Table of Contents

Chapter 1: Overview ................................................................................................................................. 5
  Requirements for Data Authentication Systems .................................................................................. 5
  Introduction to DocuSign SA ................................................................................................................ 5
    Environments Supported by DocuSign SA ......................................................................................... 6
    Applications that Work with DocuSign SA ....................................................................................... 6
    DocuSign SA Components ............................................................................................................... 7
  DocuSign SA Guides .............................................................................................................................. 7
  Intended Audience ............................................................................................................................... 8
  Organization of this Guide .................................................................................................................... 8

Chapter 2: Introduction to DocuSign SA Web App .................................................................................. 9
  About DocuSign SA Web App .............................................................................................................. 9
    Advantages of DocuSign SA Web App ............................................................................................ 10
  Integration with Document Management Solutions ........................................................................... 10
  Integration with External Identity Providers ....................................................................................... 10

Chapter 3: Installing and Deploying the DocuSign SA Web App ......................................................... 11
  DocuSign SA Web App Installation ..................................................................................................... 11
  Installing and Configuring the DocuSign SA Client ....................................................................... 11
  Installing the DocuSign SA Web App Component ....................................................................... 16
  Configuring the Mail Server ........................................................................................................... 20
  Configuring IIS for Kerberos Ticketing ............................................................................................ 21
  Configuration Required if PDF Forms are Used ........................................................................... 22
  Uninstalling the DocuSign SA Web App ............................................................................................ 33

Chapter 4: Using DocuSign SA Web App for Signing and Validating Documents .......................... 34
  Logging in to the DocuSign SA Web App ......................................................................................... 34
  Selecting a Document ....................................................................................................................... 36
    Selecting a File from a Cloud-Based File storage Service ............................................................. 36
  Viewing and Signing Documents ..................................................................................................... 37
    Viewing Pages ............................................................................................................................... 40
  Using PDF forms ............................................................................................................................. 41
  Signing a PDF Document .................................................................................................................. 42
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Signing Office 2010/2013/2016 Documents</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>Viewing Signature Details</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>Validating Signatures</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Post-Signing Actions</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>Using a Volatile Graphical Signature</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>User Experience when using Volatile Graphical Signatures</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Signing in a Point of Sale Mode of Operation</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>Chapter 5: Using the DocuSign SA Web App as part of the DocuSign Cloud Signature Workflow</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>General</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Flow of Operations</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Enabling use of DocuSign Cloud in the Organizational Workflow</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>DocuSign Cloud Settings</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>DocuSign SA Web App settings</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>Chapter 6: Adding Graphical Signatures and Modifying User Settings</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Adding Graphical Signatures</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Modifying User Settings</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>Logging off the DocuSign SA Web App</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>Chapter 7: Signing Documents from Cloud-based File Storage Services</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Signing Documents from Dropbox</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Signing Documents from OneDrive</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>Signing Documents from Google Drive</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>Invoking the Signature Operation from Within Google Drive</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>Signing Documents from Box</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Invoking the Signature Operation from Within Box</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Chapter 8: Signing Office 365 documents using DocuSign SA Web App</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Enabling Signing O365 Documents using the DSA Web App</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Required Software Components</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Step 1: Deploying and Configuring the DocuSign WebApp Proxy</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Step 2: Registering the DocuSign SA O365 App in the Office 365 SharePoint Cloud</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>Step 3: Modifying and Deploying the DocuSign SA O365App</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Step 4: Updating the DSAApp Proxy Settings</td>
<td>82</td>
</tr>
</tbody>
</table>
User Signature Flow for Signing Documents located in the Office 365 SharePoint cloud... 82
Chapter 9: Configuring the DocuSign SA Web App ................................................................. 84
Modifying DocuSign SA Web App Parameters ................................................................. 84
  Modifying Image Setting Parameters .......................................................... 84
  Defining SMTP Settings ................................................................. 85
  Defining Extended Authentication Settings ........................................ 85
  Defining API Integration Settings .................................................... 88
  Defining General Settings .......................................................... 89
  Defining DocuSign Cloud Settings ................................................ 92
  DTM Settings section ....................................................................... 92
  Login Policy Settings section .......................................................... 93
  Configuring Web App for ADFS with WS Federation Protocol ......................... 94
  Running the Microsoft Federation Utility .................................................... 95
Configuring DocuSign SA Web App in SAML2 Protocol Mode ........................................ 100
  Configuring DocuSign SA Web App to use the SAML2 protocol .................. 101
  Running the Saml2ConfigUtility ....................................................... 101
  Configuration at the Identity Provider System ............................................ 104
Configuring DocuSign SA Web App in Passive Mode for using JWT tokens ....................................................... 104
  Configuring Web App for JWT based user authentication with OpenID Connect Protocol ................................................................. 105
Configuring DocuSign SA Web App to use Microsoft Active Directory Kerberos authentication ................................................................. 106
  Setting Web App’s web.config File .................................................. 106
  Using the setspn utility to establish a Kerberos based authentication scheme 106
  Using Internet Explorer to Access DocuSign SA Web App Based on Kerberos Ticketing ...................................................................... 107
  Using Google Chrome to access DocuSign SA Web App based on Kerberos Ticketing 108
Configuring DocuSign SA Web App to Work with PDF Forms ........................................ 108
Integrating with Cloud-Based File Storage Services ................................................. 109
  Defining Cloud Storage Settings ....................................................... 109
  Creating an Application for Enabling Access to User Accounts in the Cloud 110
Chapter 1: Overview

Over the last four decades, the biggest challenge of IT departments in many organizations was moving to a paperless work environment. Seemingly, there was tremendous success in this regard. Today, most transactions in the business world are performed electronically:

- Documents are written using word processing programs.
- Messages are sent via email.
- Inventories and purchases are tracked using Enterprise Resource Planning (ERP) systems.
- Medical information is stored in Electronic Medical Record (EMR) systems.

Although these transactions are performed in a paperless environment, organizations have still not managed to find an easy way to get rid of the paper used for data authentication (signing the authenticity of the data). Today, although organizations have invested large amounts of funds and other resources in creating paperless environments, their workers are still printing every transaction, signing it, and saving the printed copy. These organizations require a digital method for data authentication.

By moving to a viable electronic data authentication system, organizations can reduce their printing, archiving, shipping, and handling costs. In addition, better and more competitive customer service can often be provided.

Requirements for Data Authentication Systems

A viable data authentication system must meet the following specifications:

- **Security** – The system must ensure that no one other than the data creator can tamper with or change the data in any way.
- **Third-party validation** – The system must enable any third party to validate the authenticity of the data. If a dispute arises between the parties (the data creator and recipient), any third party must be able to validate the data authenticity in order to settle the dispute.
- **System independence** – Data authentication must be independent of the system that created the data. Users must be able to validate the authenticity of the data using a known standard that is independent of any specific system.
- **Validation over time** – Users must be able to validate data authenticity at any point in time. Authenticity cannot expire at any point.

Currently, the only data authentication method known to support all of these requirements is the Public Key Infrastructure (PKI) method of authenticating data, simply called “digital signatures”.

Introduction to DocuSign SA

DocuSign Signature Appliance is a PKI-based, off-the-shelf digital-signature solution that can be integrated with a wide range of applications. In this way, DocuSign SA enables organizations to embed digital signatures in various documents, forms, and transactions. DocuSign SA is a turnkey, hardware-based solution that is easily and quickly deployed in the network and provides cost-effective digital-signature capabilities for the organization.
DocuSign SA includes all the components needed for PKI-based digital-signature deployment. You do not need to install any other device or integrate any other component for the system to work.

**Environments Supported by DocuSign SA**

DocuSign SA integrates with leading user management systems, including Microsoft Active Directory and a variety of LDAP (Lightweight Directory Access Protocol) based directories, such as IBM Tivoli. This integration ensures no overhead in managing the digital-signature system and signature credentials (i.e., the private keys that are needed in a PKI environment), solving one of the main problems of legacy digital-signature systems. System managers, network managers, and end-users can continue to use the IT infrastructure in the same manner as before DocuSign SA was installed.

DocuSign SA stores the signature credentials in a secure server, ensuring that the signer has exclusive access to his or her signature credentials, while still maintaining a centrally managed solution. This is necessary in order to fulfill the security requirement of the data authentication system.

Another option is to use the DocuSign SA Cloud service. An organization can register its users to the service and thus enable them to digitally sign content without having to deploy the DocuSign SA appliance on the organizational premises.

**Applications that Work with DocuSign SA**

An increasing number of applications can work with DocuSign SA as their digital-signature layer without needing any further integration, including:

- Microsoft Office 2010/2013/2016 (Word, Excel, and PowerPoint)
- Microsoft InfoPath 2010/2013
- Adobe Acrobat/Reader
- Microsoft SharePoint 2010/2013/2016
- XML
- TIFF files
- Microsoft Outlook and Outlook Express
- Adobe Server forms (for signing web forms)
- AutoCAD
- Lotus Notes
- OpenText
- Web applications
- Any application that has a print option can use DocuSign SA to generate a PDF file and sign it.

For information on using DocuSign SA with other applications, contact DocuSign technical support.
DocuSign SA Components

DocuSign SA includes the following components:

- **DocuSign Signature Appliance** – The DocuSign Signature Appliance hardware and software, connected to the organization’s network.
- **Client** – The DocuSign SA Client software, installed on the users’ computers.
- **Administrator** – The DocuSign SA Administrative software that includes the DocuSign SA Microsoft Management Console (MMC) snap-in, installed on the administrative computer.
- **DocuSign SA Connector for SharePoint** – This connector enables adding digital signature functionality to documents managed by Microsoft SharePoint, or using digital signatures within any workflow procedure that is based on Microsoft SharePoint.
- **DocuSign SA Web App** – This application is deployed in the Microsoft Web Server of the organization and enables users to sign documents without installing any client component. DocuSign SA Web App can use either the local DocuSign SA appliance or a remote DocuSign SA. Applications can interact with the DocuSign SA Web App and add a digital signature to documents using a web based interface.
- **DocuSign SA Mobile App** – This mobile application, which can be installed on Android-based devices or Apple iOS devices, enable users to sign documents using their mobile devices. The mobile devices interface directly with Appliance via a DocuSign SA RESTful interface. The DocuSign SA Mobile App can interface with either the DocuSign SA Cloud, the organizational DocuSign Signature Appliance or DocuSign SA’s Trial system.
- **DocuSign SA Signature APIs** – Developers can use local and network APIs to integrate their applications with DocuSign SA Central appliances and the DocuSign SA Cloud service.

DocuSign SA Guides

DocuSign SA documentation includes the following guides:

- **DocuSign Signature Appliance Administrator Guide** – Provides all the information necessary for an administrator to install and manage Appliance in the various environments in which DocuSign SA can operate.
- **DocuSign Signature Appliance User Guide** – Provides all the information necessary for an end user to use DocuSign SA. Includes information about special add-ins for various applications such as Microsoft Office.
- **DocuSign Signature Appliance Connector for SharePoint User Guide** – Provides all the information necessary for implementing and using the DocuSign SA Connector for SharePoint.
- **DocuSign Signature Appliance Web App User Guide** – Provides all the information necessary for deploying DocuSign SA Web App in the organization’s environment.
- **DocuSign Signature Appliance Signature APIs Developer’s Guide** – Provides all the information necessary for developers to integrate their application with DocuSign SA.
DocuSign Signature Appliance Web App Guide

- **DocuSign Signature Appliance Mobile App Deployment Guide** – Provides all the information necessary for deploying the DocuSign SA Mobile App.

**Intended Audience**

This guide is intended for users wishing to implement the DocuSign SA Web App. It is assumed that readers have prior knowledge of DocuSign SA.

**Organization of this Guide**

This guide is organized as follows:

- **Chapter 1: Overview** – Provides an overview and introduction to DocuSign SA.
- **Chapter 2: Introduction to DocuSign SA Web App** – Provides an introduction to DocuSign SA Web App.
- **Chapter 3: Installing and Deploying the DocuSign SA Web App** – Describes how to install and deploy DocuSign SA Web App.
- **Chapter 5: Using the DocuSign SA Web App as part of the DocuSign Cloud Signature Workflow** – Describes how the DocuSign Web App can be used as part of the DocuSign Cloud Signature Workflow.
- **Chapter 6: Adding Graphical Signatures and Modifying User Settings** – Describes how each end-user can add graphical signatures and modify various user settings such as his/her signature appearance.
- **Chapter 7: Signing Documents from Cloud-based File Storage Services** – Describes the end user experience for the following Cloud-based file storage services: Dropbox, OneDrive, Google Drive, and Box.
- **Chapter 8: Signing Office 365 documents using DocuSign SA Web App** – Describes how documents that are stored in Office 365 can be forwarded to be signed in DocuSign SA Web App and then saved back to Office 365.
- **Chapter 9: Configuring the DocuSign SA Web App** – Describes how to customize the DocuSign SA Web App.
- **Index** – Provides a comprehensive index of the topics discussed in this guide.
Chapter 2: Introduction to DocuSign SA Web App

About DocuSign SA Web App

Using DocuSign SA Web App, users can digitally sign and validate documents without having to install any client software. Instead, the signature operation uses the DocuSign Signature Appliance or appliances that were configured as part of the DocuSign SA Client installation within the DocuSign SA Web App platform. You can optionally configure DocuSign SA Web App to use the DocuSign SA appliances deployed in another private cloud.

DocuSign SA Web App can be used from a wide variety of devices including iPads, iPhones, Samsung Galaxy smart phones, etc. DocuSign SA Web App serves many users in parallel.

DocuSign SA Web App supports the following document types:

- PDF files.
  
  Note that the PDF file must be no larger than 30 MB. In addition, password protected PDF files are not supported yet.

- Microsoft Word (.docx and .doc) and Microsoft Excel (.xlsx and .xls) documents.
  
  - MS documents that contain no signature fields are converted to PDF format when they are loaded to the DocuSign SA Web App.
  
  - You can globally configure whether MS documents (.docx or .xlsx) that contain at least one signature fields are converted to PDF when loaded, or remain in their original format.
    
    Note that the maximum supported file size is 30 MB, or 200 pages. Even files that are close to the maximum limit may not load properly.

Starting from DocuSign SA version 7.5, the DocuSign SA Web App can be configured to upload forms based on PDF files. The end user can enter data in the various fields. The end user’s signature applies also to the specific data entered in the form’s fields.

Starting from Web App version 8.0, Web App can be deployed as part of the DocuSign Cloud Signature Ceremony to enable end users performing a transaction to use the DocuSign Cloud Application. Upon a signature operation, the users are redirected to the organization or Trust Service Provider to perform the actual digital signature operation which is based on having the Web App use the DocuSign Signature Appliance.

In this type of configuration, the document is viewed as part of the DocuSign Cloud application. Upon redirection to the organization or Trust Service Provider, the user is requested to login to the DocuSign Signature Appliance for accessing his/her personal keys for the purpose of a digital signature operation.
Advantages of DocuSign SA Web App

No database – The DocuSign SA Web App does not include nor does it depend on a database deployed in the web server. This means that it can be easily deployed and upgraded to a new version.

High availability – It is very easy to create a high availability farm containing several web servers, each hosting a DocuSign SA Web App, for signing documents through a web interface.

No memory footprints – The DocuSign SA Web App manages all files and other elements such as graphical images in memory, and does not leave any footprints of files or other sensitive data in the local hard disk of the DocuSign SA Web App’s platform.

End User Authentication – The end user can be authenticated to the DocuSign SA Web App using either a user ID & password mechanism, or an Active Directory Kerberos ticketing mechanism. The end user can also be authenticated by an external Identity Provider. For more information, see Integration with External Identity Providers.

Integration with Document Management Solutions

The DocuSign SA Web App can be integrated with applications such as a document management application. In these types of scenarios, the application uses a special plug-in that performs a Sign with DocuSign operation. When an end user wishes to view a file and sign it, the end-user is redirected to the DocuSign SA Web. After signing is complete, the document management application retrieves the signed file from the DocuSign SA Web App. The exact interface between the application and the DocuSign SA Web App is described in the DocuSign SA Web App API section of the DocuSign SA Programmers Guide.

In this type of usage, the DocuSign SA Web App is named DocuSign SA Web Agent.

Integration with External Identity Providers

The DocuSign SA Web App can be integrated with External Identity Providers (IDP) such as Microsoft ADFS.

In the context of using an IDP, the DocuSign SA Web App (together with the DocuSign SA Appliance) is considered a Service Provider (SP). In these cases, the end user is authenticated based on the External Identity Provider mechanism, and the proof of authentication, which is either a SAML ticket or a JWT (JSON Web Token), is forwarded to DocuSign SA for establishing a sustained end user account in DocuSign SA.
Chapter 3: Installing and Deploying the DocuSign SA Web App

This chapter describes how to:

♦ Install DocuSign SA Web App.
♦ Deploy DocuSign SA Web App.

DocuSign SA Web App Installation

DocuSign SA Web App can be installed as a web component using the following Web Server product:

Microsoft IIS version 7 and above on a computer running Windows 2008 R2 and above.
Both administrative operations and user operations are based on web-based applications, thus the end user does not have to install any software on the end PC. However, a DocuSign SA Client must be installed on the server on which the DocuSign SA Web App component is installed. The recommended order of installation is:

1. Installing and Configuring the DocuSign SA Client.
2. Installing the DocuSign SA Web App Component.

Installing and Configuring the DocuSign SA Client

DocuSign SA Web App accesses DocuSign SA through the DocuSign SA client installation. Therefore, before installing the DocuSign SA Web App component, you must:

1. Install a DocuSign SA Client version 8.2 or above on the server that will host the DocuSign SA Web App component.

Note: The DocuSign SA Web App can only work with a DocuSign SA Appliance version 6.2 or above.
In addition:

- It is mandatory to have .NET Framework 4 installed on the hosting platform.
- If you are using a Windows 2012/2016 system, you must install ASP .NET v4.5.
- Microsoft IIS should also include the URL Rewrite module v2.0 or above.
  Please refer to Microsoft documentation for instructions how to install this module.


Configuring the DocuSign SA Client for DocuSign SA Web App

To configure the DocuSign SA Client to work with DocuSign SA Web App:

1. Access the DocuSign SA control panel by selecting Start > Programs > DocuSign > Signature Appliance > Signature Appliance Control Panel. The Signature Appliance Control panel appears.
2. Select **Client Configuration**. The DocuSign SA configuration utility’s main window appears.

3. Click to the left of the **Client** node.

4. Click the **Appliances** node.

![DocuSign SA Configuration Utility - Local Computer Configuration](image)

**Figure 1** DocuSign SA Configuration Utility – Client Configuration – Appliances Parameters
5. In the Appliances screen:
   a. Add the DocuSign SA appliance IP or DNS name to the Appliance list.
   b. If the list contains more than one appliance to provide high availability, you must specify the first one as the Preferred appliance.
   c. Check the Client Optimization checkbox.
   d. If DocuSign SA is deployed in an Active Directory environment, the AdMode parameter should be set to true (refer to AdMode in Chapter 9: Configuring the DocuSign SA Web App).
   e. If DocuSign SA is deployed in an LDAP environment, in the Prompt for logon method drop-down select User Pwd Server Side (AD/LDAP).
   f. If DocuSign SA is integrated with an External Identity provider (such as an ADFS Identity Provider that uses SAML2 protocol or JWT), in the Prompt for logon method drop-down make sure that Auto is selected.
      The client authentication will be determined by the Web App configuration (that is, as configured in the web.config file).
      In addition, make sure the DocuSign SA client is not configured to work in SAML Active Mode. For more information refer to the DocuSign SA Client documentation.

6. Select File → Apply (save to registry) to apply the changes.

7. Click the Login dialog node.
8. In the Login dialog screen:
   a. Uncheck the Permit known applications only checkbox.
   b. Select Disable login dialog.
9. Select File → Apply (save to registry) to apply the changes.
10. Click the **Timeouts** node.

![DocuSign SA Configuration Utility - Local Computer Configuration](image)

Figure 3  DocuSign SA Configuration Utility – Client Configuration – Timeout Parameters

11. In the **Timeouts** screen:

- Set the **Configuration reload interval** to 0.
- It is recommended to reduce the **Connect failure block time** to 30.
12. Select **File → Apply (save to registry)** to apply the changes.
13. Select **File → Exit** to exit the DocuSign SA configuration utility.

**Note:** Following any update to the DocuSign SA Client configuration, it is recommended to perform a reset by running an `iisreset` command so the new parameters will take effect.

### Installing the DocuSign SA Web App Component

To install the DocuSign SA Web App component:

1. Deploy the DocuSign SA Web App package to a directory in the Web Server.
2. Launch the **Internet Information Services (IIS) Manager** program that manages the Microsoft Web Server.
3. Right click **Sites** in the **Connections** pane and select **Add Web Site**.

![Internet Information Services (IIS) Manager](image)

*Figure 4  Internet Information Services (IIS) Manager*

The **Add Web Site** window appears.
4. Perform the following:
   a. In the Site name field, enter a name for this web site. The recommended name is **DSAWebApp**.
   b. In the Application pool field:
      - If you are using a Windows 2012/2016 system, select **.NET v4.5**.
        Note that ASP .NET 4.5 is not installed by default, and should be installed in the machine prior to Web App deployment.
      - Otherwise, select **ASP .NET v4.0**.
        If you are unable to select the application pool ASP.NET v4.0, configure ASP .NET v4.0 as part of the Microsoft IIS deployment (refer to Microsoft documentation for instructions).

**Note:** In the application pool's Advanced Settings, we recommend setting the Load User Profile attributes in the Process Model section to true, to avoid any permission related problems:
c. In the **Physical path** field, select the location in the server’s hard disk where the DocuSign SA Web App will be deployed. For example: `c:\inetpub\WebApp`.

d. In the **Host name** field, define a host name (such as sign.org.com), or leave the field empty.

e. In the **Type** field that is in the **Binding** section, select **HTTP** or **HTTPS**.

   For initial testing purposes you can use the HTTP protocol, but on a production system HTTPS should be used in order to secure the communication between the user’s browser and the web server, which handles the uploaded documents and the user’s credentials.

f. In the **Port** field, specify a port number. For testing purposes you can use port 80.

g. If you selected **HTTPS** as the Binding type, the screen refreshes and an **SSL Certificate** field appears. In the **SSL Certificate** field, select an already uploaded SSL certificate and a server key that will be used to identify this web site. For more information on how to enroll for an SSL key and certificate and upload them to the Microsoft IIS web server, please refer to Microsoft documentation.

![Add Web Site Window](image)

*Figure 6 Add Web Site Window – Specifying an SSL Certificate for the HTTPS Protocol*
5. Click **OK**. The Web site is deployed.

6. If you selected **HTTPS** as the Binding type:
   a. Select the new site under **Sites** in the **Connections** pane.

![Figure 7 Sites Node in Internet Information Services (IIS) Manager](image)

   b. Click **SSL Settings** in the right pane (Figure 7). The SSL Settings window appears.

   c. Check the **Require SSL** checkbox.

   **Note:** Whether you choose to use HTTPS or HTTP communication, make sure there is connectivity between users and the DocuSign SA Web App as well as communication between the DocuSign SA Web App and the DocuSign SA appliance. If there are any networking related issues such as a firewall, users will not be able to communicate with DocuSign SA Web App, and DocuSign SA Web App will not be able to interface with the DocuSign SA appliance.

### Redirecting HTTP to HTTPS

If you specified HTTPS connection to the DocuSign SA Web App, you can redirect all HTTP connections as follows:

1. Select the new site under **Sites** in the **Connections** pane (Figure 7).
2. Click **Error Pages** in the right pane. The Edit Custom Error Page window appears.
3. Enter **403** in the **Status code** field.

4. Select **Respond with a 302 redirect**, and enter the URL of the HTTPS web site.

**Configuring the Mail Server**

As part of the DocuSign SA Web App’s flow of operation, the DocuSign SA Web App can send signed documents through emails to recipients. To enable this option, you must configure the SMTP server settings, as described in *Defining SMTP Settings*.

You can optionally configure the Windows Server platform to include an SMTP server by turning on the **SMTP Server** feature (refer to Microsoft documentation). If you do so, make sure to setup the SMTP server so it will restart following every startup of the server machine.
Configuring IIS for Kerberos Ticketing

To enable the end user to authenticate using the Kerberos ticketing mechanism, IIS and the platform on which the DocuSign SA Web App is installed, must be configured to enable this mode.

Some general guidelines are provided below. Refer to Microsoft documentation for instructions on how to provide support for Kerberos based authentication.

**Settings in the Domain:**

1. The machine on which the Web App and IIS are installed should be part of the Microsoft Domain.
2. You may need to enable delegation by selecting the Web App server Properties > Delegation tab > Trust this computer for Delegation to any Service (Kerberos only).
   
   Please refer to Microsoft documentation.

**Settings in IIS:**

You may need to set the following Web Server settings. Please refer to Microsoft documentation:

1. In **Authentication**, perform the necessary settings for enabling the Kerberos based authentication.
2. In **Bindings**, make sure the host name of the Web App site is the same as the computer name of the Web Server that will run it.

Refer to *Chapter 9: Configuring the DocuSign SA Web App* for additional configuration actions related to using the Kerberos Ticketing mechanism.

**Using an Alternate Web Site**

If not all users in the organization have the ability to use Kerberos from their devices (for example, when using mobile phones), you can use the Alternate Web App site option, available from DocuSign SA Web App version 8.4. Starting from version 8.4, you can define an Alternate Web App site to which the user is redirected if he cannot use Kerberos. The Alternate Web App will ask the user for a User ID and a password for authenticating to the DocuSign SA.

In this case, the following should be configured in addition to the configurations described in *Configuring IIS for Kerberos Ticketing*:

1. In the Web Server Settings, enable **Anonymous** in addition to **Windows Authentication**.
2. In the Web.Config file of Web App, define a Windows authentication mode under the **system.web** section, as follows: `<authentication mode="Windows">`.
3. In the Web.Config file of Web App, enable Anonymous mode under the **system.web** section as follows: `<anonymousIdentification enabled="true" />`.
4. Refer to *Chapter 9: Configuring the DocuSign SA Web App* for additional configuration actions related to the **AlterWebAppSiteURL**
Configuration Required if PDF Forms are Used

If you wish to use PDF forms, DocuSign SA implementation is based on a product called RAD PDF.

The RAD PDF product requires deploying Microsoft SQL on the application server.

Follows the instructions in the following sections:

- Installing an SQL Server
- Configuring the SQL Server
- Installing RAD PDF

Installing an SQL Server

1. Make sure at least .NET Framework 4 is installed on the hosting platform.

2. Install an SQL Server 2008R2 or above. Keep the default settings in all installation steps, except in the following steps:

   a. When required to indicate whether to install a new SQL server or to upgrade an SQL server, select the option of installing a new SQL Server.

![Figure 9 SQL Server Installation Center - Installation](image-url)
b. When required to enter a name for this SQL server instance, the recommended name to enter is **COSIGN**.

*Figure 10  SQL Server Installation – Instance Configuration*
c. When required to select the authentication mode, select **Windows authentication mode**. Make sure to use the administrator of the local machine.

![Figure 11 SQL Server Installation – Database Engine Configuration](image)

3. After installation is complete, make sure to install the SQL Server Management Studio. You can download it free from the Microsoft website.

**Configuring the SQL Server**

Perform the following tasks:

1. **Create a RadPdf Database instance to manage DocuSign SA RadPdf data**
2. **Enable the IIS Service Account that runs Web App to Access the RadPdf Database**
3. **Create Tables in the RadPdf Database**
Create a RadPdf Database instance to manage DocuSign SA RadPdf data

Perform the following to create the RadPdf database instance that keeps information as part of the Form Filling process.

1. Launch the SQL Server Management Studio.
2. Choose the SQL server instance you created in Installing an SQL Server, and click Connect.

![Connecting to SQL Server](image)

Figure 12 Connecting to SQL Server

3. Right click Databases in the Object Explorer pane and select New Database.

![Selecting to Create a new SQL Database](image)

Figure 13 Selecting to Create a new SQL Database

4. In the DB field enter the name RadPdf and confirm.
Enable the IIS Service Account that runs Web App to Access the RadPdf Database

The service account in IIS that will run Web App should also have access to the RadPdf database. To enable it to do so:

1. In the SQL Server Management Studio, in the left explorer window, expand your server’s Security > Logins.

   ![SQL Server Logins](image)

   
   **Figure 14** Viewing SQL Server Logins

2. Search for the service account that is used to run Web App (in this case, NT AUTHORITY\NETWORK SERVICE).

   If you do not find the service account (for example, if you are using the App Pool identity), you can either:

   a. Modify the service account in IIS to one of the listed services.

   or

   b. Add the current service account to the list in Management Studio, as follows:

   ♦ Right click the Logins directory and choose New Login...:
Enter the login name. In an App Pool Identity case it should look like the following if ASP.NET v4.0 is the name of your pool:

When finished, click OK.

The service account should appear on the list.

3. Right click the relevant SQL account and select Properties. In the Properties window, click User Mapping. You should see a list of all the databases existing in the server:
4. Check the **RadPdf** server.
5. In the bottom of the *User Mapping* window, check the **public** and **db_owner** memberships:

6. Click **OK**.
7. Close the SQL Server Management Studio.

**Create Tables in the RadPdf Database**
1. Under the DocuSign SA WebApp deployment directory, locate `radpdf.sql`.
2. Double click `radpdf.sql` and the SQL Server Management Studio will be launched.
3. Login to the SQL Server Management Studio.
4. Click the **Execute** button. This will run the script that creates the tables in the RadPdf Database you created.
5. When finished, close the SQL Server Management Studio.

**Installing RAD PDF**

1. Download the RAD PDF installer from the RAD website:  

2. When queried, indicate you wish to install the ‘Complete’ package.

3. Run the installation. Before finishing the installation, make sure the **Start PDF RAD Service** checkbox is checked.
4. If you want to change some RAD PDF Administration settings, such as when documents will be deleted from the database, run the RAD PDF Administration tool. A shortcut to the tool was added to your Start menu.

![Figure 21 RAD PDF Administration Tool](image)

Note that you must first provide the SQL Maintenance Connection String before you can actually make any changes. For example: `Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=RadPdf`.

For instructions on how to create the string, refer to [Creating the Maintenance Connection String for the RadPdf Database](#). Refer to [Chapter 9: Configuring the DocuSign SA Web App](#) for additional configuration actions related to working with PDF forms.
Creating the Maintenance Connection String for the RadPdf Database

Perform the following to create the string you need for changing RAD PDF admin settings.

1. Create a new empty text file.
2. Rename the extension, from **txt** to **udl** (Microsoft Data link file type).
3. Double click the UDL file. The following Data Link Properties window appears:

![Data Link Properties](image)

4. Perform the following in the Data Link Properties window:
   a. Select the SQL Server from the list (usually, this is your computer name).
   b. Select **Use Windows NT Integrated security**.
   c. Select the RadPdf database.
   d. Click **Test Connection**. A success message similar to the following should appear:
e. Click **OK**.

f. Click **OK** again to close the window.

5. Right click the UDL file and select **Open with Notepad**. The contents of the file should be similar to the following:

![UDL File Contents displaying the SQL Connection String](image)

*Figure 24  UDL File Contents displaying the SQL Connection String*
6. Locate the text after the Provider value. This text is the connection string. In the example shown above it is:

   **Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=RadPdf;Data Source=ERL2-PC**

   Note that the **Data Source** will appear only if the RAD PDF database is installed on another PC. Otherwise, the connection string will be similar to the following:

   **Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=RadPdf**

**Uninstalling the DocuSign SA Web App**

To uninstall the DocuSign SA Web App:

1. Open the *Internet Information Services (IIS) Manager* program that manages the Microsoft Web Server.

2. In the connection pane, right-click the web site under **Sites** and click **Remove**.

3. Go to the location where the DocuSign SA Web App files are located and delete that directory.

4. Uninstall the DocuSign SA Client.

Note that because the DocuSign SA Web App leaves no footprints of files or other elements such as graphical images, you do not need to delete any temporary files as part of the uninstall procedure.
Chapter 4: Using DocuSign SA Web App for Signing and Validating Documents

Using DocuSign SA Web App, you can digitally sign and validate documents without having to install any client software. The workflow is as follows:

2. Select a document.
3. View the document contents.
4. Sign empty signature fields and/or create signatures.
5. Validate existing signatures, if desired.
6. Save or send the signed copy.

Logging in to the DocuSign SA Web App

In order to view documents, sign them and validate them, you must first login to the DocuSign SA Web App. Login is performed based on your DocuSign SA user ID and password.

To login to DocuSign SA Web App:

1. Access DocuSign SA Web App by entering its URL, such as: https://DocuSignSAweb.org.com.

   The DocuSign SA Web App login page appears:
2. In the User name / Email field:
   - Enter your User name if DocuSign SA Web App is using a local DocuSign SA appliance or appliances.
   - If DocuSign SA is deployed in an Active Directory environment, enter your Active Directory User ID and your domain, in the following email format: <Active Directory user ID>@<domain name>.

3. Enter your password in the Password field.

4. Click **Sign in**.

The Select Document window appears (Figure 26).

Note that if you do not perform any action in DocuSign SA Web App for a few minutes, you will be automatically logged off.

Note that if DocuSign SA Web App is installed and configured to use Kerberos Ticketing in an Active Directory environment, or is setup to use an External Identity Provider, you will be either already authenticated by the domain, or redirected to be authenticated by the External Identity provider. In either case, the above login window will not appear.
Selecting a Document

The window that appears immediately after logging in is the Select Document window.

![Select Document Window](image)

You can select a document in either of two ways:

- **From your PC or device** – Click **Browse** to select a local document. This option is not available when using an iPad or iPhone.
- **From the Cloud** – Click to access one of the Cloud-based file storage services and select your file. This method is especially useful for mobile devices (tablets and smartphones) that do not have a local Browse functionality. For details, refer to Selecting a File from a Cloud-Based File storage Service.

Selecting a File from a Cloud-Based File storage Service

1. In the Select Document window (Figure 26), select one of the following:
   - Dropbox
   - Google Drive
   - Box
   - Microsoft OneDrive
You are redirected to the Login window of the selected service.

2. Log into the service.

3. Optionally select the Remember me option if you want the browser to remember your credentials.

4. The first time you access the Cloud-based file storage service from DocuSign SA Web App, right after logging in to Dropbox you may be asked to confirm that you grant DocuSign SA Web App access to the service.

![Confirm DocuSign SA Web App Access to Dropbox](image)

5. Allow access.

DocuSign SA Web App will save your login information so that you can automatically log into the service without having to enter credentials each time.

Also, depending on the Cloud-based file storage service, you will be notified by email that the DocuSign SA application would like to connect with your Dropbox account.

Refer to **Chapter 7: Signing Documents from Cloud-based File Storage Services** for a description of the end user experience when signing using a Cloud-based file storage service.

Refer to **Integrating with Cloud-Based File Storage Services** for instructions on how to register an organizational account for the various Cloud-based file storage providers.

**Viewing and Signing Documents**

After uploading a document to DocuSign SA Web App, you can view all its pages, locate existing signature fields (both signed and empty ones) and sign empty signature fields, as well as create & sign new signature fields.
Starting from DocuSign SA Web App version 7.1, you can also upload Word and Excel 2007/2010/2013/2016 documents, and produce Office 2007/2010/2013/2016 signatures. However, you cannot add new signatures to these documents, but only sign empty existing signature fields or view the details of signed field.

The file you loaded appears in the DocuSign SA Web App window.

- If the document does not contain any empty signature fields, by default it will appear with a signature-creation frame hovering in the middle of the window. When you browse through the pages, that frame will appear on every page. The frame enables you to create a signature field on any page, position and resize it, and then sign inside it.

Figure 28  Example of a Loaded Document Without Empty Signature Fields
If the document has pre-existing empty signature fields, by default it will appear without a signature-creation frame. You will be able to sign the empty fields but will not able to move or resize them.

![Example of a Loaded Document With Empty Signature Fields](image)

**Note:** Within DocuSign SA Web App you cannot create an empty signature field. If you do not sign inside a signature-creation frame, no field is created.
However, documents loaded into DocuSign SA Web App can contain empty signature fields, and you can sign those fields within DocuSign SA Web App.

**Viewing Pages**

Use the select-page carousel to browse through the document.

The five page icons correspond to page numbers. Click the left arrow for the previous five pages, or the right arrow for the next five pages. Blue arrows above and below a page icon indicate the currently-displayed page.

If a PDF document has pre-existing empty signature fields, the select-page carousel indicates a page containing empty signature fields using a **sign** icon instead of the page number.

Use the zoom icons to zoom in and out.

Note that for an Office 2010/2013/2016 document there is no indication which pages are signed. Instead, general information about the document signatures is provided.

*Note:* Due to a limitation, the Word/Excel document does not show the visible signature in DocuSign SA WebApp. However, when the file is opened by Office tools, the visible signature is displayed.
Using PDF forms

Starting from DocuSign SA version 7.5, you can also view, fill-in and sign PDF forms in DocuSign SA Web App.

You can view the PDF form and fill-in the fields such as text boxes and option menus, as shown in the example below.

![Example of PDF Form Appearance in DocuSign SA Web App](image)

After clicking **Continue**, the filled-in document will be presented for the purpose of signing, like a regular PDF file presented for signing.
Signing a PDF Document
You can either sign a pre-existing empty signature field, or create a new field and sign it.

1. If you wish to sign a pre-existing field, skip to step 2. If you are creating a new signature field, perform the following:
   a. Locate the hovering signature-creation frame (see Figure 28). If the signature-creation frame is not displayed, click **Add another signature**.
   b. Optionally, move the frame to the desired location on the page, as follows:
      - In a PC, click inside the frame and drag.
      - In a mobile device, touch inside the frame and drag.
   c. Optionally, resize the frame, as follows:
      - In a PC, click a side or a corner and drag.
      - In a mobile device, touch a side or a corner and drag.

2. Click/tap **Sign** in the signature frame. The Signature Appearance window appears.

   Note that if you enabled using a *Volatile Graphical Signature*, a Signature Appearance window similar to Figure 39 or Figure 40 appears.

---

**Figure 31 Signature Appearance Window**
3. Select a graphical signature from the carousel. Alternatively, click the radio button to **Sign without a graphical signature**.

Note that a user's graphical images and default settings are kept in the user's account in the DocuSign SA appliance.

4. Optionally, add a reason and title to the signature, and define your signature appearance. To do so, click/tap **Add reason and change signature appearance**. The window refreshes to display the following:

   ![Signature Appearance](image)

   **Figure 32** Adding Reason and Defining Appearance

5. In you created a new field, edit the appearance as desired by selecting whether to show the date and time, the signer's title, etc.

   If you are signing a pre-existing field, you can only enter the reason and title if those are required.

6. Click/tap [Apply].
The document will display a digital signature, similar to the following:

![Sample Signature](image)

*Figure 33 Sample Signature*

7. You can optionally click the signature to view its details. Refer to [Viewing Signature Details](#).

Note that also PDF forms can be signed. The signature applies to the information entered by the user in the form, in addition to the regular content.

**Note:** Starting from version 8.5, it is possible to perform the following operation upon a signature field:

Certify a signature – For more information, read about the certify PDF operation in the *Prepare & Sign – Signing PDF and non-PDF Files* chapter of the *DocuSign SA user Guide*.

### Signing Office 2010/2013/2016 Documents

Using DocuSign SA Web App, you can sign existing signature fields in Office documents. Note that you cannot create new signature fields.

By default, if an Office document contains at least one signature or signature field, it is not converted to PDF when you upload it into DocuSign SA Web App and you can sign it in DocuSign SA Web App.

DocuSign SA Web App supports the signing of signature fields that were created by the Microsoft Signature Line provider or DocuSign SA Signature Line Provider. However, it is recommended to create signature fields using the Microsoft Signature Line provider to avoid rendering problems. If a signature field was created using the DocuSign SA Signature Line Provider, the text “Software is required” appears in the location of the empty signature field.

To define whether an Office document containing signature fields is converted to PDF when it is uploaded to DocuSign SA Web App, refer to *OfficeBackwardCompatibility*. 
**Note:** As described in the DocuSign SA client guide, a user may sign a Word/Excel document only if the certificate chain is trusted. That is, the DocuSign SA ROOT certificate must be trusted in the Web App platform. You can use the **Install Signature Appliance CA Certificate** option of the DocuSign SA Configuration utility to instruct Web App to trust the DocuSign SA ROOT certificate.

To sign an Office document:

1. Optionally view the signature information appearing in the top right corner. In the example shown below, the information is that the document contains two unsigned signature fields.

2. Click/tap **Sign Document**.

   If the document has more than one empty field, a list of the signature field identifiers (taken from the **Suggested Signer** attribute of the signature fields) is displayed. Select a signature field and click/tap **Close**.
The Signature Appearance window appears.

Note that if you enabled using a Volatile Graphical Signature, a Signature Appearance window similar to Figure 39 or Figure 40 appears.

3. Click/tap Add Reason and change signature appearance if you wish to add more information to the signature, such as a reason.

4. Click/tap Apply. The document is signed.

The information in the upper right corner is updated.
Viewing Signature Details
You can view the signature details of each signature.

- In a PDF document, click any signature, whether pre-existing or created by you, to view its details.
  A Details window similar to the following is displayed.

![Signature Details for a PDF Document](image)

For an Office 2007/2010/2013 document, a signature summary report appears in the top right corner of the screen.
Click **valid** to view information about each signed field.

![Signature Summary Report for an MS Office Document](image)

A Signature Details window similar to the following is displayed.
Validating Signatures

You can view the validity status of each signed signature field.

- In a PDF document:
  Each signature displays in its top left corner an indication of the validity of the signature (see Figure 33). A green checkmark indicates the signature is valid, and a red X indicates it is invalid.

  In addition, clicking/tapping a signature field displays the Signature Details window (Figure 34), which lists the status of the signature and provides information about the user certificate.

- In a Microsoft Office 2007/2010/2013/2016 document:
  A signature summary report appears in the top right corner of the screen.

  Click valid in the summary report to view information about each signed field, including its validity status (see Figure 36).
Post-Signing Actions

When you finish viewing and signing the entire document, click/tap **Done**. The following window appears.

![What would you like to do next?](image)

*Figure 37 What Would You Like to Do Next Window*

You can select any of the following options:

- **Save in <file storage service>** – Save a signed copy of the file back in the Cloud-based file storage service. The copy is a new file, and must be given a new name (you cannot overwrite the original file). This option appears only if you retrieved the file from a Cloud-based file storage service. The name of the option indicates the specific file storage service from which you retrieved the file.

- **Download copy** – Download a signed copy to your computer’s Downloads folder. The copy is a new file but its name is the name of the original file. This option may not appear when using a mobile device (for example, on iPads).

- **Send via email** – Send the file as an attachment using your default mail application. When you select this option, an email message frame appears within the window. You can enter multiple recipients, separated by semicolons, in the To field. Enter any text in the Subject field and main body of the email. Clicking/tapping Send will send the message with the signed document added as an attachment.

Starting from DocuSign SA Web App version 7.1, an auto-complete mechanism can be used to assist in filling in previously-used email addresses.
Using a Volatile Graphical Signature

Starting from DocuSign SA Web App version 8.0, it is possible to add to the signing ceremony a graphical signature that is not fetched from the DocuSign Signature Appliance. This graphical signature is entered by the user during the signing ceremony, and incorporated into the digital signature.

The graphical signature, called a volatile signature, is available for any signature operation performed on the document, and deleted when the document is closed.

To enable using volatile signatures:

Enable volatile signatures for Web App by setting the value of the GrSigSelMode key.

1. Enable volatile signatures for a specific user by setting the value of Graphical signature selection mode.
User Experience when using Volatile Graphical Signatures

- If the DocuSign Web App is configured to use only volatile graphical signatures, the user will get the following window as part of the signing ceremony. The user can then use his or her PC or mobile device to sketch the graphical signature.

![Signature Appearance](image)

*Figure 39 Sign using only a Volatile Signature*

- If the DocuSign Web App is configured so that the user can either select a graphical signature kept inside the DSA, or sketch a volatile signature, the following screen is presented as part of the Signing Ceremony.
**Figure 40  Sign using a Volatile Signature if desired**

**Signing in a Point of Sale Mode of Operation**

You can set your account to work in a Point of Sale (POS) mode of operation. In this mode of operation, each empty signature field can be signed electronically (intended for external
customers), or digitally (intended for the agent, a DocuSign SA user). POS mode supports only PDF files.

Once the document is digitally signed, it is not possible to add new electronic signatures (customer signatures). Therefore make sure the document is first signed electronically wherever needed before signing it digitally.

For instructions on how to set your account to work in a Point of Sale mode of operation, refer to EnablePOSMode.

**To sign in POS mode:**

1. If you wish to sign a pre-existing field, skip to step 2. If you are creating a new signature field, perform the following:
   - Locate the hovering signature-creation frame (see Figure 28). If the signature-creation frame is not displayed, click Add another signature.
   - Click/tap Sign in the signature frame. The POS Signing window appears.

![POS Signing Window](image)

   **Figure 41  POS Signing Window**

   The page displays two tabs: **Customer** tab and **Agent** tab. Customers should use the Customer tab, which creates an electronic signature, and the DocuSign SA user should use the Agent tab, which creates a digital signature.

   By default the **Customer** tab is selected.
3. To create a Customer signature:
   a. Instruct the customer to electronically sign in the signature area (typically using the agent’s tablet or iPad).
   b. Click/tap the **Customer** tab if you wish to add additional information to the electronic signature.

   ![Signature](image)

   *Figure 42  POS Customer Signature Additional Fields*

   c. Optionally add a reason and title to the signature, and define the signature appearance.
   
   d. Click/tap **Apply**.
4. To create an Agent signature:
   a. Click the **Agent** tab. A Signature Appearance window appears.

   ![Signature Appearance Window](image)

   *Figure 43  POS Agent Signature Appearance Window*

   b. Select a graphical signature from the carousel. Alternatively, click the radio button to **Sign without a graphical signature**.

   Note that a user’s graphical images and default settings are kept in the user’s account in the DocuSign SA appliance.

   c. Optionally, add a reason and title to the signature, and define your signature appearance. To do so, click/tap **Add reason and change signature appearance**. The window refreshes to display the following:
d. In you created a new field, edit the appearance as desired by selecting whether to show the date and time, the signer's title, etc.

If you are signing a pre-existing field, you can only enter the reason and title if those are required.

e. Click/tap **Apply**.
The document will display a digital signature, similar to the following:

![Sample Signature](image)

*Figure 45 Sample Signature*

5. You can optionally click the signature to view its details. Refer to [Viewing Signature Details](#).
Chapter 5: Using the DocuSign SA Web App as part of the DocuSign Cloud Signature Workflow

General

This chapter describes the workflow of DocuSign cloud users who use Web App as part of the DocuSign cloud signing process, and the settings required to enable this scenario.

**Note:** The DocuSign SA Web App relevant screens are available in the following languages, if any of these language are set in the DocuSign Cloud:
- French
- German
- Italian
- Japanese
- Spanish
- Portuguese-Brazil
- Greek
- Dutch

Flow of Operations

The redirection from the DocuSign Cloud to the local DocuSign Signature Appliance and back, occurs as follows:

1. The end user connects to the DocuSign Cloud to start signing transactions or documents in the cloud. The user’s settings and configuration define that the signatures of the document will be based on digital signatures that are created using a local DSA of the organization or TSP (Trust Service Provider). A TSP is a trust service that enables end users to access their signature keys remotely in a protected manner.
   As part of the configuration, the URL of the DocuSign SA Web App is defined in DocuSign.
When the user finishes to click **Sign** on all signature fields, the user presses **Continue**. This starts the interaction of the DocuSign Cloud with the DocuSign SA Web App in order to enable the user access to his/her account in the local DocuSign Signature Appliance deployed inside the organization or TSP.
3. The user is presented with a DocuSign SA Web App frame (shown below), for logging in to the DocuSign Signature Appliance. The user needs to enter his/her user ID and password.

Note that if the organization uses a different authentication method, such as Kerberos or SAML, the user may automatically be logged on to the DocuSign Signature Appliance, depending on his local workstation or mobile device configuration.

**Note:** Starting from version 8.5, if the user's password has expired, and the DocuSign Signature Appliance is deployed in Directory Independent mode, the user will have the option of changing his/her password and then continuing with the signing process.
After a successful authentication process, the DocuSign Web App extracts from the DocuSign cloud the relevant hash value that corresponds to the required digital signature, which was created using the user’s key inside the DocuSign Signature Appliance. A window similar to the following appears:

![Login Page](image)

Figure 47  DocuSign SA Web App Login Page
5. If the document has several signature fields, the user may need to enter credentials for each signature field. This depends on how the DocuSign Signature Appliance is deployed. For example, if the DocuSign Signature Appliance is deployed in a Common Criteria mode of operation, the user may need to enter a different One Time Password (OTP) for each digital signature operation.
6. After the user finishes all digital signature operations, the user is redirected back to the DocuSign Cloud to finalize the operation. The user can then choose to download the signed PDF document as shown below:

![Figure 49 Signing Completed](image)

7. Using a PDF reader or Prepare&Sign, the user can inspect the digital signatures inside the PDF document.

**Note:** As part of version 8.5, the workflow can trigger an eSealing operation that can perform several signature operations as a batch processing job, without requiring any interaction with the end user.

**Enabling use of DocuSign Cloud in the Organizational Workflow**

This section describes the tasks required if you wish to use the DocuSign Cloud for your organizational transactions and workflows.

**DocuSign Cloud Settings**

To configure the DocuSign Cloud settings, contact DocuSign support. They will:

- Enable DocuSign to interact with the DocuSign Signature Appliance through the DocuSign SA Web App. This will enable your DocuSign account to redirect the communication with DocuSign to the local Web App and back to the DocuSign Cloud.
- Configure the URL of Web App at the DocuSign cloud to enable this redirection.
DocuSign SA Web App settings
Refer to the section *Defining DocuSign Cloud Settings.*
Chapter 6: Adding Graphical Signatures and Modifying User Settings

Using DocuSign SA Web App, you can add graphical signatures and modify various user settings such as your signature appearance.

Adding Graphical Signatures

You can add personal graphical signatures using any of the following methods:

♦ Sketching  
♦ Typing  
♦ Uploading a graphic file

The graphical signatures you add are kept in your user account inside the DocuSign SA appliance.

To add graphical signatures:

1. Open the drop-down list adjacent to your user name, in the top right corner of the DocuSign SA Web App window (Figure 50).

2. Select My Signatures.
3. Add a signature using any of the following methods:

- **Sketch** – Use the mouse in a desktop computer, or sketch with your finger upon a mobile device. You can then save or cancel the sketch.
- **Type** – Select a font from among nine fonts and type in the desired text. You can then save or cancel the signature.
- **Upload** – Upload a jpeg or bmp file.
Modifying User Settings

You can modify and update your user settings that affect the DocuSign SA Web App session. All these parameters are kept in your user account inside the DocuSign SA appliance.

To modify user settings:

1. Open the drop-down menu adjacent to your user name, in the top right corner of the DocuSign SA Web App window (Figure 50).
2. Select Settings.

3. Optionally, set any of the following:
   - **Title** – Enter a default title.
   - **Change Password** – Change your password, if relevant (for example, if DocuSign SA is installed in a Directory Independent environment).
   - **Default Signature Appearance** – Specify the appearance of signature fields you will be creating. Set the appearance by selecting or deselecting the following:
     - **Show graphical signature**
     - **Show date and time**
     - **Show my name**
     - **Show my title**
     - **Show signature reason**
 Reasons for Signing – You can maintain a list of reasons that are displayed whenever a reason needs to be added to a digital signature. You can add reasons to the list and remove reasons from the list, edit existing reason and select a reason to be the default reason.

 Advanced Settings – Define the following advanced settings, applying to all signature fields you will be creating:
  - Signature type – Defines whether the signature type will be Digital, Electronic, or Invisible Digital.
  - Date format – Defines the date format of the signatures.
  - Time format – Defines the time format of the signatures.
  - Display format – Defines whether only a date is displayed as part of the digital signature, or both date and time are displayed.

 Operational Mode – Define whether DocuSign SA Web App works in standard operation mode or in Point of Sale mode:
  - Standard – The regular mode of operation described in this guide.
  - Point of Sale – A mode of operation where external users supply an electronic signature while being overseen by an organizational representative, such as a Sales representative, who digitally signs the overall transaction. For more information, refer Signing in a Point of Sale Mode of Operation.

 Graphical signature selection mode – Enable the user to add a volatile graphical as part of the signing ceremony:
  - My Signatures – The regular mode of operation, where the user needs to select a graphical image from the graphical images stored inside the DocuSign Signature Appliance.
  - One Time Signature – The user sketches a volatile graphical signature as part of the signing ceremony (refer to Volatile Graphical Signature).
  - Both – the user can choose to either select a graphical signature or sketch a volatile graphical signature.
Logging off the DocuSign SA Web App

- Starting from DocuSign SA Web App version 7.5, a user can log out of his/her session by clicking the Log Off option.

![Log Off the DocuSign SA Web App](image)

After selecting the Log Off option:

- When DocuSign SA Web Agent is integrated with a Document Management system, the user is directed back to the Document Management System.
- If an external Identity Provider is used, you can configure to which the web page the user is redirected upon logoff. For information, refer to the description of LogoutRedirectURL. In any case, whenever you logoff and an external Identity Provider is used, the external Identity Provider is notified regarding the user logoff operation.
- In all other cases, the DocuSign SA Web App Login page is displayed.
Chapter 7: Signing Documents from Cloud-based File Storage Services

As mentioned above, the document that needs to be signed can be extracted from a Cloud-based file storage service, signed using the DocuSign SA Web App, and then loaded back to the Cloud-based file storage system or sent via email.

This chapter describes the end user experience for the following Cloud-based file storage services: Dropbox, OneDrive, Google Drive, and Box.

In Box and Google Drive, you can sign with the DocuSign SA Web App from within Box or Google Drive. That is, after you select a certain document, you are redirected to the DocuSign SA Web App to digitally sign the document, and after the document is signed, a signed copy is saved to either Box or Google Drive.

Note that your organization needs to perform some integration tasks to register the DocuSign SA Web App as an application in the relevant Cloud-based file storage service. These integration tasks are described in Integrating with Cloud-Based File Storage Services.

Signing Documents from Dropbox

When you first attempt in DocuSign SA Web App to sign documents that are stored in Dropbox, you are asked to confirm that you grant DocuSign SA Web App access to your data:

![Figure 54 Confirm DSA Access to Dropbox](image)

After you click **Allow**, you will receive an email notifying you that the DocuSign SA Web App application will interface your Dropbox account.

You can now proceed to view, sign and save the file in DocuSign SA Web App.
Signing Documents from OneDrive

When you first attempt in DocuSign SA Web App to sign documents that are stored in OneDrive, you are asked to confirm that you grant DocuSign SA access to your data:

![Confirm DSA Access to OneDrive](image)

Figure 55  Confirm DSA Access to OneDrive

After you click Yes you can proceed to view, sign and save the file in DocuSign SA Web App.

Signing Documents from Google Drive

When you first attempt in DocuSign SA Web App to sign documents that are stored in Google Drive, you are asked to confirm that you grant DocuSign SA access to your data.

After you click Allow Access you can proceed to view, sign and save the file in DocuSign SA Web App.
Invoking the Signature Operation from Within Google Drive

Every end-user can configure Google Drive so that he/she can invoke a digital signature operation upon a selected document from within Google Drive.

In order to have CoSign appear in your Google Drive menus, you need to connect the CoSign Digital Signatures application to Google Drive, as described in this section.

**Note:** As a prerequisite, a DocuSign SA administrator must first configure the Google Drive CoSign application, as described in *Enabling Users to Sign from Within Google Drive*.

To add CoSign to Google Drive Menus:

1. Click a file, select **Open with** and select **Connect more apps**.
2. Search for the **CoSign Digital Signatures** application.
   - The search results window appears:

   ![Figure 56 Connecting the DocuSign SA Web App Application in Google Drive](image)

3. Click **+ Connect**. The following window appears:

   ![Figure 57 DocuSign SA Web App Connected to Google Drive](image)
4. Select whether to make **CoSign Digital Signatures** application the default application for files it can open.

5. Click **OK**.

6. Restart your browser.

The CoSign Digital Signatures application now appears in the **Open with** menu of every PDF, Word or Excel file you select in Google Drive, as shown in Figure 58:

![Figure 58 Opening CoSign from Within Google Drive](image)

Selecting **CoSign Digital Signature** redirects you to the DocuSign SA Web App with the selected file appearing in the window. After finishing the signature ceremony, the updated signed document is written back to the Google Drive storage. You can select whether the updated signed file overwrites the original file or is saved as a new file.
Signing Documents from Box

When you first attempt in DocuSign SA Web App to sign documents that are stored in Box, right after logging in to Box you are asked to confirm that you grant DocuSign SA Web App access to your data:

![Confirm DSA Access to Box](image)

*Figure 59  Confirm DSA Access to Box*

After you authorize access you can proceed to view, sign and save the file in DocuSign SA Web App.
**Invoking the Signature Operation from Within Box**

Every end-user can configure Box so that he/she can invoke a digital signature operation upon a selected document from within Box.

In order to have DocuSign SA appear in your Box menus, you need to add the *DocuSign SA Digital Signatures* application to your apps, as described in this section.

As a prerequisite, a DocuSign SA administrator must first configure the Box DocuSign SA application, as described in *Enabling Users to Sign from Within Box*.

**To add DocuSign Signature Appliance to the Box menus:**

2. Go to your *Apps* section.
3. Search for the *DocuSign Signature Appliance* application.
   The search results window appears:

   ![Figure 60 Searching for the DocuSign SA App in Box](image)

4. Click the *DocuSign Signature Appliance* icon. The following window appears.
5. Click + Add.
The DocuSign SA Digital Signatures application now appears in the right-click menu and **Integrations** menu of every file you select in Box, as shown in Figure 62.

![DocuSign Signature Appliance Web App](image)

**Figure 62 Opening DocuSign Signature Appliance from Within Box**

Selecting **Digitally Sign with DocuSign Signature Appliance** redirects you to DocuSign SA Web App with the selected file appearing in the window. After finishing the signature ceremony, the updated signed document is written back to the Box storage. You can select whether the updated signed file overrides the original file or is saved as a new file.
Chapter 8: Signing Office 365 documents using DocuSign SA Web App

The DocuSign SA Web App can be used for signing documents that are located inside the Office 365 SharePoint Cloud.

In this scenario, users of the organization that are connected to the Office 365 SharePoint Cloud select a document for signing, are redirected to the local DocuSign Web App deployment, and sign the document using the local DocuSign Signature Appliance. The signed document is saved back to Office 365.

This chapter describes how to setup the two additional software components that enable using the DocuSign SA Web App to sign documents located in the Office 365 SharePoint Cloud.

Enabling Signing O365 Documents using the DSA Web App

Required Software Components

In addition to the deployment of the DocuSign SA Web App and the DocuSign Signature appliance the following two software components are required:

- **DocuSign SA WebApp Proxy**
  A server runs the DocuSign SA WebApp Proxy which is configured to run with a specific DocuSign SA WebApp and with a specific URL of a server running DocuSign SA Web App.
  The Proxy uses the Web App in a Web Agent mode of work. In this scenario, the Web Agent retrieves the document from the O365 SharePoint Cloud and sends it to the DocuSign SA Web App through the API channel.
  When the user finalizes the signature operation, the document is replied back to the O365 SharePoint Cloud.
  Documents such as PDF documents, DOCX and .XLSX can be signed.

- **DocuSign SA O365App**
  This is an application that is deployed in Office 365 and can be used to forward the document signer’s request to the DSA App Proxy.

Step 1: Deploying and Configuring the DocuSign WebApp Proxy

The DocuSign SA WebApp Proxy application can be installed on the same server on which the DocuSign SA Web App is deployed.

1. In the server on which the DocuSign SA Web App is deployed, run CoSignAppProxy.exe to extract the CoSign App Proxy files to the default directory
   C:\InetPub\DSAAppProxy.

2. Create a DocuSign SA Web App Proxy site as follows:
   a. In the IIS Manager, click **Add Web Site**.
   b. Set the site name to **DSAAppProxy**.
c. Set the physical directory to `C:\InetPub\DSAppProxy`.

d. Set the application pool to **ASP.NET v4.0 (Integrated)**.

e. Add only HTTPS (443) binding, and select an appropriate certificate that matches the DNS name for this site.

3. If you are installing the DocuSign SA App Proxy, we recommend installing the DocuSign SA Web App site in a virtual directory and not in the main DNS of the Web Site, while deploying the DocuSign SA App Proxy in the main web site address.

   To adapt the DocuSign SA Web App as recommended:

   a. In the IIS Manager, right click the DSAppProxy site, and select **Add Virtual Directory**.

   b. Set the alias to `DSAWebApp`.

   c. Set the physical directory to `C:\InetPub\WebApp`.

   d. Click **OK** to save the settings.

   e. Right click the CoSignWebApp directory, and select **Convert to Application**.

   f. Click **OK** to save the settings.

4. Edit the CoSign Web App web.config file under `C:\InetPub\WebApp`, so that the Web App is installed in a web agent mode of operation, as follows:

   a. In the `APISettings` section add `<add key="EnableAPIRequests" value="true"/>`.


**Step 2: Registering the DocuSign SA O365 App in the Office 365 SharePoint Cloud**

Perform the following to register the DocuSign SA O365 App in the Office 365 SharePoint cloud.

1. Go to your Office 365 website’s `/_layouts/15/appregnew.aspx` page (for example [https://dsaqa.sharepoint.com/sites/tests/_layouts/15/appregnew.aspx](https://dsaqa.sharepoint.com/sites/tests/_layouts/15/appregnew.aspx))

2. Generate a new Client ID and Secret as follows, and save them somewhere. You will need them at a later stage.
Figure 63 Generating a New Client ID and Secret

a. Choose a **Title** for the app and make sure you provide the correct **App Domain**. This should be the DNS entry that was defined for the DSA App Proxy installation, in *Step 1* of the process.

b. Click **Generate** to generate the Client Id, and click **Generate** to generate a Client Secret.

A success message similar to the following appears

```
The app identifier has been successfully created.
Client Id: 24358450-e678-464a-9630-6c67468899
Client Secret: EZxwN-trpZDpXfLsqNldU8Tizb92Z0/ptTsW4CAAl=
Title: CoSign
App Domain: www.cosign.com
Redirect URI:
```

Figure 64 App Identifier Successfully Created
3. Go to **O365 → SharePoint → AppCatalog → Apps for SharePoint** to view the list of all installed apps, and make sure the app was properly registered.

```plaintext
Figure 65  List of Registered Apps
```

### Step 3: Modifying and Deploying the DocuSign SA O365App

1. Run the following command in a Windows CMD command line window:
   ```plaintext
   ConfigCoSignApp.exe "<file path for DSAAppl file>" "<DSAAppProxy site URL>" "<Client ID>"
   ```
   For example:
   ```plaintext
   ```
   In the example:
   - **DSAAppl.app** is the file path for the DSAAppl file
   - **https://proxy.org.com** is the DSAApplProxy URL
   - **12345678-abcd-abcd-abcd-123456789abc** is the Client ID

2. A new **DSAAppl.app** file is created with the updated values. The original file will now be named **DSAAppl - Original.app** and can be saved as a backup.

3. Upload the app to the organizational app catalog. For help, contact DocuSign support.
Step 4: Updating the DSAApplProxy Settings

In this step you need to edit the DSA App Proxy web.config file under C:\InetPub\DSAApplProxy and add the Client ID and Secret that were created in Step 2 during DocuSign SA O365 App registration in SharePoint, as well as add the CoSign Web App site DNS.

1. In the <appSettings> section, add the following keys:
   
   ```xml
   <add key="ClientId" value="Generated Client ID" />  
   <add key="ClientSecret" value="Generated Client Secret" />  
   <add key="COSWAPath" value="https://<DNS>/DSAWebApp/"/>
   ```

   The DSAAppl should now point to the custom DSA App Proxy and DSA Web App applications.

User Signature Flow for Signing Documents located in the Office 365 SharePoint cloud

After performing all the configurations described in Enabling Signing O365 Documents using the DSA Web App, users can sign documents inside the O365 SharePoint Cloud using the DSA Web App. The operational flow of the user is as follows.

1. Go to the SharePoint Office 365 Cloud deployment and select a document to be signed.
2. Right click the document, and select the Sign operation.
Alternatively, use the menu bar for this operation.

You will be directed to the DSA App Proxy, which will use the DocuSign SA Web App for the signing operation.

3. You can also use the **Review Signatures** option. In this case you will be redirected to the DSA App proxy and then to the DocuSign SA Web App for the purpose of signature validation.
Chapter 9: Configuring the DocuSign SA Web App

This chapter describes the DocuSign SA Web App customizations that a DocuSign SA admin can perform. These include:

- Modifying various DocuSign SA Web App parameters. All DocuSign SA Web App related configuration is performed in the web.config file, located in the home directory of the DocuSign SA Web App application.
- Configuring DocuSign SA Web App in passive ADFS mode. This is a prerequisite for service providers who wish to offer digital signature services to organizations that deploy Active Directory with ADFS.
- Registering the DocuSign SA Web App as an application in the various Cloud-based file storage services. This is a prerequisite if you wish to use DocuSign SA Web App to sign files stored in a Cloud-based file storage system.

Modifying DocuSign SA Web App Parameters

You can modify various DocuSign SA Web App parameters by editing the web.config file, located in the home directory of the DocuSign SA Web App application.

**Important:** Because the web.config file contains all the DocuSign SA Web App definitions, it is recommended to back up the file before making any changes to it.

Modifying Image Setting Parameters

The default Image Settings configuration is as follows:

```xml
<ImageSettings>
    <!-- add here Image settings -->
    <add key="ReducePercent" value="100"/>
    <add key="ReduceIterations" value="1"/>
    <add key="GrSigSelMode" value="0/1/2" />
</ImageSettings>
```

The ReducePercent and ReduceIterations parameters provide the instructions for shrinking an uploaded graphical signature image if its size is larger than 30K. ReducePercent specifies by how much to shrink the image at each iteration. ReduceIterations specifies the maximum number of iterations.

The GrSigSelMode parameter enables users to insert a [Volatile Graphical Signature](#) as part of the signing ceremony. This key can have the following values:

- 0 – Only graphical signatures from the appliance are enabled (Default)
- 1 – Only volatile graphical signatures are enabled
- 2 – Both types of graphical signatures are enabled
Defining SMTP Settings

The SMTP Settings section is by default empty.

```xml
<SMTPSettings>
    <!-- add here SMTP settings-->
</SMTPSettings>
```

You can enter the following parameters into the section. If the parameters already appear in your web.config file, you can change their values as needed.

- To set the SMTP host IP/Name:
  ```xml
  <add key="SMTPHost" value="THE_VALUE" />
  ```

- To set the SMTP Port:
  ```xml
  <add key="SMTPPort" value="THE_VALUE" />
  ```

- To set the sender address:
  ```xml
  <add key="SMTPMailFrom" value="THE_VALUE" />
  ```

- To set the mail message body:
  ```xml
  <add key="SMTPMessageBody" value="THE_VALUE" />
  ```

- To set the mail Subject:
  ```xml
  <add key="SMTPMailSubject" value="THE_VALUE" />
  ```

- To set the mail reply mode, enter one of the following:
  - None – no Sender name or email address are listed in the email.
  - Admin – the Sender name and email address are as defined in the SMTPSenderName and SMTPSenderAddress parameters.
  - User – the Sender name and email address are supplied in the email dialog.
    ```xml
    <add key="SMTPMailSenderMode" value="<None|Admin|User>" />
    ```

- To set the email Sender Address (relevant only if SMTPMailSenderMode is set to User):
  ```xml
  <add key="SMTPSenderAddress" value="THE_VALUE" />
  ```

- To set the email Sender Name (relevant only if SMTPMailSenderMode is set to User):
  ```xml
  <add key="SMTPSenderName" value="THE_VALUE" />
  ```

Defining Extended Authentication Settings

The Extended Authentication Settings section is empty by default.

```xml
<ExtAuthSettings>
    <!-- add here extended authentication settings-->
</ExtAuthSettings>
```

If DocuSign SA is configured to use extended authentication, configure extended authentication parameters in the ExtAuthSettings section.
To force Prompt for sign:

\[<\text{add key}="\text{PromptForSign" value}="\text{No/Each/Once"} />\]

- No – The user will not be requested to provide a password for every digital signature operation. This is the default behavior.
- Each – The user will be required to enter a signature password for every signature operation (as illustrated in Figure 68). This option is suitable for “event-based OTP”.
- Once – The user will be required to enter his/her password once upon login. This option is suitable for “time-based OTP”.

**Note:** If **Prompt For Sign** is NOT enforced by the DocuSign Signature Appliance, you will need to define the appropriate parameters in the DocuSign SA Client. Refer to the DocuSign SA Client documentation.

Figure 68  Example of Password Requested Upon Every Signature Operation

To force extended password usage:

\[<\text{add key}="\text{ExtPwdRequired" value}="\text{Logon/OTP/Both"} />\]

- Logon – No extended authentication mechanism is used. The sign password is the login password. In this case only one password text-box is presented in the login screen for entering the password, and the password is used for both login and for signing. If you set **PromptForSign** to Each, the user will be asked to reenter his login password during each signing process. This is the default value.
◆ **OTP** – An extended authentication mechanism is used, such as an OTP (One Time Password) device. In this case, two password text-boxes are presented in the login screen, one for the login password and one for the extended authentication password, as shown in Figure 69. If you set `PromptForSign` to Each, the user will be asked to enter the extended authentication password during each signing process. In this case, you must make sure that the DocuSign SA appliance is configured to use extended authentication.

◆ **Both** – The sign password is a concatenation of the login password and extended authentication password entered in the login screen. In this case, two text-boxes are presented in the login screen, one for the login password and one for the extended authentication password. If you set `PromptForSign` to Each, the user will be asked to enter only the extended authentication password during each signing process, and the application will then concatenate that extended authentication password with the login password, and pass it as the sign password.

◆ To enable providing an extended password with the login page:

```xml
<add key="DisplayExtPwdOnLogin" value="true/false" />
```

This mode of work is relevant only when used in conjunction with the DocuSign Cloud.

◆ **True** – The user must enter the extended password in his login page. This extended password will be used for every signature operation.

◆ **False** – The login page does not include a field for entering an extended password. This is the default value.
Defining API Integration Settings

By default, the API Settings section is empty. This section enables applications to redirect users to the DocuSign SA Web App for the purpose of signing documents during their session with the application.

```
<APISettings>
  <!-- add here API integration settings -->
</APISettings>
```

The following configuration parameters can be defined (for more information, refer to section VI of the DocuSign SA Programmer Guide).

- To enable API request handling:
  ```
  <add key="EnableAPIRequests" value="true/false" />
  ```
  - false means that no application can interact with the DocuSign SA Web App. This is the default setting.
  - true means that an application can redirect their users to communicate with the DocuSign SA Web App for the purpose of signing documents.

- To enable Google Drive integration (default=true)
  ```
  <add key="EnableGoogleDriveIntegration" value="true/false" />
  ```
  For more information, refer to Enabling Users to Sign from Within Google Drive.
To enable Box integration (default=true)

<add key="EnableBoxIntegration" value="true/false" />

For more information, refer to Enabling Users to Sign from Within Box.

To define an API request session timeout (default=30 min.)

<add key="APIRequestTimeOut" value="30" />

If the application does not collect back the signed document after this timeout, the document will be erased from the memory of the DocuSign SA Web App.

**Defining General Settings**

The following general settings can be set in the General Settings section.

- To enable working with the DocuSign cloud, set the *EnablePEN* parameter to true (default = false). Only then the section Defining DocuSign Cloud Settings is relevant.

<add key="EnablePEN" value="true/false" />

- To enable working with Web App, set the *EnableAPP* parameter to true (default = true). If this parameter is set to false, make sure that the DocuSign SA Web App is interacting with the DocuSign cloud by setting *EnablePEN* to True, otherwise the entire Web App deployment is useless.

<add key="EnableAPP" value="true/false" />

- To enable using localized text as part of the DocuSign cloud interface, set the *EnableLocalization* parameter to true (default = true).

<add key="EnableLocalization" value="true/false" />

- To enable using an Alternate web site, mainly in the case that your organization uses Kerberos authentication but there are user devices that do not have Kerberos capability (such as mobile phones), you can define an Alternate Web Site to which users who are not able to connect using Kerberos are redirected. Note that in this case, the Alternate Web Site should be configured in Anonymous Windows Authentication mode, as described in Using an Alternate Web Site.

To enable using an Alternate Web Site, add the following entry:

<add key="AlterWebAppSiteURL" value="THE_VALUE" />

This parameter is relevant in cases where DocuSign SA is installed in an Active Directory environment.

- The following parameters specify whether a link exists for the Forgot My Password functionality, as well as its name and URL:

  - To activate a "Forgot my password" link:

    <add key="ForgetPwdVisible" value="true/false" />

  - To set the "Forgot my password" link text:

    <add key="ForgetPwdText" value="THE_VALUE" />

  - To set the "Forgot my password" link URL:

    <add key="ForgetPwdUrl" value="THE_VALUE" />
The following parameters specify whether a link exists for the Sign Up functionality, as well as its name and URL:

- To activate a "Sign up" link:
  
  ```
  <add key="SignUpVisible" value="true/false" />
  ```

- To set the "Sign up" link text:
  
  ```
  <add key="SignUpText" value="THE_VALUE" />
  ```

- To set the "Sign up" link URL:
  
  ```
  <add key="SignUpUrl" value="THE_VALUE" />
  ```

The following additional parameters can be setup:

- To enable or disable the change password functionality:
  
  ```
  <add key="EnableChangePwd" value="true/false" />
  ```

  **Note:** In Directory Independent installations, the change password option is enabled by default.

- To set the default domain name (default value = ")
  
  ```
  <add key="DefaultDomain" value="domainname" />
  ```

  This parameter is relevant in cases where DocuSign SA is installed in an Active Directory environment. If the default domain name is specified using `DefaultDomain`, the user will not be required to supply the DocuSign SA Domain Name.

- To enable working with AD (default value = "false")
  
  ```
  <add key="ADMode" value="true/false" />
  ```

  This parameter is relevant in cases where DocuSign SA is installed in an Active Directory environment. If the DocuSign SA Client is configured to use DocuSign SA in an Active Directory environment, the DocuSign SA Web App will work even without setting `ADMode` to true. However, if the `ADMode` parameter is set to true, the GUI of the DocuSign SA Web App will be slightly changed, for example, the user settings will not include the change password option.

The following additional parameters are relevant for DocuSign SA Web App V7.1 and above:

- To enable or disable the Point of Sale (POS) functionality globally for all users:
  
  ```
  <add key="EnablePOSMode" value="true/false" />
  ```

  The default value is false, meaning POS mode is not globally enabled. To enable POS mode for an individual user, refer to Operational Mode in the Modifying User Settings section.

  When the value is true, all users have the capability to execute a POS signature process.

- To enable or disable the new Office capability, in which Office 2007/2010/2013 documents are not converted to PDF if they contain one or more signature fields:
  
  ```
  <add key="OfficeBackwardCompatibility" value="true/false" />
  ```

  The default value is false which instructs the system not to convert Word and Excel 2007/2010/2013 documents to PDF if they contain one or more signature fields.
The following additional parameters are relevant for DocuSign SA Web App V7.5 and above:

- As part of the functionality of DocuSign SA Client/DocuSign SA Signature APIs version 7.5, it is possible to handle Signature Locators as signature fields (for more information about Signature Locators, refer to the DocuSign Signature Appliance Signature APIs Developer's Guide). If you enable this option, the DocuSign SA Web App will show Signature Locators as signature fields.

To enable or disable the new functionality use the following parameter:

```xml
<add key="EnableLocators" value="true/false" />
```

The default value is `false`, which instructs the system not to handle Signature locators.

- The following additional parameters are relevant for DocuSign SA Web App V8.0 and above:

  Use the `EnableIECompatibilityMode` parameter to enable accessing Web App while running Internet Explorer 11 in Compatibility Mode, which simulates an older version of Internet Explorer.

  ```xml
  <add key="EnableIECompatibilityMode" value="true/false" />
  ```

  The default value is `false`, which instructs the system that when accessing WebApp while running Internet Explorer, only Internet Explorer 11 can be used, and Compatibility Mode cannot be used.

- As part of the functionality of DocuSign SA Client/ DocuSign SA Signature APIs version 7.5, it is possible to redirect the user to a specific Logoff URL if either an External Identity Provider is used or the Active Directory Kerberos Ticketing mechanism is used. To do so, use the following parameter:

  ```xml
  <add key="LogoutRedirectURL" value="<URL>">
  ```

  If a URL value is supplied, the user will be redirected to that URL upon clicking the Log Off option.

- The parameter `UsrNamePlaceholder` can be used for the label displayed near the user name entry field. Depending on the format of the local user ID, you can specify the name “User Name”, “Email address” or any other label that will help the local user to understand which user identity he is required to enter.

  ```xml
  <add key="UsrNamePlaceholder" value="Username" />
  ```

---

*Figure 70  Label of Username Field*
The parameter **ClearPolicy** can be used to define the clear-signature-field policy. This is relevant for cases where a signature is created and signed. The newly created signature will be subject to the Clear policy set in **ClearPolicy**. The value can be any of the following:

- **Anyone** – Any user can clear a **signature** field. This is the default value.
- **NoOne** – No user can clear a signature field.
- **Signer** – Only the signer of the signature field can clear the signature field.

Note that the clear action itself is not available in WebApp, but only in Prepare&Sign.

```xml
<add key="ClearPolicy" value="<Anyone|NoOne|Signer/>" />
```

The parameter **CertifyMode** can be used to define how the end users can use the **Certify** option in WebApp:

- **Disable** – The **Certify** option is not enabled in WebApp. This is the default option.
- **Enable** – The **Certify** option is enabled in WebApp. As part of the signature ceremony, the user can perform a certify operation when applicable.
- **Enforce** – The end user can only perform the **Certify** operation, and no other operation. Consider this option well if you plan on setting it.

```xml
<add key="CertifyMode" value="<Disable|Enable|Enforce/>" />
```

The parameter **CertifyType** can be used to define the default **Certify** type. The available options include:

- **1** – No changes are allowed. This is the default value.
- **2** – Form filling and signing is allowed
- **3** – Annotations, Form filling and Signing are allowed.

```xml
<add key="CertifyType" value="<1|2|3/>" />
```

The parameter **SrvLocation** can be used to specify the geographical location of the WebApp and Signature Appliance system. This textual parameter is used to indicate the geographical location where WebApp is deployed.

```xml
<add key="SrvLocation" value="<two letter country code/>" />
```

**Defining DocuSign Cloud Settings**

Define the following **DTM Settings** and **Login Policy Settings** to allow the DocuSign SA Web App to properly interact with the DocuSign Cloud.

**DTM Settings section**

The following two parameters enable generating a secure communication between the DocuSign SA Web App and the DocuSign Cloud. The Communication channel is based on TLS and is on outbound communication between the DocuSign SA Web App and the DocuSign Cloud. The two parameters define the client security measures when DocuSign SA Web App interacts with the DocuSign Cloud. You will need to receive these values from DocuSign Support.
The Web App application Client ID as was defined by account server configuration

```
<add key="DTMClientID" value="" />
```

The Web App application Client Secret as was defined by account server configuration

```
<add key="DTMClientSecret" value="" />
```

To enable setting Long Term Validation information as part of the digital signature, you can use the DTMDssMode parameter, with any of the following options:

- None – DSS mode is not in use.
- OcspCrl – OCSP is used for the user certificate level, and CRL is used for the rest of the certificates in the chain.
- CrlOnly – Only CRLs are used.
  Set the value as follows:

```
<add key="DTMDssMode" value="None|OcspCrl|CrlOnly" />
```

Login Policy Settings section

In the Login Policy Settings section you can set certain policies related to the login pages that are presented to the user as part of the integration of the DocuSign cloud and the local interaction with the DocuSign Appliance through the DocuSign SA Web App.

- Enforce username given by the DocuSign Cloud:
  - If true, the user must logon to the DocuSign Signature appliance with the same user name used when connecting to the DocuSign cloud, and cannot change it
  - If false, the user can logon with any other user name as long as that user exists in the DSA (true by default)

```
<add key="EnforceUserName" value="true/false"/>
```

- Display username given by the DocuSign Cloud:
  - If true, the user name input box will be prefilled with the name given when connecting to the DocuSign cloud (true by default)
  - If false, the user name input box will not be prefilled.

```
<add key="DisplayUserName" value="true/false "/>
```

- Keep user session at the end of the sign operation:
  - If true, the user session is kept open, for the length of session timeout period, at the end of a sign operation so the user need not login again to sign the next document
  - If false, the session is closed at the end of the sign operation (false by default).

  Note that regardless of the KeepUserLogin parameter value, the user us required to enter his signing password (OTP) for every signing operation.

```
<add key="KeepUserLogin" value="true/false"/>
```

- Display Select Certificate screen:
  - If true, a Select Certificate screen is always displayed, even if the user has only one certificate (true by default)
If false, then if the user has only one certificate, a Select Certificate screen will not be displayed.

```xml
<add key="DisplayCertApprove" value="true/false"/>
```

### Configuring DocuSign SA Web App in Passive ADFS Mode – Using the WS Federation Protocol

DocuSign SA Web App can be deployed by a service provider to offer individuals and organizations digital signature services. Any end user who wishes to use the service must be added to the service either manually or automatically through DocuSign SA APIs. The end user is authenticated to the service using a User ID and password mechanism.

The service provider can offer digital signature services to organizations that deploy Active Directory with ADFS. By configuring a trust between the service provider and the organization that wishes to use the services, any user that belongs to the organization can communicate automatically using the ADFS ticketing mechanisms (based on the SAML standard).

Using the ADFS ticketing mechanism, the end user first uses the local Active Directory services to authenticate and receive an ADFS ticket. The ADFS ticket is then used as a mechanism to connect to DocuSign SA Web App, with the initial communication creating a new account for the user (including a digital signature key and a certificate). The end user can then continue using the account to sign documents.

This section describes how a digital signature service provider should configure DocuSign SA Web App to enable users of an organization that deployed ADFS to use the DocuSign SA Signature service. The description is based on using ADFS with the WS Federation protocol.

The following configuration is required for every organization that wishes to use the Web App service. It consists of two tasks:

- **Configuring Web App for ADFS with WS Federation Protocol**
- **Running the Microsoft Federation Utility**

For more information about the ADFS functionality provided by the DocuSign SA appliance, refer to the ADFS sections in the *DocuSign Signature Appliance Administrator Guide*.

**Note:** Alternatively, you can use the SAML2 protocol to enable users to use the DocuSign SA Signature service. This is discussed in [Configuring DocuSign SA Web App in SAML2 Protocol Mode](#).

### Configuring Web App for ADFS with WS Federation Protocol

1. Rename `web.config.adfs` to be `web.config`, thus overwriting the original `web.config` file.
2. In the `web.config` file, add a line in the `ADFSSettings` section for every organization you wish to trust, as follows:

   ```xml
   <ADFSSettings>
   <add key="org1-ID" value="https://adfs.org1.com/adfs/ls/"/>
   <add key="org2-ID" value="https://adfs.org2.com/adfs/ls/"/>
   </ADFSSettings>
   ```
Where:

- **key** is the identification of an organization.
- **value** is the External Identity Provider URI for the ADFS service of this originating organization.
- The URL for users wishing to connect to the Web App Service must be of the following type:

  \[https://<web app DNS>/org/<identification of the originating organization>,\]

  where the 

  \(<identification of the originating organization>\) has the same value as the key in the web.config file.

  For example:

  \[https://webapp.service.com/org/Company-Z\].

- As part of DocuSign SA Web App v7.5 an additional parameter can be used:

- If you are using a Default Identity Provider, you can set a default redirect URL to the identity provider as follows:

  \(<\text{add key}="ADFSDefaultRedirect" \text{value}="<URL>"/>\)

### Running the Microsoft Federation Utility

The administrator deploying DocuSign SA Web App must run the Fedutil utility for every new organization wishing to use the digital signature service through ADFS.

1. Enable the Fedutil mechanism as follows:

   - If the PC hosting the DocuSign SA Web App is running Windows 2012/2016, you need only select the **Windows Identity Foundation** option in the Control Panel’s **Turn Windows features on or off** section.

2. In the Microsoft IIS server on which DocuSign SA Web App is installed, go to `C:\Program Files (x86)\Windows Identity Foundation SDK\v4.0` and execute `fedutil.exe`. The Federation Utility Wizard welcome page appears.
3. In the Welcome page do the following:
   a. In Application configuration location, specify the path to the web.config file for your DocuSign SA Web App website. For example: C:\inetpub\WebApp\web.config.
   b. In Application URI, specify the URI for your website. For example: https://webapp.service.com/.
   c. Click Next. The Security Token Service page appears.
In the Security Token Service page:

4. In the Security Token Service page:
   a. Select **Use an existing STS**.
   b. Specify the `FederationMetadata.xml` file that was provided to you by the originating organization that deployed the local ADFS deployment.
   c. Click **Next**. The Security Token Encryption page appears.
5. In the Security Token Encryption Page:
   a. Select **No encryption**.
   b. Click **Next**. The Offered Claims page appears.
6. The Offered Claims page displays information related to Claims that are defined in the local ADFS of the originating organization. Review the claims and click **Next**.

The Summary page appears.
7. In the Summary page click **Finish**.

The Web App’s `web.config` file is updated, and information is created under the `FederationMetadata` directory (the created file in the `FederationMetadata` directory is the same for all customers).

8. Backup both the `web.config` file and the `FederationMetadata` directory.

9. The `FederationMetadata` directory contains the `2007-06\FederationMetadata.xml` file. You may need to send the file to your end customer to be used as part of the **Trusted Relaying Party** configuration in the customer’s ADFS server.

**Configuring DocuSign SA Web App in SAML2 Protocol Mode**

As discussed in the previous section, DocuSign SA Web App can enable users to login based on an External Identity Provider. Various Identity Provider solutions use the SAML2 protocol mechanism instead of WS Federation for managing all information required to enable SAML based user authentication. DocuSign SA Web App can optionally be configured to use SAML2 protocol instead of WS Federation.
Configuring DocuSign SA Web App to use the SAML2 protocol

1. In the web.config file, set the following in the General Settings section:
   
   \(<add key="ADFSMode" value="true"/>
   \(<add key="SAML2Mode" value="true"/>

   ♦ If you are using a Default Identity Provider, set a default redirect URL to the identity provider as follows:
   \(<add key="ADFSDefaultRedirect" value="<URL>"/>

Running the Saml2ConfigUtility

Additional configuration is required in order to establish SAML2 protocol based access to DocuSign SA Web App. Because there are various possible External Identity Providers, the following description is written as generically as possible.

1. Create a folder named MetadataDocs under the WebApp folder (for example: C:\inetpub\WebApp\MetadataDocs).

2. Make sure the MetadataDocs directory may be accessed by the user account that executes the Web App Web Server.

3. Locate the Saml2ConfigUtility in the WebApp\bin directory.

4. Execute the saml2configutility.exe utility as follows:
   
   saml2configutility.exe -cf <config file name> -op <operation> [options]
   
   Where
   
   ♦ config file name is the full path name to DocuSign SA Web App’s web.config file.
   
   ♦ operation is one of the following operations:

     ♦ 1 – Add WebApp configuration information as a Service Provider, to Web App’s web.config file. This operation should be executed first (before either operation 3 or operation 2 is executed)
     
     ♦ 2 – Import External Identity Provider information. Information will be imported from the Meta Data file provided by the Identity Provider. This information will be used to build a trust between the Identity Provider and DocuSign SA.
     
     ♦ 3 – Export Web App information (as a Service Provider) to a Meta Data file that will be sent to the Identity Provider. This Meta Data file will be used by the Identity Provider to build trust between the identity Provider and DocuSign SA.

   ♦ options depend on the type of operation, as described in:

   ♦ The saml2configutility.exe Options related to operation = 1
   ♦ The saml2configutility.exe Options related to operation = 2
   ♦ The saml2configutility.exe Options related to operation = 3
The `saml2configutility.exe` Options related to operation = 1

- `[-a <allowed audience URI>]` – Supply the URI of the Service Provider. If the default Web App is used, then supply the URL of the DocuSign SA Web App (for example `https://webapp.domain.com`). This parameter is mandatory.

- `[-sig <signing certificate thumbprint>]` – Enter the certificate thumbprint of the DocuSign SA Web App services’ SSL certificate (refer to step (4)(g) in Installing the DocuSign SA Web App Component). This parameter is mandatory.

  In the case of a certificate thumbprint, the specific account needs to have permission to access the certificate and its key located inside the Microsoft certificate store.

The `saml2configutility.exe` Options related to operation = 2

- `[-m <Identity Providers Meta Data Path>]` – Enter the path for the Meta Data files directory (for example `C:\inetpub\WebApp\MetadataDocs`). This parameter is mandatory. Use it only for adding the first IDP.

- `[-id <Identity Provider identification>]` – The identification for the Identity provider. (for example `http://fsweb.contoso.com/adfs/services/trust`). This parameter represents the ID of the federation partner to which this configuration pertains. This ID must have a match in one of the metadata files that are known by the service provider (entityID according to the SAML2 standard). This parameter is mandatory.

- `[-n <Identity Provider Name>]` – The unique identification for the Identity Provider. Use the DNS representation for the Identity provider. This parameter is optional.

- `[-sou <Identity Provider Sign-On EndPoint URL>]` – The URL end Point for the Identity Provider that will be accessed by the end user. This parameter is optional. You can use this parameter to define a default URL different from the one defined in the Metadata file.

- `[-sob <Identity Provider Sign-On EndPoint Binding>]` – The binding of the end point of the Identity Provider to be accessed by the end user. Use the value `post` if the EndPoint is accessed via an HTTP Post command, or `redirect` if the EndPoint is accessed via a redirect command. This parameter is optional. You can use this parameter to define a default value different from the one defined in the Metadata file.

- `[-slu <Identity Provider Logout EndPoint URL>]` – The URL end Point for the Identity Provider that will be accessed by the end user for the Logout operation. This parameter is optional. You can use this parameter to define a default value different from the one defined in the Metadata file.

- `[-slb <Identity Provider Logout EndPoint Binding>]` – The binding of the end point of the Identity Provider to be accessed by the end user for the logoff operation. Use the
value **post** if the EndPoint is accessed via an HTTP Post command or **redirect** if the EndPoint is accessed via a redirect command.

This parameter is optional. You can use this parameter to define a default value different from the one defined in the Metadata file.

**The saml2configutility.exe Options related to operation = 3**

- `-mf <metadata.xml path>` – The file path for the Meta Data file that was prepared by the Identity Provider for DocuSign SA Web App. This file must be placed in the C:\inetpub\WebApp directory.
  This parameter is mandatory.
- `[-s]` – Sign the Metadata file if the IDP requires to do so.

The following parameters are related to the contact person or division that provides the Web App service (the SP).

- `[-cn <Contact Name>]` – The first name of the contact person running the Web App Services.
- `[-cs <Contact Surname>]` – The surname of the contact person running the Web App Services.
- `[-cc <Contact Company>]` – The company name of the company running the Web App Services.
- `[-ce <Contact Email>]` – The email of the contact person running the Web App Services.
- `[-cp <Contact Phone>]` – The phone number of the contact person running the Web App Services.
- `[-ct <Contact Type>]` – The contact type of the contact person running the Web App Services.

Note that some External Identity Providers use a different method of interaction with the Service Provider (in this case, the DocuSign SA Web App). In this mode, the user communicates directly with the SP via a SAML token provided by the IDP. In this case, you will need to directly edit the web.config file and set the allowUnsolicitedResponses as in the following example:

```xml
<saml2>
  <identityProviders metadata="METADATALOCATION">
    <add id="IdProvider" allowUnsolicitedResponse="true">
      <endpoints>
        <endpoint type="SignOn" url="http://www.example.com/signon" binding="POST"/>
        <endpoint type="Logout" url="http://www.example.com/logout" binding="POST"/>
      </endpoints>
    </add>
  </identityProviders>
</saml2>
```
<identityProviders>
</saml2>

Configuration at the Identity Provider System

The Meta Data file of the Identity Provider must be generated and sent to the DocuSign SA Web App administrators, while the Meta Data file that represents the DocuSign SA Web App as a Service Provider must be imported to the Identity Provider system.

Please use the Identity Provider documentation to perform these tasks.

Note that browser related caching problem may occur when the user perform a logoff operation. In such cases, even though the user performed a logoff operation, the user is immediately logged on again due to the browser reusing cached information.

Please consult with the Identity Provider’s Web Services documentation to solve this problem. For example, if the identity Provider deployment is based on IIS, you can consult the following link: http://social.technet.microsoft.com/wiki/contents/articles/1600.ad-fs-2-0-how-to-change-the-local-authentication-type.aspx.

Configuring DocuSign SA Web App in Passive Mode for using JWT tokens

DocuSign SA Web App can be deployed by a service provider to offer digital signature services to individuals and organizations. Any end user who wishes to use the service must be added to the service either manually or automatically through DocuSign SA APIs. The end user is authenticated to the service using a User ID and password mechanism.

The service provider can offer digital signature services to organizations that have in place an Identity Provider (IDP) that provides JWT tokens based on successful authentication to the IDP. By configuring a trust between the service provider and the organization that wishes to use the services, any user that belongs to the organization can communicate automatically with the service provider using the the JWT mechanism.

The user authentication process is as follows:

1. The end user connects to the DocuSign SA Web App.
2. The DocuSign SA Web App redirects the user to the OpenID server authentication endpoint.
3. The user authenticates himself and gets a temporary code.
4. The user is redirected back to the DocuSign SA Web App with the code.
5. The DocuSign SA Web App sends a request, to the OpenID provider’s token endpoint, to exchange the code for an ID token.
6. The DocuSign SA Web App tries to authenticate the user against the DocuSign SA using the token.
7. Depending on the token’s validity, the user is redirected either to the requested page or to an error page.

The following section describes how to configure DocuSign SA Web App to enable users to use the DSA services based on JWT. Refer to Configuring Web App for JWT based user authentication with OpenID Connect Protocol.
For more information about the JWT functionality provided by the DocuSign SA appliance, refer to the JWT sections in the DocuSign Signature Appliance Administrator Guide.

**Configuring Web App for JWT based user authentication with OpenID Connect Protocol**

1. In the `web.config` file, set the following in the `JwtSettings` section:
   ```xml
   <add key="JwtMode" value="true"/>
   ```

2. The following additional parameters can be setup:
   - To set the Discovery Document URL, add the following parameter:
     ```xml
     <add key="DiscoveryDocumentURL" value="THE_VALUE" />
     ```
     It there is no Discovery Document URL is recommended to use an empty string (""). The Discovery Document URL replies with all relevant attributes that can be used for enabling the end user to interact with the IDP.
   - To set the Token URL, add the following parameter:
     ```xml
     <add key="TokenURL" value="THE_VALUE" />
     ```
     By default, the following value is used "https://account.docusign.com/oauth/token" This parameter is ignored if `DiscoveryDocumentURL` has a non empty value.
   - For the Scopes parameter, do not change the default setting which is “openid”:
     ```xml
     <add key="Scopes" value="openid" />
     ```
   - To set the OpenID provider's application client ID, add the following parameter:
     ```xml
     <add key="ClientId" value="THE_VALUE" />
     ```
     The value should be provided by the operator of the IDP.
   - To set the OpenID provider's application client secret, add the following parameter:
     ```xml
     <add key="ClientSecret" value="THE_VALUE" />
     ```
     The value should be provided by the operator of the IDP.
   - To set the OpenID provider's redirect URL, add the following parameter:
     ```xml
     <add key="JwtRedirectUrl" value="THE_VALUE" />
     ```
     The value should be provided by the operator of the IDP.
Configuring DocuSign SA Web App to use Microsoft Active Directory Kerberos authentication

To enable using Active Directory Kerberos authentication, some required configurations need to be performed. Some of the configurations are required in the Active Directory domain and some are required on the Microsoft IIS deployment of the DocuSign SA Web App.

Setting Web App’s web.config File

1. Modify the web.config file of the DocuSign SA Web App as follows:
   a. In the Configuration > System.web section, change the authentication settings from Forms to Windows.
   b. In the same section, add an authorization sub-section in order to disable anonymous access.

   ```xml
   <authentication mode="Windows">
   <forms loginUrl="~/" timeout="10"/>
   </authentication>
   <authorization>
   <deny users="?"/>
   </authorization>
   ```

2. Make sure the ADMode parameter is set to true. Refer to ADMode in Chapter 9: Configuring the DocuSign SA Web App.

3. If the DocuSign SA Web Agent is used together with the Active Directory Kerberos ticket mechanism, you will need to enable Anonymous communication for the communication between the Document management system and DocuSign SA Web Agent. To do so, add the following under the <authentication> line shown in the preceding step.

   ```xml
   <anonymousIdentification enabled="true"/>
   ```

Using the setspn utility to establish a Kerberos based authentication scheme

To enable end-users to connect to the DocuSign SA Web App using an Active Directory based Kerberos mechanism, the Microsoft setspn command line utility can be used.


The utility defines Service Principle Name (SPNs) for a certain computer in the Active Directory.

The utility can be executed in any PC which is not the Active Directory domain PC, using a Domain Admin account.
For example, if the PC that runs the DocuSign SA Web App is named **DocuSign-SA-Web-App-PC** and the domain is named **Domain.com**, the PC should have the following SPNs:

- HOST/DocuSign-SA-Web-App-PC
- HOST/DocuSign-SA-Web-App-PC.Domain.com
- HTTP/DocuSign-SA-Web-App-PC
- HTTP/DocuSign-SA-Web-App-PC.Domain.com
- RestrictedKrbHost/DocuSign-SA-Web-App-PC
- RestrictedKrbHost/DocuSign-SA-Web-App-PC. Domain.com

1. Run the utility with the –L flag as follows, to view the current SPN listing of the Web App PC:
   ```bash
   setspn -L <web server name>
   ```
   In our example, run: `setspn -L DocuSign-SA-Web-App-PC`

2. To add missing SPNs, run the utility as follows for every missing item:
   ```bash
   setspn -s <SPN value> <Server name>
   ```
   For example, to set the HTTP/DocuSign-SA-Web-App-PC SPN for the DocuSign-SA-Web-App-PC server, run:
   ```bash
   ```

3. Finally, it is recommended to run `setspn -L <web server name>` again to make sure that all SPNs are listed.

### Using Internet Explorer to Access DocuSign SA Web App Based on Kerberos Ticketing

Organizations that want their end users to access the DocuSign SA Web App using the Kerberos ticketing mechanism should configure the following settings in the end user’s PC:

1. **In Internet Options > Advanced**, check **Enable Integrated Windows Authentication**.
2. **In Internet Options > Security**, add the Web App site to the trusted sites in Internet Explorer.
3. **In Internet Options > Security**, select the **Internet** zone and click **Custom level**… Scroll to the bottom and check the option: **Automatically logon with current user name and password**.

It is advisable to connect using the full DNS name of the DocuSign SA Web App and not only with the PC in the Intranet.

**Note**: the DocuSign SA Web App PC should have a valid DNS name under the domain of the organization, and also should be accessed via Internet Explorer using this DNS name.
Using Google Chrome to access DocuSign SA Web App based on Kerberos Ticketing

Add the following registry keys to the end user’s Registry entry settings.

```
[HKEY_LOCAL_MACHINE\Software\Policies\Google\Chrome]
  "AuthNegotiateDelegateWhitelist"="
  "AuthSchemes"="basic,digest,ntlm,negotiate"
  "AuthServerWhitelist"=""
```

Where:

- **AuthNegotiateDelegateWhitelist** – Servers that Google Chrome may delegate to. You can enter multiple server names separated by commas. Wildcards (*) are allowed. If you leave this policy not set, Google Chrome will not delegate user credentials even if a server is detected as Intranet.
- **For AuthSchemes** – use the suggested input text.
- **AuthServerWhitelist** – Specifies which servers should be whitelisted for integrated authentication. Integrated authentication is only enabled when Google Chrome receives an authentication challenge from a proxy or from a server which is in this permitted list. You can enter multiple server names separated by commas. Wildcards (*) are allowed. If you leave this policy not set Google Chrome will try to detect if a server is on the Intranet and only then will it respond to IWA requests. If a server is detected as Internet then IWA requests from it will be ignored by Google Chrome.

Configuring DocuSign SA Web App to Work with PDF Forms

If you want the DocuSign SA Web App to work with PDF forms, make the following changes in the web.config file:

1. **Under appSettings**, set the RAD PDF license key. For example:
   
   ```xml
   <add key="RadPdfLicenseKey" value="DEMO"/>
   ```
   
   In a non-demo environment, enter the license key provided by RAD PDF (a base64 encoded string).

2. **Under appSettings**, set your SQL connection string. For example:
   
   ```xml
   <add key="RadPdfConnectionString" value="Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=RadPdf"/>
   ```
   
   Or, if the RAD PDF database is installed on another PC:
   
   ```xml
   <add key="RadPdfConnectionString" value="Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=RadPdf;Data Source=ANOTHER-PC"/>
   ```

3. **Under GeneralSettings**, enable FormFill mode as follows:
   
   ```xml
   <add key="FormFillMode" value="1"/>
   ```
   
   Note that the possible values are:
   
   - 0 – Disable FormFill mode
   - 1 – Enable FormFill mode
   - 2 – Automatic mode: Display files in FormFill mode only when there are fields to fill.
Integrating with Cloud-Based File Storage Services

The setup described in this section must be performed to enable users of your organization to access documents stored in Cloud-based storage services such as Google Drive or Dropbox. The following setup tasks must be performed:

- The **CloudStorageSettings** section of the web.config file must be updated to include information regarding the Cloud-based storage services. For information, refer to [Defining Cloud Storage Settings](#).
- A dedicated application must be defined for the selected Cloud-based storage service. This application will enable the DocuSign SA Web App to access the user account in the Cloud-based storage services. In the case of Google Drive or Box, you can configure these applications to enable invoking DocuSign SA from Google Drive or Box menus. For information, refer to [Creating an Application for Enabling Access to User Accounts in the Cloud](#).

**Note:** Many Cloud Storage providers, such as Box, mandate using HTTPS protocol for accessing the DocuSign SA Web. It is highly recommended to do so.

Defining Cloud Storage Settings

The Cloud Storage Settings section is by default empty.

```xml
<CloudStorageSettings>
    <!-- add here Cloud Storage settings-->
</CloudStorageSettings>
```

The following configuration parameter must be set if you wish to integrate with any Cloud-based storage systems:

- **Enable using Dropbox as a Cloud-Based File Storage**
  ```xml
  <add key="EnableDropbox" value="true/false" />
  ```
  The default is enabled.

- **Enable using Box as a Cloud-Based File Storage**
  ```xml
  <add key="EnableBox" value="true/false " />
  ```
  The default is enabled.

- **Enable using OneDrive as a Cloud-Based File Storage**
  ```xml
  <add key="EnableSkyDrive" value="true/false " />
  ```
  The default is enabled.

  **Note:** The internal naming convention for OneDrive is SkyDrive because SkyDrive was the former name of this Storage Cloud Service.
Enable using Google as a Cloud-Based File Storage

```xml
<add key="EnableGoogle" value="true/false" />
```

The default is enabled.

Note that if a single cloud-based file storage is enabled, then only the icon for this cloud storage will appear immediately after the user logs on. If no cloud-based file storage is enabled, no indication of a cloud based file storage system appears.

Set the Cloud storage callback URL (redirect URL):

```xml
<add key="CloudStorageCallbackURL" value="THE_VALUE" />
```

When the user is redirected to the Cloud-based storage service for a login operation, the Cloud-based storage system is redirected back to CloudStorageCallbackURL to continue the user session in the DocuSign SA Web App.

Specify the following value:

```
https://sign.org.com/cloudstorage/loggedtocloudstorage
```

where sign.org.com is the location where the DocuSign SA Web App is deployed.

**Creating an Application for Enabling Access to User Accounts in the Cloud**

This section describes, for each type of Cloud-based storage service, how to create an application enabling the DocuSign SA Web App to access the user account in the Cloud-based storage service.

**Creating a DocuSign SA Application in Dropbox**

1. Connect to the Dropbox development portal at [https://www.dropbox.com/developers](https://www.dropbox.com/developers) and login with a user account.

2. Select **My Apps**.

3. Click **Create app** to create a new application.
4. Select the **Dropbox API** option.
5. Select the following options:
   a. **Choose the type of access you need** – Select **Full Dropbox**.
   b. **Name your app** – The recommended app name is `<your organization name > - DSA`.
6. Click **Create app**. The following form appears:
7. In the Application forms, supply additional information such as a description and icons.

**Note:** After applying for production – there is no DropBox limit on the number of users who can use the DocuSign SA Web App application.

8. Record the **App key** and **App secret** values for entering in the web.config file.

9. In the **OAUTH2 – Redirect URIs** field, enter one of the following URI, depending on your Web App deployment characteristics, and click **Add**:

   - https://<Web App host name>/CloudStorage/LoggedToCloudStorage/DropBox
   - http://<Web App host name>/CloudStorage/LoggedToCloudStorage/DropBox,

10. Enter a similar entry in the CloudStorageSettings section of the Web Config file. For example:

    ```xml
    <add key="CloudStorageCallbackURL" value="https://<host name>/CloudStorage/LoggedToCloudStorage" />
    ```
Note that “DropBox” is omitted from the end of the URL, since this value is generic to all Cloud Service Providers.

11. Click **Update**.

12. Enter the **App key** and **App secret** values in the DropBoxSettings section of the web.config file. The DropBoxSettings section is by default empty.

   <DropBoxSettings>
   <!-- add here DropBox settings-->
   </DropBoxSettings>

   a. Enter the DropBox **App key** value:
      <add key="AppKey" value="THE_VALUE" />

   b. Enter the DropBox **App Secret** value:
      <add key="AppSecret" value="THE_VALUE" />

13. Click **Apply for production**.

   It takes about a day for Dropbox to approve your DocuSign SA app.

**Creating a DocuSign SA Application in OneDrive**

1. Connect to the OneDrive development portal at [https://account.live.com/developers/applications/index](https://account.live.com/developers/applications/index) and login in with a user account.

2. Click **Add an app**.

   ![Figure 78 Creating a OneDrive Application](image)

3. Create a new application. The recommended name is **DSA - <your organization name>**.

4. Click **Create application**.

5. In the page that appears, select the **App Settings** tab.
6. Record the **Application Id**.
7. Click **Generate New Password** and record the newly generated secret.
8. Enter these values into the `DTMClientID` and `DTMClientSecret` parameters in the `DTM Settings` section of the web.config file, located in the home directory of the DocuSign SA Web App deployment.

9. Click **Add Platform** and then click **API Settings**.

10. In the **Redirect URIs** field, enter one of the following URI, depending on your Web App deployment characteristics:
   - `https://<Web App host name>/CloudStorage/LoggedToCloudStorage/SkyDrive`
Supply additional information such as Terms of Service URL and Logo.

12. Click Save.

13. Enter the OneDrive Client ID and Client Secret values in the SkyDriveSettings section of the web.config file (the internal naming convention for OneDrive is SkyDrive because SkyDrive was the former name of this Storage Cloud Service). The SkyDriveSettings section is by default empty.

   <SkyDriveSettings>
     <!-- add here SkyDrive settings-->
   </SkyDriveSettings>

   ♦ Enter the SkyDrive Client ID value:
     <add key="ClientID" value="THE_VALUE" />

   ♦ Enter the SkyDrive Client Secret value:
     <add key="ClientSecret" value="THE_VALUE" />

14. Enter a similar entry in the CloudStorageSettings of the Web Config file. For example:

   <add key="CloudStorageCallbackURL" value="https://<host name>/CloudStorage/LoggedToCloudStorage" />

   Note that “SkyDrive” is omitted from the end of the URL, since this value is generic to all Cloud Service Providers.

Creating a DocuSign SA Web App Application in Google Drive

For information on how to integrate with Google Drive, refer to the following Google documentation:

♦ https://developers.google.com/drive/quickstart-cs
♦ https://developers.google.com/drive/enable-sdk

For assistance, contact DocuSign support.

Enabling Users to Sign from Within Google Drive

You can configure Google Drive to enable users to access DocuSign SA Web App from the Google Drive menu of a document that is stored inside the Google Drive web application. To do so:

1. Make sure the parameter EnableGoogleDriveIntegration in the API Settings section of web.config is set to true.

2. Configure the Google Drive DocuSign SA application to enable users to invoke DocuSign SA from within Google Drive. For assistance, contact DocuSign support.
Creating a DocuSign SA Web App Application in Box

1. Connect to the Box development portal at http://developers.box.com/ and login in with a user account.

2. Go to the Applications section and click Create a Box Application.

3. Provide the application name. The recommended name is <your organization name> - DSA.

4. Click Create Application.

5. Continue to configure your Box application:
Configure the OAuth2 parameters as follows:

a. Record the **Client ID** and **Client Secret** values for entering in the web.config file.

b. Enter a **redirect_uri** that points to the DocuSign SA Web App URL. Enter one of the following URLs, depending on your Web App deployment characteristics:
   - https://<Web App host name>/CloudStorage/LoggedToCloudStorage/Box
   - http://<Web App host name>/CloudStorage/LoggedToCloudStorage/Box

7. Click **Save Application**.

8. Enter the **Client Id** and **Client Secret** value in the BoxSettings section of the web.config file. The BoxSettings section is by default empty:

   ```xml
   <BoxSettings>
     <!-- add here Box settings-->
   </BoxSettings>
   ```

   Enter the **ClientID** value:
   ```xml
   <add key="ClientID" value="THE_VALUE" />
   ```

   Enter the **ClientSecret** value:
   ```xml
   <add key="ClientSecret" value="THE_VALUE" />
   ```
Enabling Users to Sign from Within Box

You can configure Box to enable users to access DocuSign SA Web App from the Box menu of a document that is stored inside the Box web application. To do so:

1. Make sure the parameter `EnableBoxIntegration` in the API Settings section of web.config is set to `true`.

2. Configure the Box DocuSign SA application to enable users to invoke DocuSign SA from within Box, as follows:
   a. In the Edit a Box Application form, click **Create a New Web App Integration**.

![Figure 83 Creating a New Web App Integration](image)
The following General Information form appears.

![General Information Form](image)

**Figure 84 Creating a Box File Action**

b. Edit the following parameters as follows. You may want to change other information as well (refer to the Box developer guides).

- **Supported Extensions** – doc, docx, pdf, xls, xlsx
- **Scoped To**: select **The parent folder of the file/folder from which this integration is invoked**.
- **Category** – Web (recommended)
- **Filetype Category** – Documents (recommended)
- **Integration Status** – Online
- **New window settings** – Check the **Popup window will open in a tab** checkbox (recommended)
Client Callback URL – A mandatory field. Enter the following:
https://<Web App host name>/CloudStorage/BoxFileAction/Box,
where
https://<Web App host name> is the organizational DocuSign SA Web App deployment location.

c. Supply the following Callback parameters:

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter Name</th>
<th>Parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get</td>
<td>RedirectUrl</td>
<td>#redirect_to_box_url#</td>
</tr>
<tr>
<td>Get</td>
<td>FileID</td>
<td>#file_id#</td>
</tr>
<tr>
<td>Get</td>
<td>AuthCode</td>
<td>#auth_code#</td>
</tr>
</tbody>
</table>

d. Specify the following Prompt Message text: The Secure Way to Sign on the Go – easily add digital signatures to PDF, Word and Excel documents

e. Click Save Web App Integration.

f. When you finish configuring the application, publish your application to Box by selecting Submit for approval (Figure 83).

![Information for listing in the Box App Gallery](https://app.box.com/services/org-ds-a)

**Figure 85 Submitting a Box App for Approval**
Index

---

A

Active Directory settings
- enable using localized test, 90
- enable working with AD, 91
- enable working with DocuSign cloud, 90
- enable working with Web App, 90
- set domain name, 91

ADFS mode
- configuring for SAML2 protocol, 102
- configuring for WS Federation, 95
- introduction, 95
- running the WS Federation utility, 96

Advantages of DocuSign SA Web App, 11

Alternate Web Site
- configuring, 22
- enabling, 90

API Integration settings, 89

Applications that work with DocuSign SA, 7

Authentication
- requirements, 6
- Authentication, extended authentication, 86

---

B

Box
- creating a DocuSign SA app in Box, 118
- enabling invoking DocuSign SA from Box, 90, 120
- invoking CoSign from Box, 76
- signing documents retrieved from Box, 75

---

C

Certify
- default certify type, 93
- usage, setting, 93

Clear signature field policy, 93

Compatibility mode
- disabling, 92
- enabling, 92

Configuring DocuSign SA Web App, 85

Customizing DocuSign SA Web App, 85

- API Integration Settings, 89
- cloud storage settings, 110
- DocuSign Cloud Settings, 93
- DTM Settings, 93
- Extended Authentication Settings, 86
- General Settings, 90
- Image Settings, 85
- Login Policy Settings, 94
- SMTP Settings, 86
- web.config file, 85

---

D

Data authentication systems, 6

Date and Time display in of signature, 69

Date format of signature, 69

Document appearance, 38

Document types supported, 10

DocuSign Cloud
- DTM Settings, 93
- enabling in organizational workflow, 64
- Login Policy Settings, 94
- signature workflow, 59
- supported languages, 59
- using with WebApp, 59

DocuSign SA
- applications that work with DocuSign SA, 7
- components, 8
- documentation, 8
- environments supported by DocuSign SA, 7

DocuSign SA client
- installing, 12
- optimizing for DocuSign SA Web App, 12

DocuSign SA Web Agent, 11

Dropbox
- creating a DocuSign SA app in Dropbox, 111
- signing documents retrieved from Dropbox, 71

DTM Settings, 93

---

E

Environments supported by DocuSign SA, 7

Extended Authentication settings, 86

External Identity Provider (IDP)
- configuration at the IDP end, 105
- integrating with, 11

---

F

Forgot My Password settings, 90

---

G

General settings, 90

Google Drive
- creating a DocuSign SA app in Google Drive, 117
- enabling invoking DocuSign SA from Google Drive, 89, 117
- invoking DocuSign SA from Google Drive, 73
- signing documents retrieved from Google Drive, 72

Graphical signature selection mode, 69

Graphical signatures
- adding, 66
- sketching, 67
typing in, 67
uploading a file, 67

---H---
HTTPS
redirecting HTTP to HTTPS, 20
specifying HTTPS settings, 19

---I---
Image reduction settings, 85
Installing DocuSign SA Web App
configuring a mail server, 21
configuring if PDF forms use, 23
configuring IIS for Kerberos Ticketing, 22
installation overview, 12
installing the DocuSign SA Client, 12
installing the Web App component, 17
using HTTPS, 19
Integrating with Cloud-based file storage services, 110
Intended audience, 9
Introduction
to digital signatures, 6
to DocuSign SA, 6
to DocuSign SA Web App, 10

---J---
JWT
configuring JWT based user authentication, 106
configuring SA Web App in passive mode for JWT, 105
JWT tokens, 105
user authentication with OpenID Connect, 106

---K---
Kerberos Authentication, enabling
der-user PC configuration, 108
der-user registry configuration, 109
introduction, 107
running the setspn utility, 107
web.config file modifications, 107

---L---
Logging in to DocuSign SA Web App, 35
Logging off
redirection to a specific URL, 92
results, 70
Login Policy Settings, 94

---O---
O365 documents, signing, 79
deploying DocuSign SA O365App, 82
deploying proxy, 79
flow of operations, 83
registering in the O365 SharePoint cloud, 80
required software components, 79
updating proxy settings, 83
OneDrive
creating a DocuSign SA app in OneDrive, 114
signing documents retrieved from OneDrive, 72
OpenID Connect
using with JWT based user authentication, 106
Operation mode
Point of Sale, 69
standard, 69
Operational mode
setting, 69
Overview of DocuSign SA, 6
Overview of DocuSign SA Web App, 10

---P---
Passive mode, configuring, 105
Password, changing, 68
PDF forms
filling-in, 42
prerequisite configuration, 23
prerequisite SQL server configuration, 25
prerequisite SQL server installation, 23
RAD PDF configuration, 23
required web.config file changes, 109
signing, 42
viewing, 42
Point of Sale mode signature, 53
POS mode
enabling for a specific user, 69
enabling globally, 91
Post-signing actions, 50
Prior knowledge, 9

---R---
RAD PDF
creating maintenance string for RadPdf database, 32
creating RadPdf database instance, 26
creating tables in RadPdf database, 29
enabling IIS service account to access RadPdf database, 27
installing, 30
required for PDF forms processing, 23
Reasons for signing, 69

---S---
SAML2 protocol
configuring for ADFS, 102
running the Saml2ConfigUtility, 102
saml2configutility executing, 102
Selecting a document, 37
from a cloud-based file management service, 37
from a local device, 37
Signature
adding graphical signatures, 66
defining date and time display, 69
defining date format, 69
defining reasons for signing, 69
defining signature appearance, 68
defining signature type, 69
defining time format, 69
Signature appearance, defining, 68
Signature Locators
disabling, 92
enabling, 92
Signature type, setting, 69
Signing a PDF document
creating a new signature, 43
post-signing actions, 50
signing a pre-existing signature field, 43
Signing an MS Office document
post-signing actions, 50
signing a pre-existing signature field, 45
Signing work flow, 35
SMTP settings, 86

—T—
Time format of signature, 69

—U—
Uninstalling DocuSign SA Web App, 34
User settings, modifying, 68
Username placeholder, 92

—V—
Validating document signatures, 49
Viewing a document, 41
Viewing signature details, 48
Volatile graphical signature, 51
user experience, 52

—W—
web.config file
location, 85
modifying, 85
WebApp deployment location, 93
WS Federation
configuring for ADFS, 95
introduction, 95
running the Fedutil utility, 96