

Table of Contents

Overview	1
Installation from Visual Studio	1
Obtain Server IP Address and DNS Name	4
Installation from Prebuilt Package	5
Create Storage Account.....	5
Create Cloud Service	6
Obtain Server IP Address and DNS Name	8
Server Configuration	8
Change Configuration after Installation	9
Running the Demo Applications	9
Using the Installed Server	10
Uninstallation	12

Overview

EVO PDF Server can be installed in an Azure Cloud Service Worker Role. After installation the server can be accessed by the assigned IP address or DNS name.

There are two options to install the EVO PDF Server in an Azure Cloud Service Worker Role: from the Visual Studio solution or from the prebuilt package. The prebuilt package is created from the Visual Studio solution and can be used in case you don't have the Azure development tools installed on your machine.

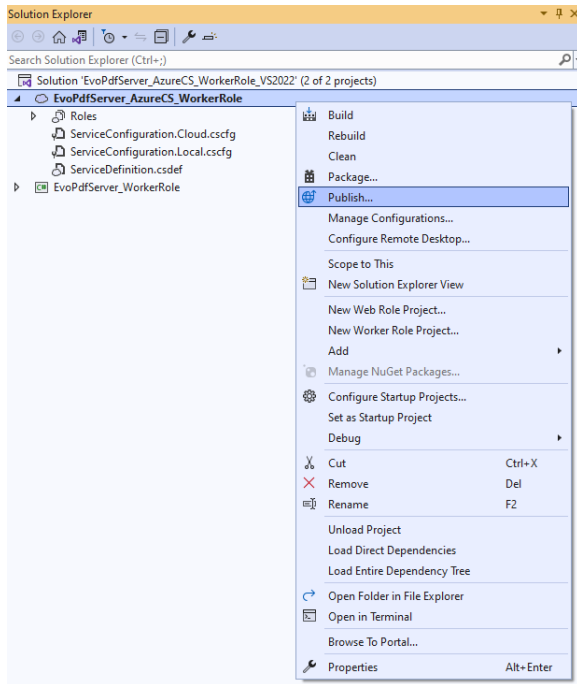
Installation from Visual Studio is guided by the Visual Studio publish wizard and it is the preferred method of installation in general.

Installation from Visual Studio

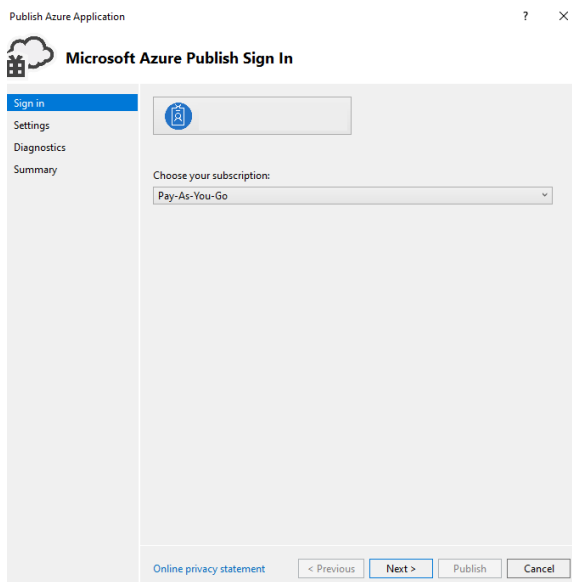
The server can be easily installed using the Visual Studio solution for Azure Cloud Service Worker Role from 'Src' subfolder.

Before publishing the service, you can set various configuration options like a service password in the 'Src\EvoPdfServer_AzureCS_WorkerRole\ServiceConfiguration.Cloud.cscfg' configuration file. The description of each configuration option can be found in a separate section of this document.

To publish the Cloud Service from Visual Studio, open the 'EvoPdfServer_AzureCS_WorkerRole.sln' solution in Visual Studio, right click the 'EvoPdfServer_AzureCS_WorkerRole' project and choose the 'Publish' option from the contextual menu.



In the first panel 'Sign In' of the publish wizard choose your Azure subscription and click 'Next'.



If you don't have an existing cloud service in your Azure account, then the 'Create Cloud Service (extended support)' panel will automatically appear and ask you to enter a name for the cloud service and to select a location for deployment.

If you already have an existing cloud service in your Azure account, then click the 'Create new' button to open the 'Create Cloud Service (extended support)' panel below.

Create Cloud Service (extended support) X

Name: ✓

DNS name: ✓

Location:

Resource group:

[Create new...](#)

OK Cancel

The cloud service name and the DNS name you enter in this panel must not contain special characters.

For the 'Resource group' you can choose 'Create new...' to enter a name for the new cloud service resource group.

Click 'OK' button to close the 'Create Cloud Service (extended support)' panel and return to the 'Common Settings' panel updated with updated settings.

Publish Azure Application ? X

Microsoft Azure Publish Settings

Sign in
Settings
Diagnostics
Summary

Common Settings | Advanced Settings

Cloud Service (extended support):

[Create new...](#)

Environment:

Build configuration:

Service configuration:

Storage account:

[Create new...](#)

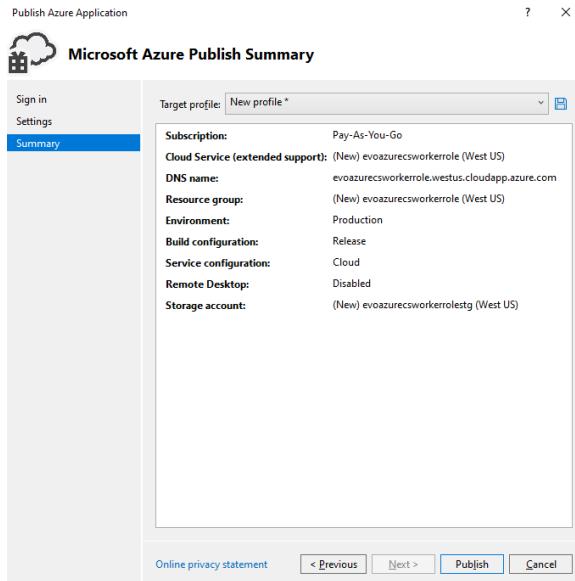
Key vault:

☐ Enable Remote Desktop for all roles [Settings...](#)

[Online privacy statement](#) < Previous Next > Publish Cancel

Leave the default value for the other options. Click 'Next' button to see the summary of your options and then click the 'Publish' button.

Install EVO PDF Server in an Azure Cloud Service Worker Role



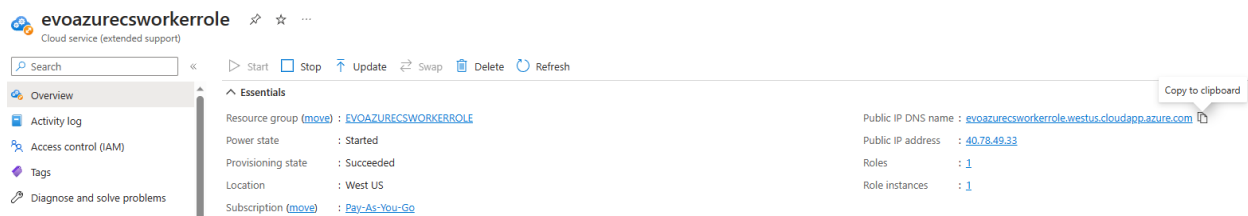
You'll be able to see the evolution of the deployment process in Visual Studio.

Obtain Server IP Address and DNS Name

When the deployment is complete you can login into your account of Microsoft Azure Portal to obtain the public IP address and the DNS name you can use to connect the client applications to server.

In Azure Portal you can find the Cloud Service you created if you click the 'All resources' item from the menu of the Azure Management Portal.

Click the link to 'Cloud service (extended support)' resource from the list of resources to open the cloud service overview panel.



In this panel you can see the public IP address of server and the assigned DNS name.

You can use the obtained IP address or the DNS name to connect the client applications to server.

Installation from Prebuilt Package

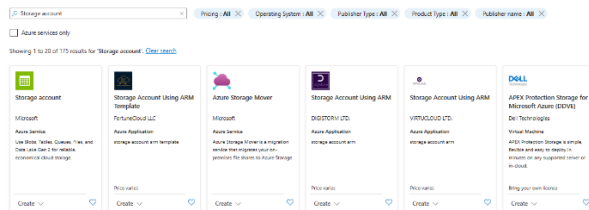
The prebuilt package for Azure Cloud Service Worker Role there is in 'Bin' folder.

Before uploading the package, you can edit the 'Bin\ServiceConfiguration.Cloud.cscfg' file to set various options like a service password or the maximum number of conversions running at the same time.

To install the package, login into your Microsoft Azure Portal account.

Create Storage Account

First you have to create a storage account for your cloud service. Click the 'Create a resource' menu item from the menu of the Azure Management Portal and search for 'Storage account' in the search text box at the top.



Click on 'Storage account' resource type from list and then click the 'Create' button to open the 'Create a storage account' page.

Home > Create a resource > Marketplace > Storage account >

Create a storage account

Basics | Advanced | Networking | Data protection | Encryption | Tags | Review

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * Pay-As-You-Go

Resource group * (New) evoaazureworkerrole2
[Create new](#)

Instance details

Storage account name ⓘ * evoaazureworkerrole2

Region ⓘ * (US) West US
[Deploy to an edge zone](#)

Performance ⓘ *
☒ **Standard:** Recommended for most scenarios (general-purpose v2 account)
☐ **Premium:** Recommended for scenarios that require low latency.

Redundancy ⓘ * Geo-redundant storage (GRS)
☒ Make read access to data available in the event of regional unavailability.

[Review](#) [< Previous](#) [Next: Advanced >](#)

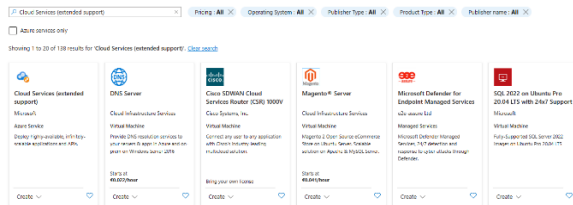
Choose your Azure subscription, choose to create a new Resource Group and enter a name for the Resource Group. You can use the same Resource group later when you deploy the cloud service.

Enter a name for the Storage Account and choose a region for the storage account location or use the suggested region. You can leave the default values for the other options.

Click 'Review' and in the next page click 'Create' button after validation to create the storage account. Wait until the storage account creation process is finished.

Create Cloud Service

The next step is to create the Cloud Service (extended support). For this click the 'Create a resource' menu item from the menu of the Azure Management Portal and search for 'Cloud Services (extended support)' in the search text box at the top.



Click on “Cloud Services (extended support)” resource type from list and then click the 'Create' button to open the ‘Create a cloud service (extended support)’ page.

Home > Create a resource > Marketplace > Cloud Services (extended support) >

Create a cloud service (extended support)

applications and services. It manages provisioning, load balancing, and health monitoring.
[Learn more about Azure Cloud Services \(extended support\)](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group *
[Create new](#)

Instance details

Cloud service name *
 ⓘ The DNS name of the cloud service is specified by the DNS name label of the public IP address and can be modified in the public IP section on the configuration tab.

Region *

Cloud service configuration, package, and service definition

Include your cloud service configuration (.cscfg), application package (.cspkg, .zip), and service definition (.csdef) files. We will store them in your selected storage account and use them to configure your cloud service. If you are coming from Cloud Services (classic), you will need to make some changes to your service configuration (.cscfg) and service definition (.csdef) files as a pre-requisite to deploy a cloud service (extended support). [Learn more about these pre-requisites](#)

Package/configuration/service definition location
☒ From local
☐ From blob

Storage account *

Upload a package (.cspkg, .zip) *

Upload a configuration (.cscfg) *

Upload a service definition (.csdef) *

Select the Azure Subscription and the Resource Group you created with the Storage Account at the previous step.

Enter the 'Cloud service name' which you can also use later as DNS name for the cloud service.

Choose a Region for the cloud service, usually the same as the storage account region.

For 'Package/configuration/service definition location' choose 'From local' option.

Choose the Storage account you created at the previous step.

Upload the prebuilt package and the configuration files by clicking the corresponding 'Browse for file' fields.

Choose the package file EvoPdfServer_AzureCS_WorkerRole.cspkg', configuration file ServiceConfiguration.Cloud.cscfg' and service definition file 'ServiceDefinition.csdef' from 'Bin' local folder.

Wait until the packages are uploaded. You can check the upload progress in the Notifications panel.

Click the 'Next : Configuration' button when packages upload completed.

[Home](#) > [Create a resource](#) > [Marketplace](#) > [Cloud Services \(extended support\)](#) >

Create a cloud service (extended support) ...

Basics **Configuration** Tags Review + create

Define network connectivity, security, and deployment configuration for your cloud service.
[Learn more about the configuration of cloud services](#) >

Configure virtual network

Azure virtual networks enable many types of Azure resources to securely communicate with each other, the internet, and on-premises networks. [Learn more about virtual networks](#) >

Virtual network ☐ (New) evopdfserver_azures_workerroleVNet (10.0.0.0/16)
[Edit address spaces and subnets](#)

Subnet ☐ (New) WebTier (10.0.0.0/24)

Configure public/internal load balancer

Public IP address * ☐ (New) evoazureworkerrole2-ip
[Create new](#)

Configure deployment

You can configure your cloud service to start after deployment and pair it with an existing cloud service for easier management of your applications during deployment cycles. [Learn more about swapping cloud service deployments](#) >

Swappable cloud service ☐ None

Start cloud service ☒

[Review + create](#) < Previous Next : Tags >

In the 'Public IP address' section click the 'Create new' button to open a popup, use the suggested 'Public IP address name' and for 'DNS name label' you can enter the cloud service name you have also used in the previous panel or you can use a different DNS name. Click 'OK' button to close the popup.

Configure public/internal load balancer

Public IP address * ☐ (New) evoazureworkerrole2-ip
[Create new](#)

Configure deployment

You can configure your cloud service to start after deployment and pair it with an existing cloud service for easier management of your applications during deployment cycles. [Learn more about swapping cloud service deployments](#) >

Swappable cloud service ☐ None

Start cloud service ☒

Create a public IP

Public IP address name * ☐ evoazureworkerrole2-ip

DNS name label ☐ evoazureworkerrole2 ✓

.westus.cloudapp.azure.com

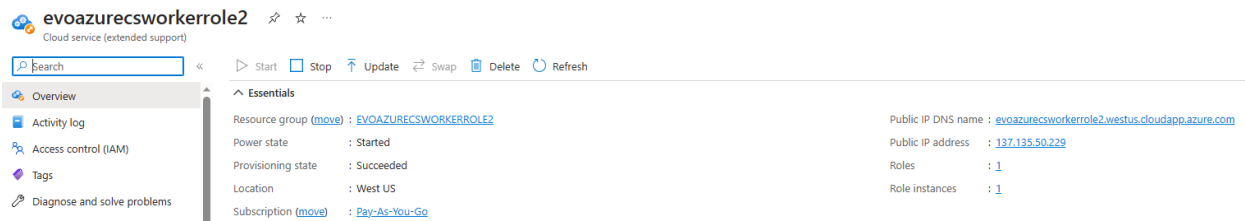
[OK](#) [Cancel](#)

Click the 'Review + create' button in the 'Configuration' page and then the 'Create' button after validation to start the cloud service deployment.

Wait a few minutes until the deployment is complete.

Obtain Server IP Address and DNS Name

When the deployment is complete click the 'Go to resource' button to open the Cloud Service 'Overview' page.



In the cloud service panel, you can see the public IP address and the public DNS name you can further use to connect the client applications to server.

Server Configuration

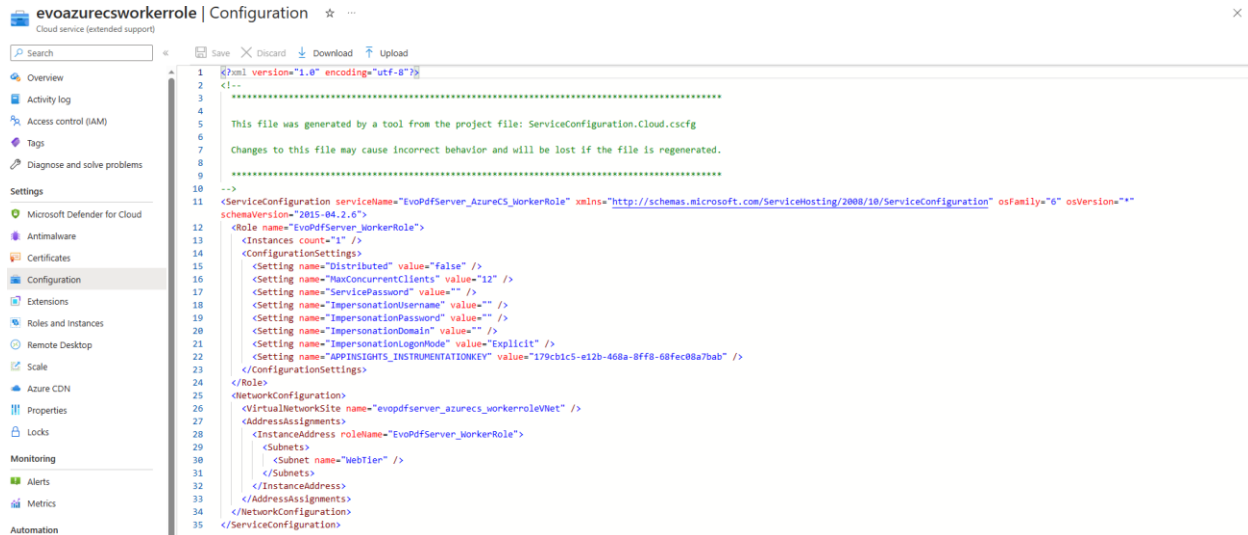
EVO PDF Server offers various configuration options you can set before Cloud Service installation in the 'Src\EvoPdfServer_AzureCS_WorkerRole\ServiceConfiguration.Cloud.cscfg' configuration file when you install from Visual Studio or in 'Bin\ServiceConfiguration.Cloud.cscfg' configuration file when you install from prebuilt package.

The possible configuration options and their default values are described below.

- **ServicePassword** option offers the possibility to protect the service with a password to prevent unauthorized access. By default, this option is not set.
- **MaxConcurrentClients** option represents the maximum number of request the server can process in parallel. If the maximum number of parallel requests is reached, the subsequent requests are queued until a slot becomes available. The default value for this option is 12.
- **Distributed** option offers the possibility to execute each request in its own process when its value is true. By default, this option is false.
- Impersonation options consisting in **ImpersonationUsername**, **ImpersonationPassword**, **ImpersonationDomain** and **ImpersonationLogonMode** options can be used to run the service under a specific user account on the hosting Windows machine. By default, they are not set.

Change Configuration after Installation

It is also possible to change the configuration options in Azure Portal after server installation. In the Cloud Service resource page click the 'Configuration' link from the left side. A page where you can edit the configuration file will open.



Edit the configuration file and for example set a value for example 'ServicePassword' option. Click the 'Save' button at the top to make the update. Wait until the operation finishes.

You have to restart the server to enable the changes. For this go back to 'Overview' page of the Cloud Service and at the top of the page click the 'Stop' button and then 'Start' button when this button gets enabled.



After that, in the client application you'll have to provide the password to be able to execute various server functions.

Running the Demo Applications

After EVO PDF Server installation you are ready to start using the client libraries in your applications. You can start with our demo applications from EVO PDF Client ZIP packages which can be downloaded directly from [EVO PDF Software Downloads Page](#) and extracted in a folder.

There are separate packages for **.NET Framework** and **.NET Core** with demo application for all components and features.

- The package for .NET Core contains the product binaries, demo Visual Studio projects with full C# code for ASP.NET Core targeting .NET Core 6.0 and later versions, the library documentation in CHM format
- The package for .NET Framework contains the product binaries, demo Visual Studio projects with full C# code for ASP.NET Web Forms and ASP.NET MVC targeting .NET Framework 4 and later versions, the library documentation in CHM format

There are separate demo projects for each major component of the library, including **HTML to PDF**, **Word to PDF**, **Excel to PDF**, **PDF to Text**, **PDF to Image**, **PDF to HTML** and **PDF Images Extractor** corresponding to the tools from **EVO PDF Toolkit** bundle.

The demo applications allow you to enter the server IP address and the port number if you installed the server as a TCP/IP service or the server URL if you have installed the server as a web service.

Using the Installed Server

EVO PDF Server was installed in an Azure Cloud Service Worker Role and it can be accessed in client applications by its IP address or DNS name.

In the demo application you can select the **Use TCP/IP** radio button and enter the IP address or the DNS name you previously obtained.

Below is a screenshot of the HTML to PDF demo client application from client package for .NET Core using the IP address of the server installed in the Azure Cloud Service Worker Role.

**EVO PDF Software**
HTML to PDF Converter
for .NET

- HTML to PDF Converter
 - Getting Started with HTML to PDF
 - HTML Content Destination and Scaling in PDF
 - Convert the Current HTML Page to PDF
 - Convert a HTML Page to PDF in Same Session
 - Merge Multiple HTML Pages into a Single PDF
 - Merge HTML with Existing PDF Documents
 - Partially Convert of a HTML Page to PDF
 - Repeat HTML Table Header and Footer in PDF
 - Headers and Footers
 - Page Breaks Control
 - Screen and Print Media Types
 - Conversion Triggering Modes
 - Hierarchical Bookmarks
 - Table of Contents
 - Live PDF Forms
 - HTML5 Features
 - HTTP GET and POST, Proxy Options
 - HTTP Headers and Cookies
 - Fonts Embedding
 - HTML Elements Location in PDF
 - HTML Elements Visibility in PDF
 - URI Links in PDF
 - Internal Links in PDF
 - File Links and Attachments
 - Text Notes in PDF
 - PDF Actions
 - HTML to PDF Elements
 - HTML to Image Elements
 - Images Quality in Generated PDF
 - Flash and Extensions Support
 - Watermarks and Stamps
 - PDF/A and PDF/X Standards
 - CMYK and GrayScale Color Spaces
 - PDF Viewer Preferences
 - PDF Security Features
- HTML to Image Converter
- PDF Creator
- PDF Editor

Getting Started with EVO HTML to PDF Converter

Convert a HTML page from an URL or a HTML String to PDF using the basic options of the EVO HTML to PDF Converter. The Full Description and a Code Sample can be accessed from the top tabs.

☒ Use TCP/IP Server IP Address: Port:

☐ Use Web Service URL:

Service Password:

☒ Convert URL or Local File ☐ Convert HTML String

HTML Page URL or Local File to Convert

HTML Viewer Options

Width: px Height: px

PDF Page Options

Size: Orientation:

Navigation Options

Timeout: sec Delay Conversion : sec

☐ Display PDF inline

Below is a screenshot of the HTML to PDF demo client application from client package for .NET Core using the DNS name of the server installed in the Azure Cloud Service Worker Role.


EVO PDF Software
HTML to PDF Converter
for .NET

HTML to PDF Converter

- Getting Started with HTML to PDF
- HTML Content Destination and Scaling in PDF
- Convert the Current HTML Page to PDF
- Convert a HTML Page to PDF in Same Session
- Merge Multiple HTML Pages into a Single PDF
- Merge HTML with Existing PDF Documents
- Partially Convert of a HTML Page to PDF
- Repeat HTML Table Header and Footer in PDF
- ▣ Headers and Footers
- ▣ Page Breaks Control
- ▣ Screen and Print Media Types
- ▣ Conversion Triggering Modes
- ▣ Hierarchical Bookmarks
- ▣ Table of Contents
- ▣ Live PDF Forms
- ▣ HTML5 Features
- ▣ HTTP GET and POST, Proxy Options
- ▣ HTTP Headers and Cookies
- ▣ Fonts Embedding
- ▣ HTML Elements Location in PDF
- ▣ HTML Elements Visibility in PDF
- ▣ URI Links in PDF
- ▣ Internal Links in PDF
- ▣ File Links and Attachments
- ▣ Text Notes in PDF
- ▣ PDF Actions
- ▣ HTML to PDF Elements
- ▣ HTML to Image Elements
- ▣ Images Quality in Generated PDF
- ▣ Flash and Extensions Support
- ▣ Watermarks and Stamps
- ▣ PDF/A and PDF/X Standards
- ▣ CMYK and GrayScale Color Spaces
- ▣ PDF Viewer Preferences
- ▣ PDF Security Features
- ▣ **HTML to Image Converter**
- ▣ **PDF Creator**
- ▣ **PDF Editor**

Getting Started with EVO HTML to PDF Converter
Convert a HTML page from an URL or a HTML String to PDF using the basic options of the EVO HTML to PDF Converter. The Full Description and a Code Sample can be accessed from the top tabs.

☒ Use TCP/IP Server IP Address: Port:
☐ Use Web Service URL:
Service Password:

☒ Convert URL or Local File ☐ Convert HTML String

HTML Page URL or Local File to Convert

HTML Viewer Options
Width: px Height: px

PDF Page Options
Size: Orientation:

Navigation Options
Timeout: sec Delay Conversion : sec

☐ Display PDF inline

Uninstallation

You can remove the Azure Cloud Service from your Azure account management portal.