Image Components for Blazor

Installation Guide

Contents

Introduction

Image Components for Blazor SDK is a dot net based document image viewing control, which can be easily integrated into your own applications.
The Blazor 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 Blazor Viewer.
Developed using CSharp, a content server and additional technologies, Image Components for Blazor 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.

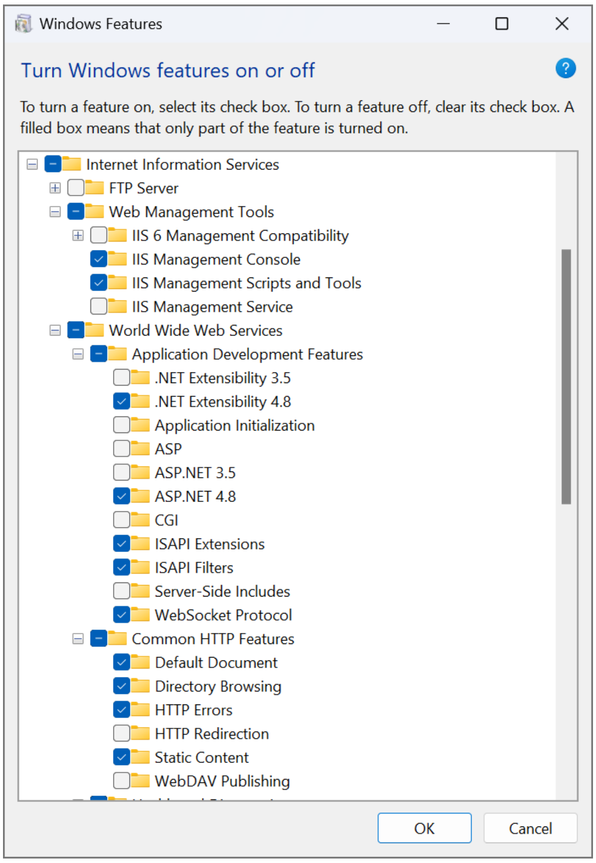
Image Components for Blazor is in constant development, other imaging features will be available soon.

Requirements

- Windows 10 or higher, 32 bit or 64 bit architecture.
- Visual Studio 2022 and above
- Some experience in Blazor development.
- Some experience in Web applications deployment.

Installation

Requirements before running the SDK installation wizard:



Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ■ Internet Information Services
  - □ FTP Server
  - ■ Web Management Tools
    - □ IIS 6 Management Compatibility
    - ☑ IIS Management Console
    - ☑ IIS Management Scripts and Tools
    - □ IIS Management Service
  - ■ World Wide Web Services
    - ■ Application Development Features
      - □ .NET Extensibility 3.5
      - ☑ .NET Extensibility 4.8
      - □ Application Initialization
      - □ ASP
      - □ ASP.NET 3.5
      - ☑ ASP.NET 4.8
      - □ CGI
      - ☑ ISAPI Extensions
      - ☑ ISAPI Filters
      - □ Server-Side Includes
      - ☑ WebSocket Protocol
    - ■ Common HTTP Features
      - ☑ Default Document
      - ☑ Directory Browsing
      - ☑ HTTP Errors
      - □ HTTP Redirection
      - ☑ Static Content
      - □ WebDAV Publishing

Enable Internet Information Services

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

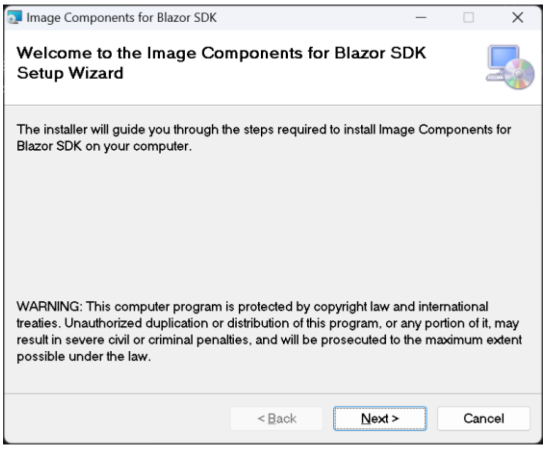


Image Components for Blazor SDK

Welcome to the Image Components for Blazor SDK Setup Wizard

The installer will guide you through the steps required to install Image Components for Blazor SDK on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

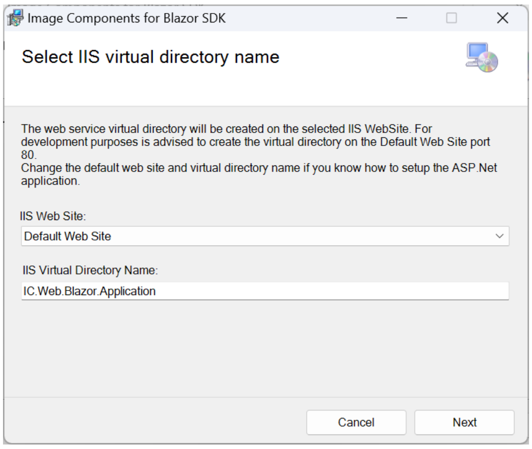


Image Components for Blazor SDK

Select IIS virtual directory name

The web service virtual directory will be created on the selected IIS WebSite. For development purposes is advised to create the virtual directory on the Default Web Site port 80.
Change the default web site and virtual directory name if you know how to setup the ASP.Net application.

IIS Web Site: Default Web Site

IIS Virtual Directory Name: IC.Web.Blazor.Application

Welcome Screen

Installation Address *

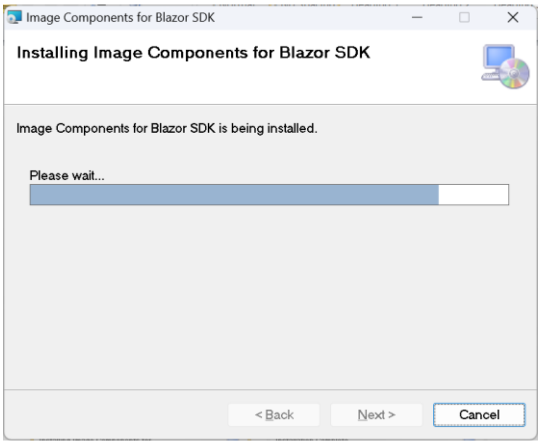


Image Components for Blazor SDK

Installing Image Components for Blazor SDK

Image Components for Blazor SDK is being installed.

Please wait...

Installing Image Components SDK

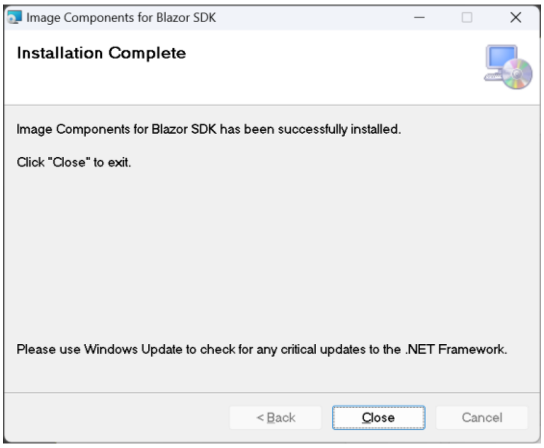


Image Components for Blazor SDK

Installation Complete

Image Components for Blazor SDK has been successfully installed.

Click "Close" to exit.

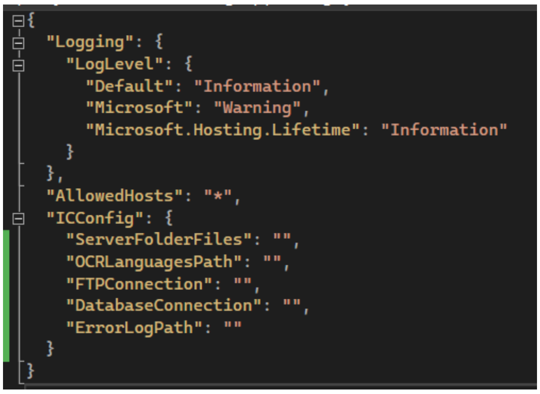
Please use Windows Update to check for any critical updates to the .NET Framework.

Installation Complete

Setting up the appsettings.json configuration

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

- Open the appsettings.json file.



```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "ICConfig": {
    "ServerFolderFiles": "",
    "OCRLanguagesPath": "",
    "FTPConnection": "",
    "DatabaseConnection": "",
    "ErrorLogPath": ""
  }
}
```

There are 4 keys in the <ICConfig> node that can be configured:

- ServerFolderFiles.
- OCRLanguagesPath.
- FTPConnection.
- DatabaseConnection.
- ErrorLogPath.

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’.

FTPConnection:
The FTP connection configuration.
Ex: `Url=MyFTPServer; Uid=MyUserName;Pwd=MyPassword;Domain=;Anonymous=false;Timeout=10000;`

DatabaseConnection:
The database connection configuration.
Ex: `ServerType=MSSQL;Server=MyServer; Database=MyDatabase;Uid=MyUserName;Pwd=MyPassword;`

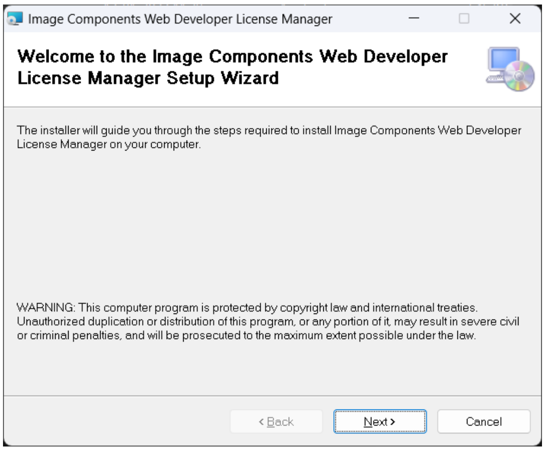
Read and write permissions must be given to the specified folders.

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

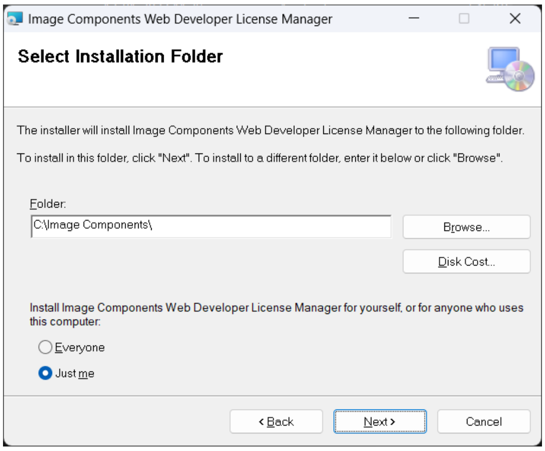


Welcome to the Image Components Web Developer License Manager Setup Wizard

The installer will guide you through the steps required to install Image Components Web Developer License Manager on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Welcome Screen



Select Installation Folder

The installer will install Image Components Web Developer License Manager to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

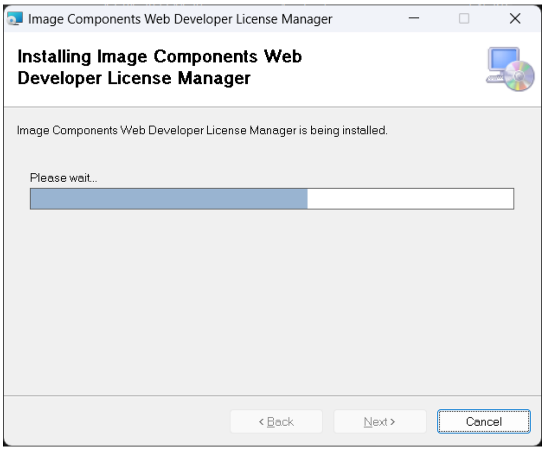
Folder: C:\Image Components\

Install Image Components Web Developer License Manager for yourself, or for anyone who uses this computer:

○ Everyone

● Just me

Installation Path

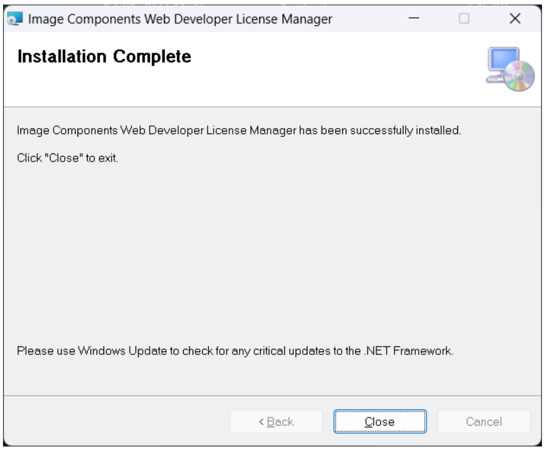


Installing Image Components Web Developer License Manager

Image Components Web Developer License Manager is being installed.

Please wait...

Installation Progress



Installation Complete

Image Components Web Developer License Manager has been successfully installed.

Click "Close" to exit.

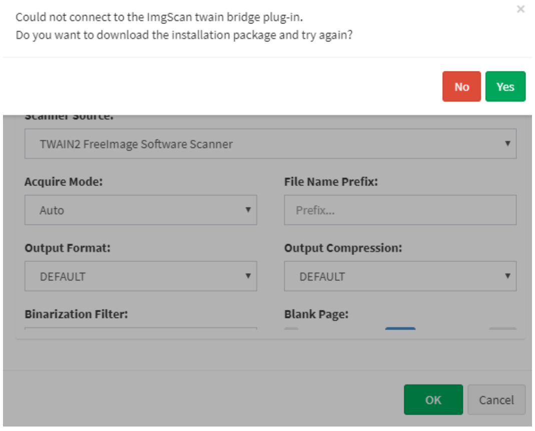
Please use Windows Update to check for any critical updates to the .NET Framework.

Installation Complete

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.



Could not connect to the ImgScan twain bridge plug-in.
Do you want to download the installation package and try again?

Scanner Source: TWAIN2 FreeImage Software Scanner

Acquire Mode: Auto

File Name Prefix: Prefix...

Output Format: DEFAULT

Output Compression: DEFAULT

Binarization Filter:

Blank Page:

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**.

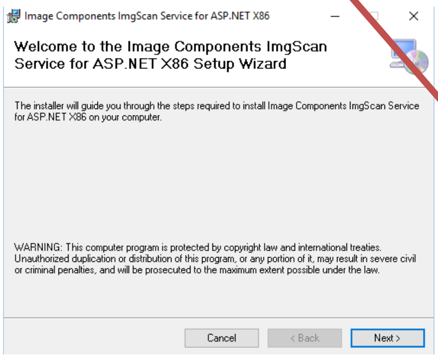


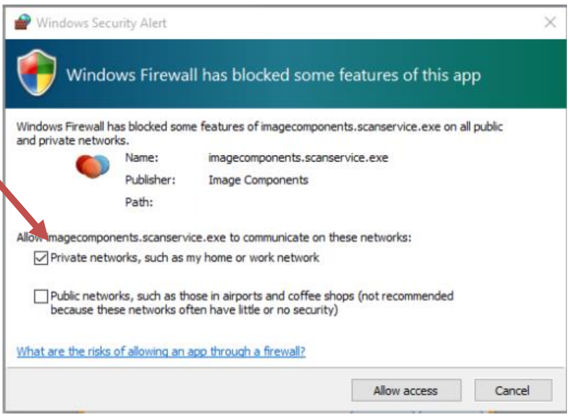
Image Components ImgScan Service for ASP.NET X86

Welcome to the Image Components ImgScan Service for ASP.NET X86 Setup Wizard

The installer will guide you through the steps required to install Image Components ImgScan Service for ASP.NET X86 on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Installation progress



Windows Security Alert

Windows Firewall has blocked some features of this app

Windows Firewall has blocked some features of imagecomponents.scanservice.exe on all public and private networks.

Name: imagecomponents.scanservice.exe

Publisher: Image Components

Path:

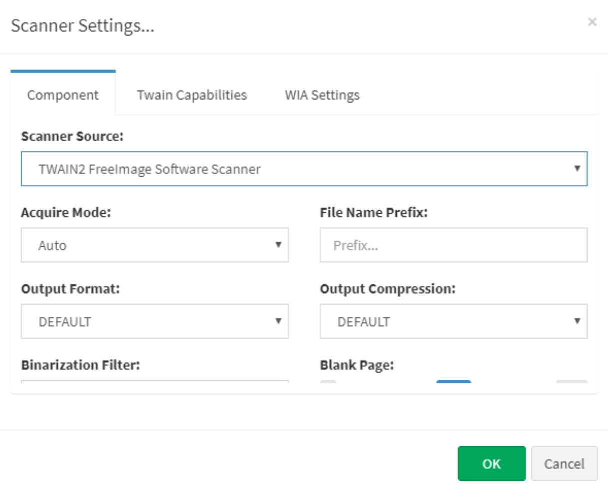
Allow imagecomponents.scanservice.exe to communicate on these networks:

☑ Private networks, such as my home or work network

☐ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing an app through a firewall?

Firewall warning dialog



Scanner Source: TWAIN2 FreeImage Software Scanner

Acquire Mode: Auto

File Name Prefix: Prefix...

Output Format: DEFAULT

Output Compression: DEFAULT

Binarization Filter:

Blank Page:

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**.

Manual 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:

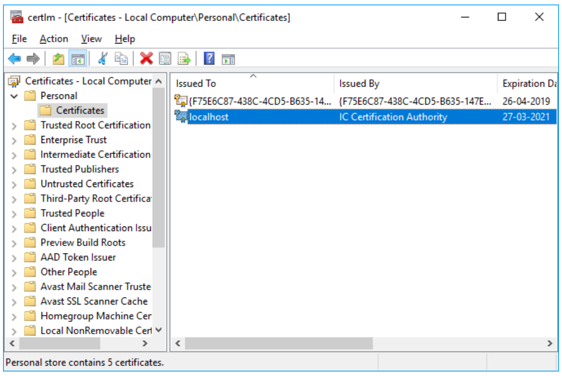
```
$params = @{
  DnsName = "IC Certification Authority"
  KeyLength = 2048
  KeyAlgorithm = 'RSA'
  HashAlgorithm = 'SHA256'
  KeyExportPolicy = 'Exportable'
  NotAfter = (Get-Date).AddYears(5)
  CertStoreLocation = 'Cert:\LocalMachine\My'
  KeyUsage = 'CertSign','CRLSign' #fixes invalid cert error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

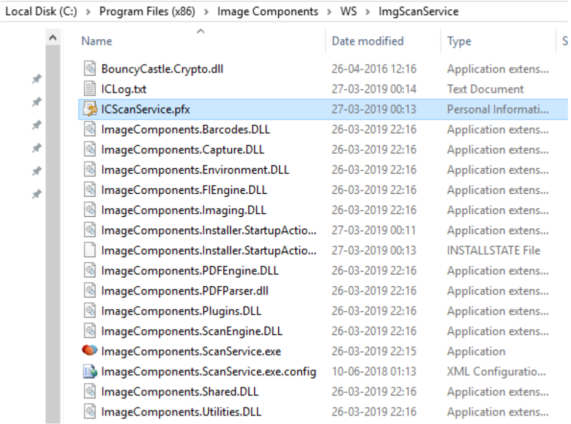
Press F5 and run the script. A valid export folder (Ex: `E:\certs\`) must exist. The *ICScanService.pfx* should have been exported to the *E:\certs\* folder with the *MyPassword* password. Add the certificate to the ImgScan service as previously shown.



Issued To	Issued By	Expiration Da
{F75E6C87-438C-4CD5-B635-14...	{F75E6C87-438C-4CD5-B635-147E...	26-04-2019
localhost	IC Certification Authority	27-03-2021

Personal store contains 5 certificates.

Manually or automatically created certificate.

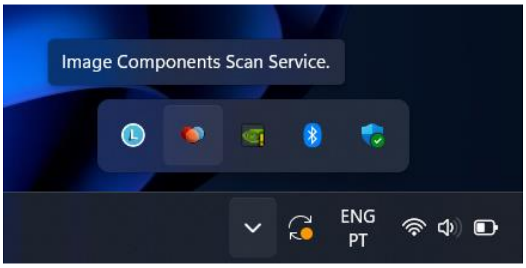

Local Disk (C:) › Program Files (x86) › Image Components › WS › ImgScanService

Name	Date modified	Type
BouncyCastle.Crypto.dll	26-04-2016 12:16	Application extens...
ICLog.txt	27-03-2019 00:14	Text Document
ICScanService.pfx	27-03-2019 00:13	Personal Informati...
ImageComponents.Barcodes.DLL	26-03-2019 22:16	Application extens...
ImageComponents.Capture.DLL	26-03-2019 22:16	Application extens...
ImageComponents.Environment.DLL	26-03-2019 22:16	Application extens...
ImageComponents.FIEngine.DLL	26-03-2019 22:16	Application extens...
ImageComponents.Imaging.DLL	26-03-2019 22:16	Application extens...
ImageComponents.Installer.StartupActio...	27-03-2019 00:11	Application extens...
ImageComponents.Installer.StartupActio...	27-03-2019 00:13	INSTALLSTATE File
ImageComponents.PDFEngine.DLL	26-03-2019 22:16	Application extens...
ImageComponents.PDFParser.dll	26-03-2019 22:16	Application extens...
ImageComponents.Plugins.DLL	26-03-2019 22:16	Application extens...
ImageComponents.ScanEngine.DLL	26-03-2019 22:16	Application extens...
ImageComponents.ScanService.exe	26-03-2019 22:15	Application
ImageComponents.ScanService.exe.config	10-06-2018 01:13	XML Configuratio...
ImageComponents.Shared.DLL	26-03-2019 22:16	Application extens...
ImageComponents.Utilities.DLL	26-03-2019 22:16	Application extens...

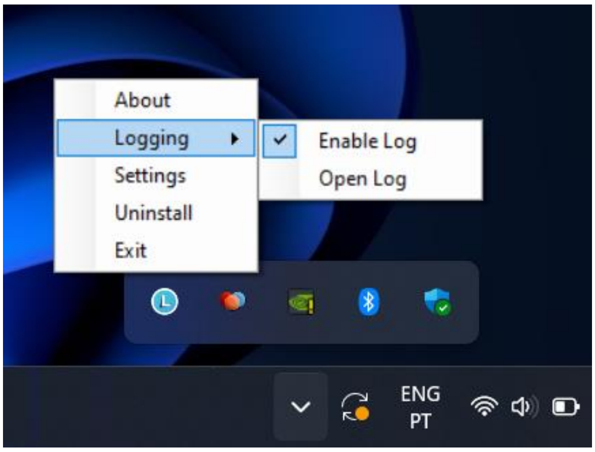
Certificate created by the installation package.

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

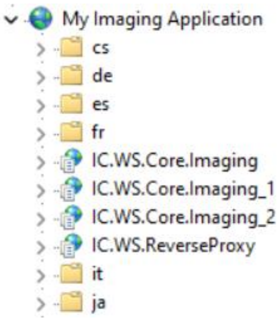
You can scale your imaging application by creating a pool of services that can respond to an increase number of users.
This can be done with a web gateway, creating a unified entry point, with the intent of distributing the unique entry point request to a listening pool of services.

Here are two great open source examples:

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

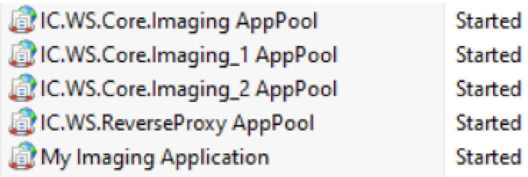
We do prefer using YARP, but this is only personal preference, as both works great.
The only disadvantage that we found was on Ocelot, as it doesn't support virtual directories or virtual web applications (web Applications inside web applications) supported by IIS.

YARP 2.0 configuration example



```
My Imaging Application
  cs
  de
  es
  fr
  IC.WS.Core.Imaging
  IC.WS.Core.Imaging_1
  IC.WS.Core.Imaging_2
  IC.WS.ReverseProxy
  it
  ja
```

Web application with 3 imaging services and the YARP reverse proxy



IC.WS.Core.Imaging AppPool	Started
IC.WS.Core.Imaging_1 AppPool	Started
IC.WS.Core.Imaging_2 AppPool	Started
IC.WS.ReverseProxy AppPool	Started
My Imaging Application	Started

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.WS.Core.Imaging/"
          },
          "destination2": {
            "Address": "http://localhost/IC.WS.Core.Imaging_1/"
          },
          "destination3": {
            "Address": "http://localhost/IC.WS.Core.Imaging_2/"
          }
        }
      }
    }
  }
}
```

Load balancing must be enabled, with session affinity setup.

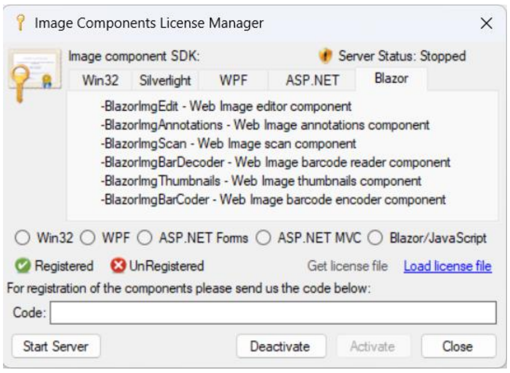
What changes in the components configuration

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.WS.Core.Imaging/api/
- New Value: http://localhost/IC.WS.ReverseProxy/Imaging/api/

Developer Licensing

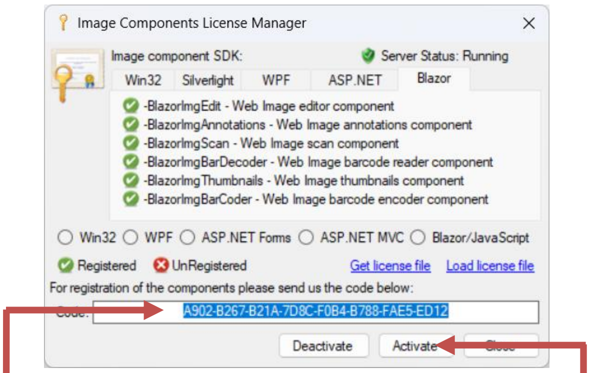
The Image Components for Blazor installation includes a License Manager application. This manager allow you to check what components are licensed or not, and your developer machine key code.
Before starting using the components, you must start the license server.
- On the Task Bar press the Start button/All Programs/Image Components/Licensing/Web License Manager.



-BlazorImgEdit - Web Image editor component
-BlazorImgAnnotations - Web Image annotations component
-BlazorImgScan - Web Image scan component
-BlazorImgBarDecoder - Web Image barcode reader component
-BlazorImg Thumbnails - Web Image thumbnails component
-BlazorImgBarCoder - Web Image barcode encoder component

For registration of the components please send us the code below:

License Server Before Start



✓ -BlazorImgEdit - Web Image editor component
✓ -BlazorImgAnnotations - Web Image annotations component
✓ -BlazorImgScan - Web Image scan component
✓ -BlazorImgBarDecoder - Web Image barcode reader component
✓ -BlazorImg Thumbnails - Web Image thumbnails component
✓ -BlazorImgBarCoder - Web Image barcode encoder component

For registration of the components please send us the code below:

Code: A902-B267-B21A-7D8C-F0B4-B788-FAE5-ED12

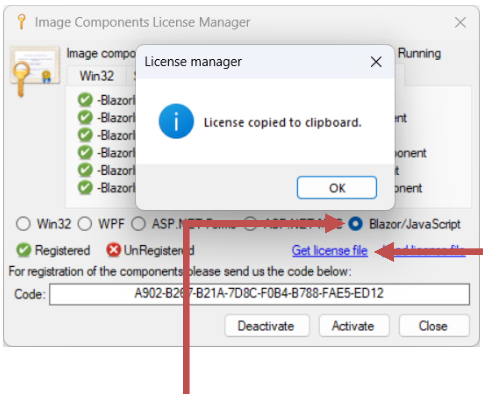
License Server Started

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 CORE, HTML, and Others:

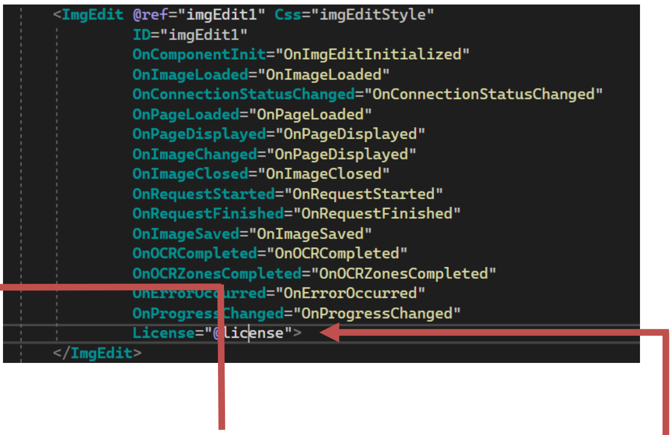
You can also add the license value manually in an ASP.NET CORE, HTML or Other project types by doing the following procedure:



License copied to clipboard.

For registration of the components please send us the code below:

Code: A902-B267-B21A-7D8C-F0B4-B788-FAE5-ED12



```
<ImgEdit @ref="imgEdit1" Css="imgEditStyle"
         ID="imgEdit1"
         OnComponentInit="OnImgEditInitialized"
         OnImageLoaded="OnImageLoaded"
         OnConnectionStatusChanged="OnConnectionStatusChanged"
         OnPageLoaded="OnPageLoaded"
         OnPageDisplayed="OnPageDisplayed"
         OnImageChanged="OnPageDisplayed"
         OnImageClosed="OnImageClosed"
         OnRequestStarted="OnRequestStarted"
         OnRequestFinished="OnRequestFinished"
         OnImageSaved="OnImageSaved"
         OnOCRCompleted="OnOCRCompleted"
         OnOCRZonesCompleted="OnOCRZonesCompleted"
         OnErrorOccurred="OnErrorOccurred"
         OnProgressChanged="OnProgressChanged"
         License="@License">
</ImgEdit>
```

Check the **Blazor/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 **License** 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.

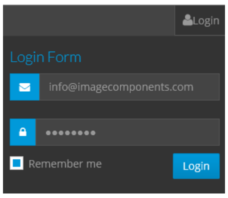
Purchase steps:

- 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

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.



Login

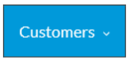
Login Form

info@imagecomponents.com

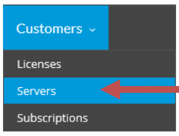
••••••••

□ Remember me

Login



Customers ⌄



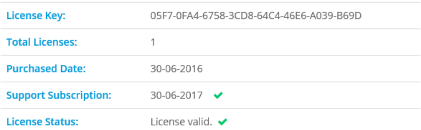
Customers ⌄

Licenses

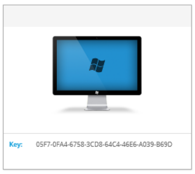
Servers

Subscriptions

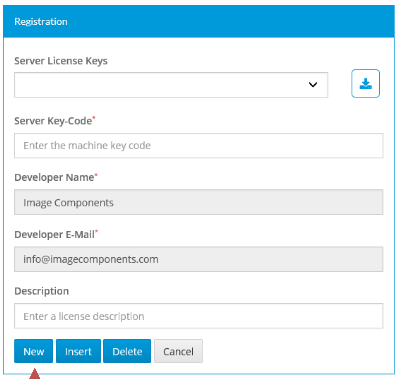
Press the 'Servers' link to manage the deployment's registrations.



License Key:	05F7-0FA4-6758-3CD8-64C4-46E6-A039-B69D
Total Licenses:	1
Purchased Date:	30-06-2016
Support Subscription:	30-06-2017 ✓
License Status:	License valid. ✓



Key: 05F7-0FA4-6758-3CD8-64C4-46E6-A039-B69D



Registration

Server License Keys

Server Key-Code*

Enter the machine key code

Developer Name*

Image Components

Developer E-Mail*

info@imagecomponents.com

Description

Enter a license description

New Insert Delete Cancel

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

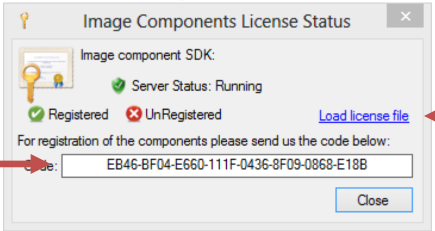


Image Components License Status

Image component SDK:

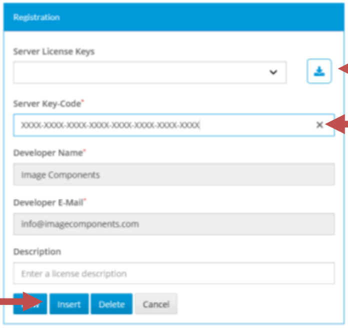
Server Status: Running

Registered UnRegistered Load license file

For registration of the components please send us the code below:

Code: EB46-BF04-E660-111F-0436-8F09-0868-E18B

Close



Registration

Server License Keys

Server Key-Code*

XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX

Developer Name*

Image Components

Developer E-Mail*

info@imagecomponents.com

Description

Enter a license description

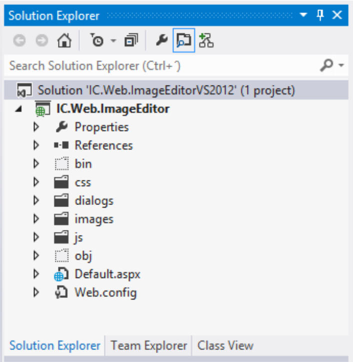
New Insert Delete Cancel

Server license manager application.

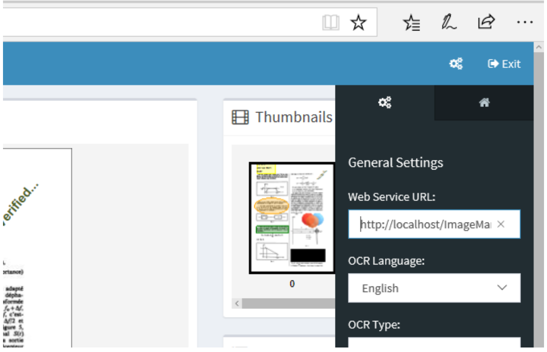
- Start the deployed server license manager application.
- Copy the **key code** from the license manager application to the **server key-code** text box.
- Press the **Insert** button.
- Select and download the new inserted server key-code.
- Load the downloaded license file using the license manager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

Running the Demo Projects

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



Demo Solution



General Settings

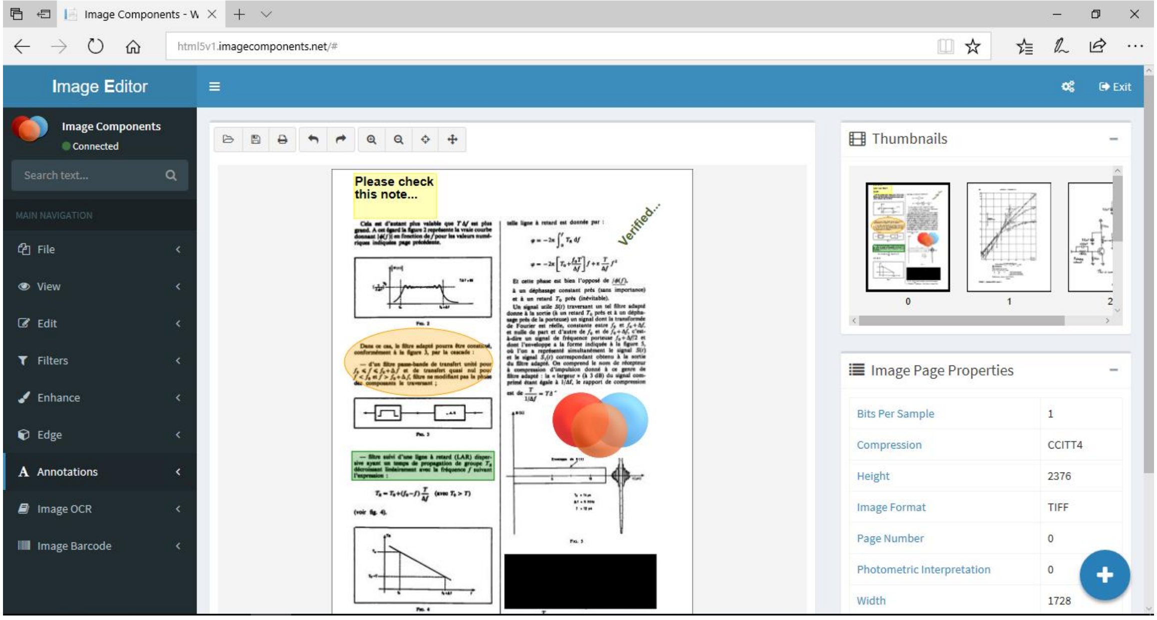
Web Service URL: http://localhost/ImageMa

OCR Language: English

OCR Type:

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.



Bits Per Sample	1
Compression	CCITT4
Height	2376
Image Format	TIFF
Page Number	0
Photometric Interpretation	0
Width	1728

Demo ASP.NET/HTML5/JavaScript Application

The demo license will not end at a certain time. Just try as long as you like.