

EdgeDoc ASP.NET Document Viewer

Developer Guide

Table of Contents

EdgeDoc ASP.NET Document Viewer Developer Guide	1
1. Overview	2
2. Quick Start.....	3
2.1 Running Demo Projects in IIS	3
2.2 Try Online Demo.....	3
3. Integrate with Visual Studio Project Templates	4
3.1 ASP.NET Empty Web Application.....	4
3.2 ASP.NET MVC4 Razor Web Application	10
4. Embed EdgeDoc Viewer into Web Page.....	18
5. Document Input & Output.....	19
5.1 Load file into EdgeDoc.....	19
5.2 Save file from EdgeDoc into your file system or database	20
6. Customize EdgeDoc Viewer.....	23
6.1 Re-locate Viewer client side files.....	23
6.2 Re-locate Viewer web service files.....	23
6.3 UI: Viewer width and height settings	24
6.4 UI: Viewer startup form view	24
6.5 UI: Viewer HTML page title	25
6.6 Comments settings.....	25
6.7 Customize toolbar and commands.....	25
6.8 Multiple languages support	26
6.9 Cached file data management	27
7. Configure EdgeDoc through Web.config appSettings	28

1. Overview

Loading, viewing, annotating, redacting, saving and printing documents and images on web browser can be an easy task with the help of RasterEdge EdgeDoc 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.

RasterEdge EdgeDoc 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. Quick Start

2.1 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 ASP.NET Document Viewer demo project folder **“/DemoProjects/XDoc.HTML5 Viewer Demo”**

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 copy the content in the Web(for .net4.0 or higher).Config to Web.config.

If your IIS server .NET framework is 2.0, please use the 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 XDoc.HTML5Viewer 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

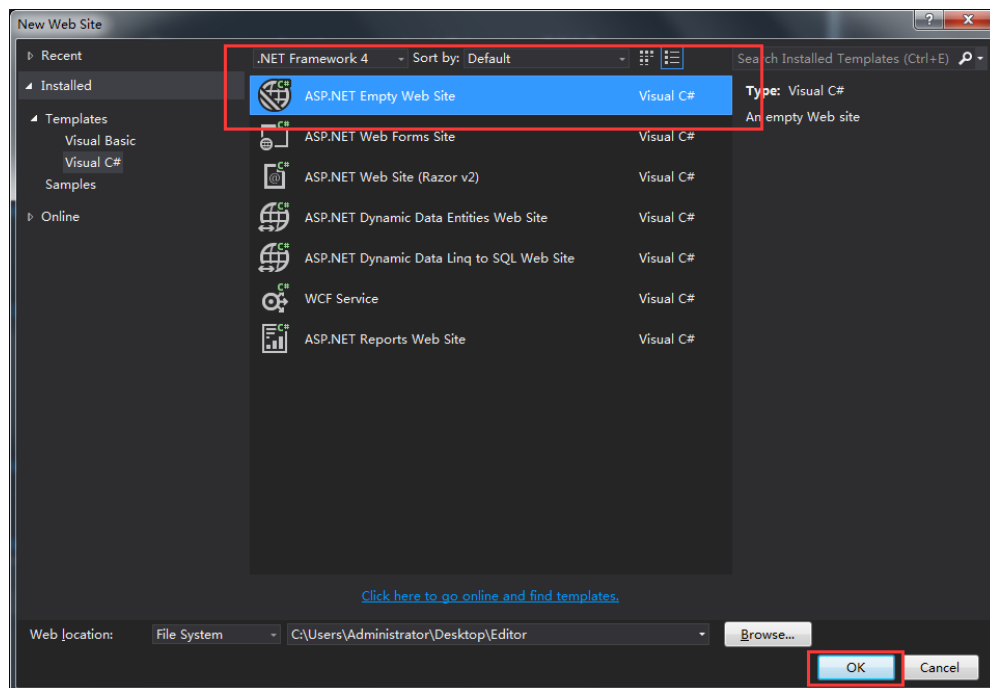
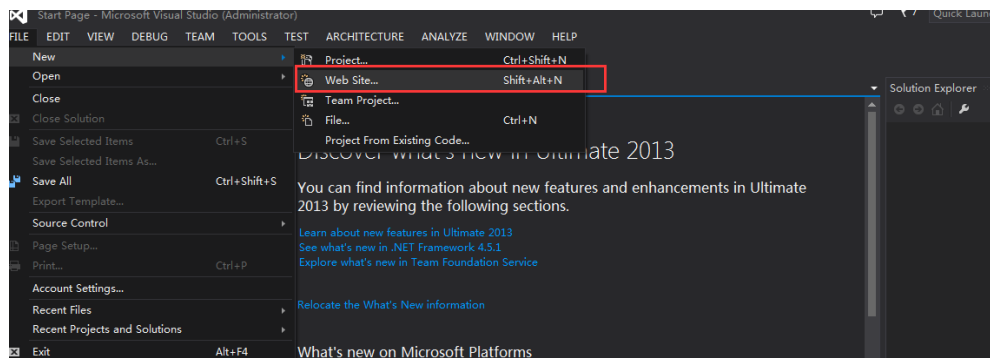
2.2 Try Online Demo

Please go to <http://www.rasteredge.com/dotnet-imaging/web-viewer-demo/> to view HTML5 Document Viewer online demo.

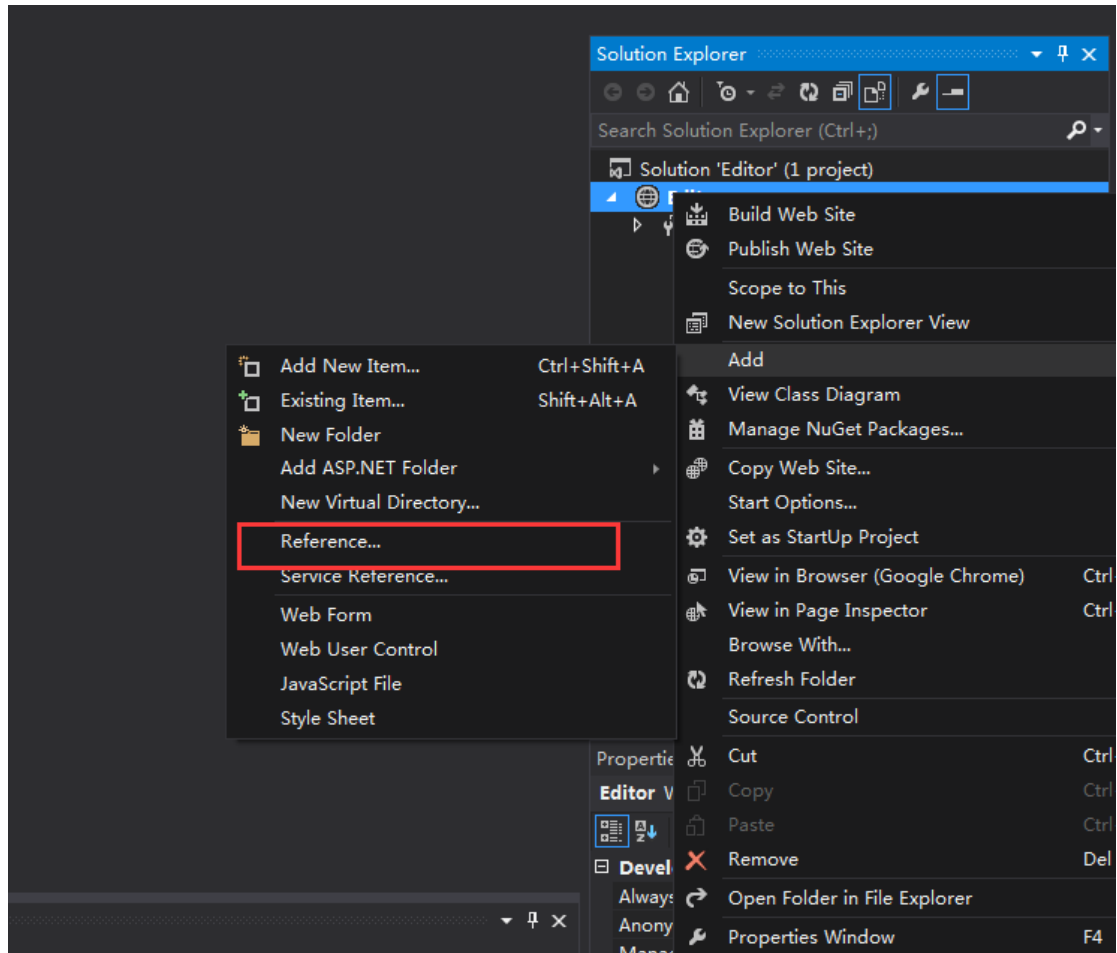
3. Integrate with Visual Studio Project Templates

3.1 ASP.NET Empty Web Application

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



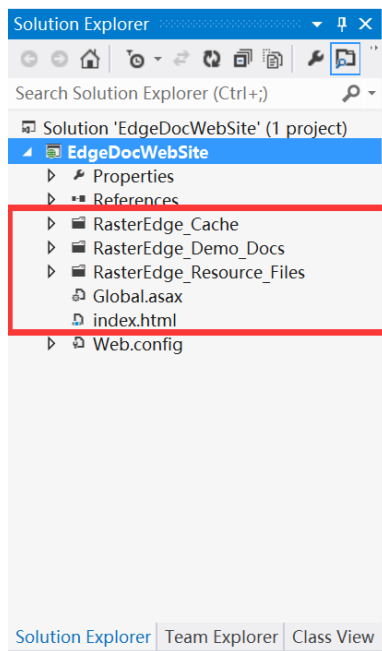
2. Add RasterEdge dll reference (add all dll names starting with RasterEdge)



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"/>
    <!-- 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="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="false"/>
    <!-- if reIntelliDrawShapesToImage is true, -->
    <!-- valid values are:      "all", "firefox", "chrome", "ie", "edge", "salari", "others"    if    "all"
included, ignore the rest;    default is "firefox, ie"    -->
    <add key="reIntelliDrawShapesToImageSupportBrowsers" value="firefox, ie"/>
    <!-- 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"/>
  </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>
    <requestFiltering allowDoubleEscaping="true">
      <requestLimits maxAllowedContentLength="41943040"/>
    </requestFiltering>
  </security>
  <staticContent>
    <remove fileExtension=".js"/>
    <remove fileExtension=".svg"/>
    <remove fileExtension=".woff"/>
    <remove fileExtension=".ttc"/>
    <remove fileExtension=".xpdf"/>
    <remove fileExtension=".fdf"/>
    <mimeMap fileExtension=".xpdf" mimeType="application/octet-stream"/>
    <mimeMap fileExtension=".fdf" mimeType="application/octet-stream"/>
    <mimeMap fileExtension=".woff" mimeType="application/octet-stream"/>
    <mimeMap fileExtension=".svg" mimeType="image/svg+xml"/>
    <mimeMap fileExtension=".ttc" mimeType="application/x-font-ttc"/>
    <mimeMap fileExtension=".js" mimeType="application/javascript"/>
  </staticContent>
  <urlCompression doStaticCompression="true" doDynamicCompression="true"/>
</system.webServer>
</configuration>
```


5. Change path

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

```
<script src="/EdgeDocWebSite/RasterEdge_Resource_Files/javascript/jquery.js"
type="text/javascript"></script>
<script
src="/EdgeDocWebSite/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp_Customi
ze.js" type="text/javascript"></script>
<script
src="/EdgeDocWebSite/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp.js"
type="text/javascript"></script>
<script src="/EdgeDocWebSite/RasterEdge_Resource_Files/javascript/RasterEdge.js"
type="text/javascript"></script>
<link rel="stylesheet"
href="/EdgeDocWebSite/RasterEdge_Resource_Files/css/process.css" type="text/css"/>
<link rel="stylesheet"
href="/EdgeDocWebSite/RasterEdge_Resource_Files/css/customize.css" type="text/css"/>
<link rel="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:
"/EdgeDocWebSite/RasterEdge_Resource_Files/UserCommandProcessHandler.ashx"
});
```

In file “Web.config”

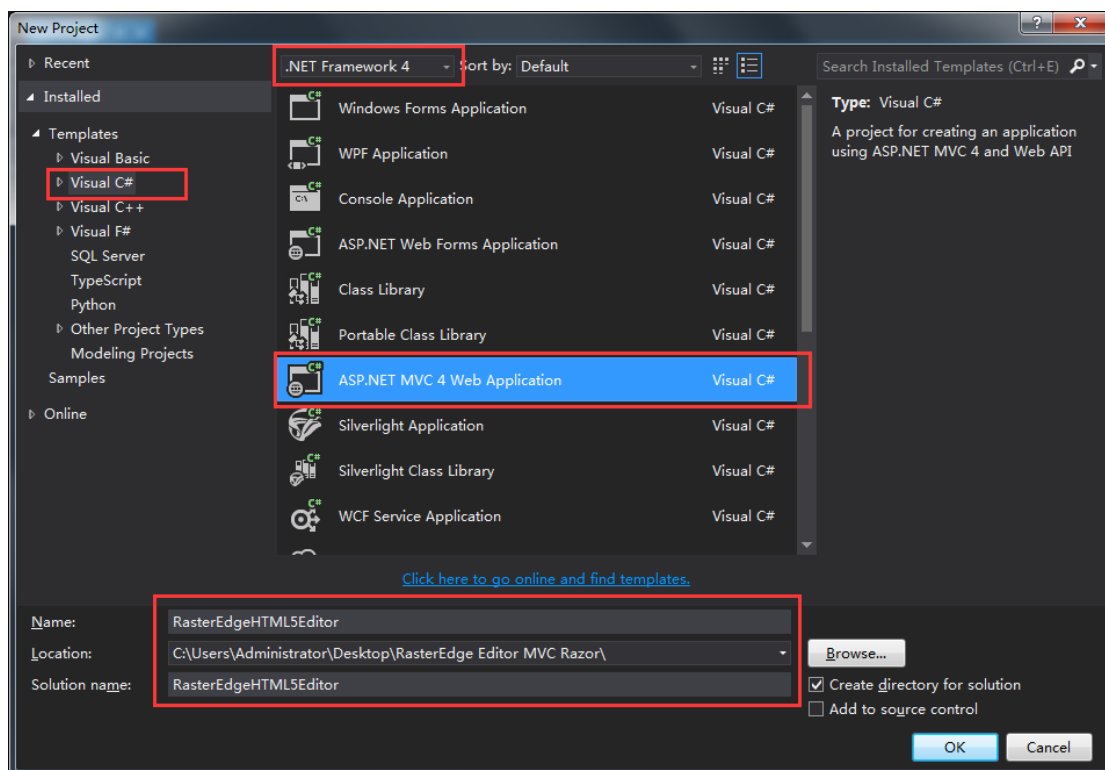
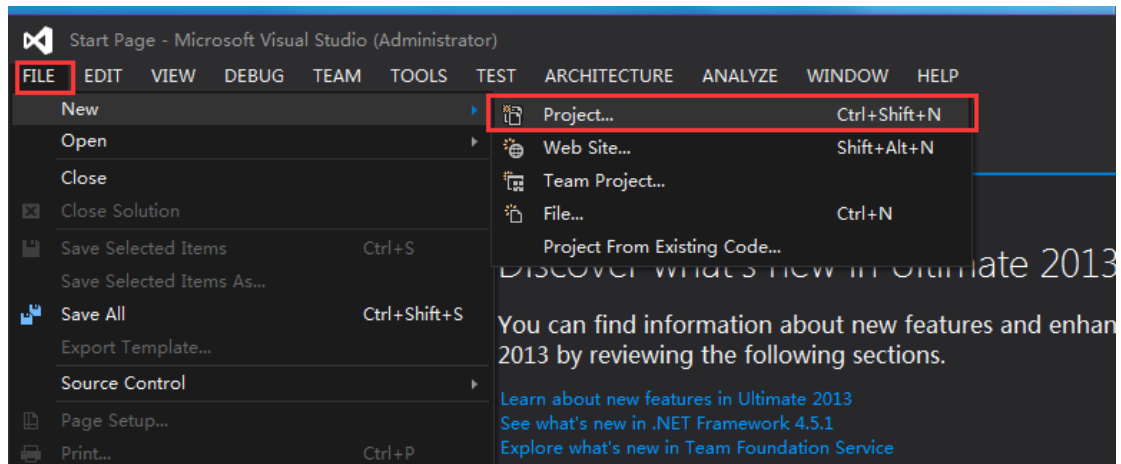
```
<!-- the cache folder -->
<add key="reCacheFolder" value="EdgeDocWebSite/RasterEdge_Cache"/>
```

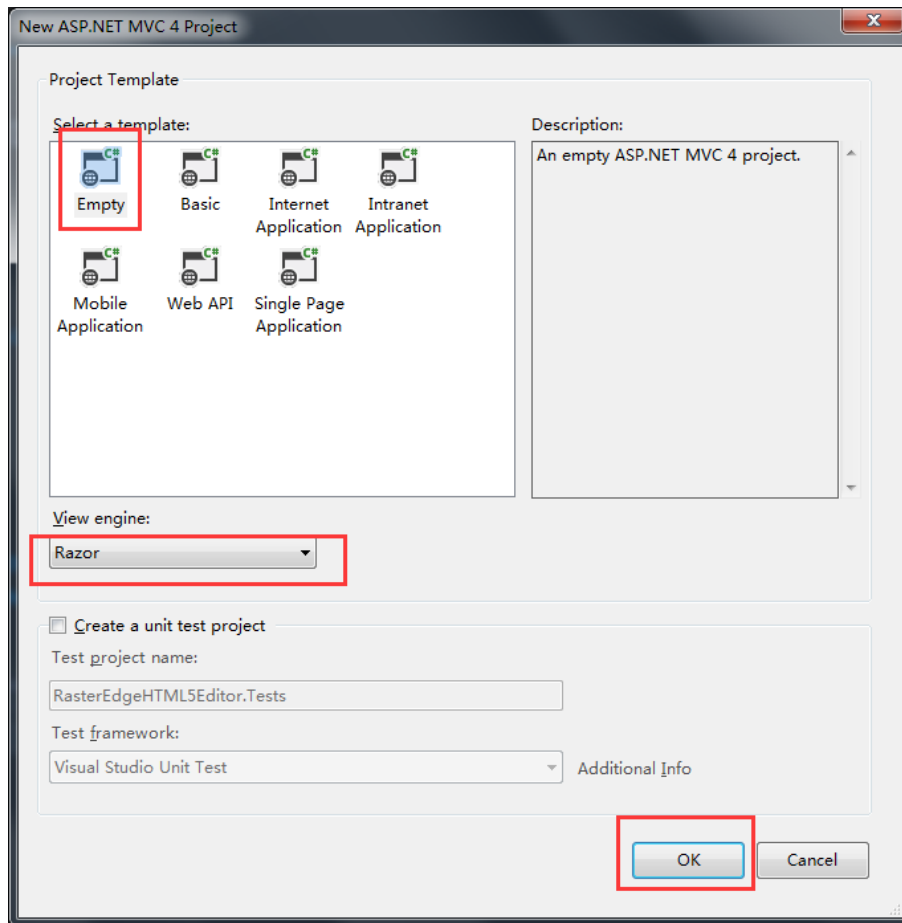
Note: reCacheFolder must use with relative path.

6. It's done. Now run the website.

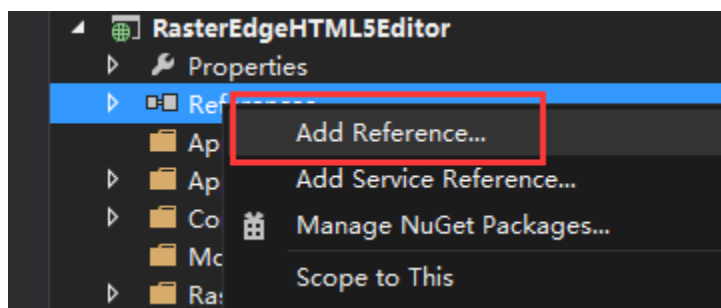
3.2 ASP.NET MVC4 Razor Web Application

1. Create MVC Project





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

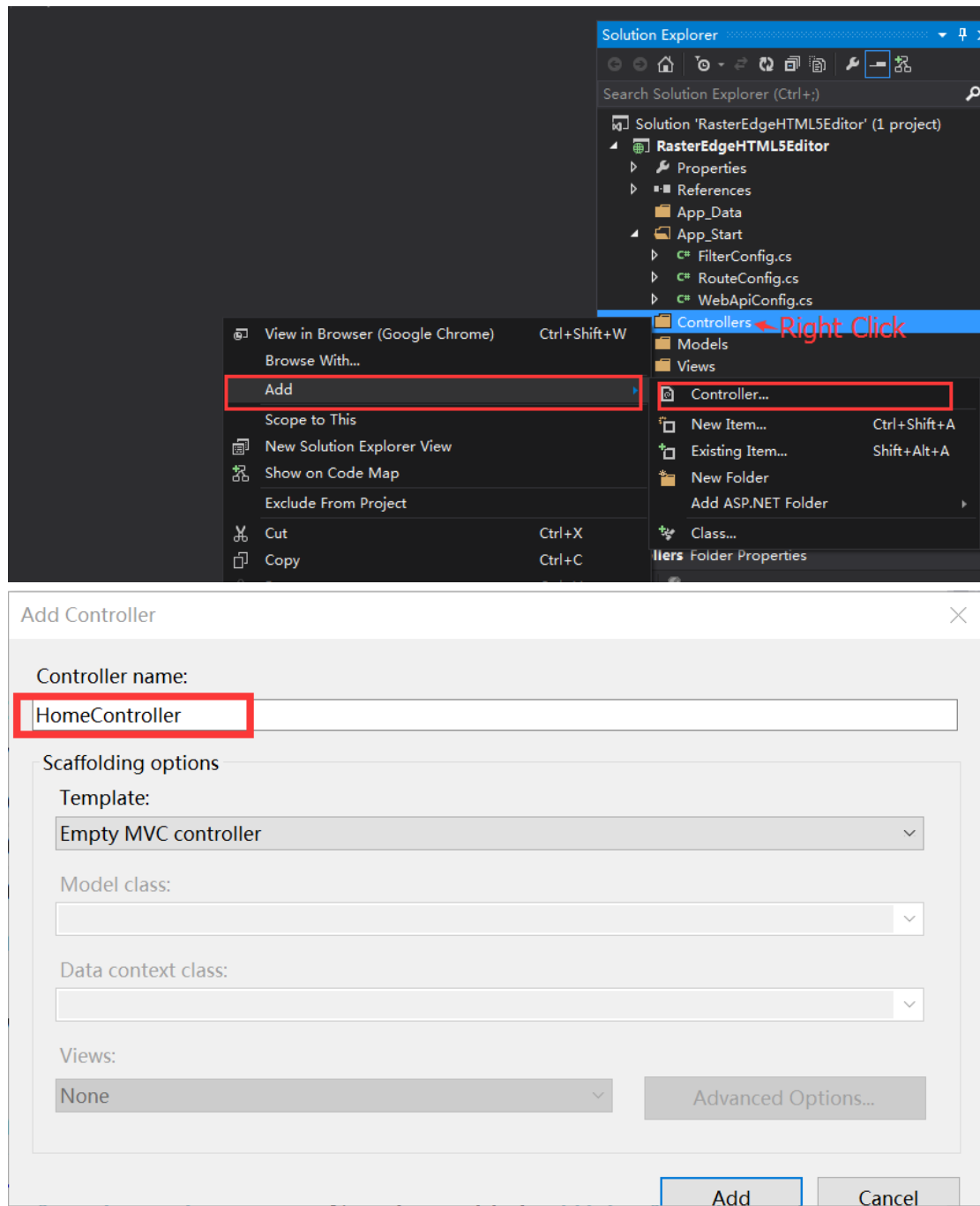


Please go to download package “/Bin”, choose right .net version. Add all **dlls** with starting name “**RasterEdge.**” to your project reference.

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

Please choose x86 dlls from Bin folder.

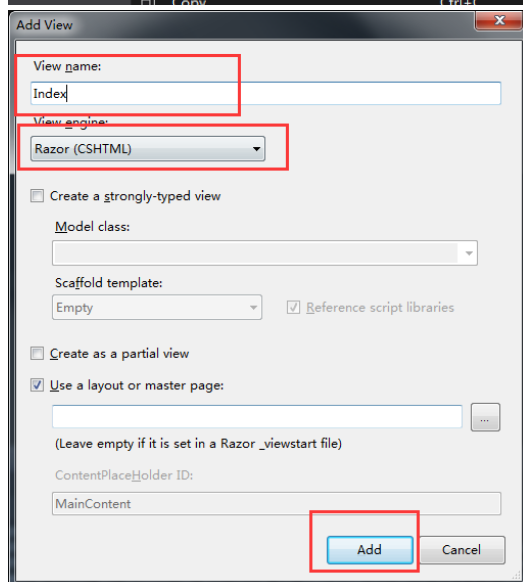
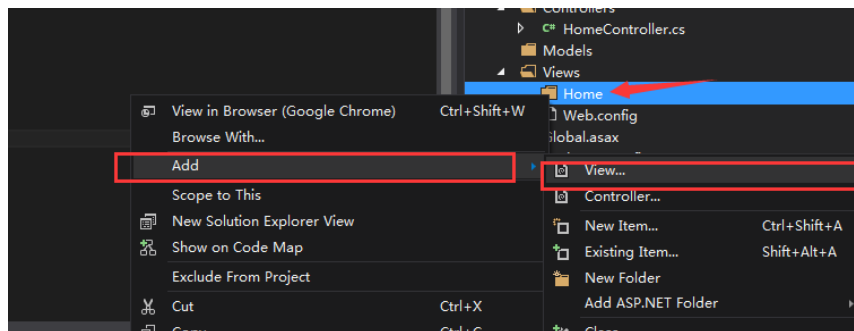
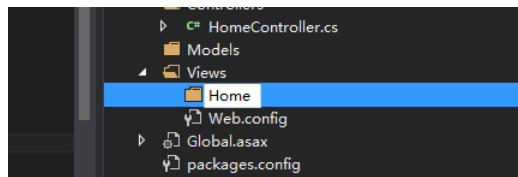
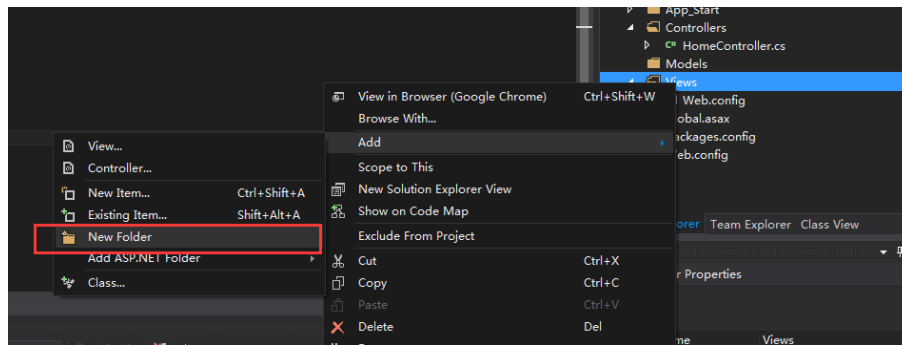
3. Create controller



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

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

4. Create View

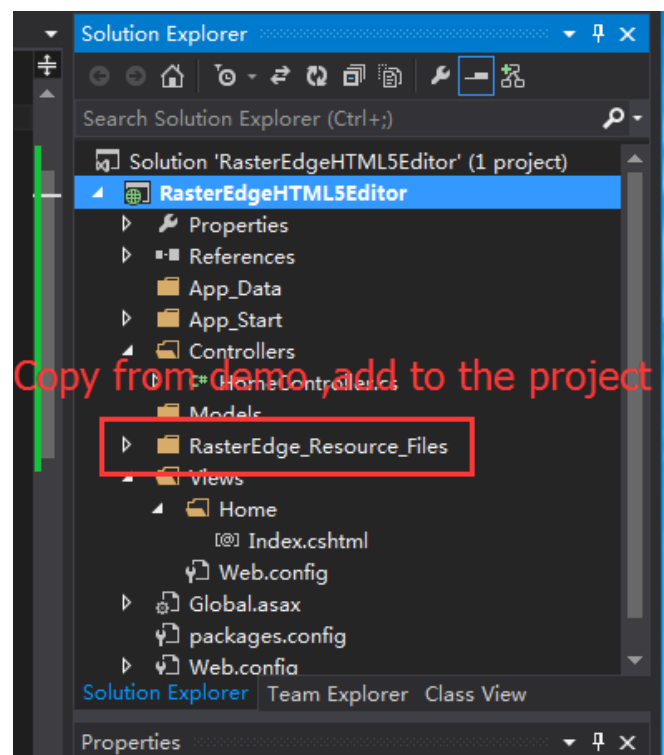


Add the html to **Index.cshtml**.

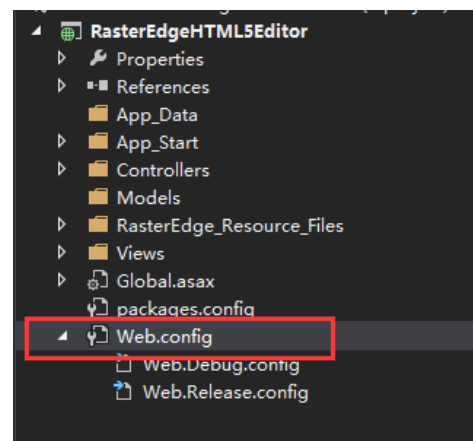
Copy the contents in file “/Developer Guide/EdgeDoc ASP.NET Document Viewer Developer Guide/MVC4 Razor project files/**Index.cshtml**” to your Index html file.

5. Add the folder “/DemoProjects/XDoc.HTML5 Viewer

Demo/RasterEdge_Resource_Files” to project.



6. Set the Web.config



```
15 <add key="reProjectRootPath" value="/" />
16 <add key="reCacheFolder" value="/RasterEdge_Cache" />
17 <add key="reResourceFolder" value="RasterEdge_Resource_Files" />
18 <!-- set the server folder if you want to open the file on server-->
19 <add key="reDefaultFolder" value="E:\\tempFile\\" />
20 <!-- set the default file when open the web first time,if set "" ,this function is disabled. (defaultFo
21 <add key="reDefaultFilePath" value="" />
22 <!-- image quality in PDF document-->
23 <add key="reImageZoom" value="2" />
24 <add key="reAnnotationImageZoom" value="2" />
25 <add key="reOutputLogData" value="true" />
26 <!-- if true, SDK will automatically convert many shape and path inside PDF page into image-->
27 <!-- default is false -->
28 <add key="reIntelliDrawShapesToImage" value="false" />
29 <!-- if reIntelliDrawShapesToImage is true, -->
30 <!-- valid values are: "all", "firefox", "chrome", "ie", "edge", "safari", "others" if "all" included
31 <add key="reIntelliDrawShapesToImageSupportBrowsers" value="firefox, ie" />
32 </appSettings>
```

AppSettings:

```
<!-- the cache folder -->
<add key="reCacheFolder" value="RasterEdge_Cache"/>
<!-- 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="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="false"/>
<!-- if reIntelliDrawShapesToImage is true, -->
<!-- valid values are:      "all", "firefox", "chrome", "ie", "edge", "salari", "others"    if    "all"
included, ignore the rest;    default is "firefox, ie"    -->
<add key="reIntelliDrawShapesToImageSupportBrowsers" value="firefox, ie"/>
<!-- 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 name="extensionlessurlhandler-integrated-4.0" path="~" verb="GET,HEAD,POST,DEBUG,PUI,DELETE,PATCH,
  </handlers>
  <security>
    <requestFiltering allowDoubleEscaping="true">
      <requestLimits maxAllowedContentLength="41943040"/>
    </requestFiltering>
  </security>
  <staticContent>
    <remove fileExtension=".js"/>
    <remove fileExtension=".svg"/>
    <remove fileExtension=".woff"/>
    <remove fileExtension=".ttc"/>
    <remove fileExtension=".xpdf"/>
    <remove fileExtension=".fdf"/>
    <mimeTypeMap fileExtension=".xpdf" mimeType="application/octet-stream"/>
    <mimeTypeMap fileExtension=".fdf" mimeType="application/octet-stream"/>
    <mimeTypeMap fileExtension=".woff" mimeType="application/octet-stream"/>
    <mimeTypeMap fileExtension=".svg" mimeType="image/svg+xml"/>
    <mimeTypeMap fileExtension=".ttc" mimeType="application/x-font-ttc"/>
    <mimeTypeMap fileExtension=".js" mimeType="application/x-javascript"/>
  </staticContent>
  <urlCompression doStaticCompression="true" doDynamicCompression="true"/>
</system.webServer>

```

system.webServer

```

<security>
  <requestFiltering allowDoubleEscaping="true">
    <requestLimits maxAllowedContentLength="41943040"/>
  </requestFiltering>
</security>
<staticContent>
  <remove fileExtension=".js"/>
  <remove fileExtension=".svg"/>
  <remove fileExtension=".woff"/>
  <remove fileExtension=".ttc"/>
  <remove fileExtension=".xpdf"/>
  <remove fileExtension=".fdf"/>
  <mimeTypeMap fileExtension=".xpdf" mimeType="application/octet-stream"/>
  <mimeTypeMap fileExtension=".fdf" mimeType="application/octet-stream"/>
  <mimeTypeMap fileExtension=".woff" mimeType="application/octet-stream"/>
  <mimeTypeMap fileExtension=".svg" mimeType="image/svg+xml"/>
  <mimeTypeMap fileExtension=".ttc" mimeType="application/x-font-ttc"/>
  <mimeTypeMap fileExtension=".js" mimeType="application/x-javascript"/>
</staticContent>
<urlCompression doStaticCompression="true" doDynamicCompression="true"/>

```

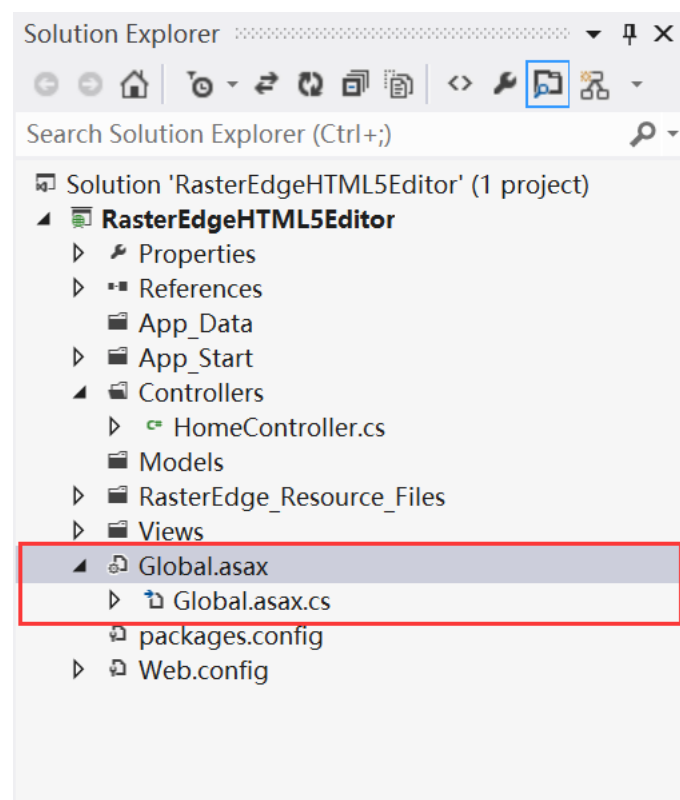

system.web

```
32 <system.web>
33 <compilation debug="true" targetFramework="4.0" />
34 <pages>
35 <namespaces>
36 <add namespace="System.Web.Helpers" />
37 <add namespace="System.Web.Mvc" />
38 <add namespace="System.Web.Mvc.Ajax" />
39 <add namespace="System.Web.Mvc.Html" />
40 <add namespace="System.Web.Routing" />
41 <add namespace="System.Web.WebPages" />
42 </namespaces>
43 </pages>
44 <httpRuntime requestValidationMode="2.0" executionTimeout="1200" maxRequestLength="40960" useFullyQuali
45 </system.web>
```

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

7. Set the cache folder auto-clear rules

Copy or add the Global.asax files to your project.



8. It is done. Now run the project.

4 Embed EdgeDoc Viewer into Web Page

In demo project, we have included two html pages to explain how to embed EdgeDoc into your existing web pages (html or aspx).

edgedoc-page-frame.html

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

edgedoc-page-popup.html

It will popup the EdgeDoc viewer inside html page.

5 Document Input & Output

5.1 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"

With EdgeDoc 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 **yourtarget** to other names, like fileid.

5.2 Save file from EdgeDoc into your 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 = "RasterEdge PDF HTML5 Viewer";
        pdfDoc.SetDescription(metadata);

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

6. Customize EdgeDoc Viewer

6.1 Re-locate Viewer client side files

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

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

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

```
<script src="/js-path/RasterEdge_Resource_Files/javascript/jquery.js" type="text/javascript"></script>  
    <script src="/js-path/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp_Customize.js"  
type="text/javascript"></script>  
    <script src="/js-path/RasterEdge_Resource_Files/javascript/RasterEdge_WebApp.js" type="text/javascript"></script>  
    <script src="/js-path/RasterEdge_Resource_Files/javascript/RasterEdge.js" type="text/javascript"></script>  
    <link rel="stylesheet" href="/css-path/RasterEdge_Resource_Files/css/process.css" type="text/css"/>  
    <link rel="stylesheet" href="/css-path/RasterEdge_Resource_Files/css/customize.css" type="text/css"/>  
    <link rel="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”**.

6.2 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.

6.3 UI: 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:

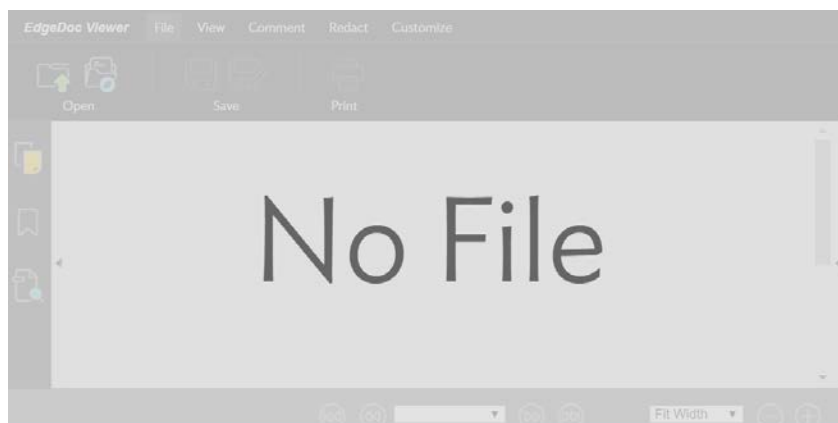
```
<div style="width: 90%; height: 600px;">  
    <iframe name="edgedoc" src="http://localhost" style="width: 100%; height:  
90%"></iframe>  
</div>
```

6.4 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"
```



6.5 UI: 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);
```

6.6 Comments 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.

6.7 Customize 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.

6.8 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>

6.9 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);
}
```

7.Configure EdgeDoc through Web.config

appSettings

reCacheFolder	<p>Specify the location of cache folder. Please use the relative path 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)
reImageZoom	Image quality in PDF document
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	<p>Set the type of the show page.</p> <p>Valid values: svg, png</p> <p>Default: svg</p>