



Image Components for ASP.NET/HTML5/JavaScript

Installation Guide

Contents

Introduction	3
Features Include	3
Requirements	3
Installation.....	4
License Manager Installation	5
Image Manager WCF Web Service configuration	6
Image Manager WebAPI Web Service configuration.....	7
<i>Enable Cross-Origin resource sharing</i>	7
<i>Configuring HTTPS binding</i>	8
<i>Setting up the Web.config app settings</i>	8
Configuring the ImgScan client service	9
<i>ImgScan client service support for https</i>	10
<i>Manually Certificate creation</i>	11
<i>ImgScan service error logging</i>	11
Scaling Image Manager Web API Web Service	12
<i>YARP 2.0 configuration example</i>	12
<i>YARP 2.0 configuration file (appsettings.json)</i>	13
<i>What changes in the components configuration</i>	13
Developer Licensing	14
<i>ASP.NET Forms license:</i>	14
<i>ASP.NET MVC & Forms license:</i>	15
<i>ASP.NET CORE, HTML, and Others:</i>	15
Purchase Options	16
Deployment.....	16
Troubleshoot	18
Running the Demo Projects	20

Introduction

Image Components for ASP.NET/HTML5/JavaScript SDK is a JavaScript based document image viewing control, which can be easily integrated into your own ASP.NET applications.

The HTML5/JavaScript toolkit provides developers with the power of the Image Components SDKs for cross-platform development.

View, annotate, and manage documents, on any device, through a fully customizable zero-footprint HTML5/AJAX Viewer.

Developed using HTML5, a content server and additional technologies, Image Components for ASP.NET/HTML5/JavaScript SDK can easily provide imaging functionality to web applications. Operates in any browser, eliminates client platform compatibility issues and runs in Internet Explorer, Safari, Chrome and Firefox.

Features Include

- View and Edit the most common image file types.
- View PDF documents.
- Open and save documents locally or in server side.
- Image page navigation.
- Image rotation and mirror.
- Zoom percentage or selectable zoom zone.
- Undo and Redo.
- Image management with append, insert, delete and move pages.
- Manual or automating cropping and clear image zones.
- De-skew, resize and border removal.
- Many image filters (Invert, grayscale, median, black and white...)
- Many enhancements (Smooth, sharpen, brightness, color, Contrast...)
- Edge detections.
- Document, page or zone OCR.
- Text search and highlight.
- Exif image read and write access.
- Local and remote load of annotations.
- Local and remote save of annotations.
- Multipage annotations support.
- Add, selected, double click and deleted events.
- Annotations insert by selection.
- Text, Stamp, Highlight, Hide, Ellipse, Embedded Image annotations.
- OCR, barcode and zoom zone annotations.
- Edit, remove and clear page annotations.
- Access to the page annotations collection.

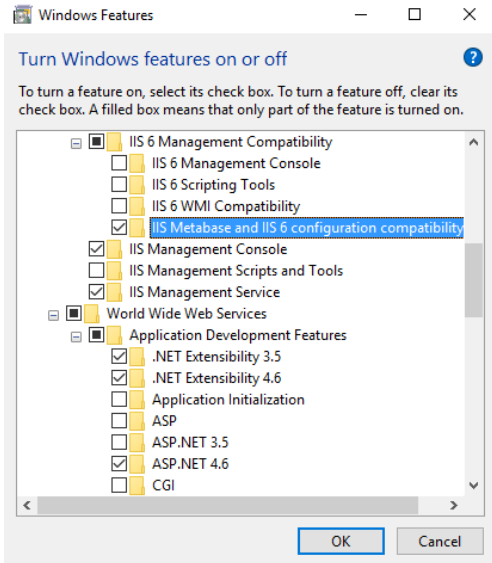
Image Components for ASP.NET is in constant development, other imaging features will be available soon.

Requirements

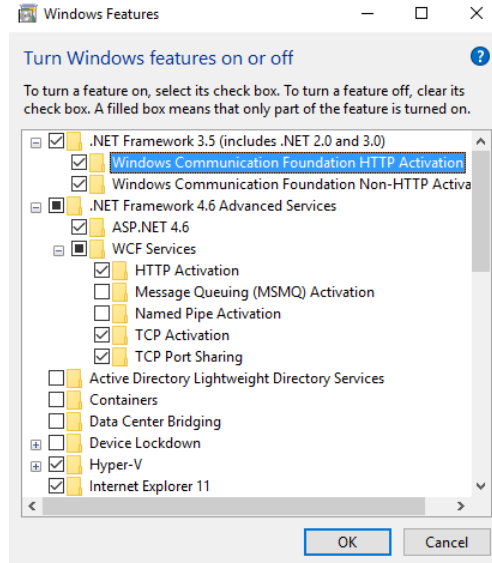
- Windows 7 or higher, 32 bit or 64 bit architecture.
- **IIS Metabase Compatibility (Required by the installation package).**
- Visual Studio 2019 and above
- Some experience in ASP.NET/HTML5/JavaScript development.
- Some experience in Web applications deployment.

Installation

Requirements before running the SDK installation wizard:

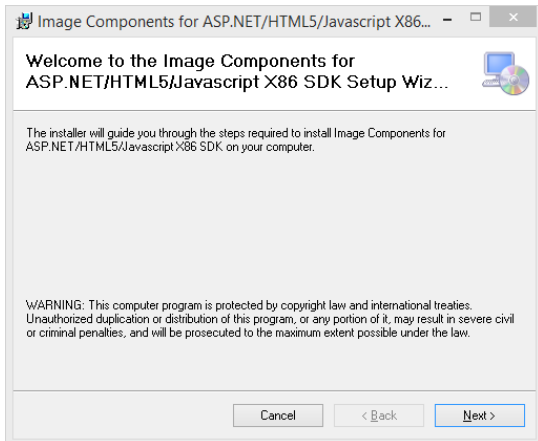


IIS Metabase and IIS 6 configuration compatibility

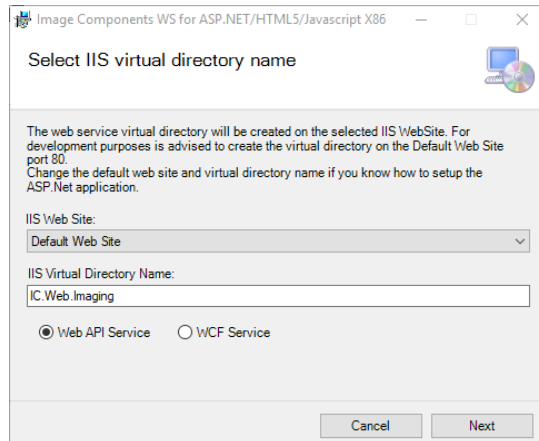


Enable WCF Services

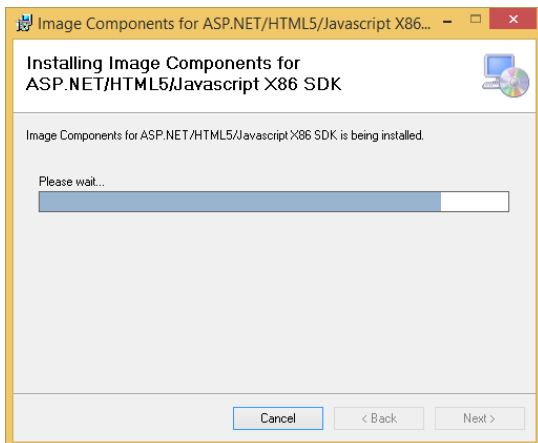
Image Components for ASP.NET SDK installation wizard X86 or X64:



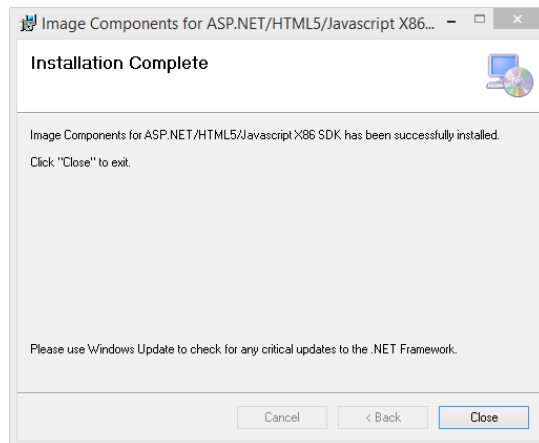
Welcome Screen



Installation Address *



Installing Image Components SDK



Installation Complete

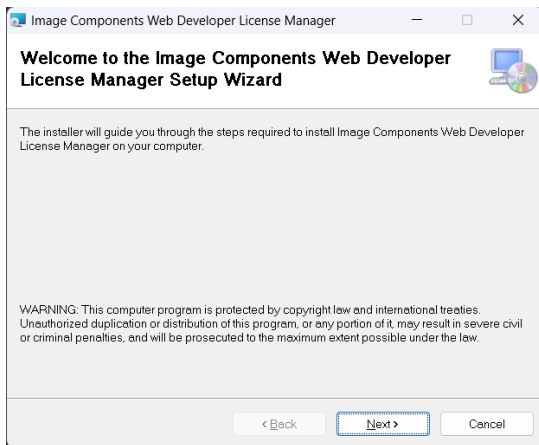
* - IIS Site, virtual directory and Web API or WCF Service type.
An application pool will be created automatically by the installer.

License Manager Installation

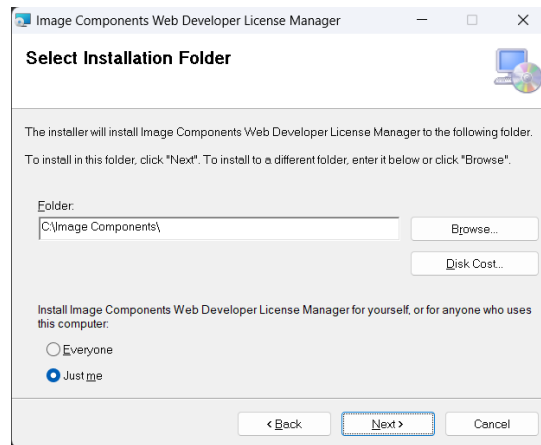
If you own a valid developer license, is required the installation of the license manager application.

In the Image Components installation path there is a folder named 'License'.

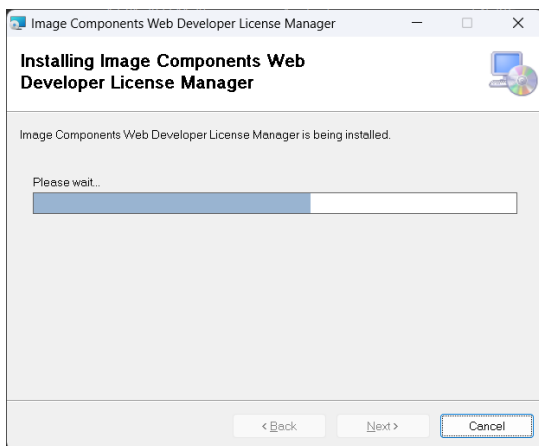
Ex: [Drive]:\[InstallationPath]\Image Components\License\ (Installation default path).
The 'License' folder contains the 'IC.Licensing.msi' file.
The installation files are also available on-line (ASP.NET/Blazor Developer License):
<http://www.imagecomponents.net/Downloads/Index/-1>



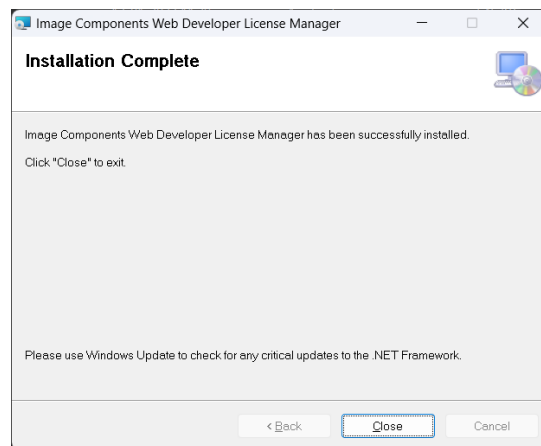
Welcome Screen



Installation Path



Installation Progress



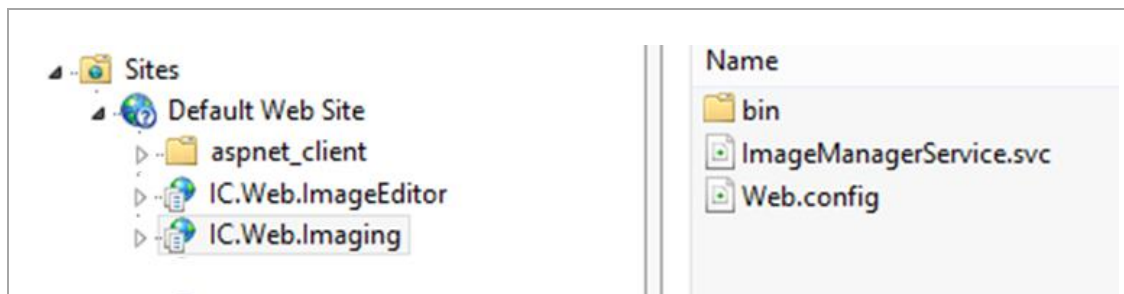
Installation Complete

Image Manager WCF Web Service configuration

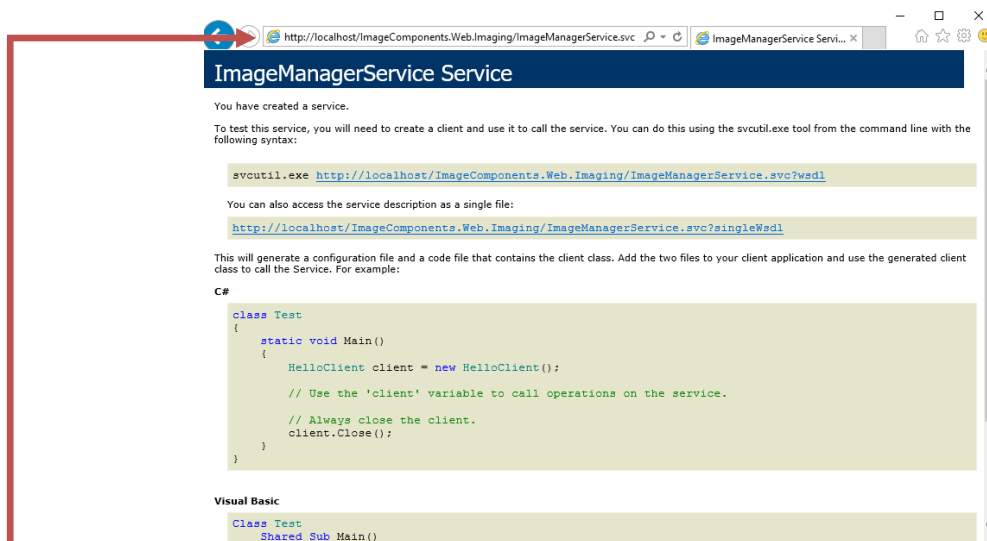
After the wizard completion, is advised to test the WCF image manager service, checking if the installation is correct and also to obtain the URL address for the configuration of the ASP.NET/HTML5/JavaScript ImgEdit component.

Browsing the WCF Image Manager Web Service:

- Open the Internet Information Services (IIS).
- Expand the 'Default Web Site'.
- Select the 'IC.Web.Imaging' virtual directory (Installation default).
- Click on the 'Content View' button.



- Right click in the 'ImageManagerService.svc' file and in the context menu press 'Browse'.



If a similar page appears in your browser, the installation was completed with success and the WCF service is configured correctly. If not, check the troubleshoot page.

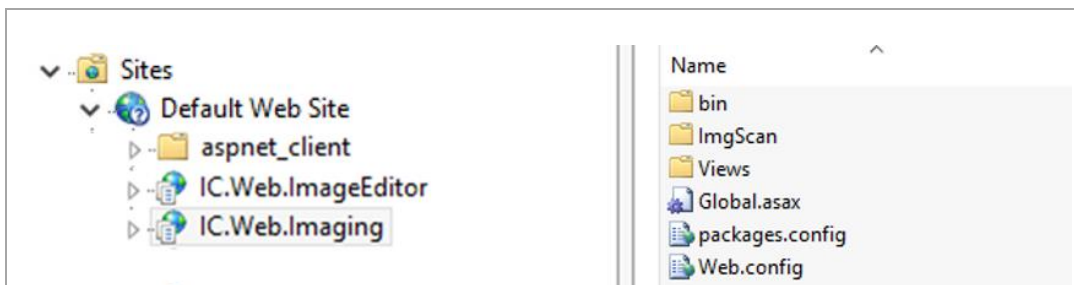
Copy the URL address and save it for later use.

Image Manager Web API Web Service configuration

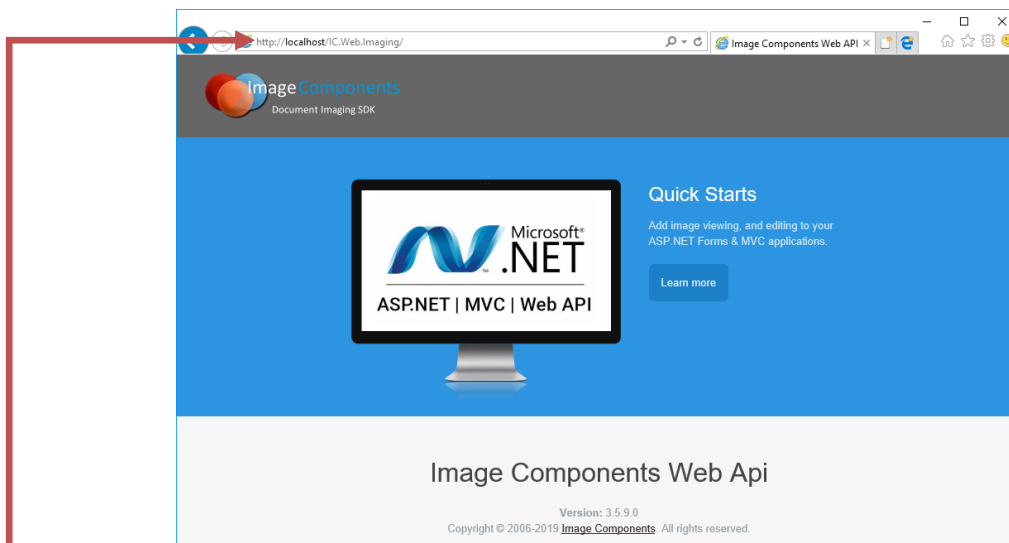
After the wizard completion, is advised to test the Web API image manager service, checking if the installation is correct and also to obtain the URL address for the configuration of the ASP.NET/HTML5/JavaScript ImgEdit component.

Browsing the Web API Image Manager Web Service:

- Open the Internet Information Services (IIS).
- Expand the 'Default Web Site'.
- Select the 'IC.Web.Imaging' virtual directory (Installation default).
- Click on the 'Content View' button.



- In the 'Actions' panel press the 'Browse' link.



If a similar page appears in your browser, the installation was completed with success and the WebAPI service is configured correctly. If not, check the troubleshoot page.

Copy the URL address and save it for later use.

Enable Cross-Origin resource sharing

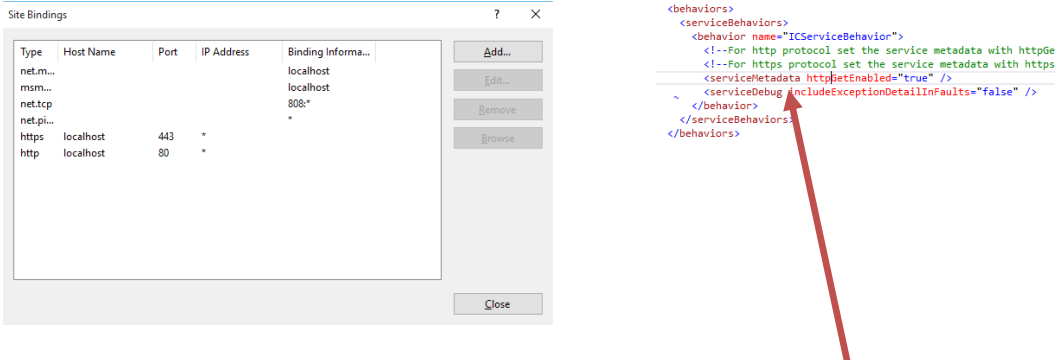
Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS is enabled by default in the WCF Image Manager service.

Wikipedia information: http://en.wikipedia.org/wiki/Cross-origin_resource_sharing

Configuring HTTPS binding

The WCF web service will be configured automatically depending of the enabled binding in the web site. **You cannot have both http and https bindings enabled on the web site.**



The Site Bindings dialog box shows the following configuration:

Type	Host Name	Port	IP Address	Binding Informa...
net.m...				localhost
msm...				localhost
net.tcp				808:*
net.pi...				*
https	localhost	443	*	
http	localhost	80	*	

The web.config code snippet shows the following configuration for the 'ICServiceBehavior' service behavior:

```
<behaviors>
  <serviceBehaviors>
    <behavior name="ICServiceBehavior">
      <!--For http protocol set the service metadata with httpGe
      <!--For https protocol set the service metadata with https
      <serviceMetadata httpGetEnabled="true" />
      <serviceDebug includeExceptionDetailInFaults="false" />
    </behavior>
  </serviceBehaviors>
</behaviors>
```

To enable the WCF service metadata for the required binding, change the web.config by setting the following service metadata keys:

HTTP: `<serviceMetadata httpGetEnabled="true" />`

HTTPS: `<serviceMetadata httpsGetEnabled="true" />`

Both web application and WCF service must have the same URI scheme.

Setting up the Web.config app settings

There are some configurations that must be made to the web.config file, in order to have access to specific functionalities.

- Open the Web.config file.

```
<appSettings>
  <!--Ex: C:\MyImageFiles\-->
  <add key="ServerFolderFiles" value="" />
  <!--Ex: C:\OCRLanguages\-->
  <add key="OCRLanguagesPath" value="" />
  <!--Ex: C:\MyLogPath\-->
  <add key="ErrorLogPath" value="" />
  <!--Ex: 60 Value in seconds-->
  <add key="ClientStayAlivePing" value="60" />
  <!--Ex: Url=MyFTPServer; Uid=MyUserName;Pwd=My
  <add key="FTPConnection" value="" />
  <!--Ex: ServerType=(MSSQL|MYSQL|SQLLITE); Serv
  <add key="DatabaseConnection" value="" />
```

There are 4 keys in the <appSettings> node that should be configured:

- ServerFolderFiles.
- OCRLanguagesPath.
- ErrorLogPath.
- ClientStayAlivePing.

ServerFolderFiles:

The folder containing the server side images or PDF files. The ImgEdit component can read and save server side documents.

OCRLanguagesPath:

The OCR languages files path. If this folder is not set, the OCR functions are not enabled.

ErrorLogPath:

If this key is set, all service logs will be written in a file to this folder. The log file has a default name of 'IC.WS.Imaging.Service.html'.

ClientStayAlivePing:

The web-service permanent connection ping interval in seconds.

FTPConnection:

The FTP connection configuration.

Ex: `<add key="FTPConnection" value="Url=MyFTPServer; Uid=MyUserName;Pwd=MyPassword;Domain=;Anonymous=false;Timeout=10000;" />`

DatabaseConnection:

The database connection configuration.

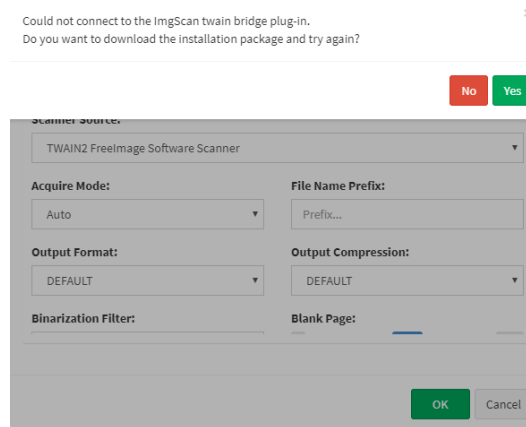
Ex: `<add key="DatabaseConnection" value="ServerType=MSSQL;Server=MyServer;Database=MyDatabase;Uid=MyUserName;Pwd=MyPassword;" />`

Read and write permissions must be given to the specified folders.

Configuring the ImgScan client service

To successfully connect the browser to a twain/WIA device, some configurations must be made. The configuration steps can differ from each browser vendor.

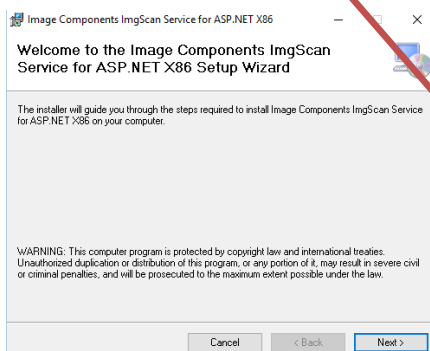
The connection from the browser to the scanning service is made using the web socket protocol.



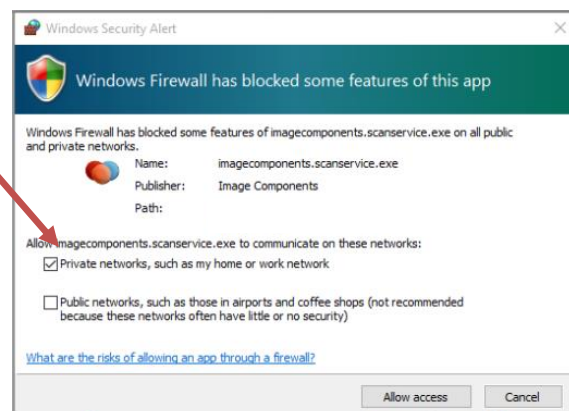
Selecting the device

After opening the scanner settings dialog, the WebImgScan component tries to connect to the <http://localhost:8182> or the <https://localhost:8183> web socket port. If the connection is unavailable, a message is displayed order to download and install the client service.

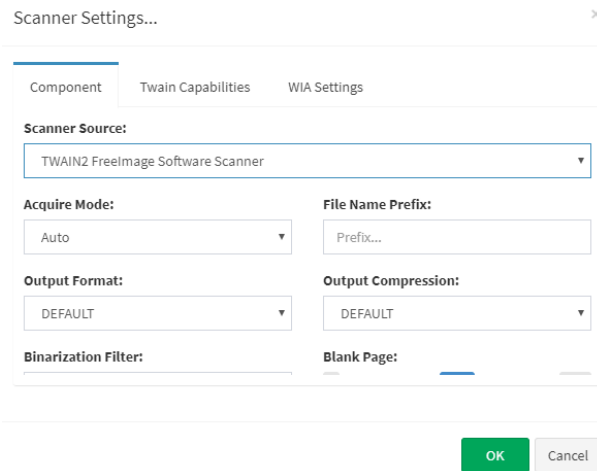
The installer dialog will display the progress and information about the service installation. Completed the installation, the windows firewall will prompt you to allow or deny the access to the network. Since the required communications is only made locally, you only need to allow access to the **private networks**.



Installation progress



Firewall warning dialog



Restart the browser or press CTRL+F5 to reload the web page.
Open the scanner settings dialog to select and configure the available twain or WIA devices.

Special configuration may also be necessary to use the older versions of Edge with Windows 10 OS.

MS Windows 10 uses an isolation technology ("AppContainer") that may prevent requests being sent from the older versions of Edge via AppScan to the ImgScan client service.

<https://msdn.microsoft.com/en-us/library/windows/apps/hh780593.aspx>

To enable loopback for Edge, use this command in Command Prompt:

```
CheckNetIsolation LoopbackExempt -a -n="Microsoft.MicrosoftEdge_8wekyb3d8bbwe"
```

To disable loopback for Edge use this command in Command Prompt:

```
CheckNetIsolation LoopbackExempt -d -n="Microsoft.MicrosoftEdge_8wekyb3d8bbwe"
```

ImgScan client service support for https

If your ASP.Net application is running under HTTPS/SSL the ImgScan Client service will also require a secure connection to the browser.

An X509 certificate containing a public and private key must be added to the service configuration.

Both endpoints **must have valid certificates** in order to communicate.

You can choose one of the following ways to create and apply a certificate to the ImgScan Client Service:

1 – Automatic creation. The installation of the ImgScan Client service will try to create a local self-signed certificate automatically. No further steps are required. The installation must be done using an Administrator account.

2 - Purchase an SSL certificate from an external trusted authority such as Symantec (VeriSign), Thawte, or GoDaddy. This has the same benefits as a domain certificate, but does not require a domain certificate authority. **The drawback** is that this is the most expensive option, which could be prohibitive.

3 - Create a self-signed certificate to be used as a trusted root certificate and create a host certificate using the trusted self-signed certificate. This certificate is only trusted on the machine where the certificate was created. This option was added to the installation package (*.msi) of the ImgScan Client service (version 3.1.2.5). The installation package **will try to automatically create and apply** a self-signed certificate on the local machine certificate store. In order to correctly create the certificate, the installation package must be run as a **machine Administrator**.

Manually Certificate creation

Open the windows power shell console on the local machine where the `ImgScan` service will be installed as an Administrator, and insert the following lines of code:

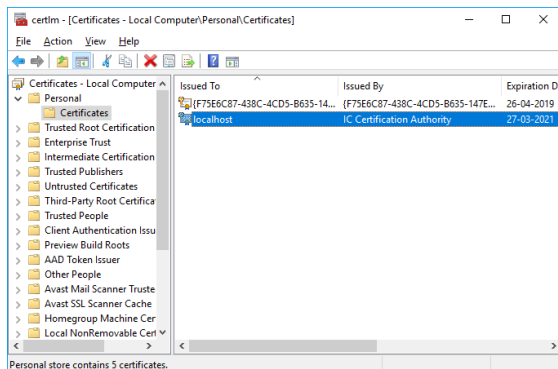
```
$params = @{
    DnsName = "IC Certification Authority"
    KeyLength = 2048
    KeyAlgorithm = 'RSA'
    HashAlgorithm = 'SHA256'
    KeyExportPolicy = 'Exportable'
    NotAfter = (Get-Date).AddYears(5)
    CertStoreLocation = 'Cert:\LocalMachine\My'
    KeyUsage = 'CertSign','CRLSign' #fixes invalid cert error
}
$rootCA = New-SelfSignedCertificate @params

$params = @{
    DnsName = "localhost"
    Signer = $rootCA
    KeyLength = 2048
    KeyAlgorithm = 'RSA'
    HashAlgorithm = 'SHA256'
    KeyExportPolicy = 'Exportable'
    NotAfter = (Get-date).AddYears(2)
    CertStoreLocation = 'Cert:\LocalMachine\My'
}
$vpnCert = New-SelfSignedCertificate @params

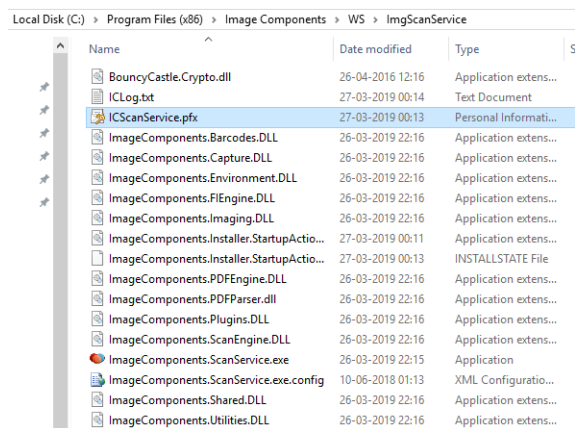
Export-Certificate -Cert $rootCA -FilePath "E:\certs\rootCA.crt"
Import-Certificate -CertStoreLocation 'Cert:\LocalMachine\Root' -FilePath "E:\certs\rootCA.crt"

Export-PfxCertificate -Cert $vpnCert -FilePath 'E:\certs\ICScanService.pfx' -Password (ConvertTo-SecureString -AsPlainText 'MyPassword' -Force)
```

Press F5 and run the script. A valid export folder (Ex: `E:\certs\`) must exist. The `ICScanService.pfx` should have been exported to the `E:\certs\` folder with the `MyPassword` password. Add the certificate to the `ImgScan` service as previously shown.



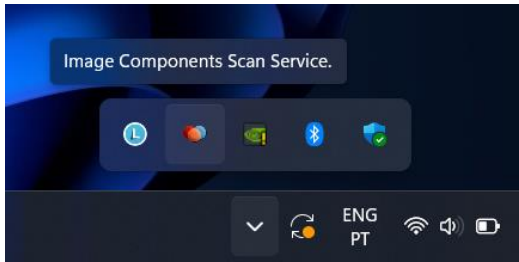
Manually or automatically created certificate.



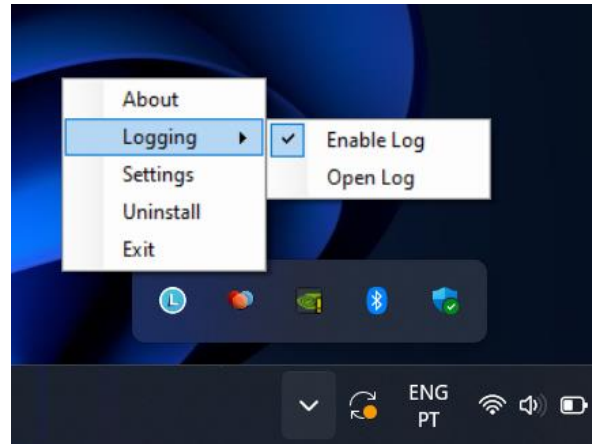
Certificate created by the installation package.

ImgScan service error logging

In order to diagnose and troubleshoot any problem with the connection or scanning, you can enable our extensive log, by right clicking the image components scan service icon on the system tray.



System Tray Icon



Enable and Open Log

Scaling Image Manager Web API Web Service

You can scale your imaging application by creating a pool of services that can respond to an increase number of users.

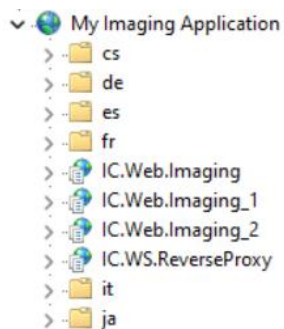
This can be done with a web gateway, creating a unified entry point, with the intent of distributing the unique entry point request to a listening pool of services.

Here are two great open source examples:

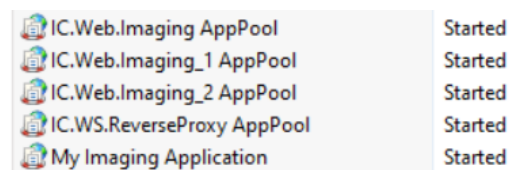
- Ocelot - <https://github.com/ThreeMammals/Ocelot>
- YARP 2.0 - <https://microsoft.github.io/reverse-proxy/>

We do prefer using YARP, but this is only personal preference, as both works great. The only disadvantage that we found was on Ocelot, as it doesn't support virtual directories or virtual web applications (web Applications inside web applications) supported by IIS.

YARP 2.0 configuration example



Web application with 3 imaging services and the YARP reverse proxy



One application pool for each service

In this IIS configuration example, we have the main application (My Imaging Application), making request to a single point (IC.WS.ReverseProxy), that in turn, redirects the request to an Imaging Service, just like a load balancing, distributing network traffic equally across a pool of resources.

YARP 2.0 configuration file (appsettings.json)

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "ReverseProxy": {
    "Routes": {
      "route1": {
        "ClusterId": "ImagingCluster",
        "Match": {
          "Path": "/Imaging/api/{**catch-all}"
        },
        "Transforms": [
          {
            "PathRemovePrefix": "/Imaging"
          },
          {
            "ResponseHeader": "Source",
            "Append": "YARP",
            "When": "Success"
          }
        ]
      }
    },
    "Clusters": {
      "ImagingCluster": {
        "LoadBalancingPolicy": "LeastRequests", // "PowerOfTwoChoices", "FirstAlphabetical", "Random",
        "RoundRobin", "LeastRequests"
        "SessionAffinity": {
          "Enabled": true, // defaults to 'false'
          "Policy": "HashCookie", // defaults to 'HashCookie' alternatively "CustomHeader"
          "FailurePolicy": "Redistribute", // defaults to 'Redistribute' Alternatively "Return503Error"
          "AffinityKeyName": "ImagingKey1",
          "Cookie": {
            "Domain": "localhost",
            "Expiration": "03:00:00",
            "HttpOnly": false,
            "IsEssential": true,
            "MaxAge": "1.00:00:00",
            "Path": "mypath",
            "SameSite": "Strict",
            "SecurePolicy": "Always"
          }
        },
        "Destinations": {
          "destination1": {
            "Address": "http://localhost/IC.Web.Imaging/"
          },
          "destination2": {
            "Address": "http://localhost/IC.Web.Imaging_1/"
          },
          "destination3": {
            "Address": "http://localhost/IC.Web.Imaging_2/"
          }
        }
      }
    }
  }
}

```

Load balancing must be enabled, with session affinity setup.

What changes in the components configuration

If you want to use the reverse proxy instead of direct connection to an image service, change the **WebServiceURL** property value.

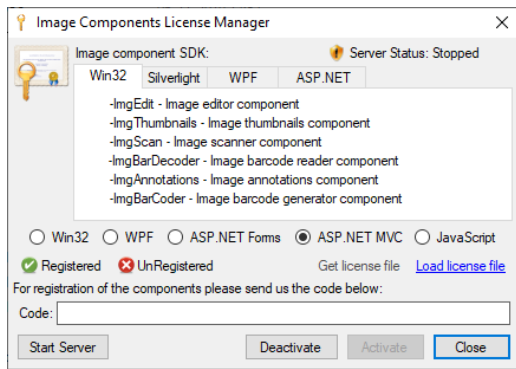
- Old Value: <http://localhost/IC.Web.Imaging/api/>
- New Value: <http://localhost/IC.WS.ReverseProxy/Imaging/api/>

Developer Licensing

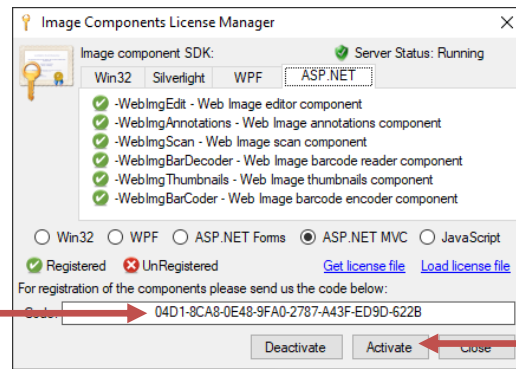
The Image Components for ASP.NET/HTML5/JavaScript installation includes a License Manager application. This manager allow you to check what components are licensed or not, and your developer machine key code.

Before starting using the components, you must start the license server.

- On the Task Bar press the Start button/All Programs/Image Components/Licensing/Web License Manager.



License Server Before Start



License Server Started

Key Code:

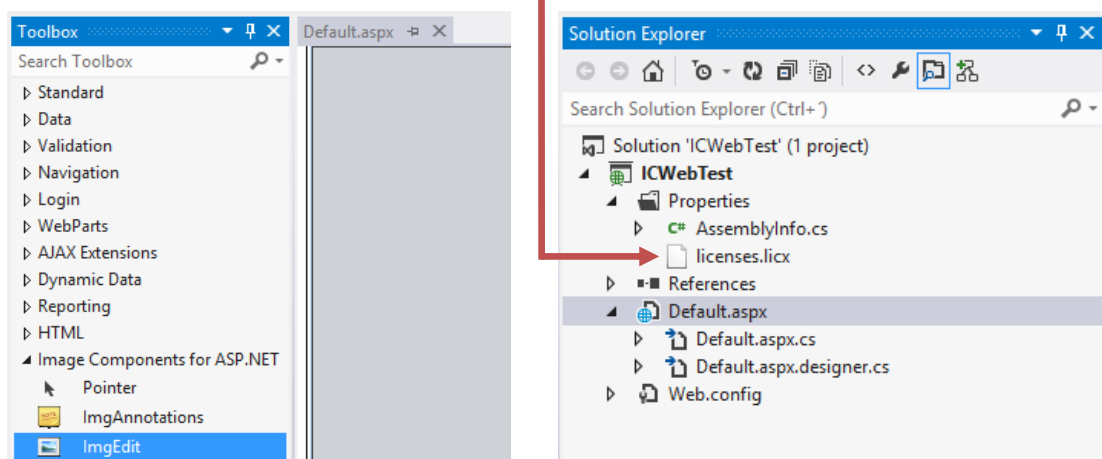
For registration, purchase and deploy purposes. It is important that you safe keep your code.

Activate:

After purchasing your will receive a confirmation. With an internet connection present, you just need to activate your components.

ASP.NET Forms license:

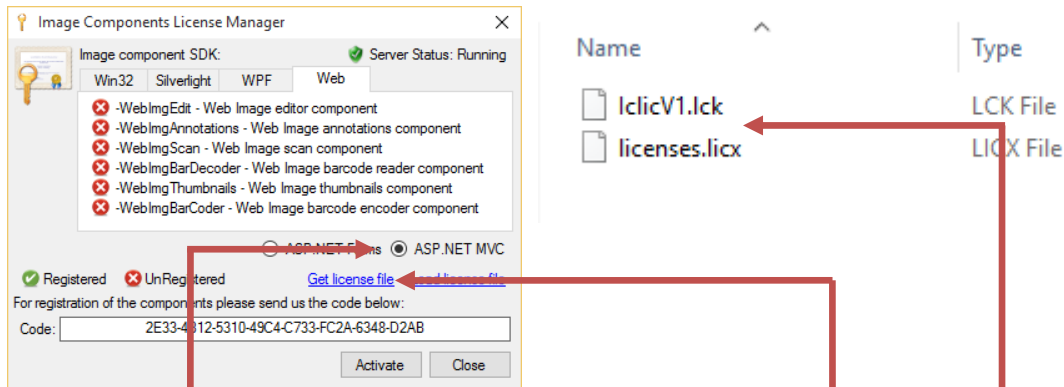
The license file is included automatically in your project, when you drag and drop the components from the Toolbox to a page.



ASP.NET MVC & Forms license:

You can also add the license file manually in an ASP.NET MVC or Forms project by doing the following procedure:

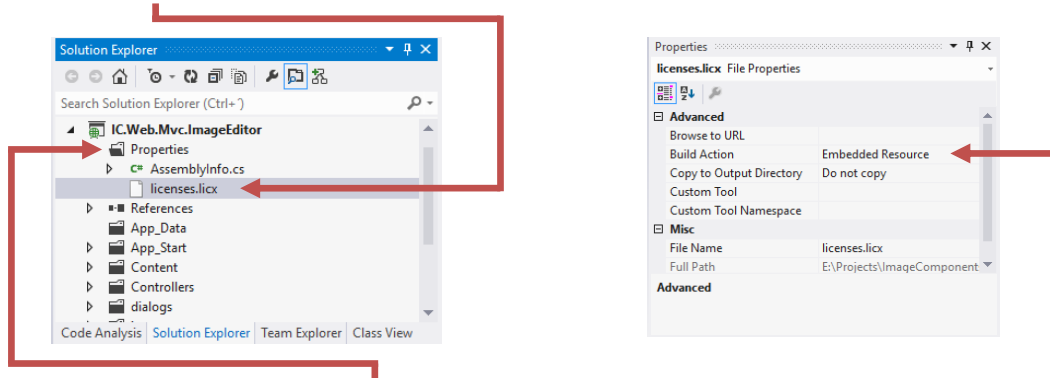
- On the Task Bar press the Start button/All Programs/Image Components/Licensing/Web License Manager.



Check the **ASP.NET MVC** or **Forms** radio button and press the Get license file link button. Select a known folder and press the OK button on the Save Image Components license dialog. Two files will be created on the selected folder:

- licenses.licx (The license file to be added to your main project)
- IcllicV1.lck (The **Silverlight** client license file.)

Add the licenses.licx file to your main project and set the **Build Action** to **Embedded Resource**.

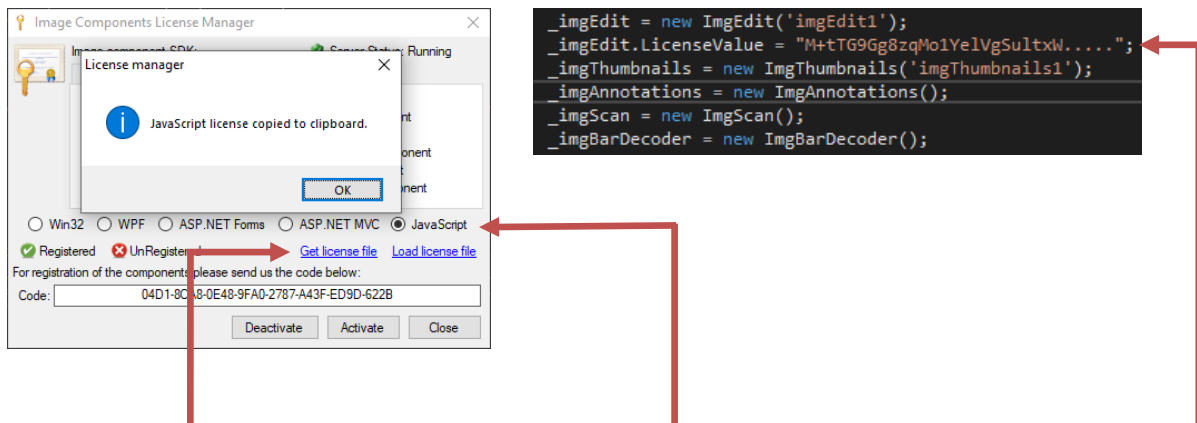


The file may be added to the **Properties/ Project** folder, root or any other folder. **Re-Build** your project.

If you are evaluating the components, **no license file** needs to be added to the project.

ASP.NET CORE, HTML, and Others:

You can also add the license value manually in an ASP.NET CORE, HTML or Other project types by doing the following procedure:



Check the **JavaScript** radio button and press the Get license file link button.

The license value will be copied to the clipboard, in order to paste to the project **LicenseValue** `_imgEdit` property.

Purchase Options

Components can be ordered over the Internet. Once you have purchased the product by completing the online order form, the full activation will be enabled on-line. You can order the fully functional version of the **Components** over the Internet using any major credit card. The ordering page is located on a secure server, ensuring that your personal information remains confidential. As soon as your order is processed, we'll provide you the online activation process. We will require your key code, to proceed with the product activation. The Components are licensed by developer, so they can be **freely** distributed with your applications.

Purchase steps:

- Download and install the latest version of the components.
- Try the components as long as you like.
- Check the following video on how to purchase and licensing:
[Purchase and Licensing](#)
- Check the following video on how to register your machine:
[Registration](#)

Deployment

The Image Manager Web Service and the License Manager tool must be included in the deployment of your application. For the ease of installation, there is a package ready to use. In the Image Components installation path there is a folder named 'Deployment'.

Ex: [Drive]:\[InstallationPath]\Image Components\Deployment\Asp.Net Framework Web Api\
(Installation default path).

The 'Deployment' folder contains the 'IC.Web.Imaging.msi' file.

The installation files are also available on-line:

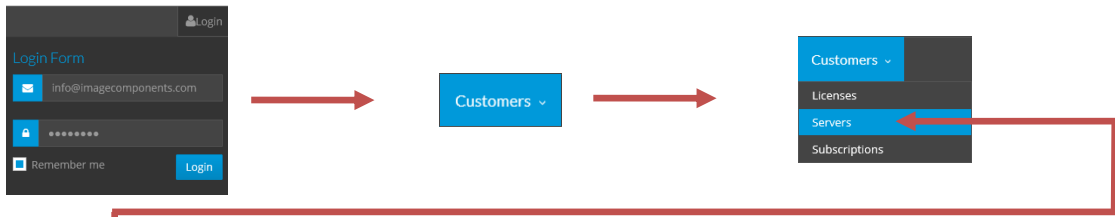
<http://www.imagecomponents.net/Downloads/Index/-1>

- ASP.NET WCF Imaging Service

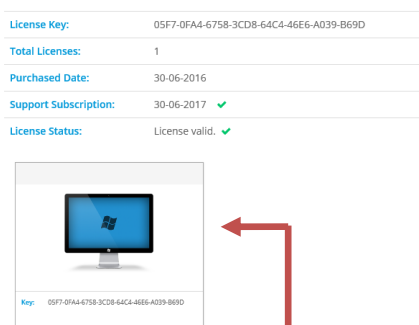
This package will install the required files in your production machines.

Your distribution must also be registered, using the following method:

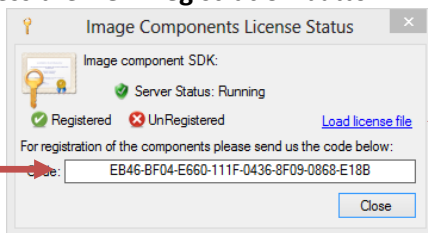
You must use your Image Components account in order to manage your licenses and registrations. We also provide a tool to register your server's deployments.



Press the 'Servers' link to manage the deployment's registrations.



- Select your purchased developer key.
- Press the **New Registration** button.



Server license manager application.

- Start the deployed server license manager application.
- Copy the **key code** from the license manager application to the **server key-code** text box.
- Press the **Insert** button.
- Select and download the new inserted server key-code.
- Load the downloaded license file using the license manager application.
- **Restart your IIS server.**

Troubleshoot

Ensure that IIS and WCF Are Correctly Installed and Registered:

MSDN: <http://msdn.microsoft.com/en-us/library/aa751792.aspx>

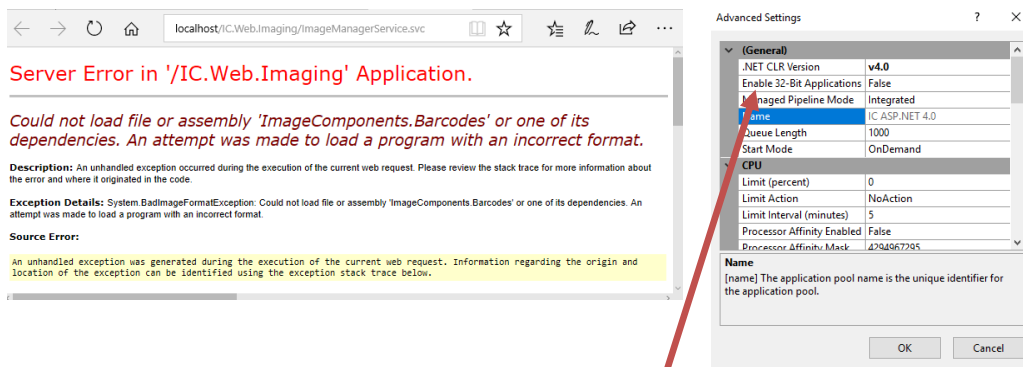
Could not load file or assembly 'ImageComponents....DLL' or one of its dependencies. The specified module could not be found.

Ensure that Microsoft Visual C++ 2015 or higher Redistributable Package (X86 or x64) is installed on your server:

- X86 - https://download.microsoft.com/download/9/3/F/93FCF1E7-E6A4-478B-96E7-D4B285925B00/vc_redist.x86.exe

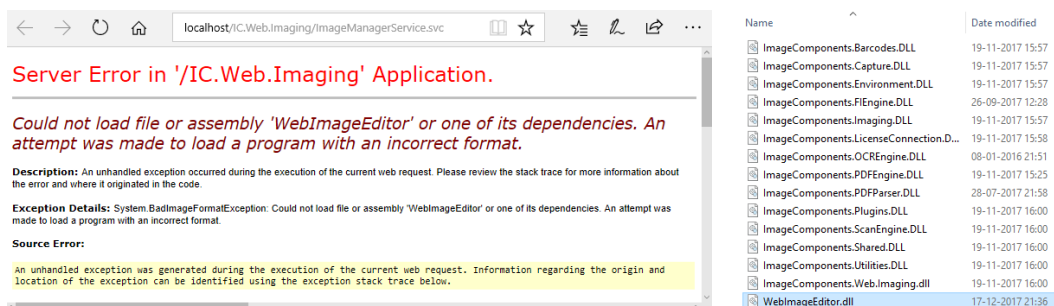
- X64 - https://download.microsoft.com/download/9/3/F/93FCF1E7-E6A4-478B-96E7-D4B285925B00/vc_redist.x64.exe

An attempt was made to load a program with an incorrect format:



Make sure that the application pool "Enable 32-bit Applications" setting running the service is correctly set to the WCF assembly's compilation platform.

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



Make sure that there aren't other assemblies located in the WCF bin folder with different platform compilation. All assemblies located in the bin folder must have the same platform compilation even if they are not needed, referenced or in use.

Failed to load http://... Response for preflight is invalid (redirect) .

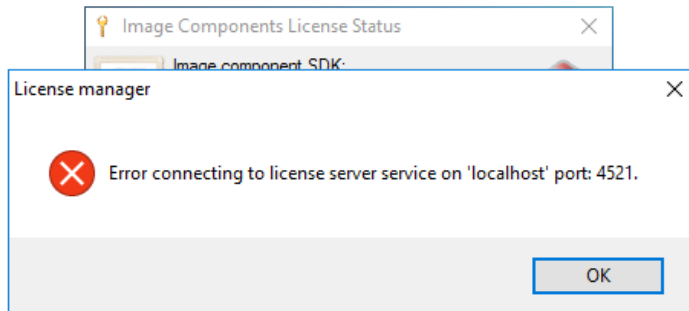
```

18:58:51.576 Navigated to https://localhost/IC.Web.Mvc.ImageEditor/
18:58:51.969 Browser: 4 Version: 537.36 Is Mobile: false
18:58:51.970 IE user agent: 0
18:58:52.216 ▶ [Violation] Added non-passive event listener to a scroll-blocking 'mousewheel' event. Consider marking event handler with 'passive' to allow the browser to remove the event listener after it has finished. See https://www.chromestatus.com/feature/5745543795965952
18:58:52.220 ▶ [Violation] Added non-passive event listener to a scroll-blocking 'mousewheel' event. Consider marking event handler with 'passive' to allow the browser to remove the event listener after it has finished. See https://www.chromestatus.com/feature/5745543795965952
18:59:28.352 Failed to load http://localhost/IC.Web.Imaging.WebApi/api/Connect: Response for preflight is invalid (redirect)

```

Make sure the application and the web service bindings are equal.

Error connecting to license server service on 'localhost' port: 4521 .



Make sure that the ICLicenseServerV1 service is running.

Name	Description	Status
Hyper-V Remote Desktop Virtualization Service	Provides a platform for comm...	
Hyper-V Time Synchronization Service	Synchronizes the system time ...	
Hyper-V Volume Shadow Copy Requestor	Coordinates the communicati...	
ICLicenseServerV1	Image Components License S...	Running

If the service status is 'Running', restart the license manager application as 'Administrator'.

The client license must be the same of the web service.

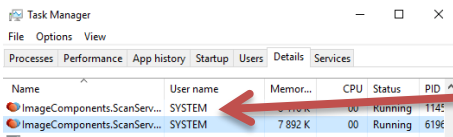
This site says...

One error occurred in the control 'WEBImgEdit'.
The error detail in the method 'onProcessReceivedData' is as follows:

The client license must be the same of the web service.
Client License key: 'Company' is different from the server license key.Source value: IMAGE COMPONENTS - Destination

Make sure that your application was compiled on the licensed developer machine, and the licenses.licx file exists on your main project as an embedded resource.

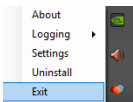
The scan doesn't start after installing the ImgScan client service.



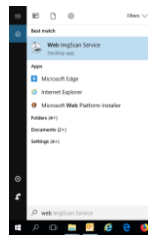
If the installation was started by double clicking the *.msi file, the scan service was launched with the SYSTEM user account context.

Choose from the following procedures:

1 – Manual restart.



Exit the ImgScan client service.

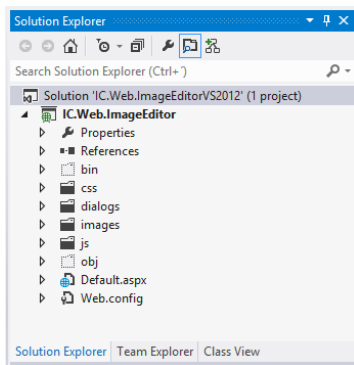


Search and run the Web ImgScan service.

2 - Restart the computer.

Running the Demo Projects

After a successful installation open and run the demo project. Go to the start menu/All Programs/Image Components/Web, and click in one of the '*.sln' Shortcut.



Demo Solution

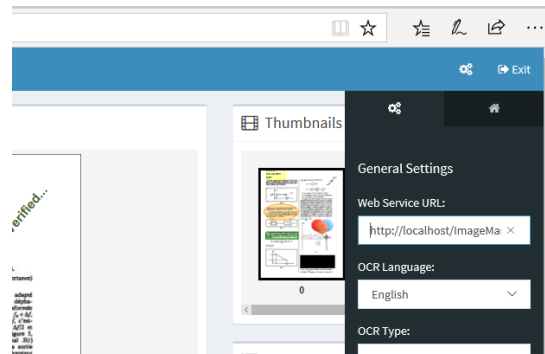
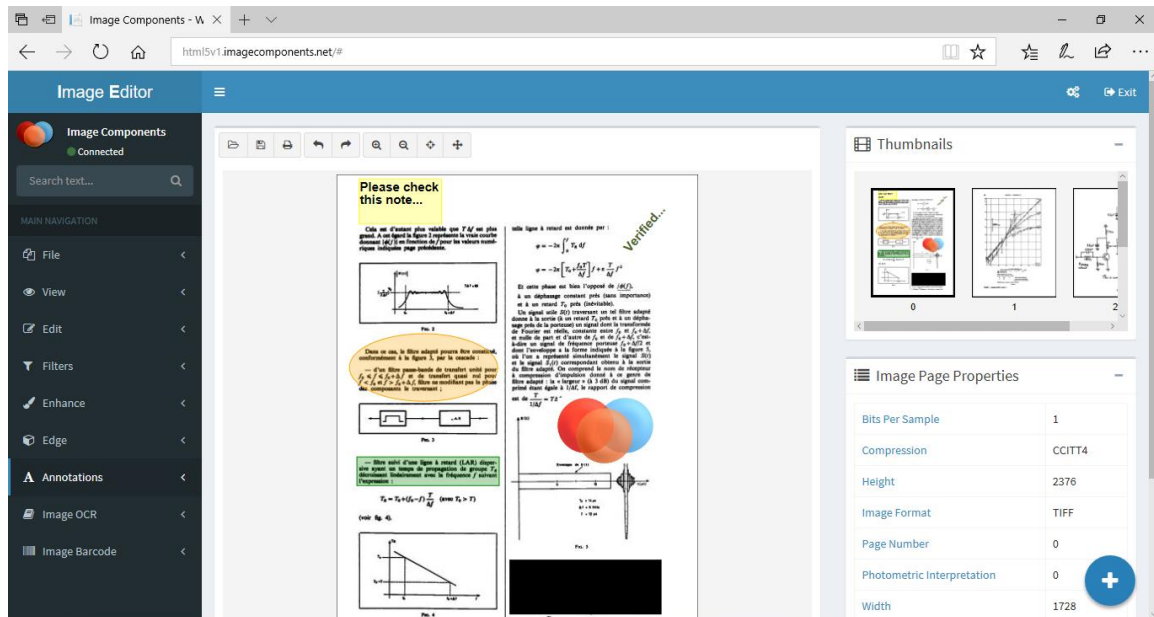


Image Manager Web Service Configuration

Paste the URL of the previous installed image manager web service into the text box dialog and press the 'OK' button. The demo application should be ready to use.



Demo ASP.NET/HTML5/JavaScript Application

In the installation package there are also Visual Studio 2010/2017 projects in visual basic and C# language. The demo license will not end at a certain time. Just try as long as you like.