

# EdgePDF Developer Guide

## Table of Contents

EdgePDF 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.....	5
4. Integrate with Visual Studio Project Templates .....	6
4.1 ASP.NET Empty Web Site .....	6
4.2 ASP.NET MVC4 Razor Web Application .....	12
4.3 ASP.NET MVC5 Web Application .....	19
5. Embed EdgePDF into HTML/ASPX Web Page .....	27
6. Document Connect with EdgePDF .....	30
6.1 Load file into EdgePDF.....	30
6.2 Save file from EdgePDF into your server file system .....	31
7. Configure EdgePDF through Web.config appSettings .....	35
8. Enable “Content Edit” functions .....	37
9. PDF Font Management .....	38
9.1 Install Adobe fonts to system .....	38
9.2 Install Adobe fonts in Runtime .....	39
10. Customize EdgePDF.....	40
10.1 Re-locate Web Editor client side files .....	40
10.2 Re-locate Web Editor web service files .....	40
10.3 UI: Web Editor width and height settings .....	41
10.4 UI: Web Editor startup form view .....	41
10.5 UI: Web Editor HTML page title.....	42
10.6 Comments settings.....	43
10.7 Define your own toolbar and commands.....	43
11. Customize, extend Annotation, Comment, Markup, Redaction .....	44
11.1 Set annotation default properties cookie live time.....	44
11.2 Enable continuous annotations.....	44
11.3 Enable continuous text redact & area redact.....	44
11.4 Create a new stamp annotation .....	45
11.5 Get user mouse click coordinates .....	46
12. Extend EdgePDF Examples .....	47
12.1 Example 1: Add watermark to all documents displayed on EdgePDF .....	47

12.2	Example 2: Apply pre text search before documents displayed on EdgePDF .....	48
13.	Common Exceptions and Errors .....	49
13.1	Could not load file or assembly 'RasterEdge.Imaging.Annotation' .....	49
13.2	Could not load file or assembly 'RasterEdge.Imaging.Font.DLL' .....	50
14.	Error logging & Report problems to us .....	51

# 1.Overview

RasterEdgeEdgePDF ASP.NET PDF Editor is a professional online PDF manipulation tools, which is compatible with both 32-bit and 64-bit operating systems and .NET Framework 3.0 & above versions.

It's easy to deploy EdgePDFASP.NET PDF Editor on various platforms such as Visual Studio .NET project and IIS (Internet Information Services), ASP.NET MVC application, Azure Cloud Service ( Azure), SharePoint site, Dotnetnuke ( DNN) project.

With EdgePDF ASP.NET PDF Editor, developers can perform a lot of utility PDF editing functionalities in ASP.NET program. Users are able to edit PDF text in ASP.NET, edit images on PDF document, edit PDF document page in ASPX webpage, set and edit PDF password, edit PDF file permissions and redact PDF text, image and page content to protect important information.

## 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 EdgePDFASP.NETPDF Editor demo project folder  
**“/DemoProjects/EdgePDF 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 or earlier version, 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 XDoc.PDF.HTML5 Editor 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 EdgePDF will generate cache files.
5. Run the web application

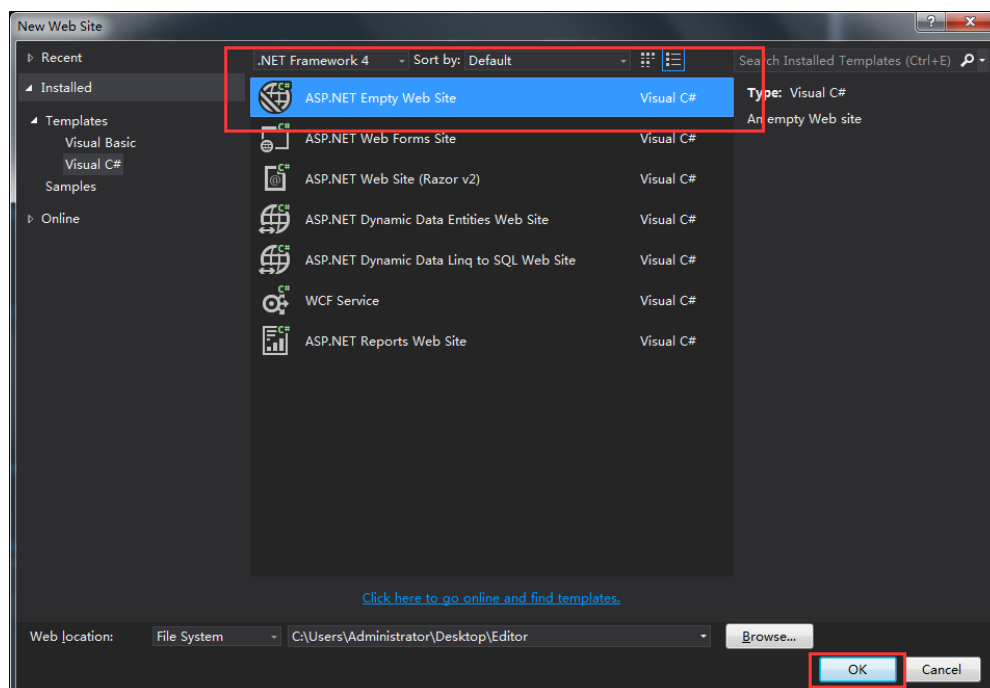
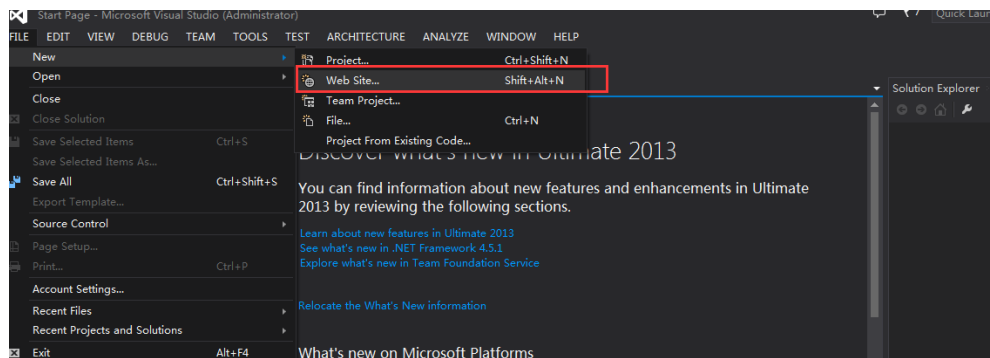
## 3.2 Try Online Demo

Please go to <http://www.rasteredge.com/online/pdf-editor/> to view EdgePDFASP.NETPDF Editor online demo.

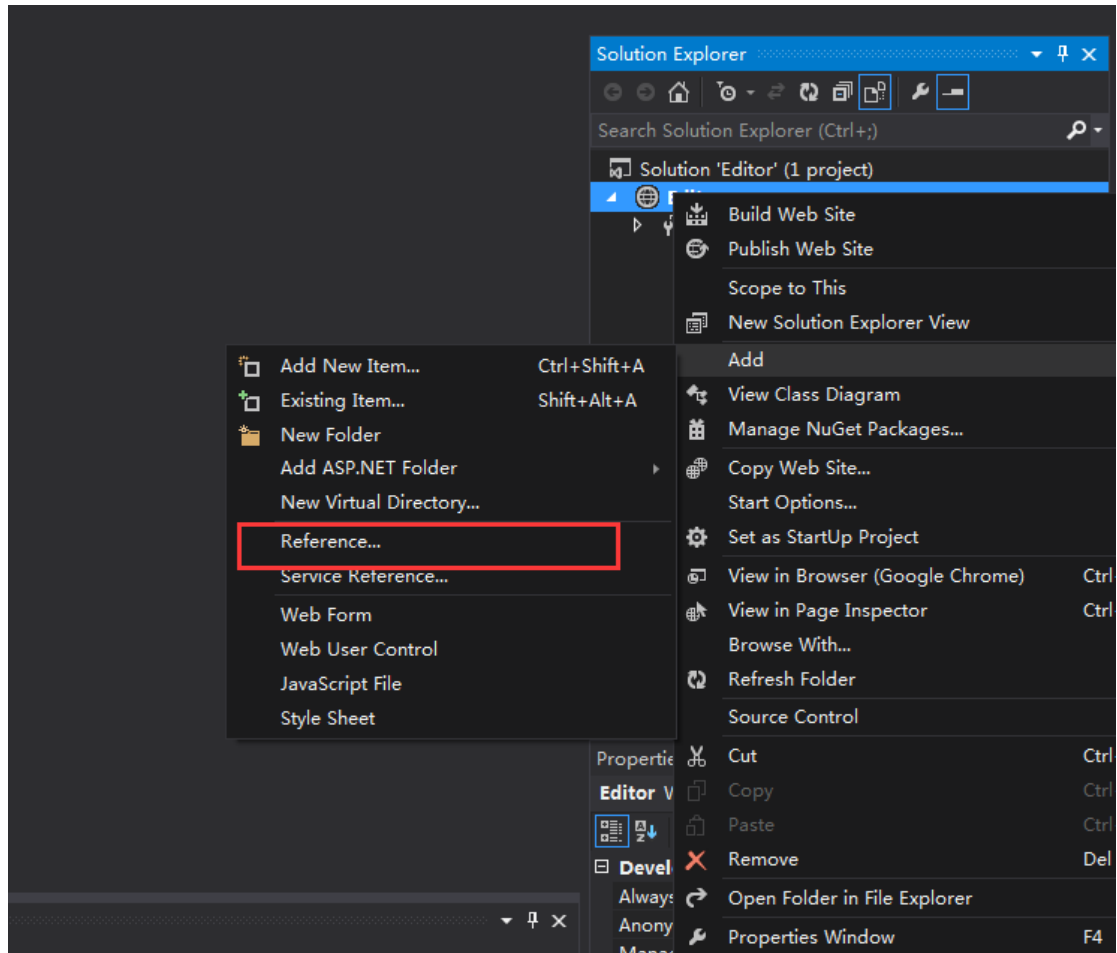
## 4. Integrate with Visual Studio Project Templates

### 4.1 ASP.NET Empty Web Site

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



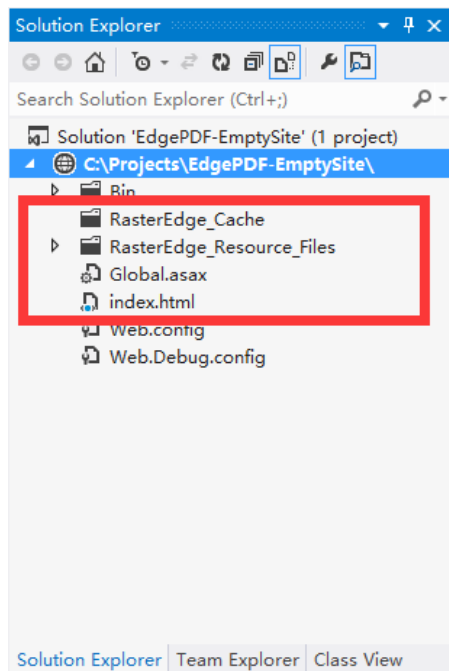
2. Add RasterEdge.dll reference (add all dll names starting with RasterEdge)



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 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,html
Default: html
-->
<add key="reDocRenderEngine" value="html"/>
<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 “index.html”, update JS, CSS file path

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

And update index.html,UserCommandProcessHandler.ashx path

```
var _rootpath = "/EdgePDFWebSite/";

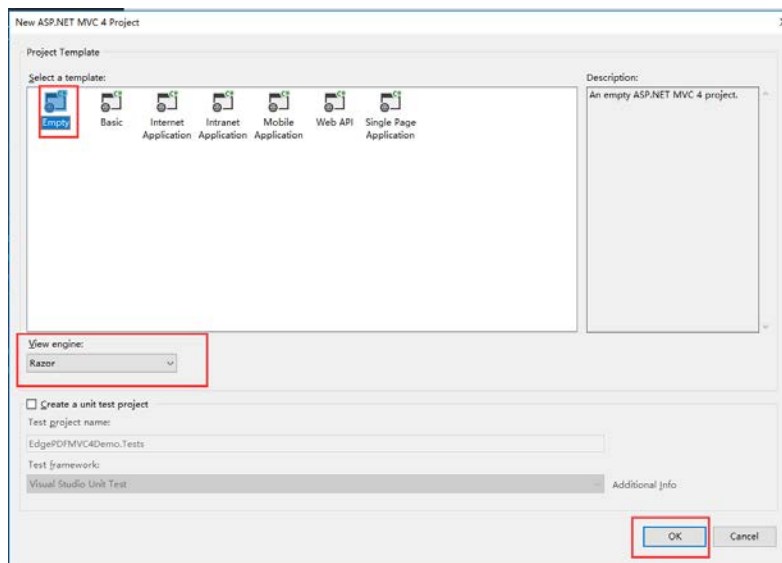
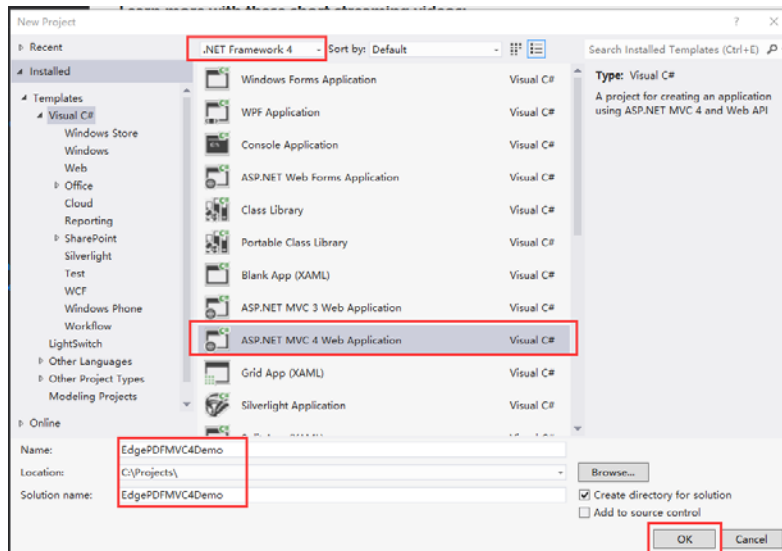
_WDPApp = new WDPOnlineApplication({
    _serverUrl: "RasterEdge_Resource_Files/UserCommandProcessHandler.ashx"
});
```

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

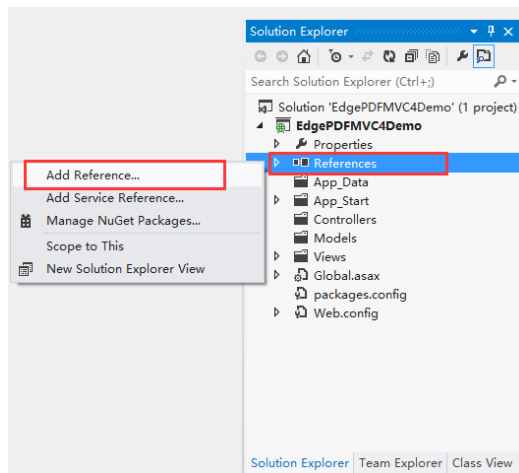
5. It’s done. Now run the website.

## 4.2 ASP.NET MVC4 Razor Web Application

### 1. Create MVC Project “EdgePDFMVC4Demo”



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

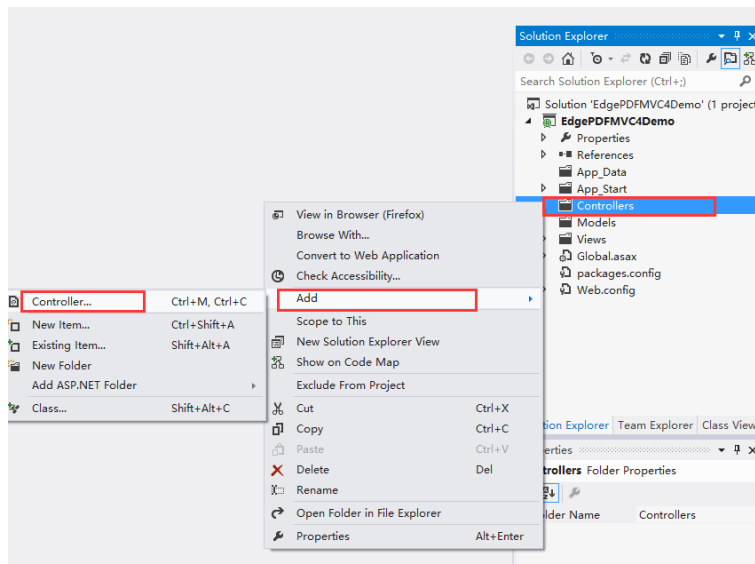


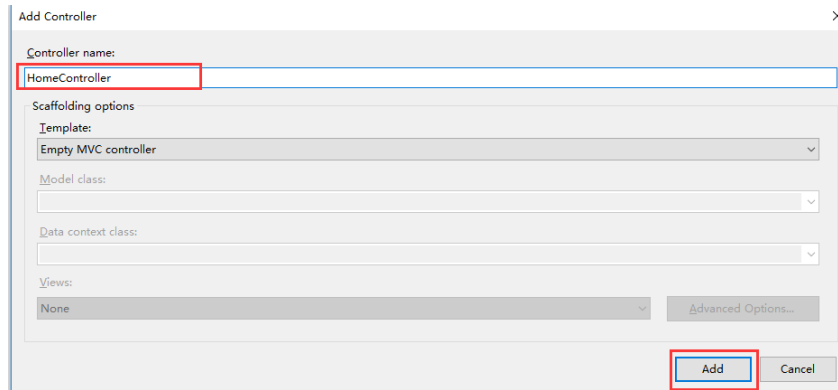
Please go to download package “/Bin”, choose right .net version. Add all **dlls** with starting name “**RasterEdge.**” to your project reference.

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

Please choose x86 dlls from Bin folder.

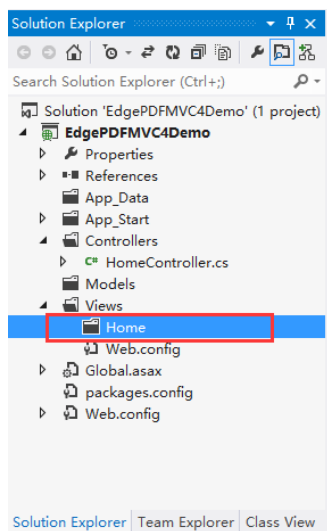
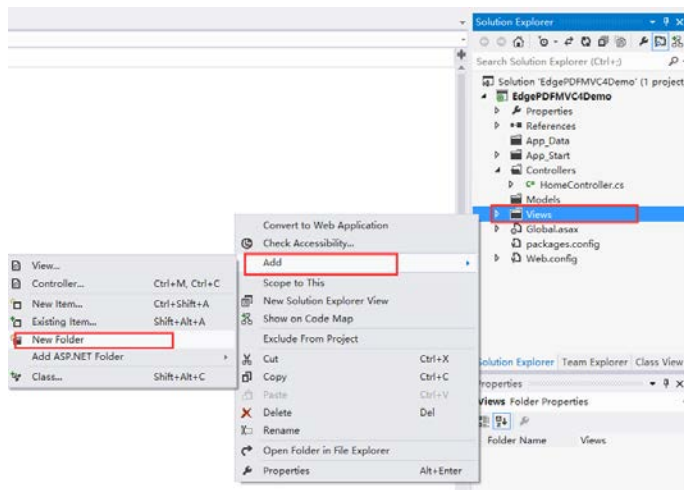
3. Create controller “**HomeController**”



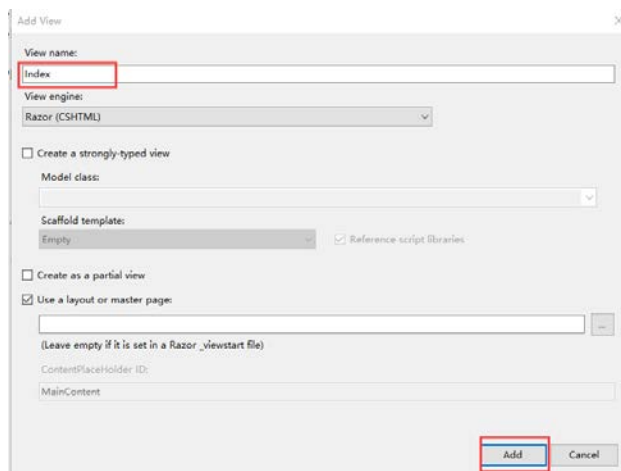
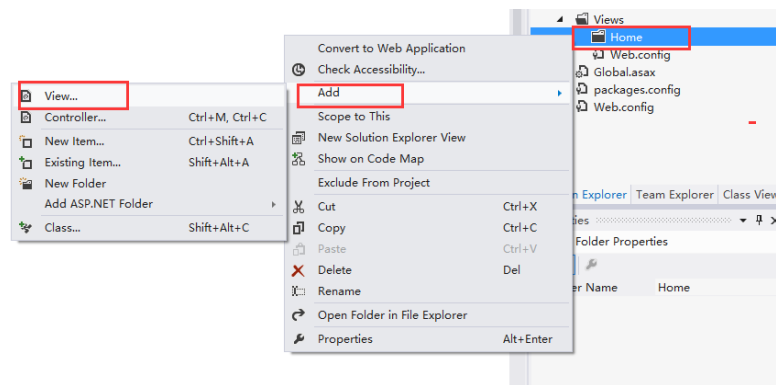


## 4. Create View

Create a new folder “**Home**” under **Views**



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

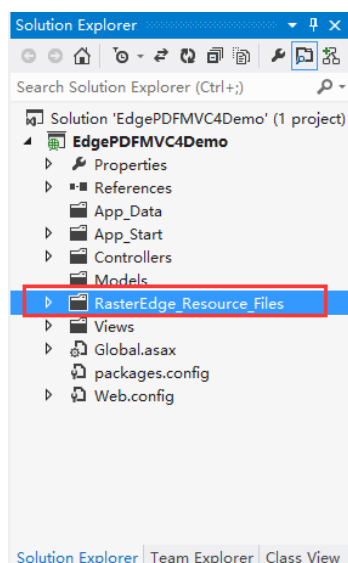


Add the html to Index.cshtml .

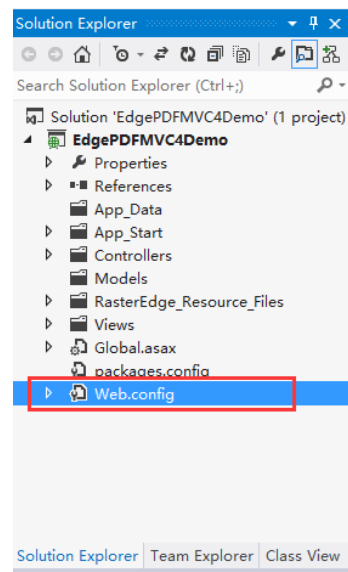
Copy the contents in file “/Developer Guide/EdgePDF/MVC4 Razor project files/**Index.cshtml**” to your Index html file.

5. Add the folder “/DemoProjects/EdgePDF 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,html
      Default: html
-->
<add key="reDocRenderEngine" value="html"/>
<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>**

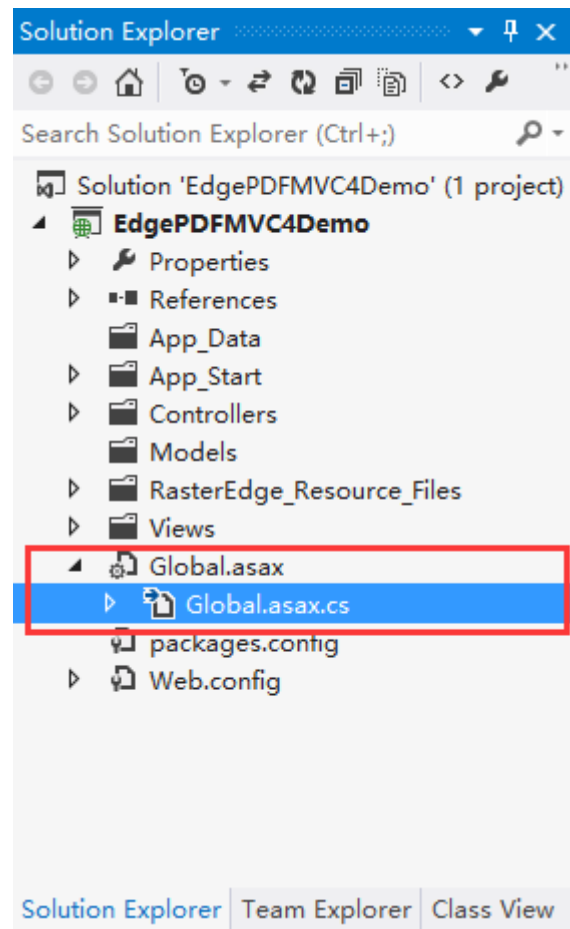
```
<security>
  <requestFilteringallowDoubleEscaping="true">
  <requestLimitsmaxAllowedContentLength="41943040"/>
</requestFiltering>
</security>
<staticContent>
  <remove fileExtension=".js"/>
  <remove fileExtension=".svg"/>
  <remove fileExtension=".woff"/>
  <remove fileExtension=".ttc"/>
  <remove 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>**

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

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

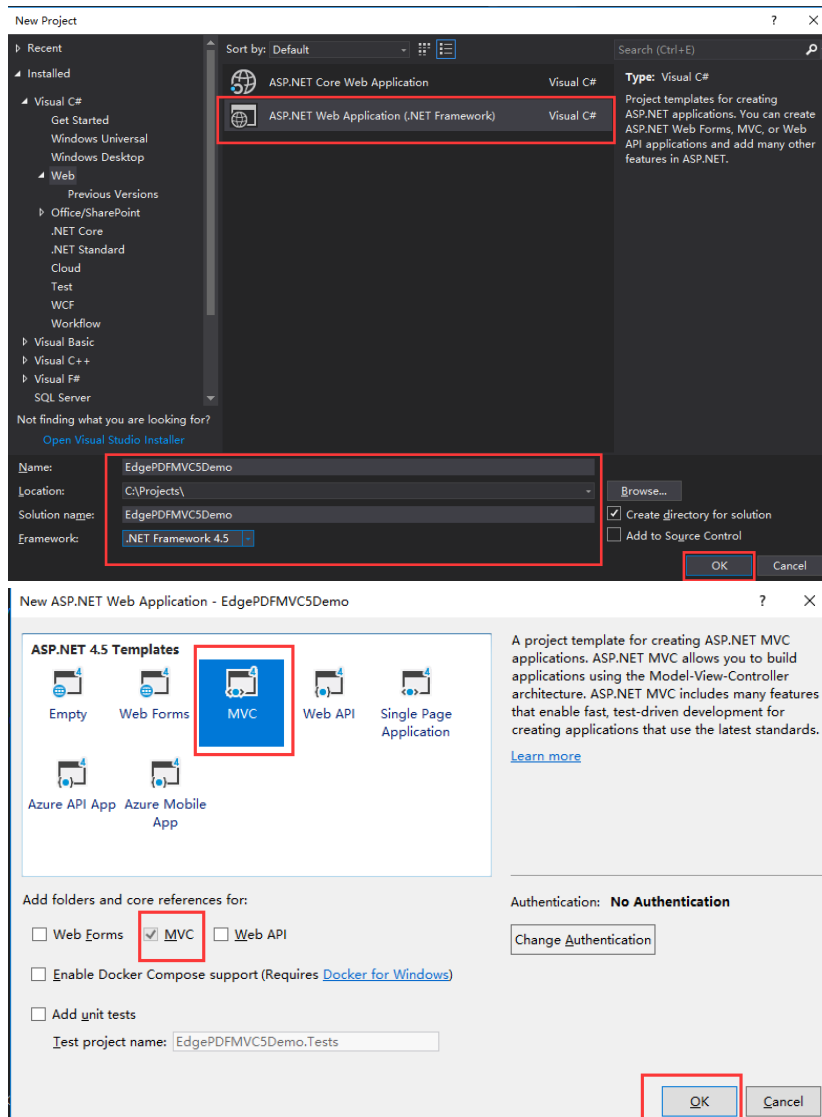
Copy or add the “[/Developer Guide/EdgePDF/MVC4 Razor project files/Global.asax.cs](#)” file to your project.



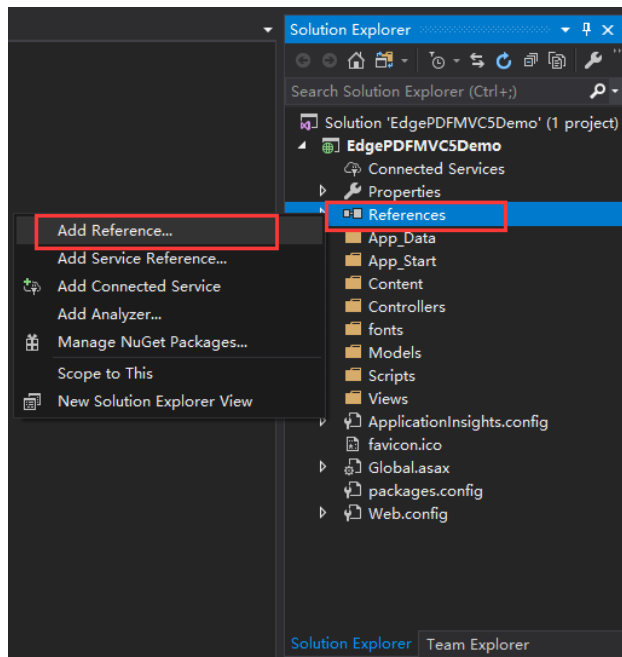
8. It is done. Now run the project.

## 4.3 ASP.NET MVC5 Web Application

### 1. Create a new MVC5 Project “EdgePDFMVC5Demo”



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

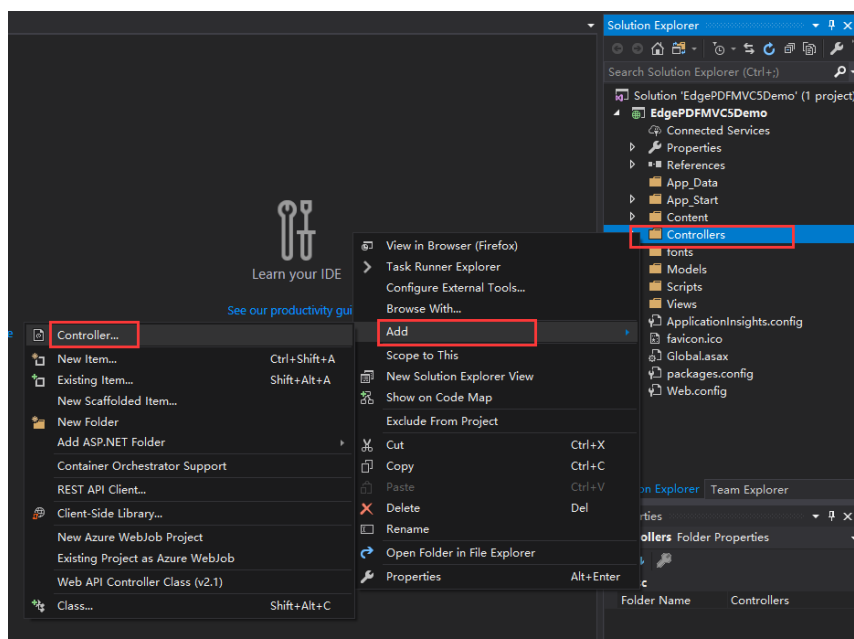


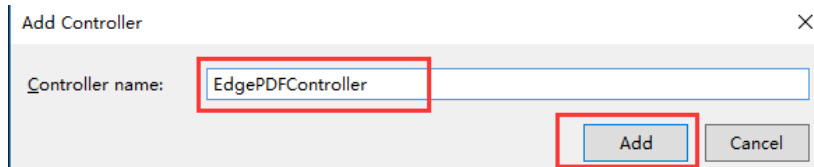
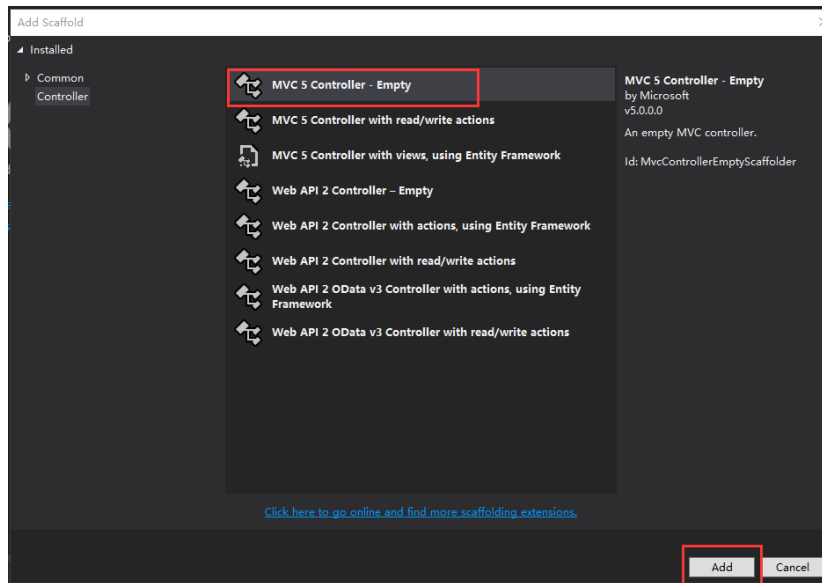
Please go to download package “/Bin”, choose **.net 4.0, x86 dlls**. Add **all dlls** with starting name “**RasterEdge.**” to your project reference.

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

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

3. Create controller “**EdgePDFController**”

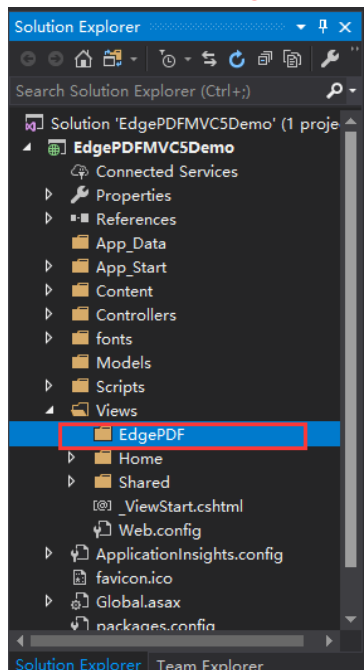




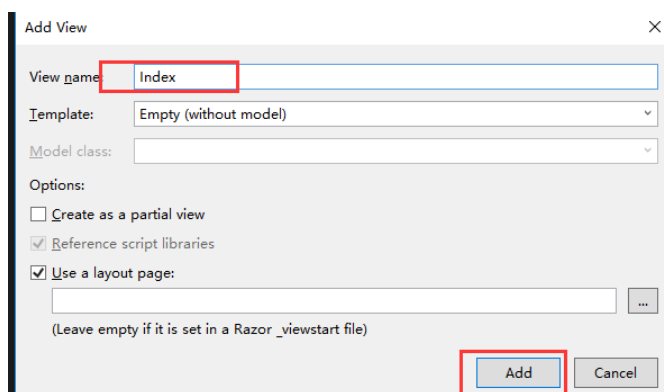
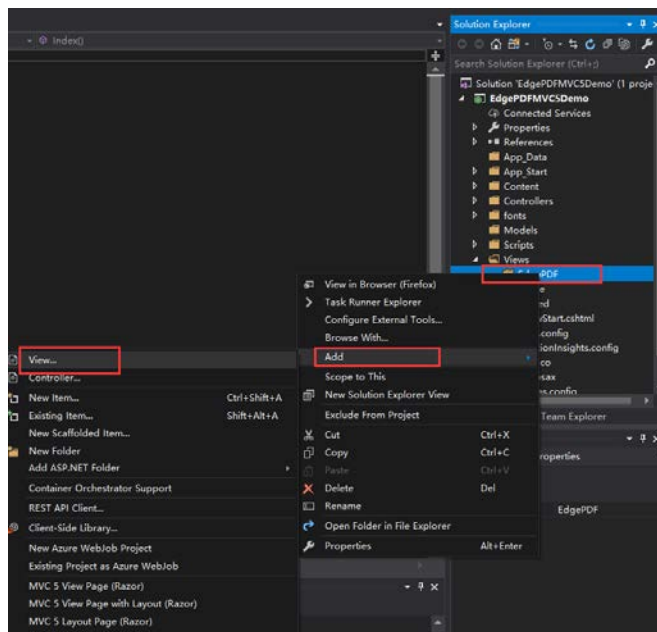
Click **“Add”**, to add a new controller **“EdgePDFController”** to your project.

#### 4. Create View

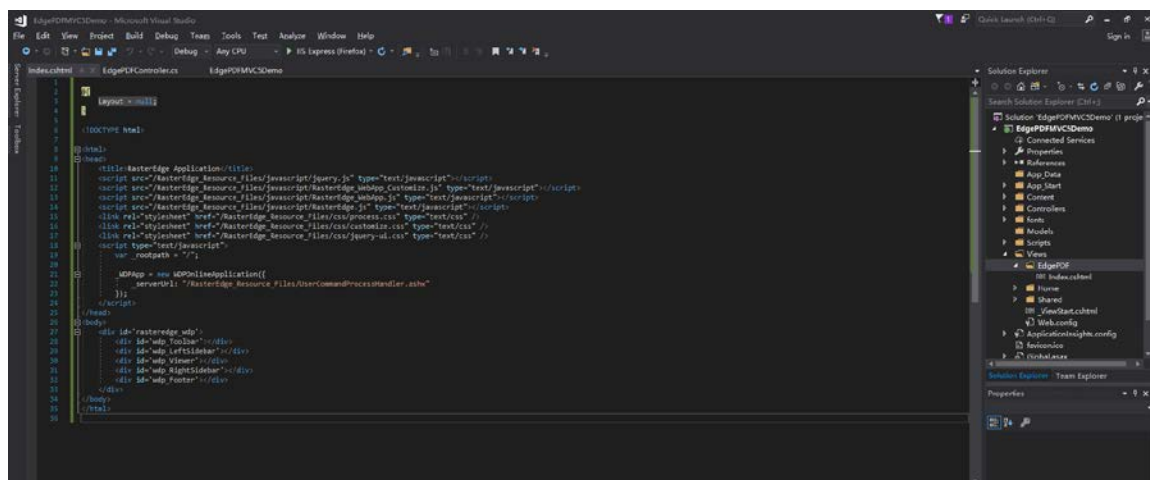
Add a new folder **“EdgePDF”** to the Views



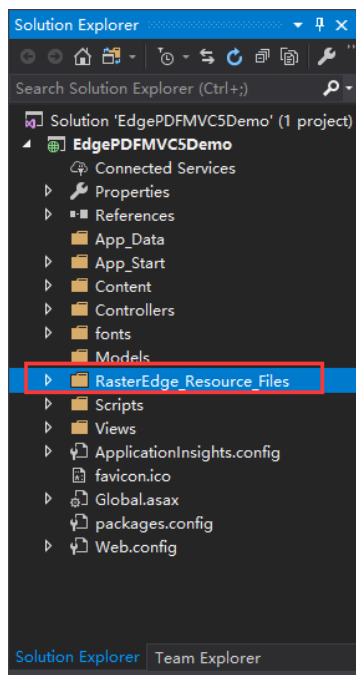
Add a new View **Index** under folder **EdgePDF**



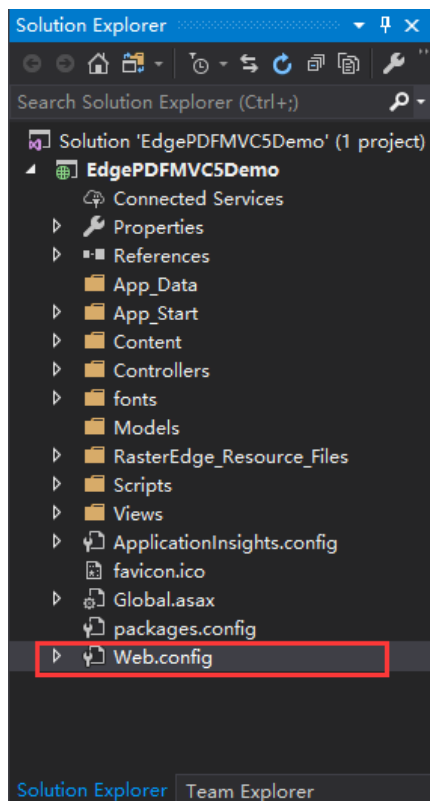
Copy the contents in file “/Developer Guide/EdgePDF/MVC5 project files/**Index.cshtml**” to your Index.html file.



5. Add the folder “/DemoProjects/EdgePDF 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,html
    Default: html
-->
<add key="reDocRenderEngine" value="html"/>
<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"/>
```

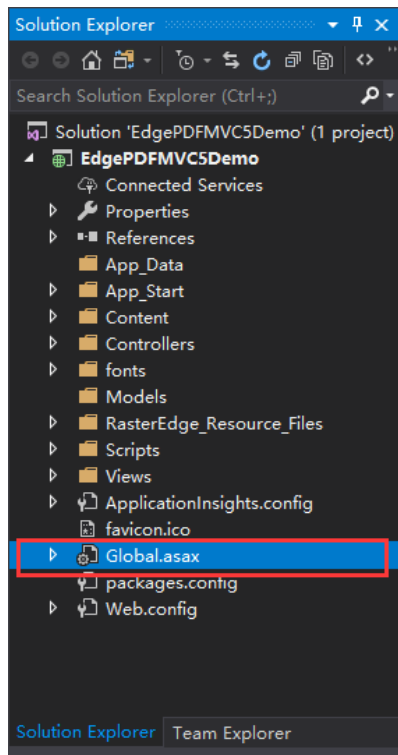
Add the following contents to tag **<system.web>**

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

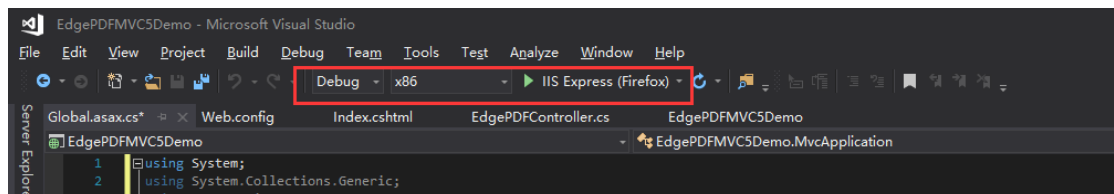
Note: If web.config has already defined tag "<httpRuntime>", you need merge the above content with existed tag contents.

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

Copy or add the “/Developer Guide/EdgePDF/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:7729/edgepdf/>”

## 5. Embed EdgePDF into HTML/ASPX Web Page

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

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

### **Sample 1: Embed editor into your html, aspx pages using DIV and**

#### **iFrame: edgedoc-page-frame.html**

It embeds EdgePDF editor 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 EdgePDF editor in your html, aspx pages:**

**edgedoc-page-popup.html**

It will popup the EdgePDF editor inside html page.

## 6. Document Connect with EdgePDF

To run the following demo code, 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 EdgePDF

*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”*

WithEdgePDFHTML5 PDF Editor, you can easily load a file from your server file system or database into viewer.

Here are the details

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

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

RasterEdge.XDoc.PDF.PDFDocument pdf = new
    RasterEdge.XDoc.PDF.PDFDocument(@"C:\\temp\" + targetkey);
// object can be PDFDocument,File bytes or File Stream
webDoc = REProcessControl.PageLoadFile(request,pdf,targetkey);
```

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 EdgePDFeditor.
4. You can change http parameter **yourtarget** to other names, like **fileid**.

## 6.2 Save file from EdgePDF into your serverfile system

Define your saving command in EdgePDF web editortoolbar

Navigate to **{EdgePDF 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 **{EdgePDF 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 = "RasterEdgeEdgePDF";
        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. Configure EdgePDF 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>
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: html, svg, png</p>

	Default: html
--	---------------

## 8.Enable “Content Edit” functions

By default, EdgePDF has disabled the Content Edit functions in the demo project.

To enable it, please follow the steps:

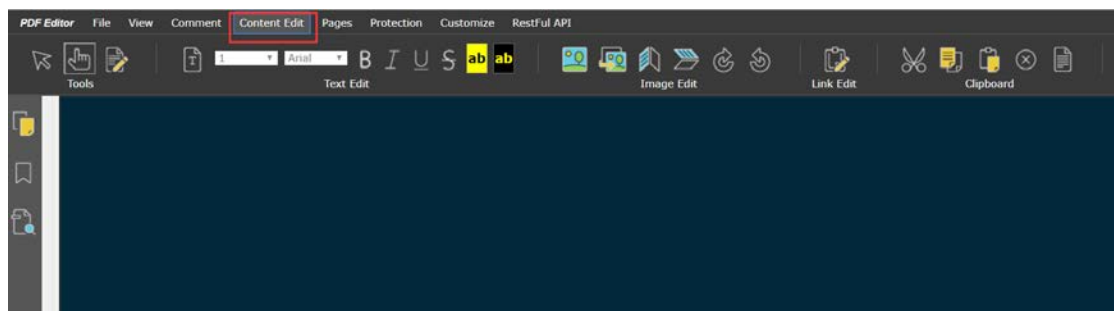
1. In file “/Web.Config”, change render engine to “html” (default is “svg”)

```
<add key="reDocRenderEngine" value="html"/>
```

2. In file “/RasterEdge\_Resource\_Files/Javascript/RasterEdge\_WebApp.js”, uncomment the following Javascript code

```
enableContentEdit();
```

3. Restart the IIS, and clear your web browser cached data. Open web browser and navigate to EdgePDF web application, you can find the “Content Edit” toolbar functions.



## 9.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"

### 9.1 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.

## 9.2 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");
```

## 10. Customize EdgePDF

### 10.1 Re-locate Web Editor 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”**.

### 10.2 Re-locate Web Editor 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.



## 10.3 UI: Web Editor width and height settings

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

Editor width and height with absolute size:

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

Editor width and height with relative size:

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

## 10.4 UI: Web Editor startup form view

By default, when editor starts, it will display a form with several tabs to allow users to

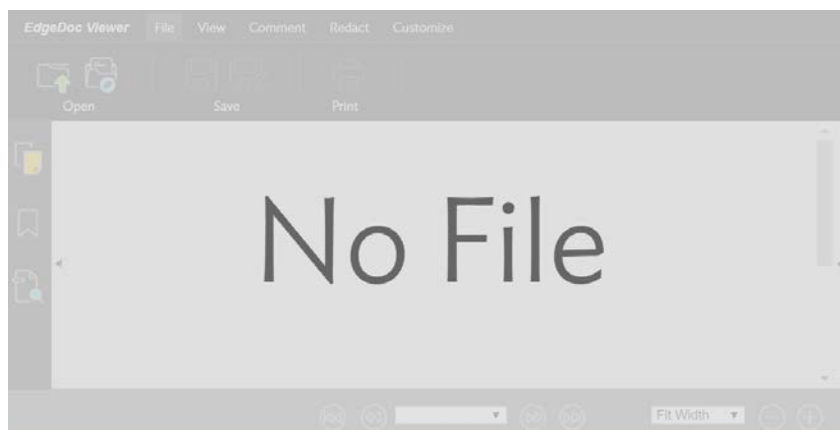
- upload a pdf file;

- upload and convert other files to pdf;
- open an online pdf file;
- combine multiple files to pdf document

Sometimes, you do **NOT ALLOW** users to upload their own files. You can change editor startup status to “**NOFILE**”.

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

```
setDialog("NOFILE");
// valid values: UPLOAD-PDF, CONVERT-2-PDF, OPEN-ONLINE-PDF, COMBINE-FILES-2-PDF, NOFILE
```



You can also remove certain tabs in the startup form:

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

```
setEditorUploadMethod(["CONVERT-2-PDF", "OPEN-ONLINE-PDF", "COMBINE-FILES-2-PDF",  
"UPLOAD-PDF"]);
```

## 10.5 UI: Web Editor HTML page title

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

In {EdgePDF 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("EdgePDFEditor: online view, comment, redact, edit, convert PDF documents.",true);
```

## 10.6 Comments settings

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

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

## 10.7 Define your own toolbar and commands

It is really easy to define your own toolbar and commands. Navigate to {EdgePDF 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.

# 11. Customize, extend Annotation, Comment, Markup, Redaction

## 11.1 Set annotation default properties cookie live time

In RasterEdge\_WebApp.js, initWebApplication(), you can set your preferred cookie expiring dates.  
**setCookieExpires(7);**//day

## 11.2 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;**

## 11.3 Enable continuous text redact & area redact

To enable continuous adding text redactions and area redactions, please do the following change:

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

## 11.4 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.

## 11.5Get 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 cooradinales 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 cooradinales under Select mode,call by system after mouse down  
    //if (location != null)  
        //      console.log("x,y:(" + location.x + "," + location.y + "),pageIndex:" +  
location.pageIndex);  
}  
-----
```

## 12. Extend EdgePDF Examples

### 12.1 Example 1: Add watermark to all documents displayed on EdgePDF

To add text watermark to all uploaded documents, in `/RasterEdge_Resource_Files/UserCommandProcessHandler.ashx`, method `"FileProcess()"`, add the following code,

```
RasterEdge.Imaging.Basic.REWatermarkSourceTexttextwatermark = new RasterEdge.Imaging.Basic.REWatermarkSourceText("Your Company",newSystem.Drawing.Font("Arial", 12F));
RasterEdge.Imaging.Basic.REWatermark watermark = new RasterEdge.Imaging.Basic.REWatermark(textwatermark);

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

Comment out the following code:

```
//webDoc = REProcessControl.PageLoadFile(request);
```

Or you can add image watermark

```
Bitmap bitmap = new Bitmap(@"...");
REWatermarkSourceImageimgSource = new REWatermarkSourceImage(bitmap);

REWatermark watermark = new REWatermark(imgSource);
watermark.RelativeWidth = 80F;
watermark.HoriAlign = REHoriAlignment.Middle;
watermark.RelativeHeight = 80F;
watermark.VertAlign = REVertAlignment.Middle;
watermark.Rotate = 45F;
watermark.ScaleMode = REScaleMode.None;
watermark.Opacity = 0.8F;
watermark.IsAbovePage = true;
```

## 12.2 Example 2: Apply pre text search before documents displayed on EdgePDF

In file /RasterEdge\_Resource\_Files/Javascript/**RasterEdge\_WebApp.js**,

```
var initWebApplication = function () {
setWebTitle(i18n['web']['EditorTitle'], true);
setApplicationTitle(i18n['web']['AppEditorTitle']);
initFileToolbar();
initViewToolbar();
initCommentsToolbar();
initEditToolbar();
initPagesToolbar();
initProtectionToolbar();
setCommentsStyle();
setCustomFileFormat();
setCookieExpires(7);//day
    //setDefaultSearchText(["Edge"], false, false);

    // Parameter 1: words list,
    // Parameter 2: is case sensitive
    // Parameter 3: is whole word
    setDefaultSearchText(["Edge"], false, false);
}
```





## 13.2 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 may 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 threadIsolated, Boolean fIgnore
System.Reflection.RuntimeAssembly._LoadAssemblyName fileName, String codebase, Evidence assemblySecurity, RuntimeAssembly locationHint, StackCrawlMark stackMark, IntPtr pPrivateData, Boolean threadIsolated, Boolean fIgnore
System.Reflection.RuntimeAssembly.InternalLoadAssemblyName(AssemblyName assemblyName, Evidence assemblySecurity, RuntimeAssembly requestingAssembly, StackCrawlMark stackMark, IntPtr pPrivateData, Boolean threadIsolated, Boolean fIgnore)
System.Reflection.RuntimeAssembly.InternalLoad(String assemblyString, Evidence assemblySecurity, StackCrawlMark stackMark, IntPtr pPrivateData, Boolean fIgnore) +27
System.Reflection.Assembly.Load(String assemblyString) +28
System.Web.Configuration.CompilationSection.LoadAssembly(String assemblyName, Boolean starDirective) +38

[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.LoadAssembly(String assemblyName, Boolean starDirective) +78
System.Web.Configuration.CompilationSection.LoadAllAssembliesFromAppDomainStaticDirectory() +217
System.Web.Configuration.CompilationSection.LoadAssembly(byte[] all) +138
System.Web.Compilation.BuildManager.GetReferencedAssemblies(CompilationSection compilationSection) +138
System.Web.Compilation.BuildManager.GetPreStartMethodsFromReferencedAssemblies() +92
System.Web.Compilation.BuildManager.CallPreStartMethods(String preStartInitPath, Boolean isReferencedAssembly) +290
System.Web.Compilation.BuildManager.ExecutePreStart() +157
System.Web.Hosting.HostingEnvironment.Initialize(ApplicationManager appManager, ApplicationHost appHost, IConfigurationFactory configPathFactory, HostingEnvironmentParameters hostingParameters, PolicyLevel policyLevel, Exception e)

[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) +2800000
System.Web.HttpRuntime.InternalRequestInit(HttpContext context) +95
System.Web.HttpRuntime.ProcessRequestInitPrivate(HttpContext context) +254
```

Version Information: Microsoft .NET Framework Version 4.0.30319; ASP.NET Version 4.7.21018.5

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>

## 14. 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.PDF.HTML5Editor.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. RasterEdgeSDK 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**.