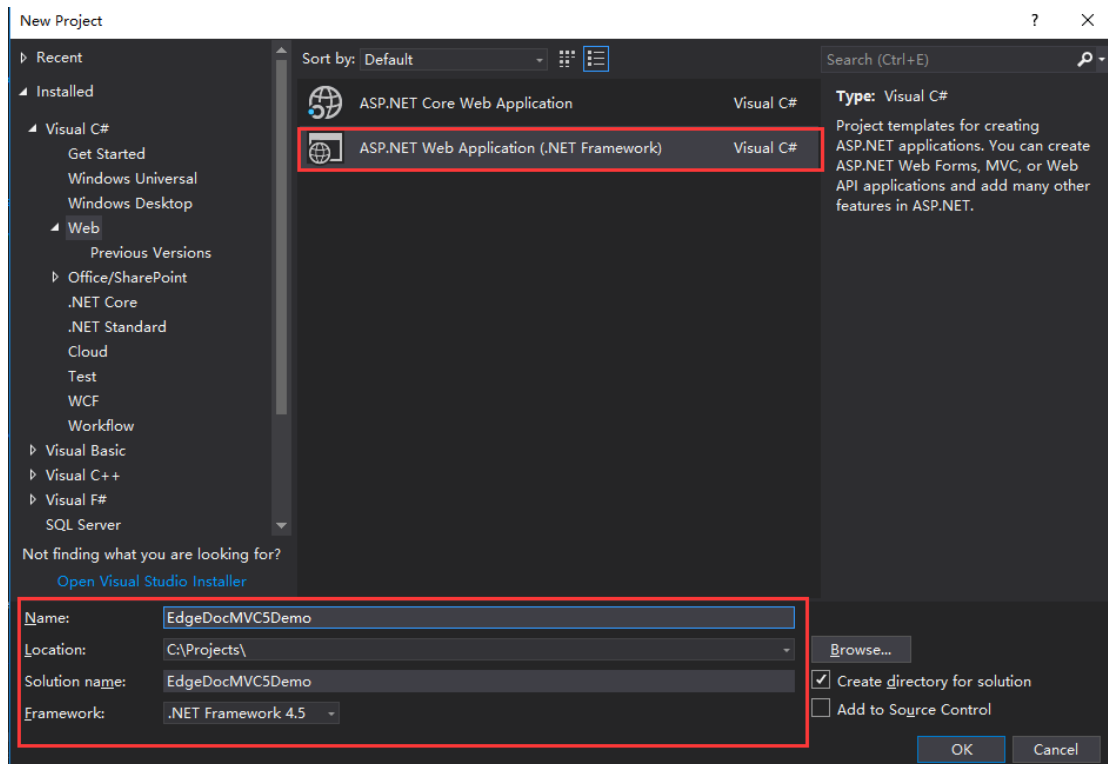
your project.



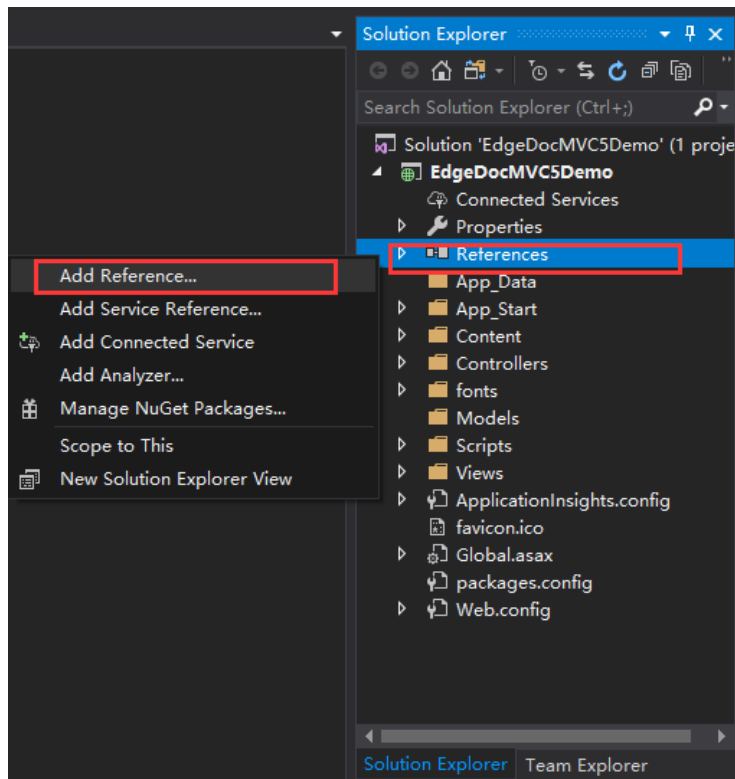
8. It is done. Now run the project.

## 4.3 ASP.NET MVC5 Web Application

### 1. Create a new MVC5 Project “EdgeDocMVC5Demo”



2. Add RasterEdgedll reference (add all dll names starting with RasterEdge)

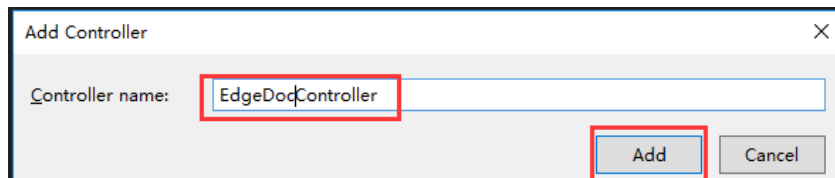
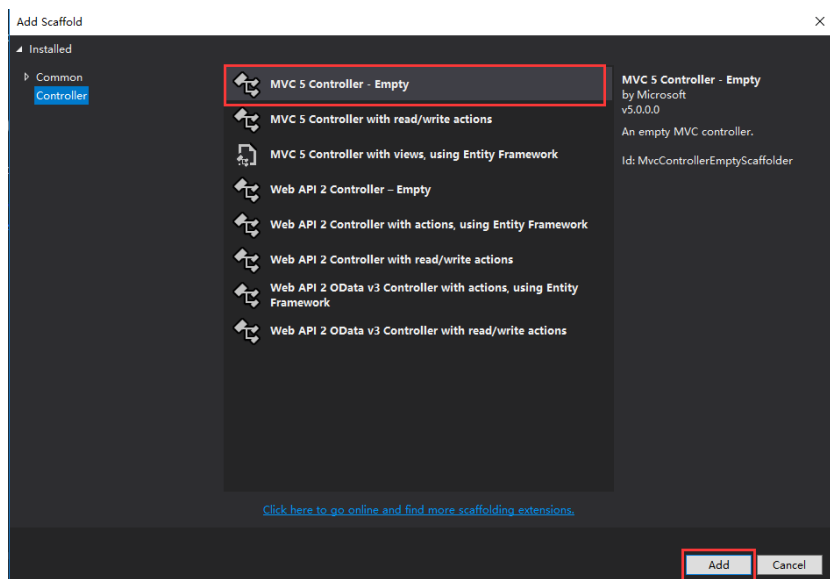
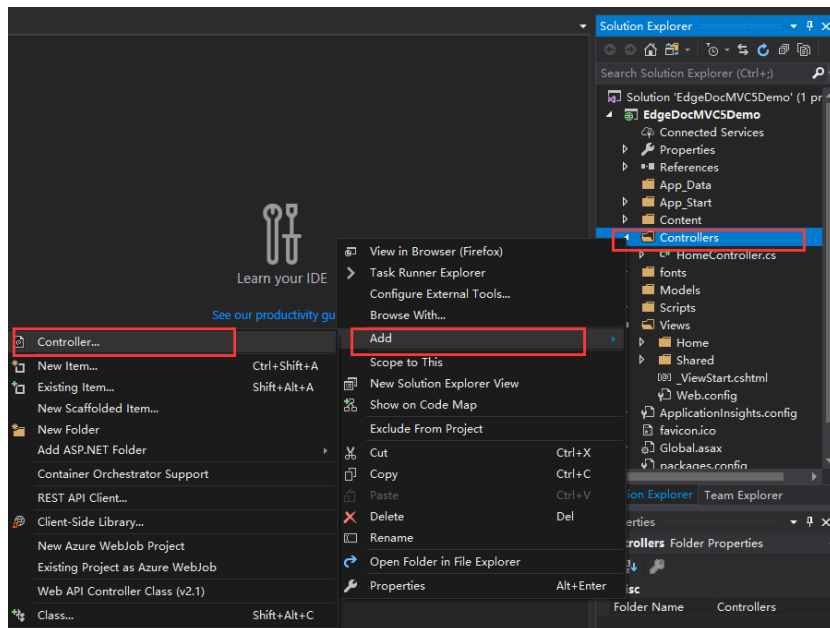


Please go to download package “/Bin”, choose **.net 4.0, x86 dlls**. Add **all dlls** with starting name “**RasterEdge.**” to your project reference.

**Note: Visual Studio embed web server (IIS Express) is for x86 only.**

**Please choose x86 dlls from Bin folder.**

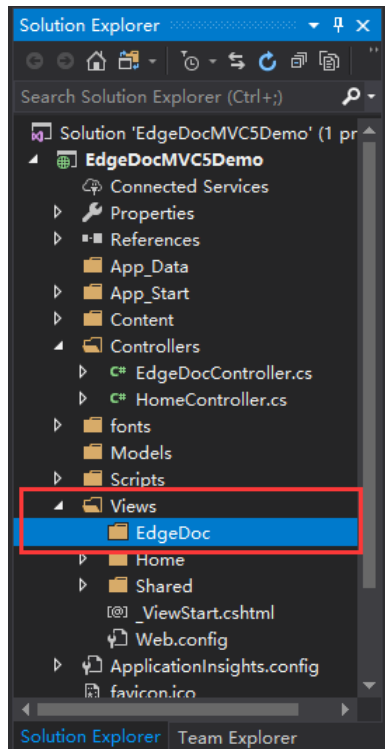
### 3. Create controller “EdgeDocController”



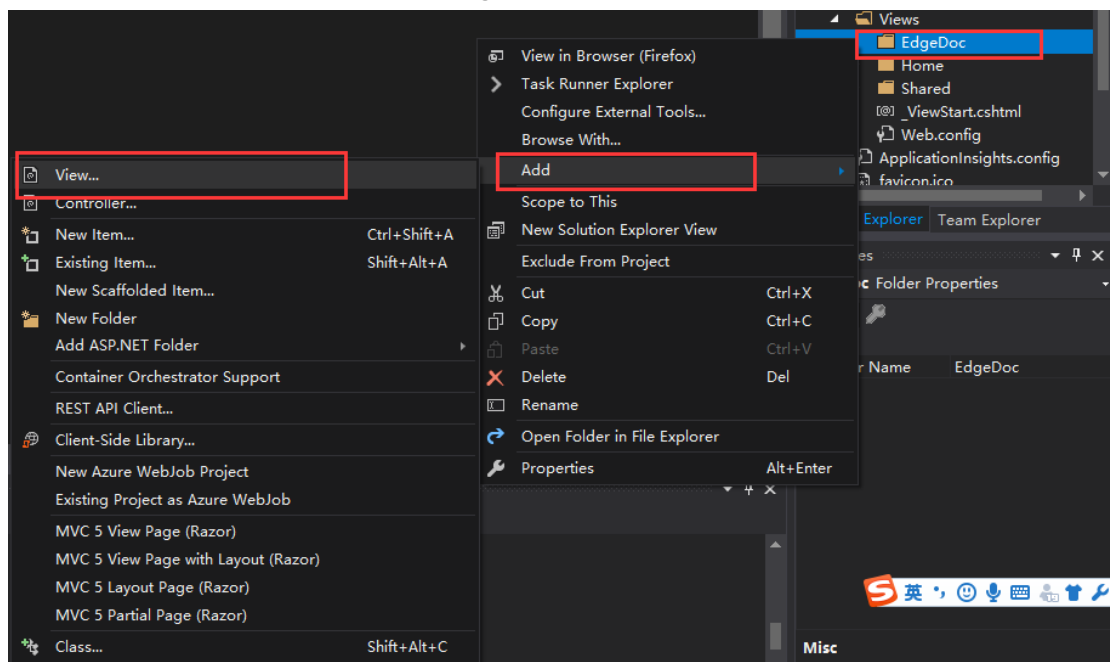
Click “Add”, to add a new controller “**EdgeDocController**” to your project.

## 4. Create View

Add a new folder “EdgeDoc” to the Views



Add a new View “Index” under folder “EdgeDoc”



Add View

View name: Index

Template: Empty (without model)

Model class:

Options:

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor \_viewstart file)

Add
Cancel

Copy the contents in file “/Developer Guide/EdgeDoc/ MVC5 project files/Index.cshtml” to your Index.html file.

EdgeDocMVC5Demo - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Any CPU Debug IIS Express (Firefox)

Index.cshtml EdgeDocController.cs EdgeDocMVC5Demo

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

```

Layout = null

```

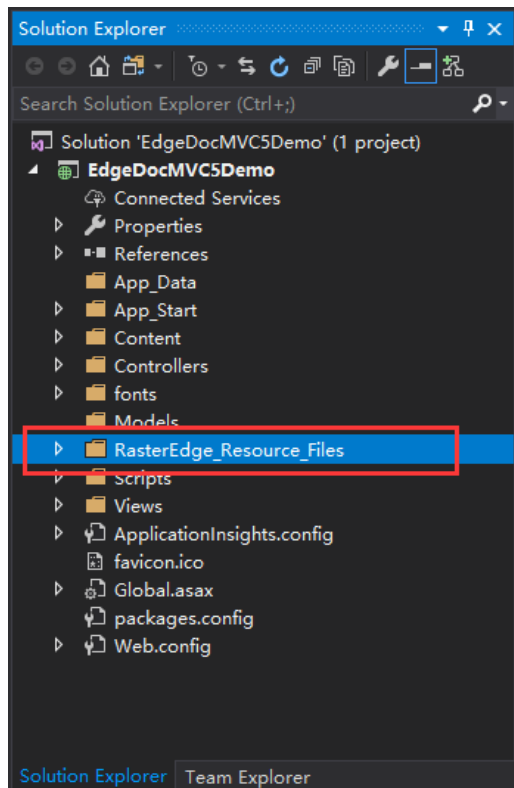
<DOCTYPE html>
<html>
<head>
<title>RasterEdge Application</title>
<script src="/RasterEdge_Resource_files/jquery.js" type="text/javascript"></script>
<script src="/RasterEdge_Resource_files/jquery/jquery-ui.js" type="text/javascript"></script>
<script src="/RasterEdge_Resource_files/jquery/jquery-ui.js" type="text/javascript"></script>
<script src="/RasterEdge_Resource_files/jquery/jquery-ui.js" type="text/javascript"></script>
<link rel="stylesheet" href="/RasterEdge_Resource_files/css/process.css" type="text/css" />
<link rel="stylesheet" href="/RasterEdge_Resource_files/css/customize.css" type="text/css" />
<script type="text/javascript">
var _rootpath = "/";
_MRAApp = new MRAOnlineApplication({
    _serverUrl: "/RasterEdge_Resource_files/UserCommandProcessor.ashx"
});
</script>
</head>
<body>
<div id="rasteredge_wdp">
<div id="wdp_toolbar"></div>
<div id="wdp_leftsidebar"></div>
<div id="wdp_center"></div>
<div id="wdp_rightsidebar"></div>
<div id="wdp_footer"></div>
</div>
</body>
</html>

```

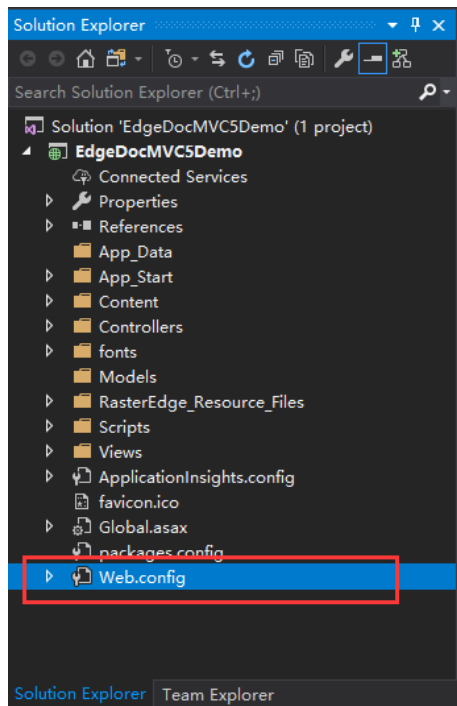


5. Add the folder “/DemoProjects/EdgeDoc Demo

Project/**RasterEdge\_Resource\_Files**”(all contents) to project.



## 6. Set the Web.config



Add the following contents to tag **<appSettings>**:

```
<!-- the cache folder -->
<add key="reCacheFolder" value="RasterEdge_Cache"/>
    <add key="reCustomStampFolder" value="RasterEdge_Resource_Files/images/stamp"/>
<!-- set the server folder if you want to open the file on server-->
<add key="reDefaultFolder" value="RasterEdge_Demo_Docs"/>
<!-- set the default file when open the web first time,if set "" ,this function is disabled.
(reDefaultFolder needed) -->
<add key="reDefaultFilePath" value=""/>
<!-- image quality in PDF document -->
<add key="rePageImageZoom" value="2"/>
<add key="reImageZoom" value="2"/>
<!-- image annotation in PDF document -->
<add key="reAnnotationImageZoom" value="2"/>
<!-- write log -->
<add key="reOutputLogData" value="true"/>
<!-- if true, SDK will automatically convert many shape and path inside PDF page into image-->
<!-- default is false -->
<!--  DEBUG ,INFO ,WARNING,ERROR -->
<add key="reLogLevel" value="DEBUG"/>
<add key="reWDPFileCacheMaxLimit" value="0"/>
<add key="reIntelliDrawShapesToImage" value="true"/>
<!-- if reIntelliDrawShapesToImage is true, -->
<!-- valid values are:      "all", "firefox", "chrome", "ie", "edge", "salari", "others"   if      "all"
```

```

included, ignore the rest;    default is "all"    -->
<add key="reIntelliDrawShapesToImageSupportBrowsers" value="all"/>
<!-- set the type of the show page.
    Valid values: svg, png
    Default: svg
-->
<add key="reDocRenderEngine" value="svg"/>
<add key="reRestfulFolder" value="c:/RasterEdge_Restful"/>
<add key="reI18NFolder" value="/RasterEdge_Resource_Files/i18n"/>
<add key="reServerURL" value=""/><!-- set the domain for remote client-->

```

Add the following contents to tag **<system.webServer>**

```

<security>
<requestFilteringallowDoubleEscaping="true">
<requestLimitsmaxAllowedContentLength="41943040"/>
</requestFiltering>
</security>
<staticContent>
    <remove fileExtension=".js"/>
<remove fileExtension=".svg"/>
<remove fileExtension=".woff"/>
<remove fileExtension=".ttc"/>
<remove fileExtension=".xpdf"/>
<remove fileExtension=".fdf"/>
<mimeMapfileExtension=".xpdf" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".fdf" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".woff" mimeType="application/octet-stream"/>
<mimeMapfileExtension=".svg" mimeType="image/svg+xml"/>
<mimeMapfileExtension=".ttc" mimeType="application/x-font-ttc"/>
<mimeMapfileExtension=".js" mimeType="application/x-javascript"/>
</staticContent>
    <urlCompressiondoStaticCompression="true" doDynamicCompression="true"/>

```

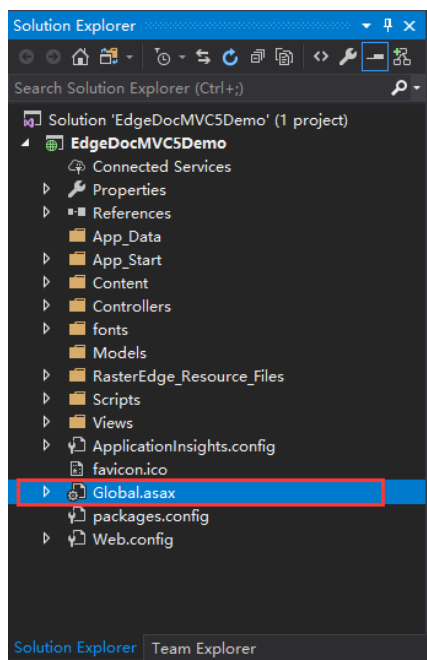
Add the following contents to tag **<system.web>**

```
<httpRuntime requestValidationMode="2.0" executionTimeout="1200"
maxRequestLength="40960" useFullyQualifiedRedirectUrl="false"/>
```

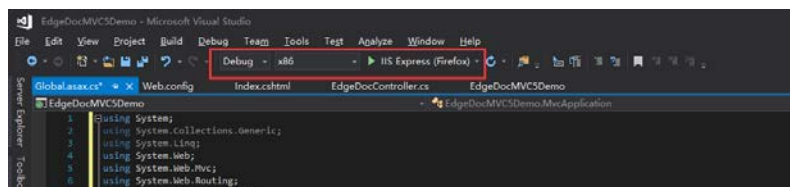
Note: If web.config has already defined tag “<httpRuntime>”, you need merge the above content with existed tag contents.

## 7. Set the cache folder auto-clear rules

Copy or add the “/Developer Guide/EdgeDoc/MVC5 project files/Global.asax.cs” file to your project.



## 8. Change target platform to “x86”



9. It is done. Now run the project. Web browser navigate to

“<http://localhost:6477/edgedoc/>”

## 4.4 Setup EdgeDoc Viewer in the IIS Application

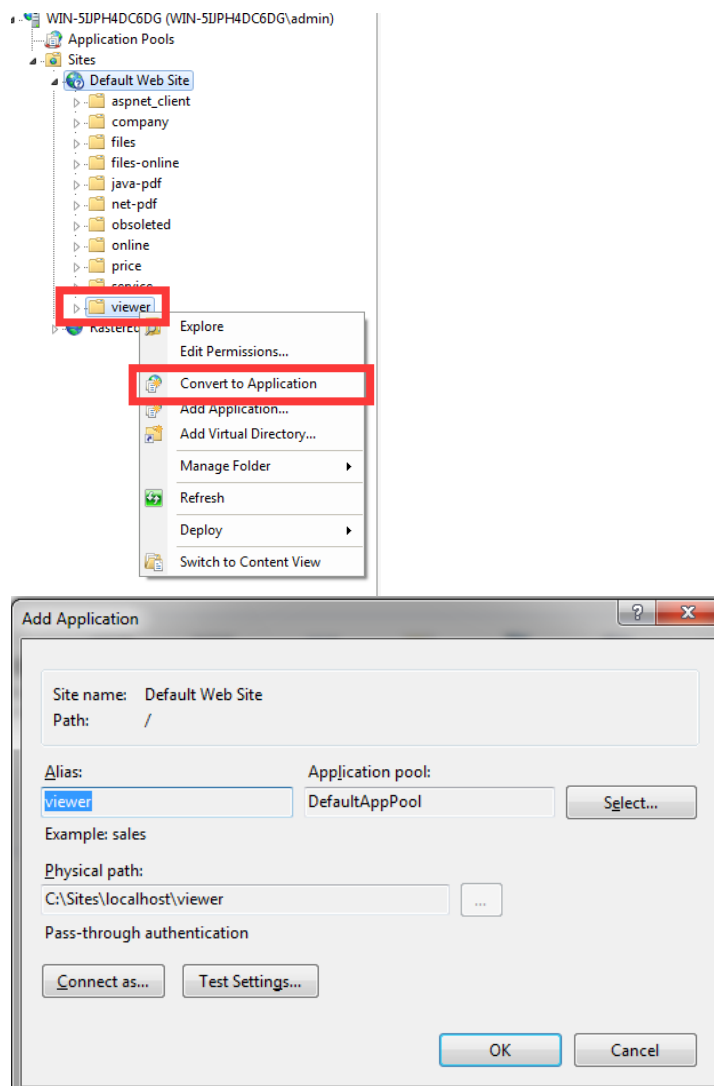
### 1. Target

Install EdgeDoc as an application to IIS site <http://localhost/viewer/>

IIS Environment:

- .net 4.0
  - x86 bit
2. Make sure localhost (<http://localhost>) is running normally. By default, localhost site is inside "C:\\inetpub\\wwwroot\\"
  3. Create a new folder "viewer", under "C:\\inetpub\\wwwroot\\".
  4. Copy all contents from "/DemoProjects/EdgeDoc Demo Project/"
  5. Grant user group "IIS\_IUSRS" full control permission to folder "C:\\inetpub\\wwwroot\\viewer\\RasterEdge\_Cache", where EdgeDoc will generate cache files.
  6. Copy .net 4.0, x86 DLLs to bin folder. (You can choose .net 4.0, x64 DLLs also to match your IIS settings)

## 7. Convert folder “viewer” to application



## 8. Update file “index.html”: Change Client side file path

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" moznoMarginBoxesmozDisallowSelectionPrint>

<head>

<title>RasterEdge Application</title>

<script src="/viewer/RasterEdge_Resource_Files/javascript/jquery.js" type="text/javascript"></script>

<script src="/viewer/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp_Customize.js" type="text/javascript"></script>

<script src="/viewer/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp.js" type="text/javascript"></script>

<script src="/viewer/RasterEdge_Resource_Files/javascript/RasterEdge.js" type="text/javascript"></script>

<link rel="stylesheet" href="/viewer/RasterEdge_Resource_Files/css/process.css" type="text/css" />

<link rel="stylesheet" href="/viewer/RasterEdge_Resource_Files/css/customize.css" type="text/css" />

<link rel="stylesheet" href="/viewer/RasterEdge_Resource_Files/css/jquery-ui.css" type="text/css" />

<script type="text/javascript">

    var _rootpath = "/viewer/";

    _WDPApp = new WDPOnlineApplication({

        _serverUrl: "RasterEdge_Resource_Files/UserCommandProcessHandler.ashx"

    });

</script>

</head>

<body>

<div id="rasteredge_wdp">

<div id="wdp_Toolbar"></div>

<div id="wdp_LeftSidebar"></div>

<div id="wdp_Viewer"></div>

<div id="wdp_RightSidebar"></div>

<div id="wdp_Footer"></div>

</div>

</body>

</html>
```

## 9. Open web browser, and navigate to <http://localhost/viewer/>

## 5.EmbedEdgeDoc Viewer into Web Page

In demo project, viewer (index.html) will render and display the user documents in a full screen web page.

Here we have also included two sample html pages to explain how to embed EdgeDoc into your existing web pages (html or aspx).

### **Sample 1: Embed viewer into your html, aspx pages using DIV and iFrame:edgedoc-page-frame.html**

It embeds EdgeDoc viewer in one div tag inside html page.

To enable this sample page,

1. you need have 3 pdf files (**1.pdf, 2.pdf, 3.pdf**) under your local system  
"C:\temp\" folder,
2. your IIS user (IIS\_IUSRS) has the read privilege for folder "**temp**".
3. Start the demo project, open web page  
<http://localhost/edgedoc-page-frame.html>.
4. Click three links on the left top corner.



How to customize this sample:

1. In “/RasterEdge\_Resource\_Files/UserCommandProcessHandler.ashx”  
method “FileProcess”,

```
else if (!string.IsNullOrEmpty(request.QueryString["yourtarget"]))
{
    // load file from your target
    string targetkey = request.QueryString["yourtarget"];
    RasterEdge.Imaging.Basic.BaseDocument pdf =
    new RasterEdge.XDoc.PDF.PDFDocument(@"C:\temp\" + targetkey);

    webDoc = REProcessControl.PageLoadFile(request, pdf, targetkey); // object can be PDFDocument, File bytes or File Stream
}
```

You can change http url parameter “yourtarget” to your defined url parameter name, and change file location (“C:\temp\”) also.

If the above settings are correct, you can open file inside folder “temp”, using url (<http://localhost/?yourtarget=1.pdf>) in web browser.

2. In “/edgedoc-page-frame.html”, you can define your file list in html

```
<a href="#" onclick="openfile('1.pdf')">Open file 1</a><br />
<a href="#" onclick="openfile('2.pdf')">Open file 2</a><br />
<a href="#" onclick="openfile('3.pdf')">Open file 3</a><br />
```

And your file open url in Javascript

```
function openfile(filesrc) {
    if (filesrc != "")
        $("#wdp_container").attr("src", "index.html?yourtarget=" + filesrc);
}
```

**Sample 2: Popup viewer in your html, aspx pages:**

**edgedoc-page-popup.html**

It will popup the EdgeDoc viewer inside html page.

## 6.Document Input &Output

To run the following demo code quickly, please do the following steps:

- Create add pdf document to “c:\temp\1.pdf”
- Make sure **IIS application** has the **read and write rights** to the folder “temp”, and file “1.pdf”

### 6.1 Load file into EdgeDoc

*This example will open a PDF file located under your server “C:/temp/1.pdf”, through url:*

<http://localhost:9090/?yourtarget=/1.pdf>.

*Next one will allow you to define a new command in toolbar. And you can save modified pdf file to original file location “C:/temp/1.pdf”*

WithEdgeDoc online document viewer, you can easily load a file from your server file system or database into viewer.

Here are the details

1. Setup EdgeDoc demo project in IIS
2. In /RasterEdge\_Resource\_Files/UserCommandProcessHandler.ashx method “**FileProcess()**”,

```
// load file by your target parameter value
string targetkey = request.QueryString["yourtarget"];

// object can be PDFDocument,File bytes or File Stream
RasterEdge.Imaging.Basic.BaseDocument pdf =
    new RasterEdge.XDoc.PDF.PDFDocument(@"C:\temp\" + targetkey);

webDoc = REProcessControl.PageLoadFile(request, pdf, "yourdoc.pdf");
```

You can use the targetKey value to locate your file from file system or database, and load file into PDFDocument object.

3. Now you navigate to “**http://localhost/?yourtarget={your file location key}**” to open file in EdgeDoc viewer.
4. You can change http parameter **yourterget** to other names, like fileid.

## 6.2 Save file from EdgeDoc into your server file system or database

Define your saving command in EdgeDoc web viewer toolbar

Navigate to **{EdgeDoc Demo Project}/RasterEdge\_Resource\_Files/Javascript/RasterEdge\_WebApp\_Customize.js**

Define your command in function initCustomize()

```
var myIcon = new CToolBarIcon();
myIcon.id="myIcon";
myIcon.event = "saveOnServer()";
myIcon.title = "Save on server";
myIcon.reqDoc = true;

var myGroup = new CToolBarIconGroup();
myGroup.name="Save Server";
myGroup.addIcon(myIcon);

var myTab = new CToolBar();
myTab.id = "Customizebar";
myTab.name = "Customize";
myTab.addIconGroup(myGroup);

addToolBarTab(myTab);
```

Define server side processing method to handle your command

```
// demo function to save the file on server
function saveOnServer() {
    // whether the file is loaded
    if (getCurrentFileId() == "")
        return;

    // get the basic save datas from API(getSaveDatas)
    var datas = getSaveDatas();
    // set the customize calling function in userCommandProcessHandler
    datas.action = "SaveFileOnServer";

    var options = {
        type: "POST",
        url: getServerHandlerUrl(),
        async: false,
        data: datas,
        success: function (result) {
            // hide the loading panel
            $("#loading").hide();
            // get the message
            var arr = eval('(' + result + ')');
            if (arr.state == "success") {
                saveFile = arr.msg;
                // open the file that store on server
                window.open(_cacheFolder + "/" + getCurrentFileId() + "/" + saveFile);
            }
            else {
                alert(arr.msg);
            }
        },
        error: function (err) {
            $("#loading").hide();
        }
    }

    // show the loading panel
    $("#loading").show();
    // send the message
    $.ajax(options);
}
```

Process your saving command in the server side

Navigate to **{EdgeDoc Demo Project}/**

**RasterEdge\_Resource\_Files/UserCommandProcessHandler.ashx**

```
public void SaveFileOnServer()
{
    string fid = _httpRequest["fid"];
    // you can process saved pdf document here, like add watermark, ...
    object document= this.SaveFile();
    if (document is RasterEdge.XDoc.PDF.PDFDocument)
    {
        RasterEdge.XDoc.PDF.PDFDocument pdfDoc = document as
RasterEdge.XDoc.PDF.PDFDocument;
        // get the upload information of the file
        RasterEdge.WDP.DocUploadInfo docinfo= this.getDocUploadInfo(fid);

        string filename = docinfo.FileName;
        // get your file open url parameters value, if needed.
        string paraFilepathValue = docinfo.GetRequestParameters("yourtarget");

        RasterEdge.XDoc.PDF.PDFMetadata metadata = pdfDoc.GetDescription();
        metadata.Producer = "RasterEdgeEdgeDoc";
        pdfDoc.SetDescription(metadata);

        try
        {
            pdfDoc.Save(@"C:\\temp\" + paraFilepathValue);
        }
        catch (Exception e)
        {
            // process error code, and return error information here
        }
    }
}
```

## Send response message to user web browser

In case, you need send some information to the user web browser (maybe error message).

In `UserCommandProcessHandler.ashx`, you need send text content to the web browser

```
public void SaveFileOnServer()
{
    ...
    catch (Exception e)
    {
        // process error code, and return error information here
        this.Context.Response.ClearContent();
        this.PrintErrorJson("fail To save(or other information yourwant,and your can do some action in
your custom js function.)");// state is error
    }
    ...
}
```

In `RasterEdge_WebApp_Customize.js`, you need display the text content

```
---
success: function (result) {
    // hide the loading panel
    $("#loading").hide();
    // get the message
    var arr = eval('(' + result + ')');
    if (arr.state == "success") {
saveFile = arr.msg;
        // open the file that store on server
        // window.open(getCacheFileFolder(getCurrentFileId()) + saveFile);
location.reload(true);
    }else if(arr.state=="error"){
        // TODO   your action here
    }
    else {
        alert(arr.msg);
    }
},
error: function (err) {
    $("#loading").hide();
}
```

## 7. PDF Font Management

RasterEdge SDK will process the PDF document font resources using the following order

1. Font resource embedded in the PDF document
2. Manual installed font resource
3. Runtime installed font resource
4. Windows system font resource "C:\Windows\Fonts\"

In following content, we will explain "Manual installed font resource" and "Runtime installed font resource"

### 7.1 Install Adobe fonts to system

Step 1: Install Adobe Reader DC to the target PC.

Website: <https://get.adobe.com/reader/>

Step 2: Move into the folder %Adobe Reader DC Install Folder%\Resource\Font\

Step 3: Install font files in the folder.

- MinionPro-Regular.otf
- MinionPro-Bold.otf
- MinionPro-It.otf
- MinionPro-BoldIt.otf
- MyriadPro-Regular.otf
- MyriadPro-Bold.otf
- MyriadPro-It.otf
- MyriadPro-BoldIt.otf

a. **Copy** font files in the above folder to the system folder.

Eg. C:\Windows\Fonts\

or

b. **Right click** \*.otf file to open the menu, and then choose "**Install**" in the menu.



## 7.2 Install Adobe fonts in Runtime

Sample Code:

```
String fontFolder = @"C:\Program Files (x86)\Adobe\Acrobat Reader DC\Resource\Font\";

PrivateFontMgr handler = PrivateFontMgr.Handler;
// install Adobe font Minion Pro Regular
handler.InstallPrivateFont(new PrivateFontInfo("Minion Pro", FontStyle.Regular), fontFolder +
"MinionPro-Regular.otf");
// install Adobe font Minion Pro Bold
handler.InstallPrivateFont(new PrivateFontInfo("Minion Pro", FontStyle.Bold), fontFolder +
"MinionPro-Bold.otf");
// install Adobe font Minion Pro Italic
handler.InstallPrivateFont(new PrivateFontInfo("Minion Pro", FontStyle.Italic), fontFolder +
"MinionPro-It.otf");
// install Adobe font Minion Pro Bold Italic
handler.InstallPrivateFont(new PrivateFontInfo("Minion Pro", FontStyle.Bold |
FontStyle.Italic), fontFolder + "MinionPro-BoldIt.otf");
// install Adobe font Myriad Pro Regular
handler.InstallPrivateFont(new PrivateFontInfo("Myriad Pro", FontStyle.Regular), fontFolder +
"MyriadPro-Regular.otf");
// install Adobe font Myriad Pro Bold
handler.InstallPrivateFont(new PrivateFontInfo("Myriad Pro", FontStyle.Bold), fontFolder +
"MyriadPro-Bold.otf");
// install Adobe font Myriad Pro Italic
handler.InstallPrivateFont(new PrivateFontInfo("Myriad Pro", FontStyle.Italic), fontFolder +
"MyriadPro-It.otf");
// install Adobe font Myriad Pro Bold Italic
handler.InstallPrivateFont(new PrivateFontInfo("Myriad Pro", FontStyle.Bold |
FontStyle.Italic), fontFolder + "MyriadPro-BoldIt.otf");
```

## 8. Customize EdgeDocViewer

### 8.1 Re-locate Viewer client side files

Change the location of index.html, in file index.html

```
var _rootpath = "/home/viewer/";  
  
_WDPApp = new WDPOnlineApplication({  
    _serverUrl: "/RasterEdge_Resource_Files/UserCommandProcessHandler.ashx"  
});
```

Change the location of **JS**, **image**, and **CSS** files, in the file index.html

```
<scriptsrc="/js-path/RasterEdge_Resource_Files/javascript/jquery.js"type="text/javascript"></script>  
<scriptsrc="/js-path/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp_Customize.js"type="text/javascript"></script>  
<scriptsrc="/js-path/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp.js"type="text/javascript"></script>  
<scriptsrc="/js-path/RasterEdge_Resource_Files/javascript/RasterEdge.js"type="text/javascript"></script>  
<linkrel="stylesheet"href="/css-path/RasterEdge_Resource_Files/css/process.css"type="text/css"/>  
<linkrel="stylesheet"href="/css-path/RasterEdge_Resource_Files/css/customize.css"type="text/css"/>  
<linkrel="stylesheet"href="/css-path/RasterEdge_Resource_Files/css/jquery-ui.css"type="text/css"/>
```

**Note:** folder **“images”** must always be under the same location with folder **“css”**.

### 8.2 Re-locate Viewer web service files

Contents of viewer web service files:

```
/Bin/  
/RasterEdge_Cache/  
/RasterEdge_Resource_Files/UserCommandProcessHandler.ashx  
/Global.asax  
/Web.Config
```

To change the location of folder **“RasterEdge\_Cache”**

In Web.config file, change the value of appSetting **“reCacheFolder”**, to the new location. Please use the **relative path** with your web application.

## 8.3UI: Viewer width and height settings

It is easy to define Viewer width and height with absolute size or relative size.

Viewer width and height with absolute size:

```
<div style="width: 1200px; height: 800px;">  
<iframe name="edgedoc" src="http://localhost" style="width: 100%; height: 100%"></iframe>  
</div>
```

Viewer width and height with relative size:

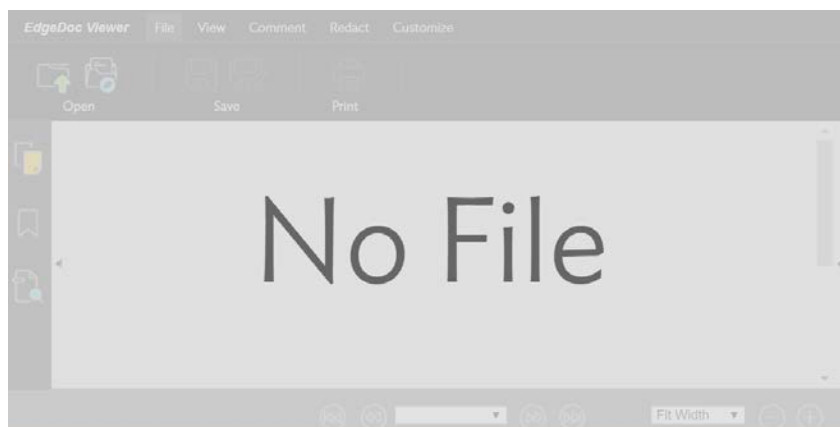
```
<div style="width: 90%; height: 600px;">  
<iframe name="edgedoc" src="http://localhost" style="width: 100%; height: 90%"></iframe>  
</div>
```

## 8.4 UI: Viewer startup form view

By default, when viewer starts, it will display a form to allow user to upload a file to display. Sometimes, you do not allow users to upload their own files. You can change viewer startup status to “**NOFILE**”.

In {EdgeDoc Demo Project}/ RasterEdge\_Resource\_Files/Javascript/**RasterEdge\_WebApp.js**,  
Go to function **setDialog**, change value to “**NOFILE**”.

```
setDialog(“NOFILE”);    // valid values: “NOFILE”, “DOC”
```



## 8.5UI: Viewer HTML page title

By default, Viewer html page title will show current displayed document file name. You can disable this feature.

In {EdgeDoc Demo Project}/ RasterEdge\_Resource\_Files/Javascript/**RasterEdge\_WebApp.js**

Go to function **setWebTitle**, use the following JS code:

true: it will show current displayed document file name.

false: it will not.

```
setWebTitle("EdgeDoc Viewer: online view, comment, redact, convert documents.",true);
```

## 8.6Comments settings

In {EdgeDoc Demo Project}/RasterEdge\_Resource\_Files/Javascript/**RasterEdge\_WebApp.js**,

Go to function **setCommentsStyle()**, here you can modify all comments item default settings.

## 8.7Customize toolbar and commands

It is really easy to define your own toolbar and commands. Navigate to **{EdgeDoc Demo Project}/ RasterEdge\_Resource\_Files/Javascript/RasterEdge\_WebApp.js**. All toolbar, command, command groups are defined in this file. You can delete, re-order any of them.

## 8.8 Multiple languages support

EdgeDoc ASP.NET Document Viewer allows you to define multiple languages for your users. By default, the web application text content is in **English**.

If you need support multiple languages, please go to **{EdgeDoc Demo Project}/RasterEdge\_Resource\_Files/i18n/** to define your preferred languages.

Once you have defined your languages, according to your web application users' web browser language setting, EdgeDoc will automatically choose the right language for your users.

To reload language file without restart IIS, you can use this http parameter: <http://localhost:8080/index.html?relang=en-us>

To ignore web browser language setting and use the your defined the language, you can use this http parameter:

<http://localhost:8080/index.html?lang=en-us>

## 8.9 Cached file data management

EdgeDoc provides 4 settings to manage the cached file data.

File Life Max Time	If max life time is reached, the file cached data is ready for deletion
File Cache Clean Time	If time is reached, EdgeDoc will delete all ready for deletion cached data
User Request Max Time	If max life time is reached, the user request cached data is ready for deletion
User File Request Cache Clean Time	If time is reached, EdgeDoc will delete all ready for deletion user cached data

You can set the above 4 settings at **Global.asax**, in method **TimerTask()**

```
private void TimerTask()
{
    RegisterCacheEntry();
    Scheduler.ClearAllFolder();

    Scheduler.StartFileCacheCleanTask(
        FileCacheCleanType.FileLifeMaxTime, new IntervalTime(3, 0, 0));
    Scheduler.StartFileCacheCleanTask(
        FileCacheCleanType.FileCacheCleanTime, new IntervalTime(3, 0, 0));
    Scheduler.StartUserCacheCleanTask(
        UserCacheCleanType.UserRequestMaxTime, new IntervalTime(3, 0, 0));
    Scheduler.StartUserCacheCleanTask(
        UserCacheCleanType.UserFileRequestCacheCleanTime, new IntervalTime(3, 0, 0));

    Logger.LogSystem("Cache Auto clear Start", LogType.INFO);
}
```

## 9. Customize, extend Annotations, Comments, Markups

### 9.1 Set annotation default properties cookie live time

In RasterEdge\_WebApp.js, initWebApplication(), you can set your preferred cookie expiring dates.

**setCookieExpires(7);**//day

### 9.2 Enable continuous annotations

To enable continuous annotation adding, please do the following change:

1. Open /RasterEdge\_Resource\_Files/Javascript/RasterEdge\_WebApp\_Customize.js
2. In JS function "initCustomize", change **supportContinuousAnnotationAdding(false)**; to **true**;

## 9.3 Create a new stamp annotation

To create a new **dynamic** stamp annotation

1. Define a new stamp annotation icon in “Comment” toolbar using css. You may need change stamp toolbar icon location

```
.stampVOID {background-size:100%  
100%;background-image:url('../images/stamp/VOID.png') ;background-repeat: no-repeat;}
```

2. Insert new stamp annotation to the toolbar using JS. In JS file  
/RasterEdge\_Resource\_Files/Javascript/RasterEdge\_WebApp\_Customize.js, in function  
“function initCustomizedStampCommands()”, add the following code:

```
{ name: 'VOID', width: 120, height: 64, color: "#c00000", cssClass: "VOID" },
```

3. Run the web program, you will find a new dynamic stamp annotation “VOID” added to the stamp annotations panel.

To create a new **image** stamp annotation

1. Define a new stamp annotation icon in “Comment” toolbar using css. **You may need change stamp toolbar icon file location**

```
.stampCustomStamp {  
    background-size:100% 100%;  
background-image:url('../images/stamp/CustomStamp.png');  
    background-repeat: no-repeat;  
}
```

2. Insert new stamp annotation to the toolbar using JS. In JS file  
/RasterEdge\_Resource\_Files/Javascript/**RasterEdge\_WebApp\_Customize.js**, in function  
“function **initCustomizedStampCommands()**”, add the following code:

```
{ name: 'CustomStamp', width: 120, height: 64, cssClass: "CustomStamp",  
CustomStampName: "CustomStamp.png" },
```

3. In Web.Config file <appSettings>, Define all of your image stamp annotation image files server side location

```
<add key="reCustomStampFolder" value="RasterEdge_Resource_Files/images/stamp"/>
```

4. Run the web program, you will find a new image stamp annotation added to the stamp annotations panel.



## 9.4Get user mouse click coordinates

To get the user mouse last click coordinates, please use the following sample JavaScript codes in file

/RasterEdge\_Resource\_Files/Javascript/**RasterEdge\_WebApp\_Customize.js**

```
-----RasterEdge_WebApp_Customize.js-----

function getLastCoordinates() {
    // get the coordinates of the last time under Select mode
    var location = getCoordinatesInDocPage();
    if (location != null)
        console.log("x,y:(" + location.x + "," + location.y + "),pageIndex:" + location.pageIndex);
}
var getCoordinates = function (location) {
    // get the coordinates under Select mode, call by system after mouse down
    //if (location != null)
    //    console.log("x,y:(" + location.x + "," + location.y + "),pageIndex:" +
location.pageIndex);
}
-----
```

## 10. Enable Watermark in the EdgeDoc

Sometimes you need display watermarks on all the documents displayed in the EdgeDoc. We provide easy job to enable watermark in the EdgeDoc.

If you want to display watermark on all web users uploaded documents, you can add the following code in file [/RasterEdge\\_Resource\\_Files/UserCommandProcessHandler.ashx](#)

In method **FileProcess()**;

```
        if (request.Files.Count > 0 || request.Form.Count > 0)
        {
            // load from upload (new file or add file)
            //webDoc = REProcessControl.PageLoadFile(request);

RasterEdge.Imaging.Basic.REWatermarkSourceTexttextwatermark =
            new RasterEdge.Imaging.Basic.REWatermarkSourceText("Rasteredge_2018_New",new
System.Drawing.Font("Arial", 12F));
RasterEdge.Imaging.Basic.REWatermark watermark = new
RasterEdge.Imaging.Basic.REWatermark(textwatermark);

watermark.RelativeHeight = 80F;
watermark.RelativeWidth = 80F;
watermark.HoriAlign = RasterEdge.Imaging.Basic.REHoriAlignment.Middle;
watermark.VertAlign = RasterEdge.Imaging.Basic.REVertAlignment.Middle;
watermark.Rotate = 45F;
watermark.ScaleMode = RasterEdge.Imaging.Basic.REScaleMode.None;
watermark.Opacity = 0.8F;
watermark.IsAbovePage = true;

            // load from upload (new file or add file)
webDoc = REProcessControl.PageLoadFile(request,watermark);

        }
```

You can enable **text watermark** or **image watermark** in the EdgeDoc. For watermark class API, please go to developer guide [/Developer Guide/XDoc.Basic/Basic Reference -- General Watermark.pdf](#)

## 11. Configure EdgeDoc through Web.configappSettings

reCacheFolder	<p>Specify the location of cache folder. Please use the <b>relative path</b> with your web application.</p> <p>Grant user "IIS_IUSRS" full control permission to this folder, where HTML5 Document Viewer will generate cache files.</p>
reResourceFolder	the resource folder(js,image,css)
reDefaultFolder	set the server folder if you want to open the file on server
reDefaultFilePath	set the default file when open the web first time,if set "" ,this function is disabled. (reDefaultFolder needed)
rePageImageZoom	<p>Default is 2.</p> <p>If a PDF document page will be rendered into one image file, this setting will decide page image zooming value.</p>
relImageZoom	<p>Default is 2.</p> <p>Zooming value for all images inside PDF document pages</p>
reAnnotationImageZoom	Image annotation in PDF document
reOutputLogData	<p>Output log data file.</p> <p>Default is false.</p>
reLogLevel	Level of log data output
reIntelliDrawShapesToImage	<p>If SDK find too many shapes and paths inside one PDF page, SDK will automatically convert the whole PDF page into a single image.</p> <p>Default is false.</p>
reIntelliDrawShapesToImageSupportBrowsers	<p>if reIntelliDrawShapesToImage is true,</p> <p>Valid values: "all", "firefox", "chrome", "ie", "edge", "salari", "others"</p> <p>If "all" included, ignore the rest;</p> <p>Default is "firefox, ie"</p>
reDocRenderEngine	Set the type of the show page.

	<div>Valid values: svg, png</div> <div>Default: svg</div>
--	---

# 12. Common Exceptions and Errors

## 12.1 Could not load file or assembly

### 'RasterEdge.Imaging.Annotation'

If your used RasterEdge DLLs version does not match your .net platform, you will view this Exception “**Could not load file or assembly ‘RasterEdge.Imaging.Annotation’ or one of its dependencies.**”

```
Server Error in '/' Application.

Could not load file or assembly 'RasterEdge.Imaging.Annotation' or one of its dependencies. An attempt was made to load a program with an incorrect format.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.BadImageFormatException: Could not load file or assembly 'RasterEdge.Imaging.Annotation' or one of its dependencies. An attempt was made to load a program with an incorrect format.

Source Error:

An unhandled exception has occurred during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Assembly Load Trace: The following information can be helpful to determine why the assembly 'RasterEdge.Imaging.Annotation' could not be loaded.

WRN: Assembly binding logging is turned OFF.
To enable assembly bind failure logging, set the registry value [HKLM\Software\Microsoft\Fusion!EnableLog] (DWORD) to 1.
Note: There is some performance penalty associated with assembly bind failure logging.
To turn this feature off, remove the registry value [HKLM\Software\Microsoft\Fusion!EnableLog].

Stack Trace:

[BadImageFormatException: Could not load file or assembly 'RasterEdge.Imaging.Annotation' or one of its dependencies. An attempt was made to load a program with an incorrect format.]
System.Reflection.RuntimeAssembly._loadAssemblyName(String fileName, String codebase, Evidence assemblySecurity, RuntimeAssembly locationHint, StackFrame& stackMark, IntPtr pPrivateBinder, Boolean throwOnFileNotFound, Boolean forIntrop, System.Reflection.RuntimeAssembly.InternalLoadAssemblyName(String assemblyName, Evidence assemblySecurity, RuntimeAssembly replaceAssembly, StackFrame& stackMark, IntPtr pPrivateBinder, Boolean throwOnFileNotFound, Boolean forIntrop) +27
System.Reflection.RuntimeAssembly.InternalLoad(String assemblyString, Evidence assemblySecurity, StackFrame& stackMark, IntPtr pPrivateBinder, Boolean forIntrop) +21
System.Reflection.Assembly.Load(String assemblyString) +35
System.Web.Configuration.CompilationSection.LoadAssemblyHelper(String assemblyName, Boolean starDirective) +18

[ConfigurationErrorsException: Could not load file or assembly 'RasterEdge.Imaging.Annotation' or one of its dependencies. An attempt was made to load a program with an incorrect format.]
System.Web.Configuration.CompilationSection.LoadAssemblyHelper(String assemblyName, Boolean starDirective) +18
System.Web.Configuration.CompilationSection.LoadAllAssembliesFromAppDomainDirectory() +217
System.Web.Configuration.CompilationSection.LoadAssembly(AssemblyInfo ai) +138
System.Web.Compilation.BuildManager.GetReferencedAssemblies(CompilationSection compConfig) +178
System.Web.Compilation.BuildManager.GetPreStartInMethodsFromReferencedAssemblies() +42
System.Web.Compilation.BuildManager.CallPreStartInMethods(String preStartInitPath, Boolean& isReferencedLoaded) +298
System.Web.Compilation.BuildManager.ExecutePreAppStart() +137
System.Web.Hosting.HostingEnvironment.Initialize(ApplicationManager appManager, ApplicationHost appHost, IConfigurationFactory configPathFactory, HostingEnvironmentParameters hostingParameters, PolicyLevel policyLevel, Exception& exception) +160
[HttpException (0x80000005): Could not load file or assembly 'RasterEdge.Imaging.Annotation' or one of its dependencies. An attempt was made to load a program with an incorrect format.]
System.Web.HttpRuntime.FirstRequestInit(HttpContext context) +10893088
System.Web.HttpRuntime.InsureFirstRequest(HttpContext context) +95
System.Web.HttpRuntime.ProcessRequestNotificationPrivate(IISWorkerRequest wr, HttpContext context) +258

Version Information: Microsoft .NET Framework Version 4.0.30319; ASP.NET Version 4.7.3190.9
```

To fix it, you need choose right RasterEdge DLLs. We have provided DLLs for **.NET 4.0 (x64, x86)**, **.NET 3.5 (x64, x86)**. And our SDK does not support **anyCPU** platform.

## 12.2 Could not load file or assembly

### 'RasterEdge.Imaging.Font.DLL'

Three RasterEdge DLLs are using C++ to develop, and the rest of them are using C#.

The three DLLs are:

- RasterEdge.Imaging.Font.dll
- RasterEdge.XImage.OCR.Tesseract.dll
- RasterEdge.XImage.Raster.Core.dll

If your system has not installed C++ library, it will throw the following Exception **“Could not load file or assembly ‘RasterEdge.Imaging.Font.DLL’ or one of its dependencies”**

```
Server Error in '/' Application.

Could not load file or assembly 'RasterEdge.Imaging.Font.DLL' or one of its dependencies. The specified module could not be found.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.IO.FileNotFoundException: Could not load file or assembly 'RasterEdge.Imaging.Font.DLL' or one of its dependencies. The specified module could not be found.

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

[FileNotFoundException: Could not load file or assembly 'RasterEdge.Imaging.Font.DLL' or one of its dependencies. The specified module could not be found.]
System.Reflection.RuntimeAssembly._LoadAssemblyName fileName, String codebase, Evidence assemblySecurity, RuntimeAssembly locationHint, StackCrawlMark stackMark, IntPtr pPrivateData, Boolean throwOnFileNotFound, Boolean forIntrospection)
System.Reflection.RuntimeAssembly._LoadAssemblyName fileName, String codebase, Evidence assemblySecurity, RuntimeAssembly locationHint, StackCrawlMark stackMark, IntPtr pPrivateData, Boolean throwOnFileNotFound, Boolean forIntrospection)
System.Reflection.RuntimeAssembly.InternalLoadAssemblyName(AssemblyName assemblyName, Evidence assemblySecurity, RuntimeAssembly requestedAssembly, StackCrawlMark stackMark, IntPtr pPrivateData, Boolean throwOnFileNotFound, Boolean forIntrospection)
System.Reflection.RuntimeAssembly.InternalLoad(String assemblyString, Evidence assemblySecurity, StackCrawlMark stackMark, IntPtr pPrivateData, Boolean forIntrospection)
System.Reflection.Assembly.Load(String assemblyString)
System.Web.Configuration.CompilationSection.LoadAssemblyManager(String assemblyName, Boolean starDirective)
[ConfigurationErrorsException: Could not load file or assembly 'RasterEdge.Imaging.Font.DLL' or one of its dependencies. The specified module could not be found.]
System.Web.Configuration.CompilationSection.LoadAssemblyManager(String assemblyName, Boolean starDirective)
System.Web.Configuration.CompilationSection.LoadAllAssembliesFromAppDomainStaticDirectory()
System.Web.Configuration.CompilationSection.LoadAssembly(byte[] assemblyData)
System.Web.Compilation.BuildManager.GetReferencedAssemblies(CompilationSection compilationSection)
System.Web.Compilation.BuildManager.GetPreStartMethodsFromReferencedAssemblies()
System.Web.Compilation.BuildManager.CallPreStartMethods(String preStartInitPath, Boolean isReferencedAssembly)
System.Web.Compilation.BuildManager.ExecutePreStartMethod()
System.Web.Hosting.HostingEnvironment.Initialize(ApplicationManager appManager, ApplicationInstance appInstance, IConfigPathFactory configPathFactory, HostingEnvironmentParameters hostingParameters, PolicyLevel policyLevel, Exception exception)

[HttpException (0x80004005): Could not load file or assembly 'RasterEdge.Imaging.Font.DLL' or one of its dependencies. The specified module could not be found.]
System.Web.HttpRuntime.FirstRequestInit(HttpContext context)
System.Web.HttpRuntime.EnsureFirstRequestInit(HttpContext context)
System.Web.HttpRuntime.ProcessRequestNotificationPrivate(IISWorkerRequest wr, HttpContext context)
```

If you are using RasterEdge .net 2.0 dll, you must install Visual C++ 2005 Redistributable Package.  
If you are using RasterEdge .net 4.0 dll, you must install Visual C++ 2010 Redistributable Package.

Please download the Microsoft VC++ redistribution package:

#### Microsoft Visual C++ 2010 Redistributable Package (x86)

<https://www.microsoft.com/en-us/download/details.aspx?id=5555>

#### Microsoft Visual C++ 2010 Redistributable Package (x64)

<https://www.microsoft.com/en-us/download/details.aspx?id=14632>

## 13 Error logging & Report problems to us

We will output two types of error log files.

1. **Web application log files.** The web controller sdk will automatically output the log information in the web application (RasterEdge.WDP.dll, RasterEdge.XDoc.HTML5Viewer.dll).

The output log file is located under “/RasterEdge\_Cache/\_log\_files”.

You can turn off the log output, by change settings in **Web.Config**

```
<add key="reOutputLogData" value="false"/>
```

2. **RasterEdge SDK inner log files.** This log file will record our SDK inner information and errors.

The output log file is located under “/RasterEdge\_Cache/\_log\_files/SDKLogger”.

You can turn off the log output, by change settings in **Web.Config**

```
<add key="reSDKLoggerEnable" value="false"/>
```

If you have any tech problems, please send us all contents in folder /RasterEdge\_Cache/ to [support@rasteredge.com](mailto:support@rasteredge.com).