

PDF Signer Server User Manual

Introduction

The main function of PDF Signer Server is to sign PDF documents using X.509 digital certificates. Using this product you can quickly sign multiple PDF files (bulk sign) by selecting input and output directory. This is ideal for bulk signing of a large number of corporate documents rather than signing each one individually.

The positioning of the signature appearance is configurable, plus on which pages of the document it should appear (first page, last page or all pages).

Links

PDF Signer Server main page: <http://www.signfiles.com/pdf-signer-server/>
Download PDF Signer Server (Free 30-Day Trial):
<http://www.signfiles.com/apps/PDFSignerServer.msi>

Warning and Disclaimer

Every effort has been made to make this manual as complete and accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this manual.

Trademarks

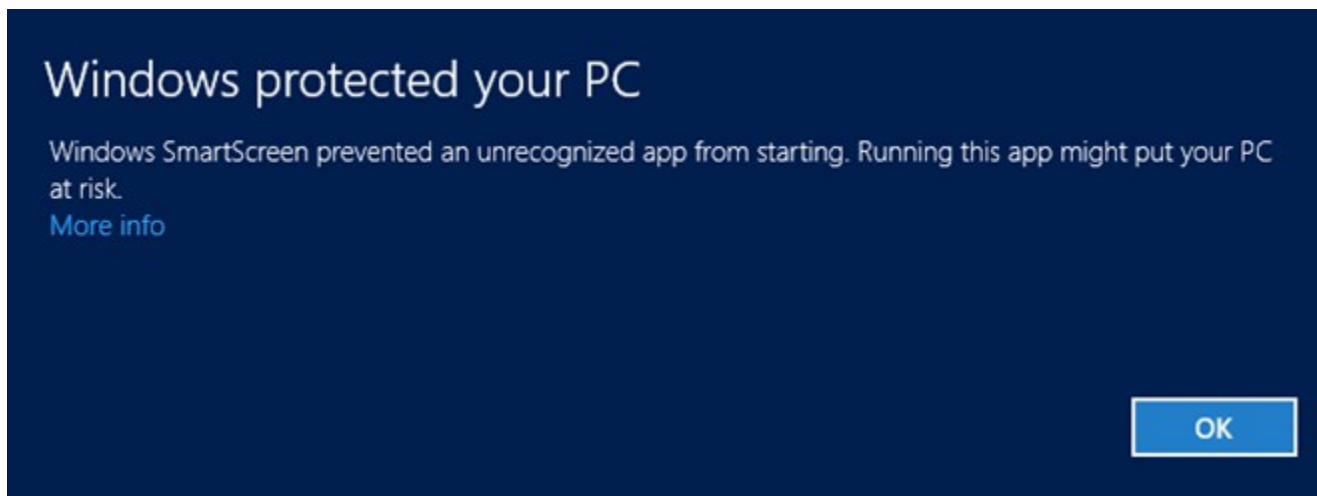
.NET, Visual Studio .NET are trademarks of Microsoft Inc.
Adobe, Adobe Reader are trademarks of Adobe Systems Inc.
All other trademarks are the property of their respective owners.

Product Installation	3
Digital Certificates	4
Digital Certificate Location.....	4
Certificates Stored on Smart Cards or USB Tokens.....	5
Select the Digital Certificate for Creating PDF Signatures.....	6
Create a Digital Certificate.....	7
Validating Digital Signatures in Adobe.....	8
Digital Signature Options	9
Digital Signature Rectangle.....	9
Set the Digital Signature Graphic.....	10
Signing Reason and Location.....	11
Using SHA256, SHA512 Hash Algorithms.....	12
Bypassing the Smart Card PIN.....	13
Certify a PDF Digital Signature.....	14
Include the CRL Revocation Information on the PDF Signature.....	15
PDF/A Standard.....	17
Time Stamping	18
Time Stamp the PDF Digital Signature.....	18
Nonce and Policy.....	18
Validating the Time Stamp Response on Adobe.....	19
Encryption	20
LTV Signatures (Long Term Validation)	22
Product Registration	23
Email Option	25
Email Workflow.....	26
FTP Option	27
FTP Workflow.....	27
Additional Actions	28
Backup Folder (Archive Folder).....	28
Error Folder.....	28
Batch Mode.....	29
Logging	30
Batch Signatures (Automatically Made Without User Intervention)	31
Custom Configuration.....	31
Digitally Sign PDF Files Using Windows PowerShell	32
Digitally Sign PDF Files Using C# or VB.NET	33

Product Installation

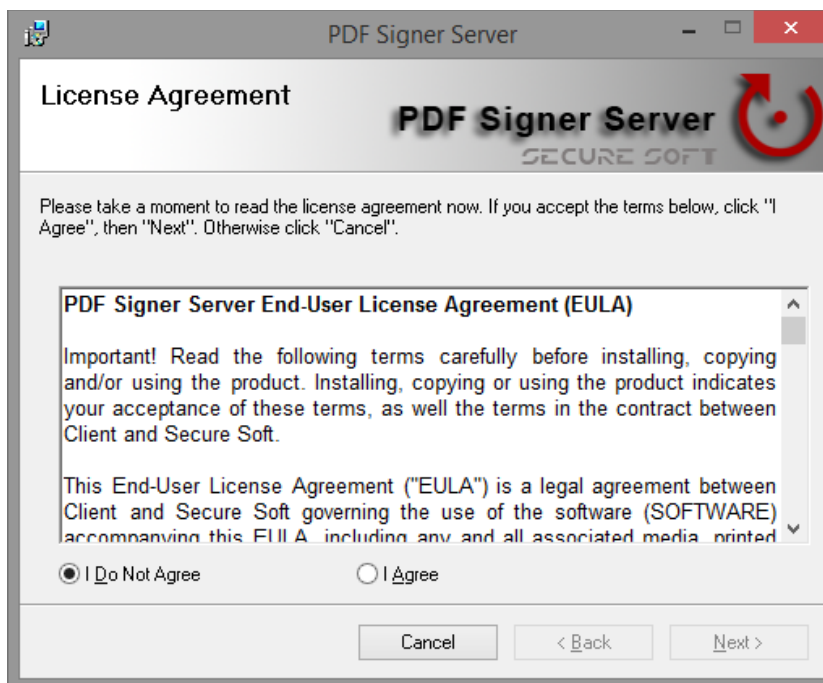
We recommend to install the product using an Administrator account.

After the setup file is verified, the operating system might request your permission to install this program.



Click More info and next click *Run anyway*.

Read the Eula and if you want to continue, select *I Agree* and click *Next* button until the setup is finished.



Digital Certificates

Digital Certificate Location

To digitally sign a PDF file a digital certificate is needed. The digital certificates are stored in two places:

- in Microsoft Store
- in PFX or P12 files

The certificates stored on **Microsoft Store** are available by opening *Internet Explorer – Tools* menu – *Internet Options – Content* tab – *Certificates* button (see below).

For PDF digital signatures, the certificates stored on *Personal* tab are used. These certificates have a public and a private key.

The digital signature is created by using the private key of the certificate. The private key can be stored on the file system (imported PFX files), on a cryptographic smart card (like Aladdin eToken or SafeNet iKey) or on a HSM (Hardware Security Module).



Signing certificates available on Microsoft Store

Another way to store a digital certificate is a **PFX (or P12) file**. This file contains the public and the private key of the certificate. This file is protected by a password in order to keep safe the key pair.

Note that a PFX/P12 file can be imported on Microsoft Store (just open the PFX/P12 file and follow the wizard).

Certificates Stored on Smart Cards or USB Tokens

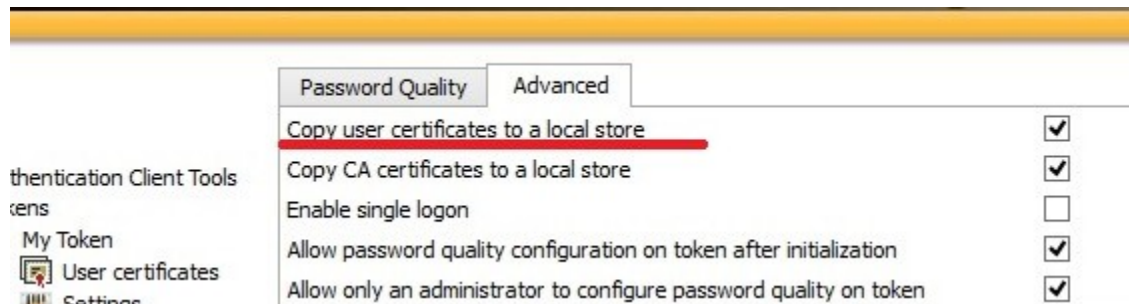
If your certificate is stored on a smart card or USB token (like Aladdin eToken), the certificate must appear on Microsoft Certificate Store in order to be used by the library.

If the certificate not appears on Microsoft Store, you must ask your vendor about how to import the certificate on the MS Store. Usually, the smart card driver or the middleware automatically install the certificate on Microsoft Certificate Store.

You should also look at the middleware options, like below:



Adding the certificate on Microsoft Certificate Store



Adding the certificate on Microsoft Certificate Store

Select the Digital Certificate for Creating PDF Signatures

To digitally sign a PDF, a digital certificate must be selected from Digital Certificates section. The digital certificate used to create the digital signature can be stored on Microsoft Store or a PFX file.

Digital Certificates

Select the digital certificate used for digital signature

Windows Certificate Store

Certificates Available on Microsoft Store

Certificate Store: Show expired certificates

Smart Card PIN:

PFX digital certificate file

PFX Certificate File

PFX file password:

Certificate Information

Issued to: Secure Soft S.R.L., Issued by: thawte SHA256 Code Signing CA, Valid until: 7/13/2017, Certificate Service Provider: Microsoft Enhanced Cryptographic Provider v1.0

Include certificate revocation information - Long Term signature (LTV)

Select the digital certificate

Create a Test Digital Certificate

If no certificates are available on the computer, a test certificate can be created from *Create a Digital Certificate* section.

This certificate can be set as the default digital certificate used for PDF signatures.

The screenshot shows a dialog box titled "Create a Test Digital Certificate" with a close button (X) in the top right corner. The main question is "Where would you like to save your self-signed test digital certificate?". There are two radio button options: "On Microsoft Certificate Store" (which is selected) and "On a password protected PKCS#12 PFX file". Below this is a section for certificate details with the following fields and values:

- Issued to (e.g. Elaine Smith)*: test certificate
- Organization Name (O=):
- Title (T=):
- Organizational Unit (OU=):
- E-mail address (E=):
- Country (C=):
- Validity period: 1 Year
- RSA Key Algorithm: 1024 bits
- Signature Algorithm: SHA1WithRSA

At the bottom of the details section, there is a checked checkbox labeled "Set as current digital certificate". At the very bottom of the dialog are "OK" and "Cancel" buttons.

Create a digital certificate

Validating Digital Signatures in Adobe

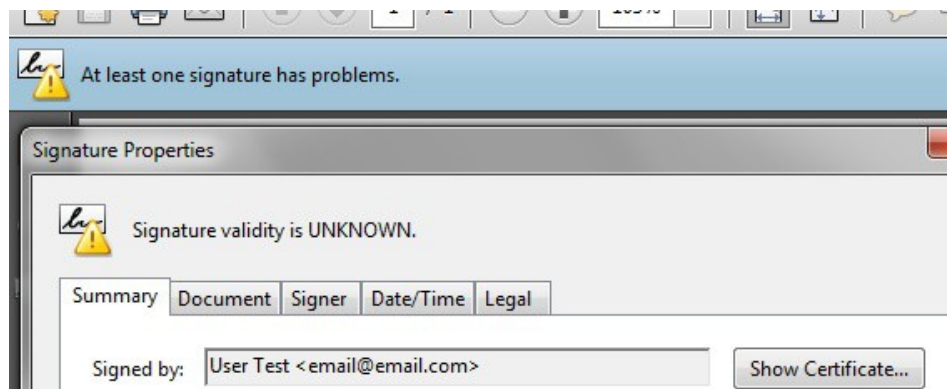
Every digital certificate is issued by a Root CA (Certification Authority). Some of the Root CA's are included by default in Windows Certificate Store (Trusted Root Certification Authorities) and only a few are included in Adobe Certificate Store. Microsoft and Adobe use different Certificate Stores different certificate validation procedures.

If the signing certificate (or the Root CA that issued the signing certificate) is not included in Adobe Store, the digital signature is considered "not trusted" when a user open a document with Adobe Reader (see example).

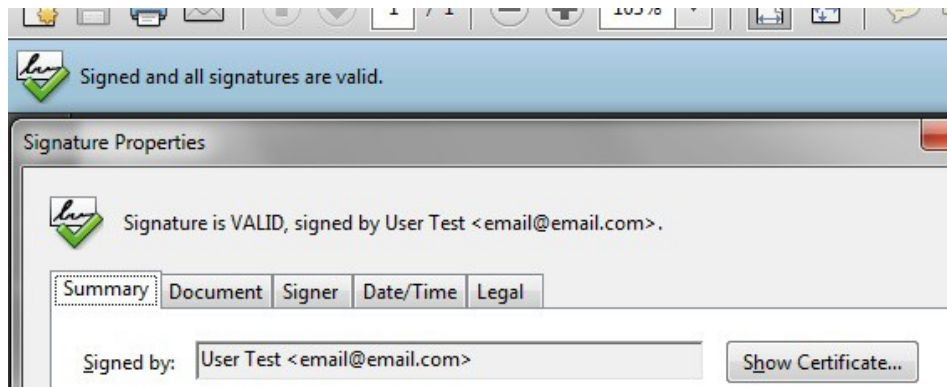
This behavior has nothing to do with the signing engine but with the Adobe certification validation procedure.

To trust a signature the user must add the signing certificate on the Adobe Certificate Store because only a few Root CA's are considered trusted by default by Adobe certificate validation engine (See this article: http://www.adobe.com/security/partners_cds.html)

To validate the signing certificate in Adobe use the methods described on this document: <http://www.signfiles.com/manuals/ValidatingDigitalSignaturesInAdobe.pdf>



Validity Unknown signature



Valid signature

Digital Signature Options

Digital Signature Rectangle

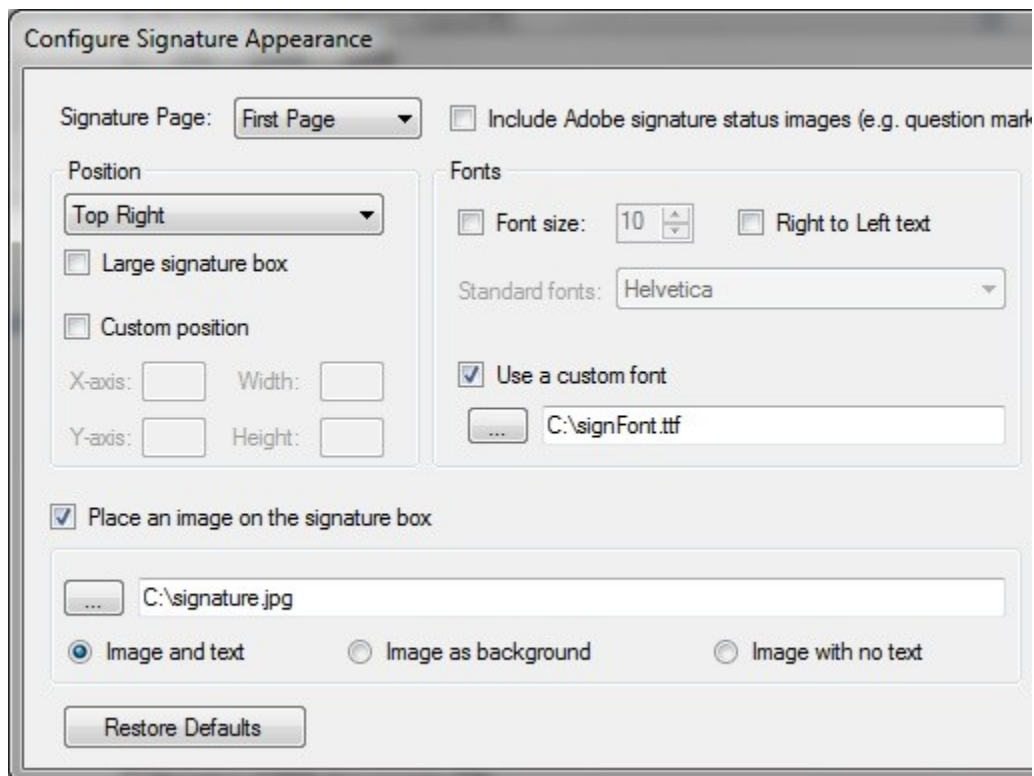
If the checkbox *Visible signature box* is checked, a signature rectangle will be inserted on the PDF document. The appearance of the digital signature can be customized from the *Signature Appearance* section.

The default text direction is left to right. To change the text direction to right to left (e.g. for Hebrew language) checkbox *Right to Left text* must be checked.

The default font file for the digital signature rectangle is Helvetica. It is possible that this font to not include all necessary UNICODE characters like **ä**, **à**, **â**. On this case you will need to use an external font.

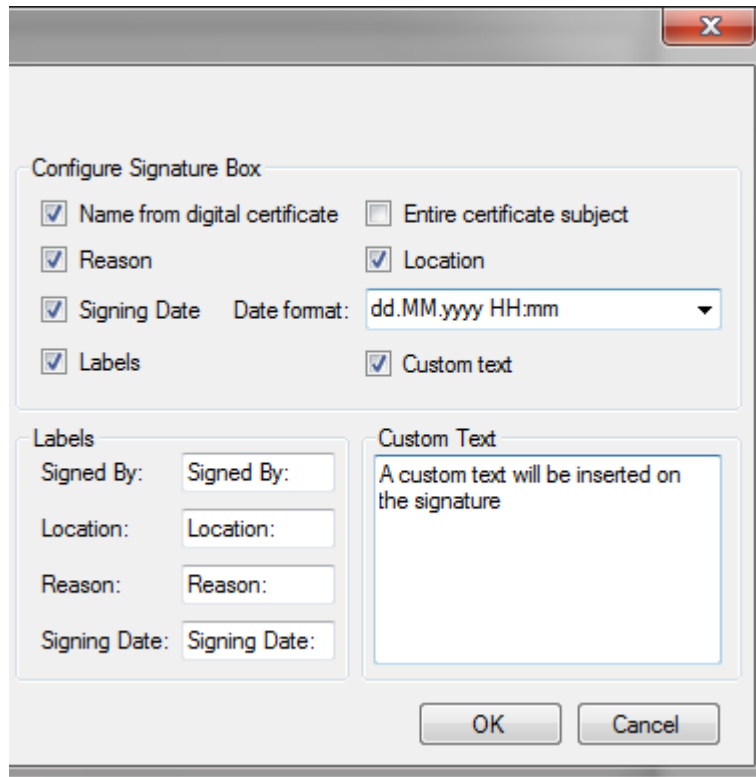
The font size is calculated based on the signature rectangle size in order to fit on the signature rectangle (it not have a fixed size). If you want to use a specific font size, it can be specified on the *Font size* section.

Observation: If the custom position will be used, the corner (0,0) is on the bottom left of the page.



Basic appearance settings

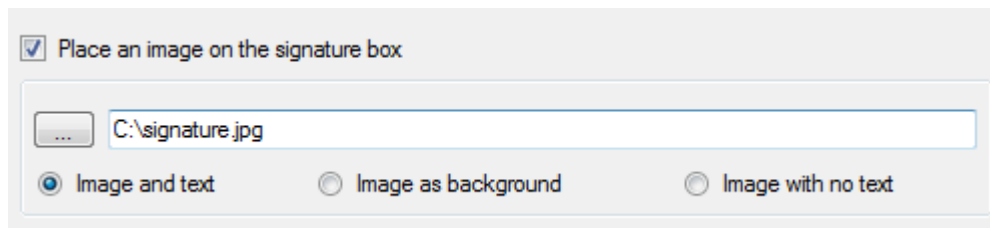
The default digital signature text contains information extracted from the signing certificate, signing date, signing reason and signing location but the digital signature text can be easily customized.



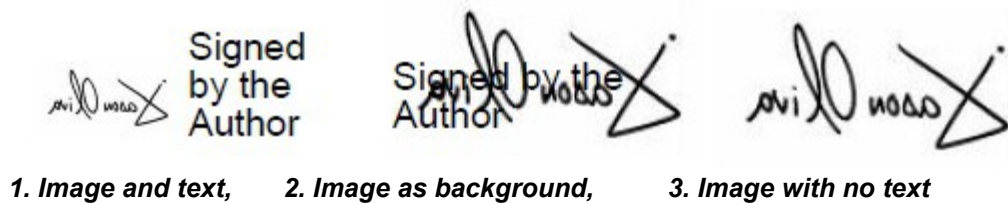
Signature text

Set the Digital Signature Graphic

The digital signature rectangle can contain text, graphic or text with graphic. To add an image on the digital signature rectangle, you can do that from *Place an image on the signature box* section.



These types of signatures are shown below:



Signing Reason and Location

The signing reason and location attributes can be set from the main interface.

Signing reason: I approve this document
Signing location: Europe branch

Signature is VALID, signed by Test Certificate <test@test.com>.

Summary Document Signer Date/Time Legal

Signed by: Test Certificate <test@test.com> Show Certificate...

Reason: I approve this document

Date: 2011/06/20 13:00:00 +03'00' Location: Europe branch

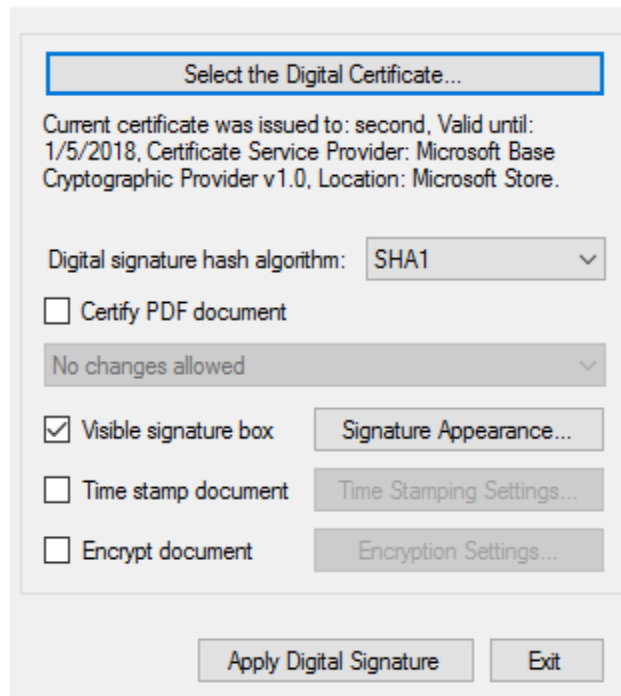
Test Certificate
2011.08.20 13:00
I approve this document
Europe branch
This is a demo version

Signed by, Reason, Location and Date properties in Adobe

Using SHA256, SHA512 Hash Algorithms

The default hash algorithm used by the library is **SHA1** but in some cases, SHA256/384/512 must be used for the digital signature and the Time Stamp Request.

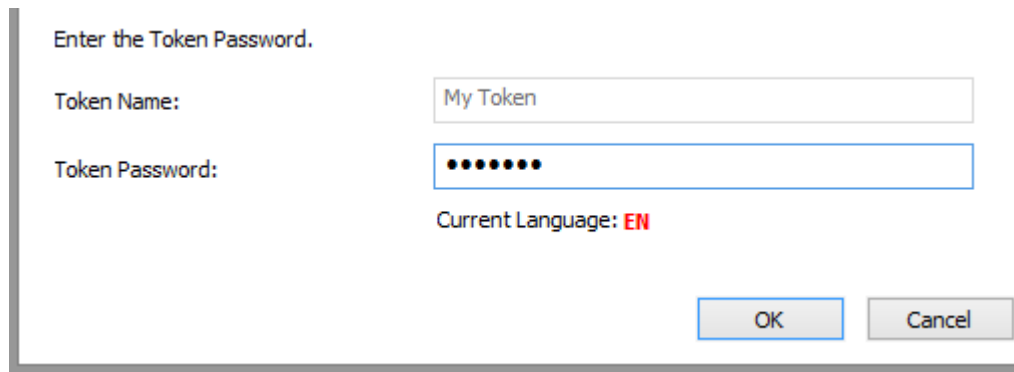
Attention: SHA-256 and SHA-512 hash algorithms are not supported by Windows XP. Note that some smart cards and USB tokens not support SHA-256 and SHA-512 hash algorithms.



Set the hash algorithm

Bypassing the Smart Card PIN

In case the digital signature must be made without user intervention and the certificate is stored on a smart card or USB token, the PIN dialog might be automatically bypassed for some models.



Enter the Token Password.

Token Name: My Token

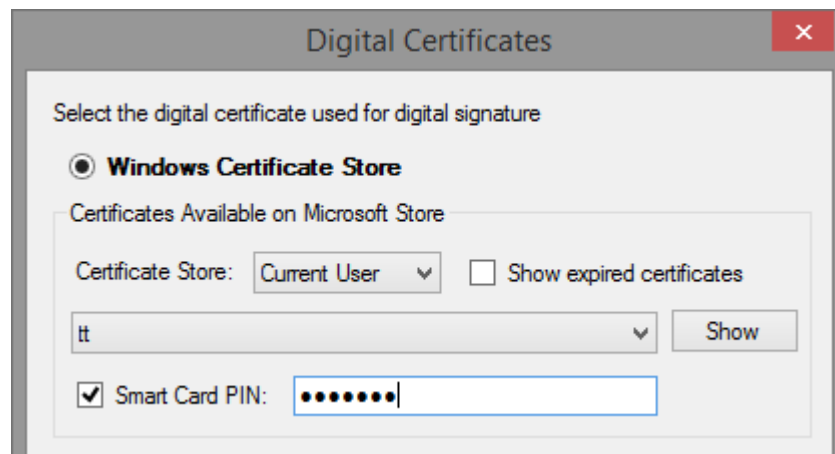
Token Password: ●●●●●●

Current Language: EN

OK Cancel

PIN dialog can be bypassed

In order to bypass the PIN dialog window, the Smart Card PIN checkbox must be checked and the right PIN to be entered. `DigitalCertificate.SmartCardPin` property must be set. This option bypass the PIN dialog and the file is automatically signed without any user intervention.



Digital Certificates

Select the digital certificate used for digital signature

Windows Certificate Store

Certificates Available on Microsoft Store

Certificate Store: Current User Show expired certificates

tt Show

Smart Card PIN: ●●●●●●

Bypassing the Smart Card PIN

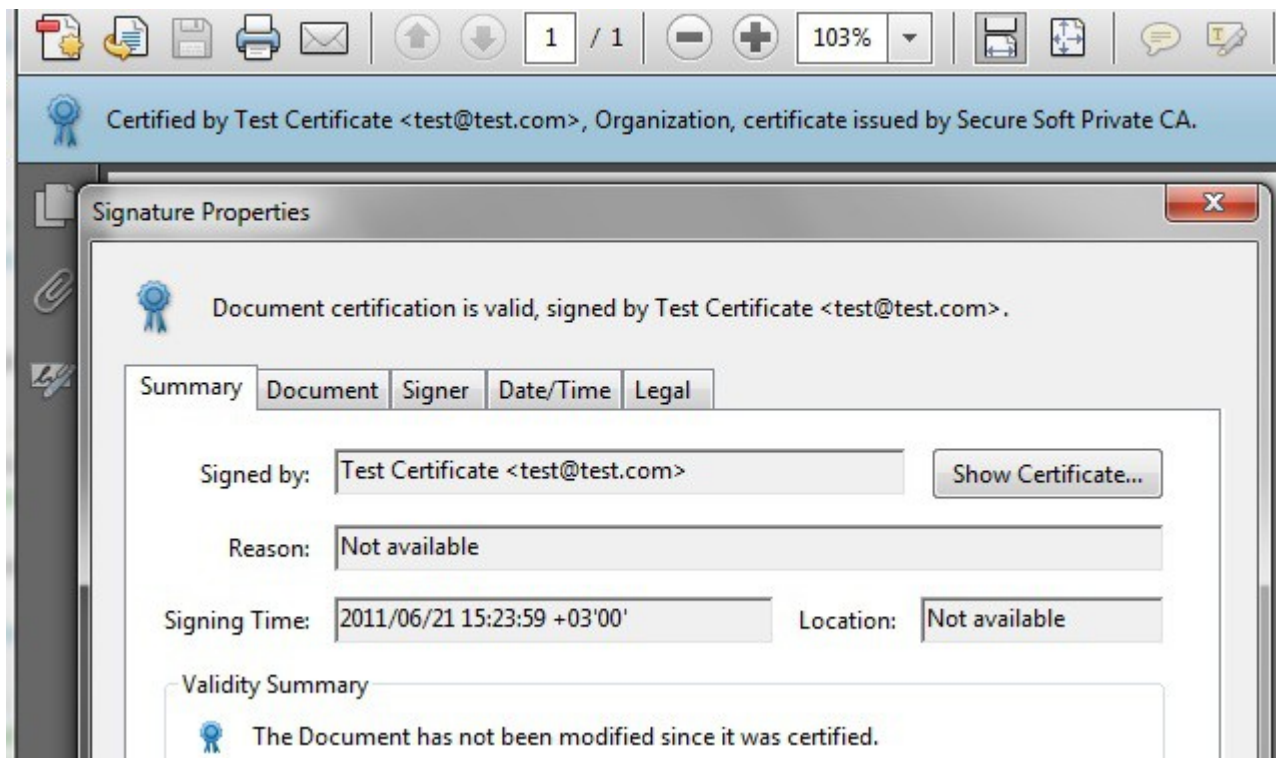
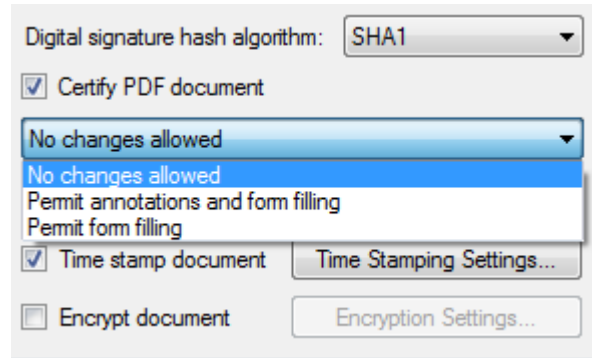
Attention: This feature will NOT work for all available smart card/USB tokens because of the drivers or other security measures. Use this property carefully.

Certify a PDF Digital Signature

When you certify a PDF, you indicate that you approve of its contents. You also specify the types of changes that are permitted for the document to remain certified.

You can apply a certifying signature only if the PDF doesn't already contain any other signatures. Certifying signatures can be visible or invisible. A blue ribbon icon in the Signatures panel indicates a valid certifying signature.

To certify a digital signature, select the certification type from the main interface.

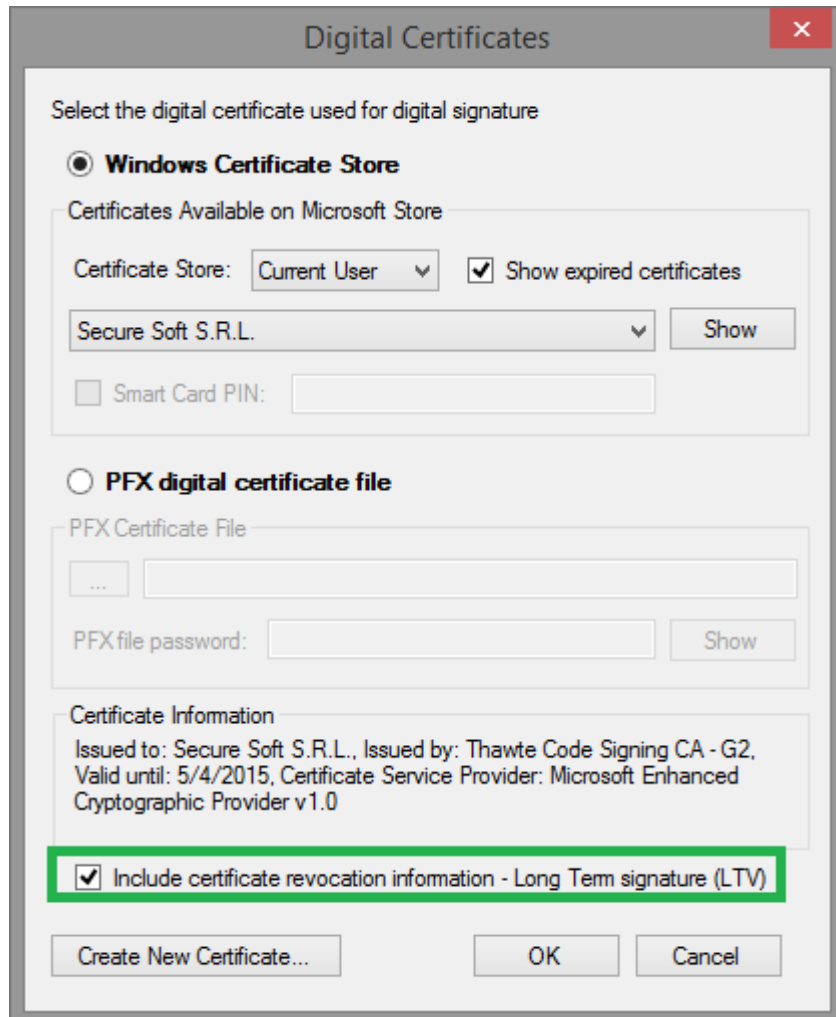


Certified signature

Include the CRL Revocation Information on the PDF Signature

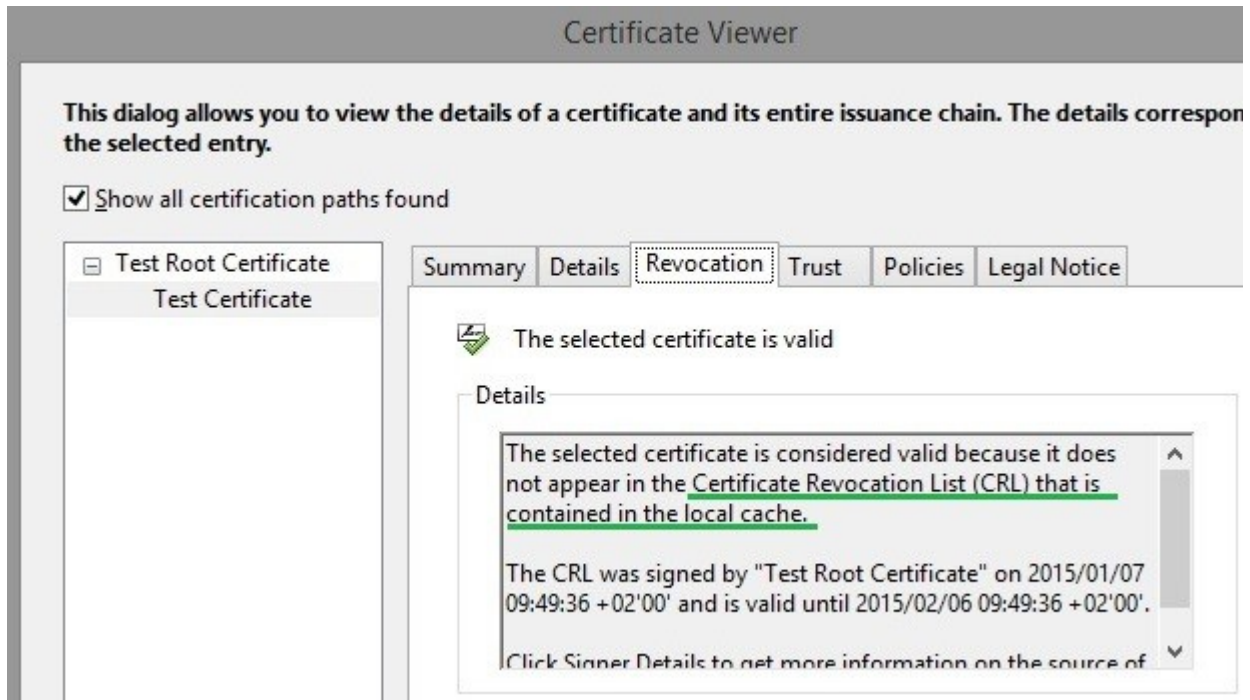
If the revocation information will not be available online, the digital signature cannot be verified by the Adobe Reader engine so it is recommended to include the CRL on the signature block.

This setting is available on the Digital Certificates window.

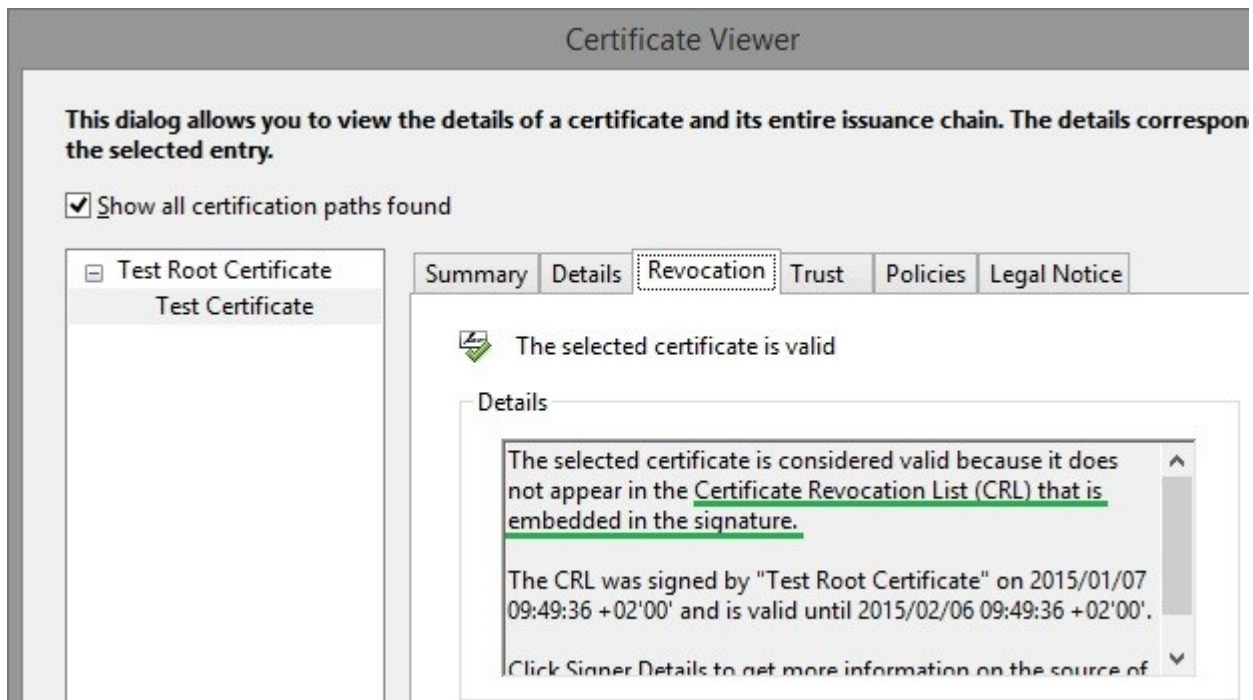


Note that some revocation information files (CRL) are very large so resulting signed file will proportionally larger.

PDF Signer Server will try to include CRL for every digital certificate from the chain.



A PDF digital signature without revocation information



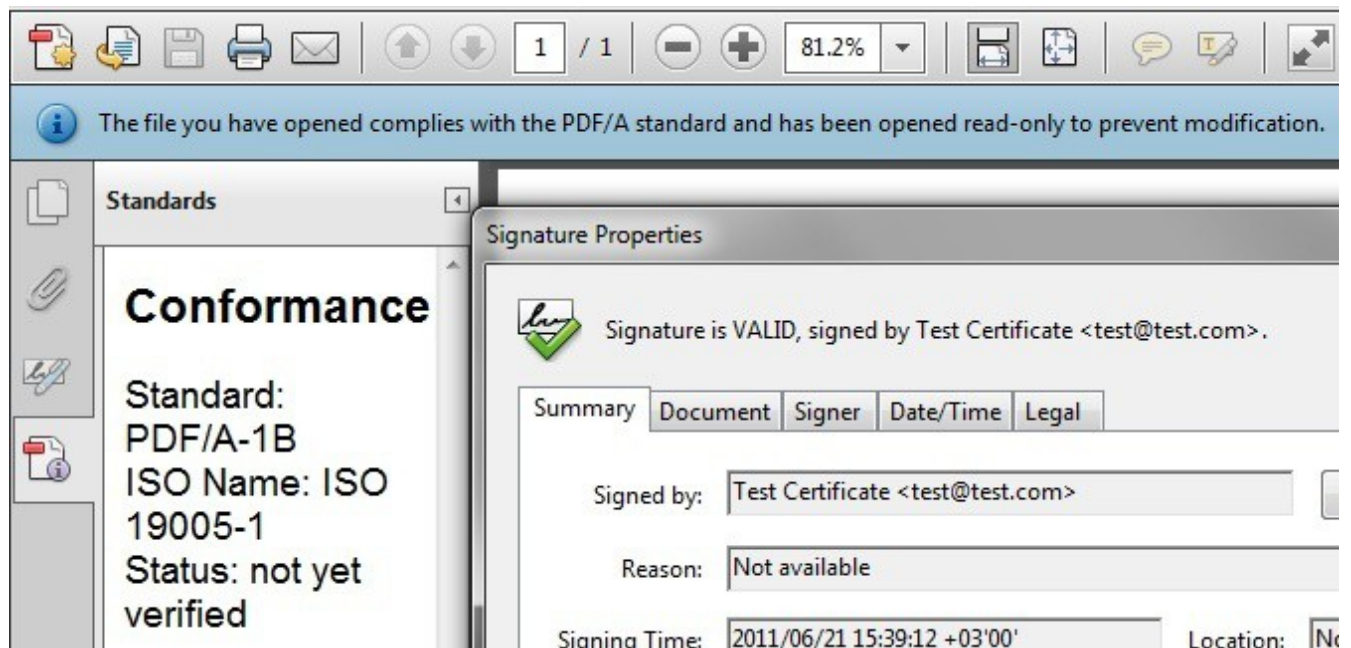
A PDF digital signature that embeds the revocation information

PDF/A Standard

PDF/A is a file format for the long-term archiving of electronic documents. It is based on the PDF Reference Version 1.4 from Adobe Systems Inc. (implemented in Adobe Acrobat 5 and latest versions) and is defined by ISO 19005-1:2005.

PDF Signer Server can digitally sign PDF/A files.

Observation: In order to save a PDF/A file, all fonts used on the PDF document must be embedded (including the font used on the digital signature rectangle). The digital signature font can be set on the Signature Appearance section.



PDF/A-1b document with digital signature

Time Stamping

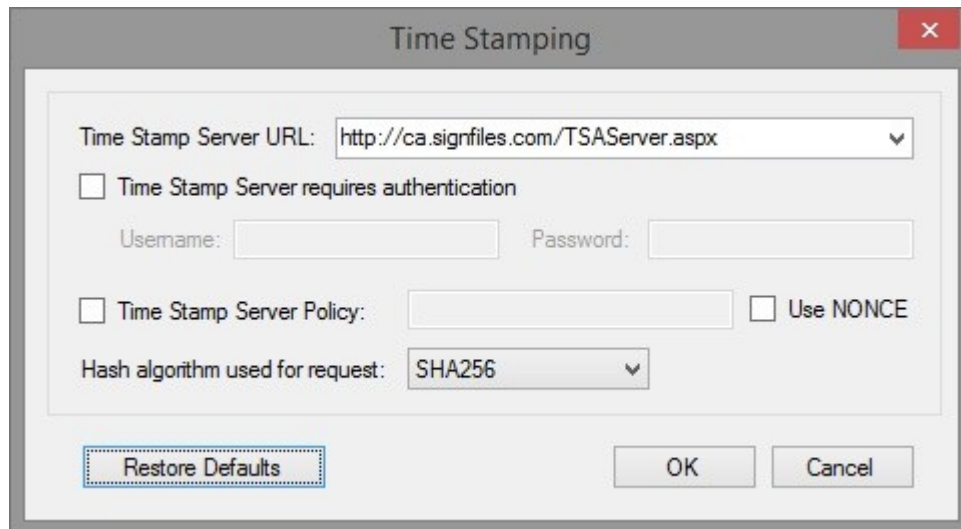
Time Stamp the PDF Digital Signature

Timestamping is an important mechanism for the long-term preservation of digital signatures, time sealing of data objects to prove when they were received, protecting copyright and intellectual property and for the provision of notarization services.

To add time stamping information to the PDF digital signature you will need access to a [RFC 3161](#) time stamping server.

A fully functional version of our TSA Authority is available for testing purposes at this link: <http://ca.signfiles.com/TSAserver.aspx> (no credentials are needed).

The Time Stamping options can be configured on the *Time Stamping* section.



The screenshot shows a dialog box titled "Time Stamping" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Time Stamp Server URL:** A text box containing "http://ca.signfiles.com/TSAserver.aspx" with a dropdown arrow on the right.
- Time Stamp Server requires authentication**
- Username:** An empty text box.
- Password:** An empty text box.
- Time Stamp Server Policy:** An empty text box.
- Use NONCE**
- Hash algorithm used for request:** A dropdown menu with "SHA256" selected.
- Buttons:** "Restore Defaults" (with a dotted border), "OK", and "Cancel".

Nonce and Policy

The **Nonce**, if included, allows the client to verify the timeliness of the response when no local clock is available. The nonce is a large random number with a high probability that the client generates it only once (e.g., a 64 bit integer).

Some TSA servers require to set a **Time Stamp Server Policy** on the Time Stamp Requests. By default, no Time Stamp Server Policy is included on the TSA request.

Validating the Time Stamp Response on Adobe

As digital signatures certificates, the time stamping responses are signed by a certificate issued by a Certification Authority.

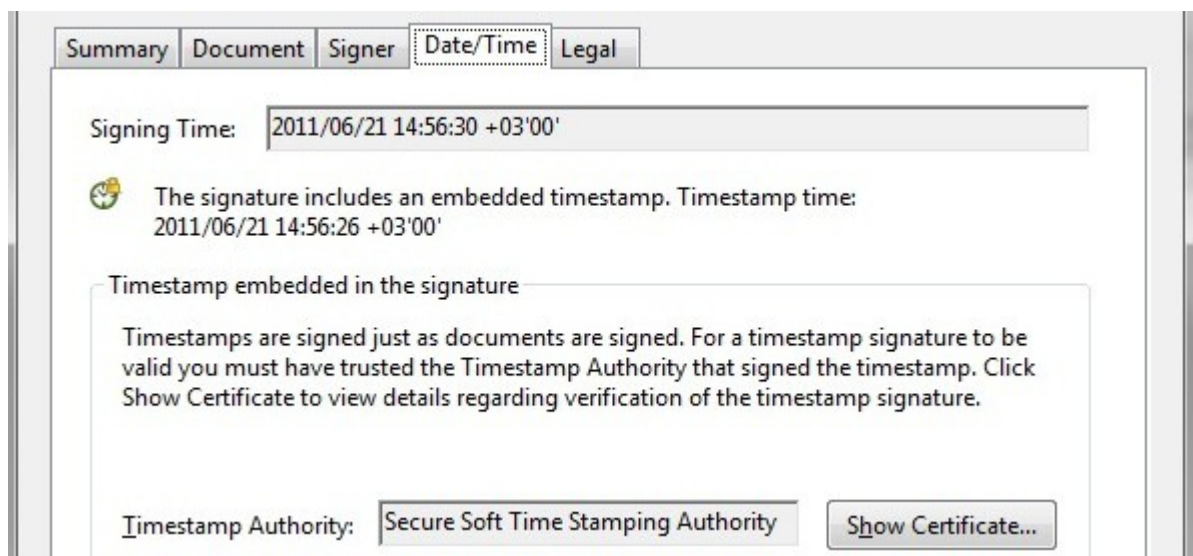
If the time stamping certificate (or the Root CA that issued the time stamping certificate) is not included in Adobe Store, the time stamping response could not be verified when a user open a document with Adobe Reader (see example).

This behavior has nothing to do with the signing engine but with the Adobe certification validation procedure.

To validate the signing certificate in Adobe use the methods described on this document: <http://www.signfiles.com/manuals/ValidatingDigitalSignaturesInAdobe.pdf>.



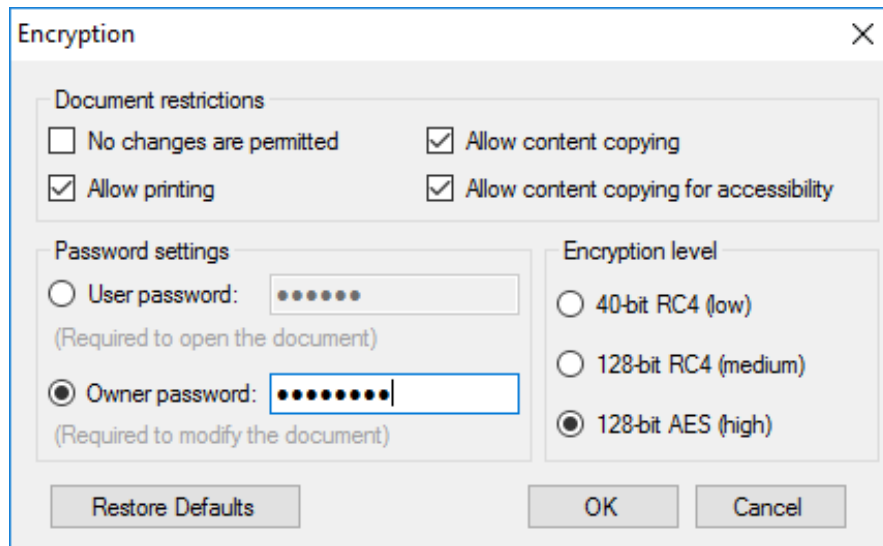
Not verified timestamp



Trusted time stamping response

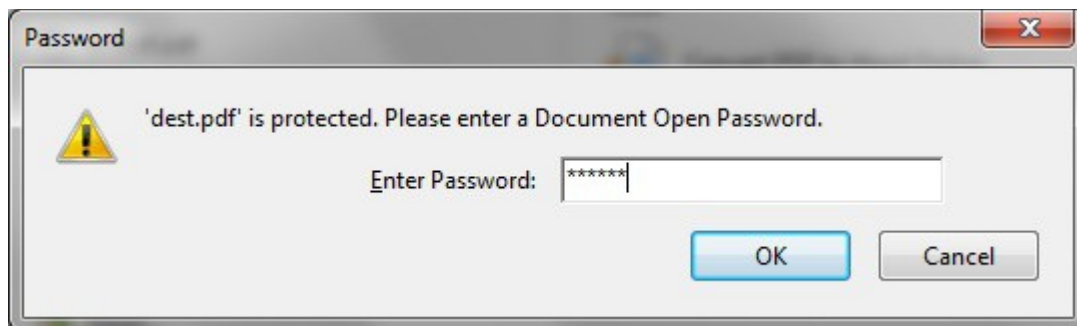
Encryption

If you want to protect the signed document by preventing actions like printing or content copying you must encrypt it. The document can be encrypted using passwords from *Encryption* section.



Encryption settings

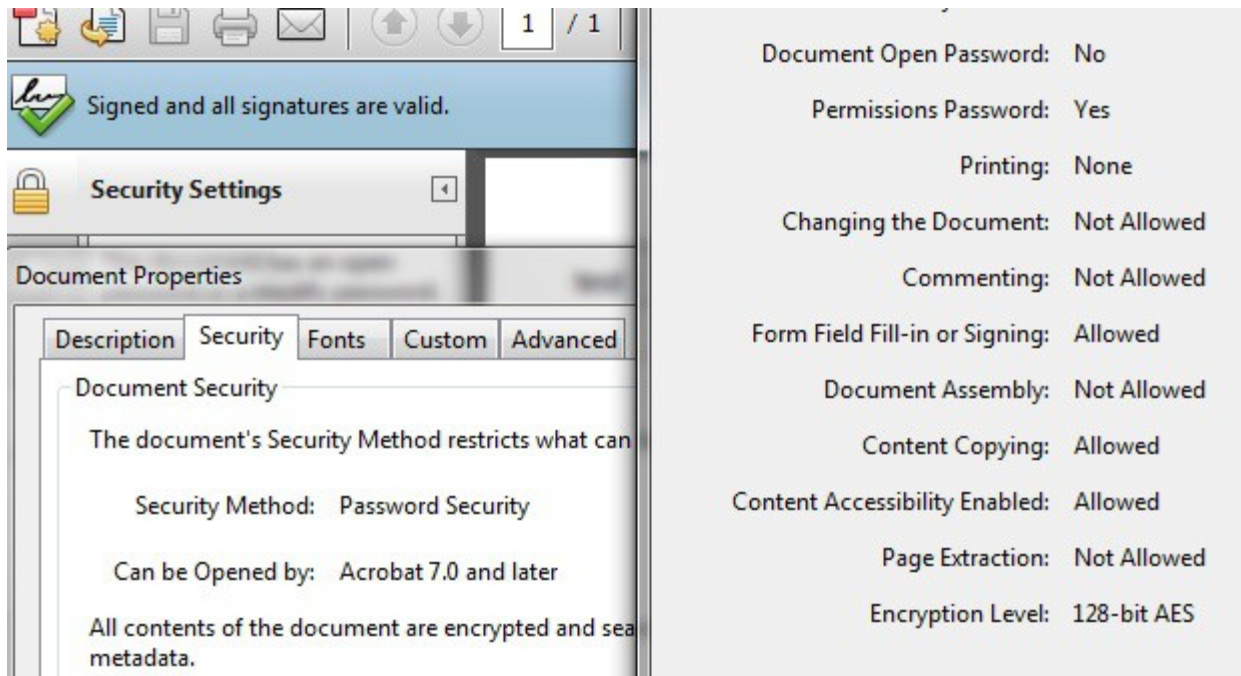
If the PDF document is signed and encrypted with an *User Password*, when the document is opened in PDF reader, the PDF document password must be entered.



Password is required to open the document

Owner Password is used to set the password that protects the PDF document for printing or content copying.

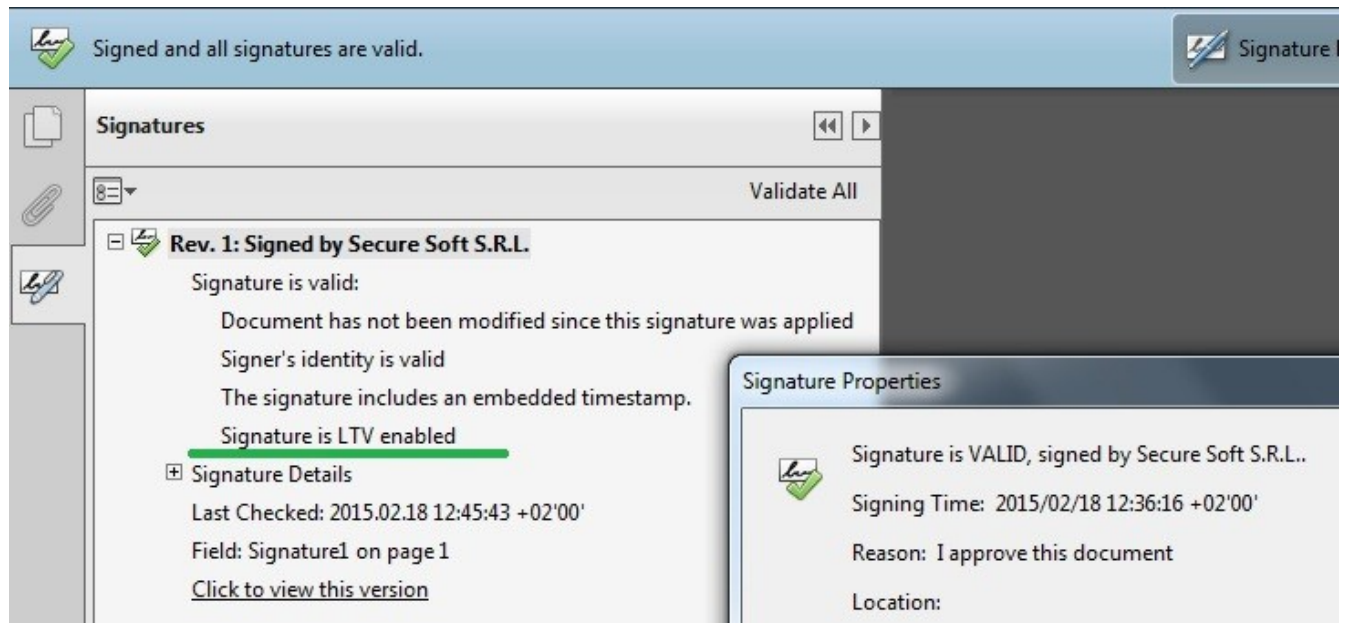
When the signed and encrypted document is opened in a PDF reader, the security settings are shown like below.



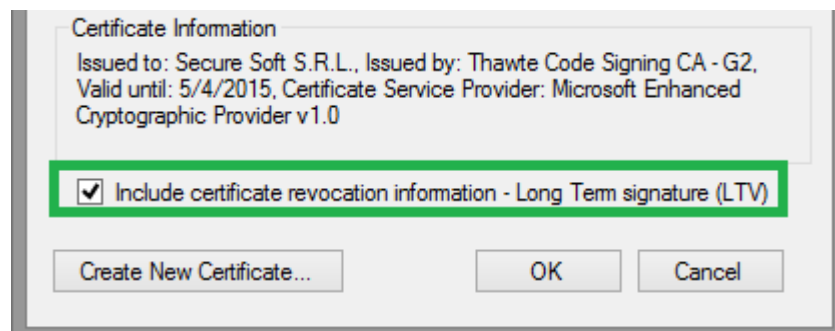
Security settings for a digitally sign and encrypted document

LTV Signatures (Long Term Validation)

PAdES recognizes that digitally-signed documents may be used or archived for many years – even many decades. At any time in the future, in spite of technological and other advances, it must be possible to validate the document to confirm that the signature was valid at the time it was signed – a concept known as Long-Term Validation (LTV).



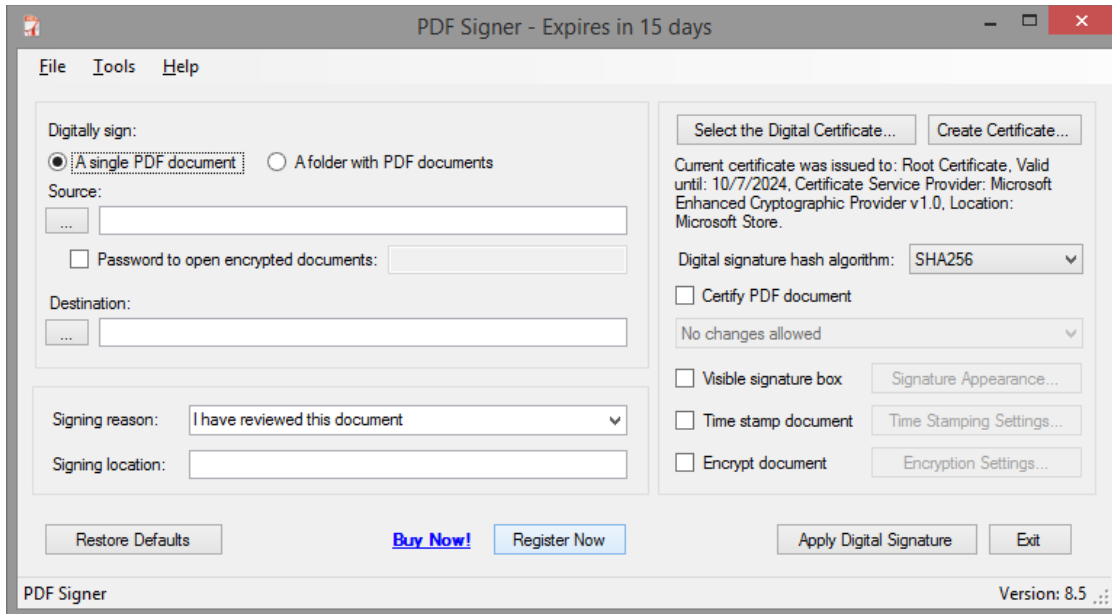
In order to have a LTV signature, be sure that on the Digital Certificates settings, the checkbox *Include certificate revocation information – Long Term signature (LTV)* is checked.



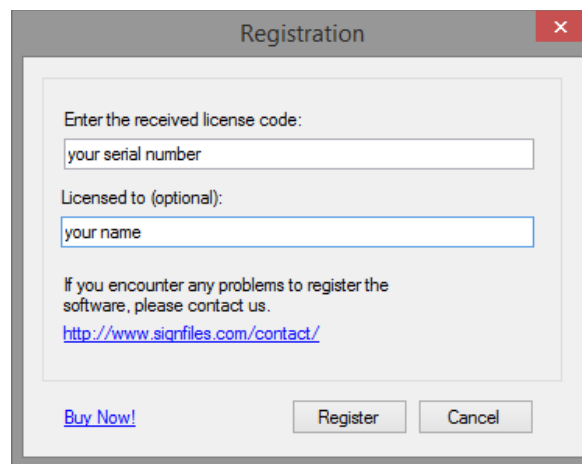
Product Registration

To register the product you will need a serial number. It can be purchased online directly from the product main page.

After you will obtain your serial number, open PDF Signer Server and click Register Now button.

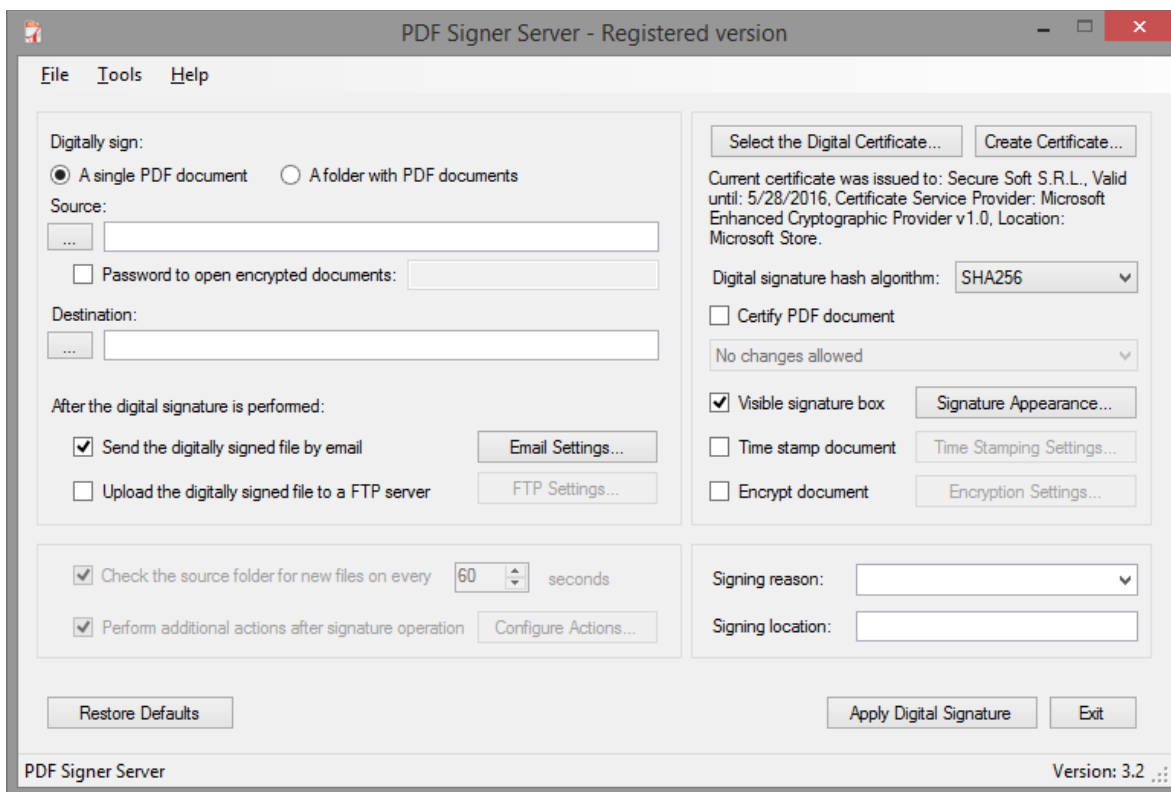
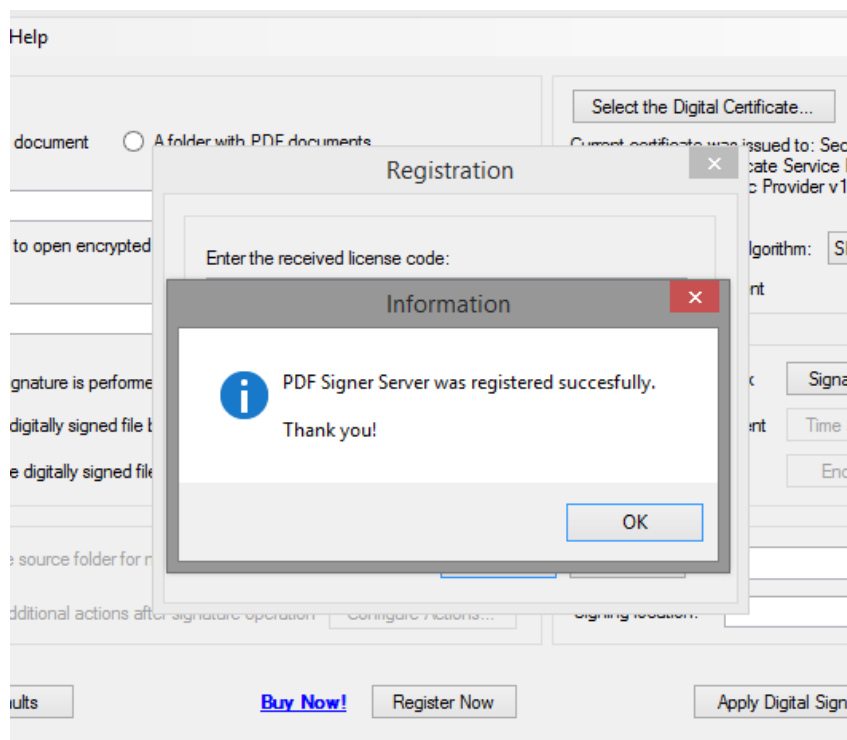


Enter the received serial on the Registration window, as below:



Click Register button.

If the serial number is correct, the product will be successfully registered.



Email Option

After the file is successfully signed, it can be sent by email to a specific email address.

For this operation are necessary some prerequisites:

- You must be sure that the Outgoing SMTP Server can be accessed and used from your location
- The Email Server Options was configured on the PDF Signer Server interface

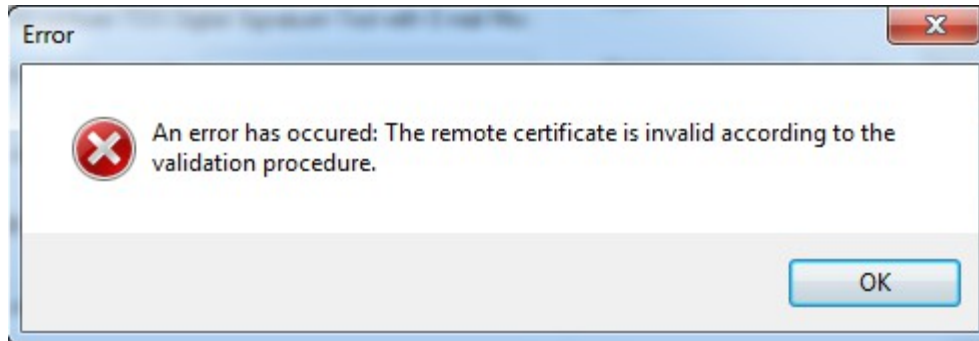
The screenshot shows the 'Email Options' dialog box with the following configuration:

- Server settings:**
 - Email Server: 172.30.57.51
 - Port: 25
 - Enable SSL
- Email Server authentication credentials (optional):**
 - Username: registration
 - Password: [masked]
 - Enable strong SSL email server certificate validation
- Email settings:**
 - From Address: office@signfiles.com
 - To Address: test_email@gmail.com
 - Email subject: Email signed File
 - Email body file: D:\body.txt
 - Send email in HTML format

The Email server options can be set from the Email Option dialog box.

Enable strong SSL email server certificate validation must be enabled only if the email server certificate is trusted, not expired and issued by a well known certificate authority (like Thawte, Comodo, GlobalSign).

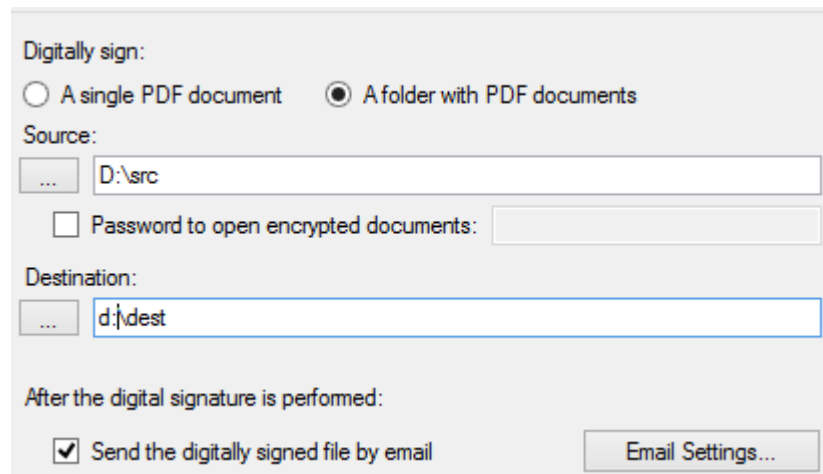
The error message above is thrown if the Server Certificate is not trusted or expired.



If this error message persists, you can disable SSL connection (**Enable SSL** checkbox).

Email Workflow

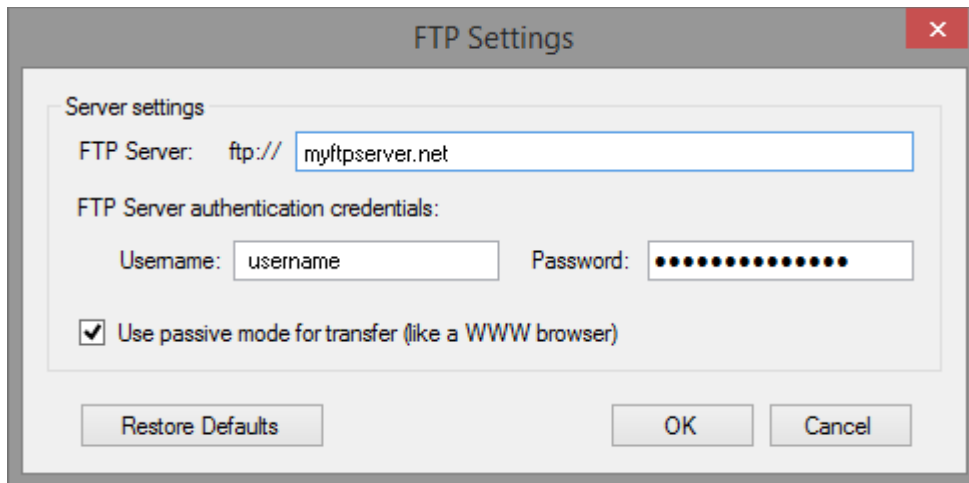
After the file is digitally signed correctly, the file is added as attachment on a new email message. This email message is passed to the email server that will try to deliver the email message to the recipient set on the Email Option window.

A configuration window for digital signing. It includes the following elements:

- Digitally sign:** Two radio buttons: "A single PDF document" (unselected) and "A folder with PDF documents" (selected).
- Source:** A text box containing "D:\src" with a browse button ("...") to its left.
- Password to open encrypted documents:** A checkbox (unchecked) followed by an empty text box.
- Destination:** A text box containing "d:\dest" with a browse button ("...") to its left.
- After the digital signature is performed:** A checkbox (checked) labeled "Send the digitally signed file by email".
- Email Settings...:** A button located at the bottom right of the window.

FTP Option

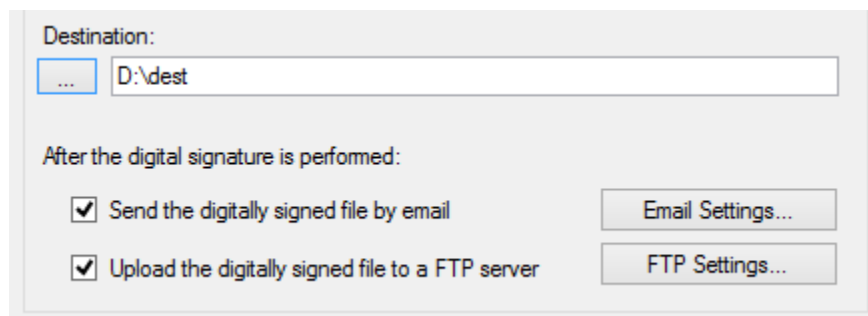
After the file is digitally signed, it can be also uploaded to a FTP account.



Attention: The FTP directory must exist on the server. If the file already exists on the FTP account it will be rewritten without any notice.

FTP Workflow

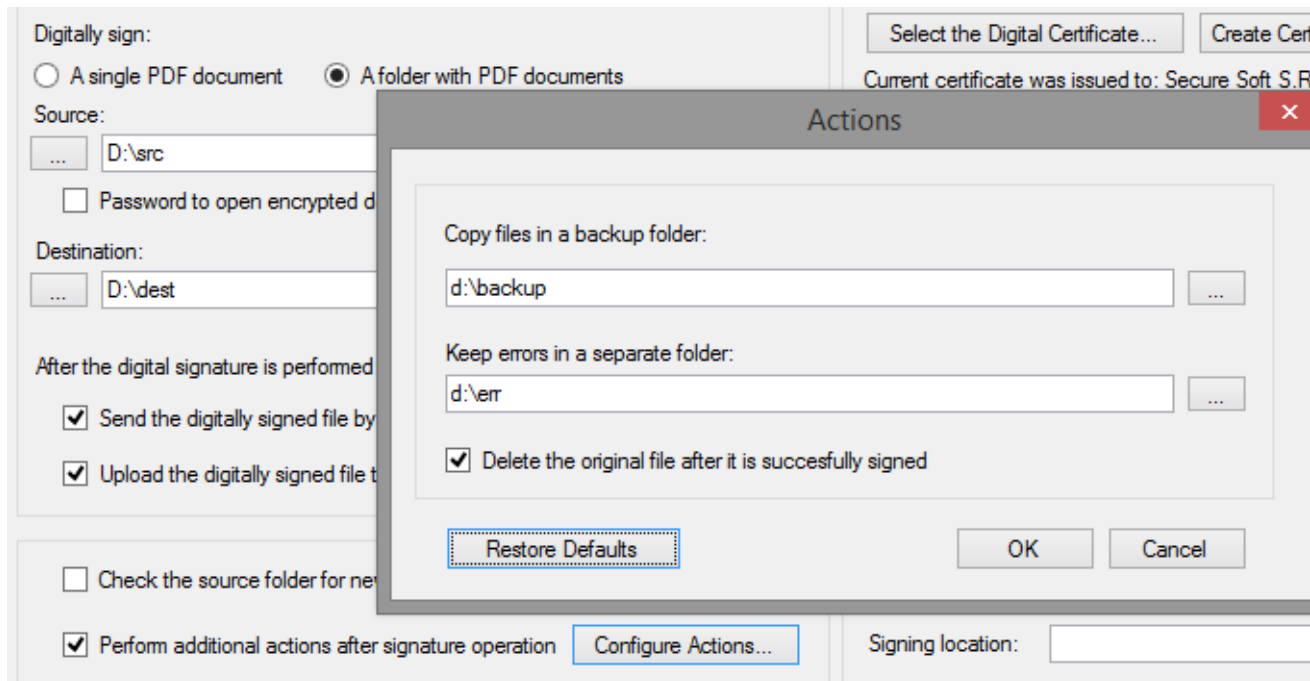
After the file is digitally signed correctly it can be optionally sent by email. After that it is uploaded to the FTP server.



Additional Actions

If *Perform additional actions after signature operation* checkbox is checked, some actions can occur after the folder is digitally signed.

Attention: these actions will not apply if “*A single PDF document*” is checked.



Backup Folder (Archive Folder)

If the backup folder is set, every file from the source folder (D:\src) will be archived on the folder d:\backup. This action will occur only if the file is successfully signed.

Attention: If the file already exists on the backup folder it will be rewritten without any notice.

Error Folder

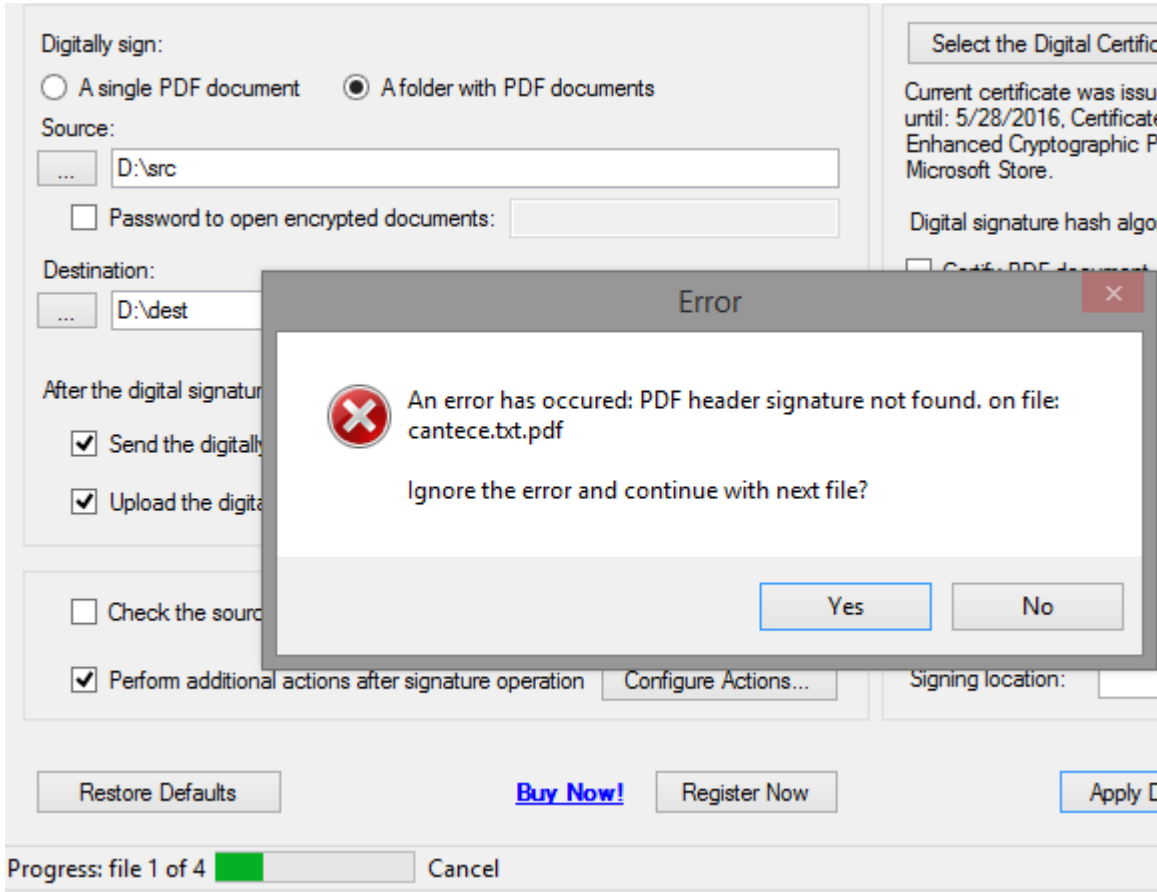
If the error folder is set, every source file that is not completely processed (either because the file is invalid or not digitally signed or not sent by email) will be saved to the error folder.

If the error folder is not set, the files will remain on the Source directory until they are manually deleted.

If “*Delete the original file after it is successfully signed*” is checked, every file that was successfully signed will be deleted from the source folder.

Batch Mode

Attention: If the checkbox *“Check the source folder for new files”* is NOT checked, every error thrown by the application will appear on the screen and the user must manually continue or discard the entire process.



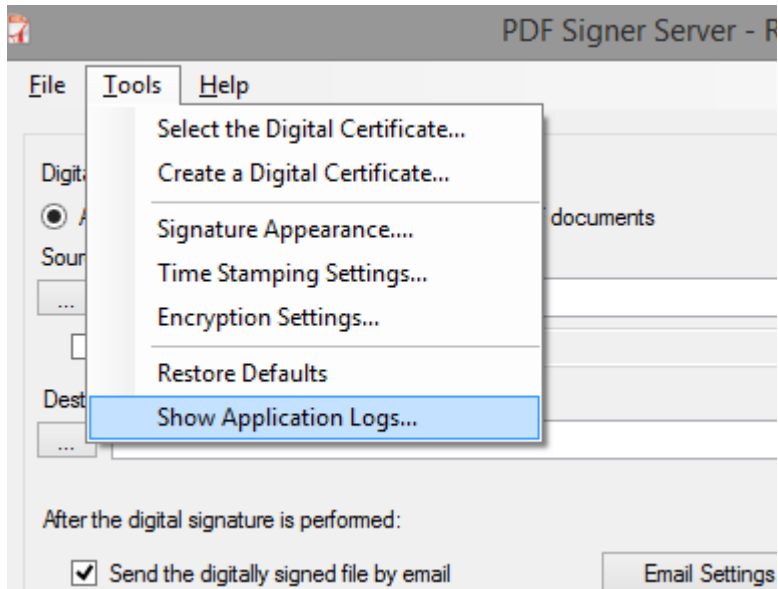
If the checkbox *“Check the source folder for new files”* IS checked, the error message boxes will not appear and the source files will be automatically send on the error folder (if it is set).

If an error will occur (no matter if it was an error on digital signature, email or ftp) that file will go to the error folder. The erroneous file will not go to the Backup folder. On this way, it is known what files are (Backup folder) or are not (Error folder) successfully processed.

If the error folder and the backup folder are set and the checkbox *“Delete the original file after it is successfully signed”* is checked, every file from the source folder will be processed. On this situation, the source folder will remain empty and will be ready for the next batch process.

Logging

Every action of PDF Signer Server is logged on a text file available here:
"%appdata%\Secure Soft\PDF Signer Server\3.2\log.txt"



Batch Signatures (Automatically Made Without User Intervention)

By default, PDF Signer Server is installed on this location:

C:\Program Files\Secure Soft\PDF Signer Server\PDF Signer Server.exe.

The command line parameters are:

PDF Signer Server.exe <source file | folder> <destination file | folder> [<XML configuration file>]

To automatically sign a **PDF file**, use the following command:

c:\Program Files\Secure Soft\PDF Signer Server>"PDF Signer Server.exe" c:\InputFile.pdf c:\SignedFile.pdf

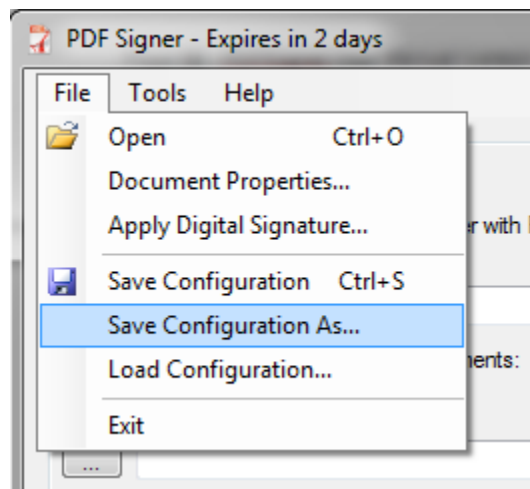
To automatically sign a **folder** that contains PDF files, use the following command:

c:\Program Files\Secure Soft\PDF Signer Server>"PDF Signer Server.exe" c:\InputFolder c:\OutputFolder

Custom Configuration

In some cases, you will need a different signature configuration (e.g. different signature appearance and digital certificates) for different PDF files/folders.

To save a specific configuration, go to *File – Save Configuration As* and save the configuration on a file. Later, you can use that file in batch mode to apply different signature configuration on your signed PDF file.



To automatically sign a **folder** that contains PDF files, using a custom configuration, use the following command:

"PDF Signer Server.exe" c:\InputFolder c:\OutputFolder c:\config-client2.xml

Digitally Sign PDF Files Using Windows PowerShell

PDF Signer Server main functions are available on *SignLib library* available at this link: <http://www.signfiles.com/sdk/SignatureLibrary.zip>

To digitally sign PDF file using Windows PowerShell, simply download the library above and inspect *Signature Library\PowerShell Scripts* folder.

The Windows PowerShell script will look below:

```
#digitally sign a PDF file using a PFX certificate created on the fly
#the script can be configured to use an existing PFX file or a certificate loaded from
Microsoft Store (smart card certificate)

if ($args.Length -eq 0)
{
    echo "Usage: signpdf.ps1 <unsigned file> <signed file>"
}
else
{
    $DllPath = 'd:\SignLib.dll'
    [System.Reflection.Assembly]::LoadFrom($DllPath)

    #create a PFX digital certificate
    $generator = new-object -typeName SignLib.Certificates.X509CertificateGenerator("serial
number")
    $pFXFilePassword = "tempP@ssword"

    $generator.Subject = "CN=Your Certificate, E=useremail@email.com, O=Organzation"
    $generator.Extensions.AddKeyUsage([SignLib.Certificates.CertificateKeyUsage]::DigitalSignatu
re)
    $generator.Extensions.AddEnhancedKeyUsage([SignLib.Certificates.CertificateEnhancedKeyUsage]
::DocumentSigning)
    $certificate = $generator.GenerateCertificate($pFXFilePassword)

    #digitally sign the pdf file
    $sign = new-object -typeName SignLib.Pdf.PdfSignature("serial number")
    $sign.LoadPdfDocument([System.IO.File]::ReadAllBytes($args[0]))
    $sign.DigitalSignatureCertificate =
    [SignLib.Certificates.DigitalCertificate]::LoadCertificate($certificate, $pFXFilePassword)

    $sign.SigningReason = "I approve this document"
    $sign.SigningLocation = "Europe branch"
    $sign.SignaturePage = 1
    $sign.SignaturePosition = [SignLib.Pdf.SignaturePosition]::TopRight

    echo "Perform the digital signature..."
    [System.IO.File]::WriteAllBytes($args[1], $sign.ApplyDigitalSignature())
}
```

How to run the Windows PowerShell script from command line:

```
powershell -executionPolicy bypass -file d:\signpdf.ps1 d:\unsigned.pdf d:\signedFile.pdf
```


Digitally Sign PDF Files Using C# or VB.NET

PDF Signer Server main functions are available on *SignLib library* available at this link:
<http://www.signfiles.com/sdk/SignatureLibrary.zip>

To digitally sign PDF file using C# or VB.NET, download the library above and inspect *Signature Library\VS2008 Projects* folder.

The C# will look like below:

```
PdfSignature ps = new PdfSignature("your serial number");

//load the PDF document
ps.LoadPdfDocument(unsignedDocument);
ps.SignaturePosition = SignaturePosition.TopRight;
ps.SigningReason = "I approve this document";
ps.SigningLocation = "Accounting department";

ps.SignaturePosition = SignaturePosition.TopLeft;

//Digital signature certificate can be loaded from various sources

//Load the signature certificate from a PFX or P12 file
ps.DigitalSignatureCertificate =
DigitalCertificate.LoadCertificate(Environment.CurrentDirectory + "\\cert.pfx",
"123456");

//Load the certificate from Microsoft Store.
//The smart card or USB token certificates are usually available on Microsoft
Certificate Store (start - run - certmgr.msc).
//If the smart card certificate not appears on Microsoft Certificate Store it
cannot be used by the library
//ps.DigitalSignatureCertificate = DigitalCertificate.LoadCertificate(false,
string.Empty, "Select Certificate", "Select the certificate for digital
signature");

//write the signed file
File.WriteAllBytes(signedDocument, ps.ApplyDigitalSignature());
```