How to Time Stamp PDF and Microsoft Office 2010/2013 Documents with the Time Stamp Server

Introduction

Time stamping 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.

Our Time Stamp Authority works as an IIS application for most Windows webservers. It means that it is not required to operate an extra TSA machine.

Links

Download Time Stamp Server for IIS: http://www.signfiles.com/apps/TSAServer.zip
Time Stamp Server Live Demo: http://ca.signfiles.com/tsa/
Time Stamp Server main page: http://www.signfiles.com/timestamping/

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, Microsoft Office, Microsoft Word 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 Features

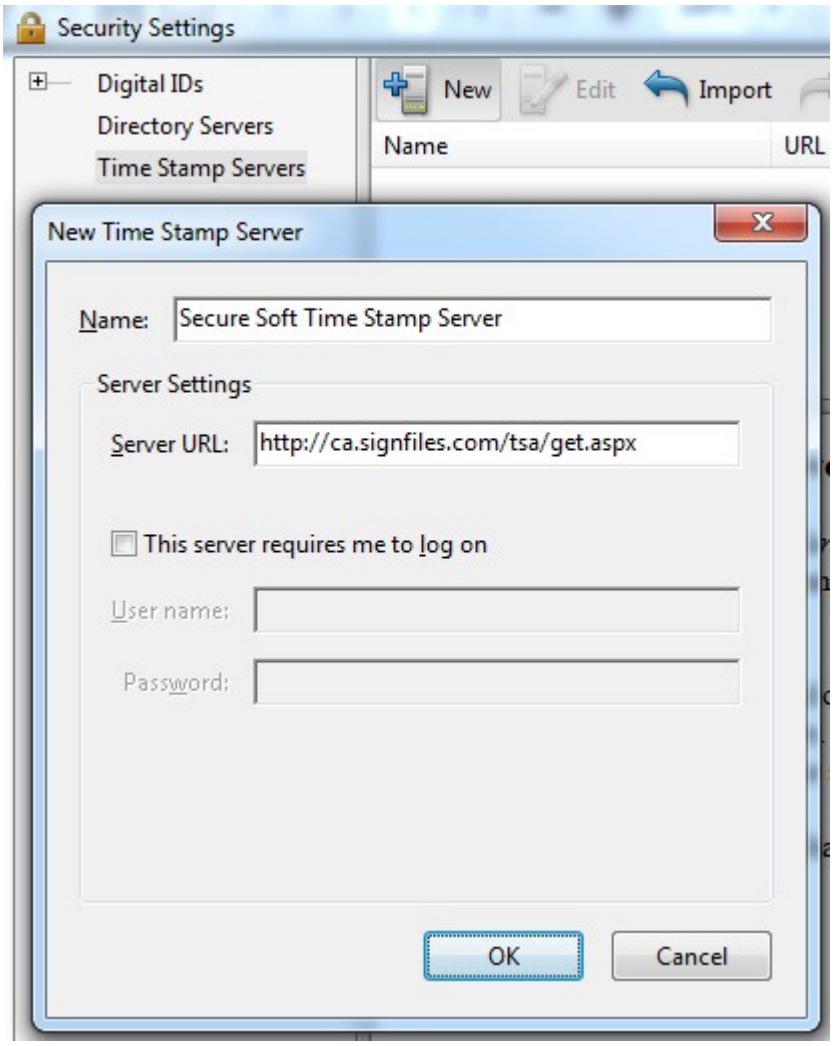
The Time Stamp Server is an ASP.NET application written in C# with .NET Framework 2.0. The entire application, including RFC3161 TSA logic, is fully written in managed code that means there are no calls to external DLL's, libraries, middleware or PKCS#11 drivers.

Benefits:
- IETF RFC 3161 TSP compliant
- Works with HSM (Hardware Security Module) devices
- X.509 standards compliant
- Works with any standards-based Certification Authority
- Uses the local available machine clock, synchronised to a UTC Time Server
- Easy to use through friendly programs like Adobe Acrobat or PDF Signer
- Supports up to 4096 bit RSA keys
- Up to 40 timestamps per second

Requirements:
- Windows XP or later operating system with IIS
- Microsoft .NET Framework 2.0

Using the Time Stamp Server with Adobe Acrobat or Adobe Reader

1. Start Adobe Acrobat or Adobe Reader.
2. Go to *Edit* menu option – *Protection* submenu - *Security Settings* option…
3. Select *Time Stamp Servers* and click *New* button.
4. On *New Time Stamp Server* window, enter the name of the TSA Server and the TSA Server URL.
5. The checkbox *This server requires me to log on* must be unchecked.



Adding the Time Stamp Server on Adobe

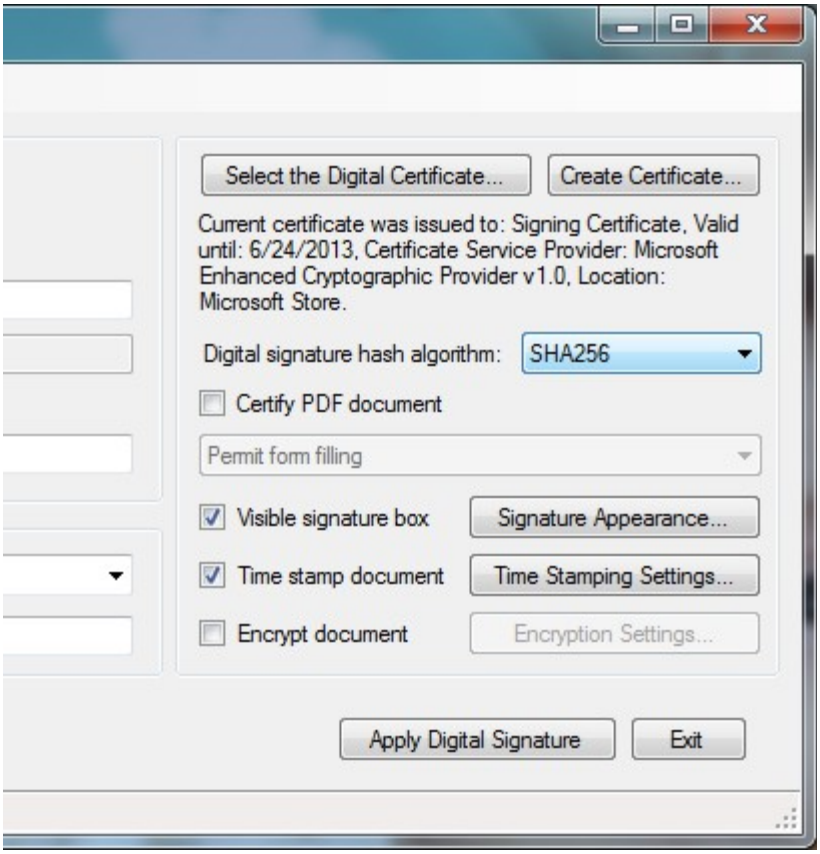
Using the Time Stamp Server with PDF Signer

Enable Time Stamping on PDF Signer

The main function of PDF Signer 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.

PDF Signer product is available at this link: http://www.signfiles.com/pdf-signer/

To enable time stamping, be sure the checkbox *Time stamp document* is checked, as below.

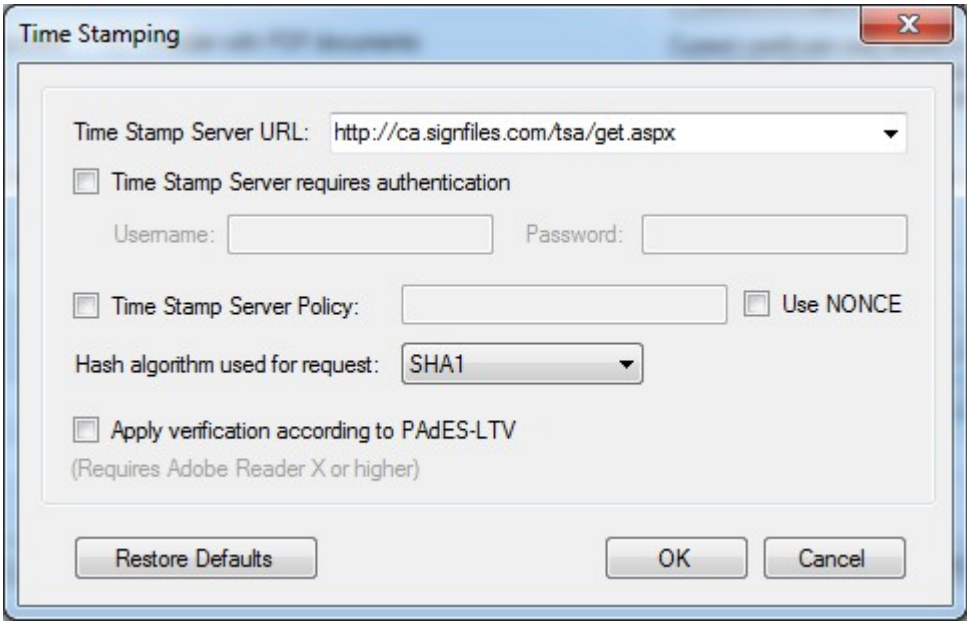


Enable time stamping on PDF Signer

Right now, the time stamp server options can be set by pressing *Time Stamp Settings* button.

Time Stamping Settings

On the Time Stamping dialog box, can be set the Time Stamp Server URL, authentication settings, Policy, Hash algorithm and PAdES-LTV options.



Time Stamping Options

If your Time Stamp Server requires **Authentication**, check the checkbox *Time Stamp Server requires authentication* and enter the username and the password.

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). To include a Nonce on the time stamping request, be sure the checkbox *Use NONCE* is checked.

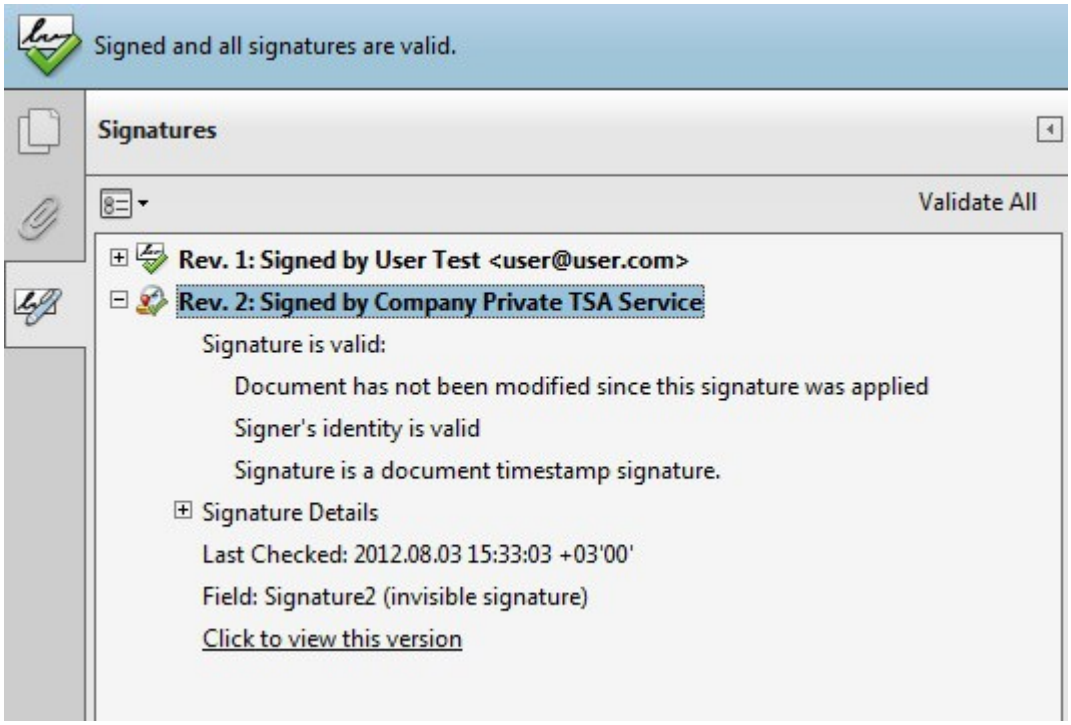
The Time Stamp Server could require a **Policy** OID on the TSA requests. To set a TSA policy OID on the time stamping requests, checkbox *Time Stamp Server Policy* must be checked and a valid OID (e.g. 1.3.6.1.4.1.13762.3) must be entered on the textbox. By default, no Policy OID is included on the TSA request.

The default (and recommended) **Hash Algorithm** used for the TSA request is **SHA1** but in some cases, SHA256/384/512 must be used (note that not all PDF readers can validate a such of signature).

Apply Verification According to PAdES-LTV (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 apply verification according to PadES-LTV for a PDF document, check the *Apply verification according to PadES-LTV* checkbox.



PAdES-LTV signed document

Validating the Time Stamping Information on Adobe

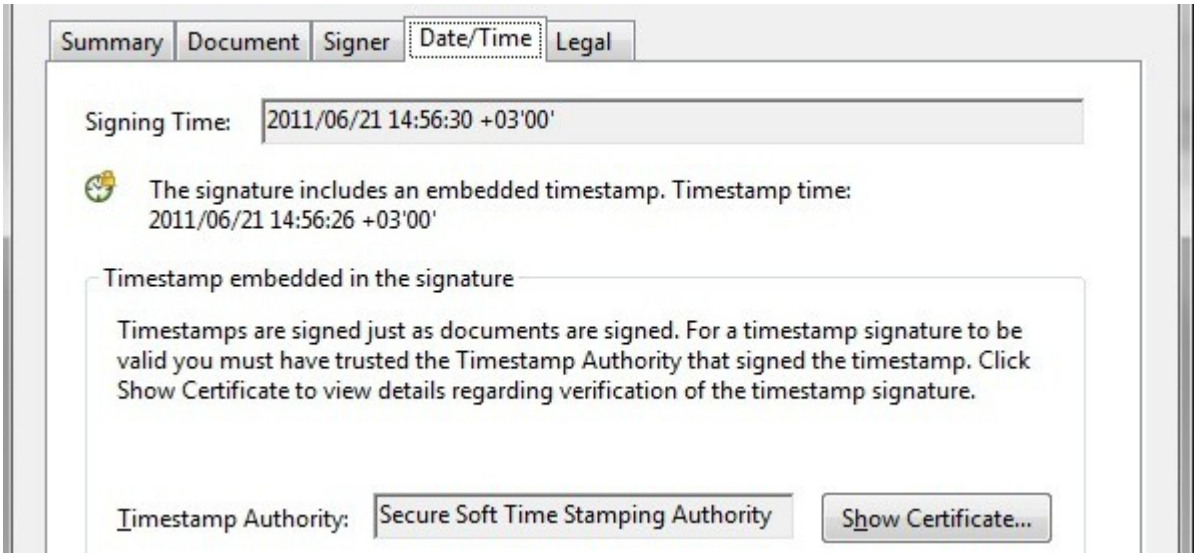
As digital signatures certificates, the time stamping responses are signed by a certificate issued by a Certification Authority.

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

Using the Time Stamp Server with Microsoft Office 2010/2013

Some important information are available on this article:
http://blogs.technet.com/b/office2010/archive/2009/12/08/digital-signitures-in-office-2010.aspx

XAdES-T Signatures

In order to create a time stamped signature (**XAdES-T** signatures), you'll need to:
- Set up the Time Stamp Server
- Configure signature policy registry to let the client systems know where to locate the timestamp server.

Once everything is configured, you can just create signatures like you normally would.

By default, Office 2010/2013 creates XAdES-EPES signatures. Registry settings are used to specify the level of signatures to create. There are two registry settings to control the type of signature Office creates, **XAdESLevel** and **MinXAdESLevel**.

XAdESLevel

Location	HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Common\Signatures
Description	This registry key specifies a requested XAdES level type signature you would like to create.
Type	REG_DWORD
List Values and Definition	0 – XAdES Off (Create XML-DSig signatures) 1 – Create XAdES-EPES signatures (Default) 2 – Create XAdES-T signatures 3 – Create XAdES-C signatures 4 – Create XAdES-X signatures 5 – Create XAdES-X-L signatures

MinXAdESLevel

Location	HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Common\Signatures
Description	This registry key specifies a minimum XAdES level that must be applied.
Type	REG_DWORD
List Values and Definition	0 – No minimum level (Default) 1 – Minimum level is XAdES-EPES 2 – Minimum level is XAdES-T 3 – Minimum level is XAdES-C 4 – Minimum level is XAdES-X 5 – Minimum level is XAdES-X-L

The *MinXAdESLevel* setting allows you to ensure that created signatures meet your required XAdES level. A XAdES-T or higher signature will fail if the timestamp server isn't available. Having a minimum setting allows scenarios where you could attempt a XAdES-T signature, but fall back to XAdES-EPES if the timestamp server is down.

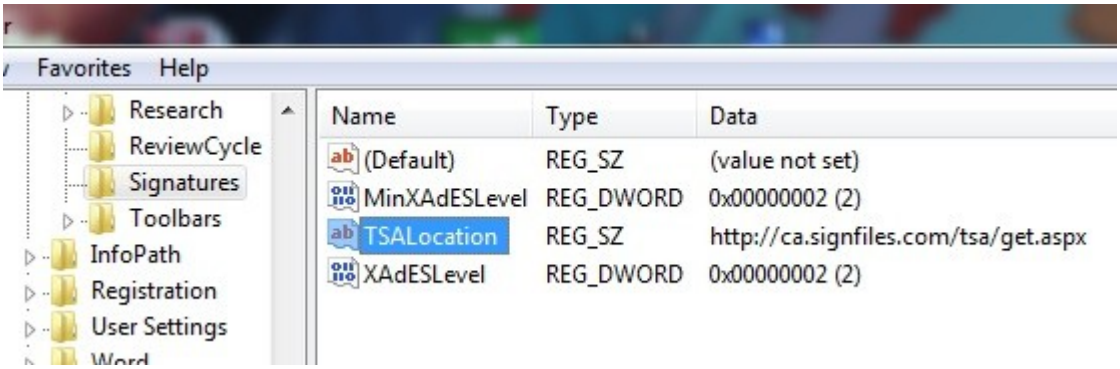
To create XAdES-T signatures and above you will need to provide Office with a time stamp server to query for time stamps:

TSALocation

Location	HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Common\Signatures
Description	This registry key specifies the URL of the time stamp server you will use to create timestamps.
Type	REG_SZ
List Values and Definition	Office needs the location of a time stamp server in order to query the server to issue time stamps. You should enter the location of your HTTP-based time stamp server (which is RFC 3161 compliant) in this registry key.

Registry Settings for XAdES-T Signatures

All registry setting must be manually entered on the Registry. However, the registry setting are available for download at this link: http://signfiles.com/apps/XAdESTInOffice.zip



Office 2010 Registry Settings

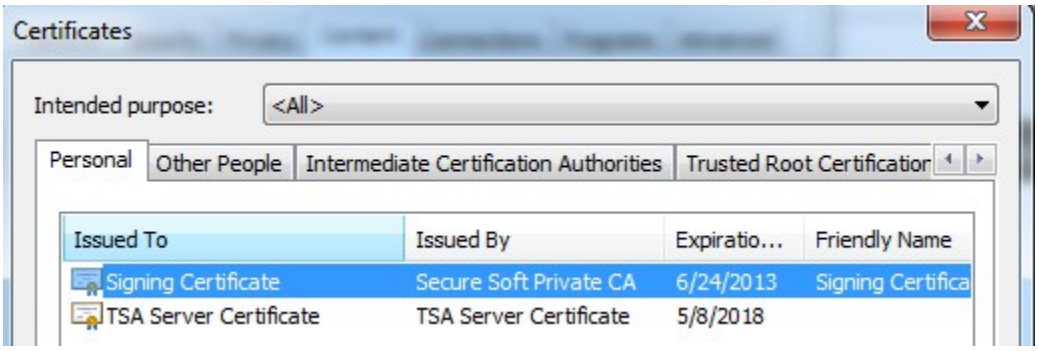
```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Common\Signatures]
"XAdESLevel"=dword:00000002
"MinXAdESLevel"=dword:00000001
"TSALocation"="http://ca.signfiles.com/tsa/get.aspx"
```

Office 2013 Registry Settings

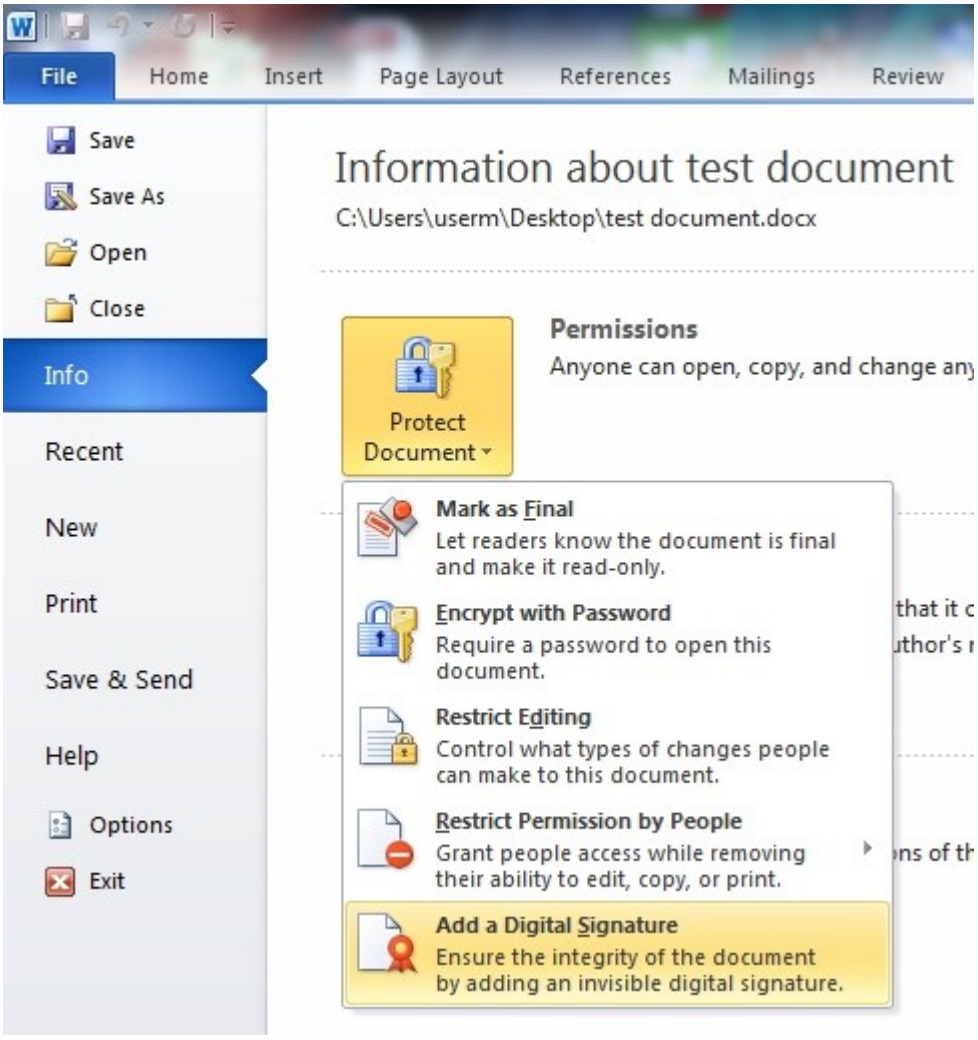
```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Common\Signatures]
"XAdESLevel"=dword:00000002
"MinXAdESLevel"=dword:00000001
"TSALocation"="http://ca.signfiles.com/tsa/get.aspx"
```

Adding a Digital Signature (XAdES-T) in a Office Document

In order to digitally sign a document, be sure you have a valid digital certificate installed on your system.



When the document is ready to be signed, go to *File – Info – Protect Document – Add a Digital Signature*, as below:



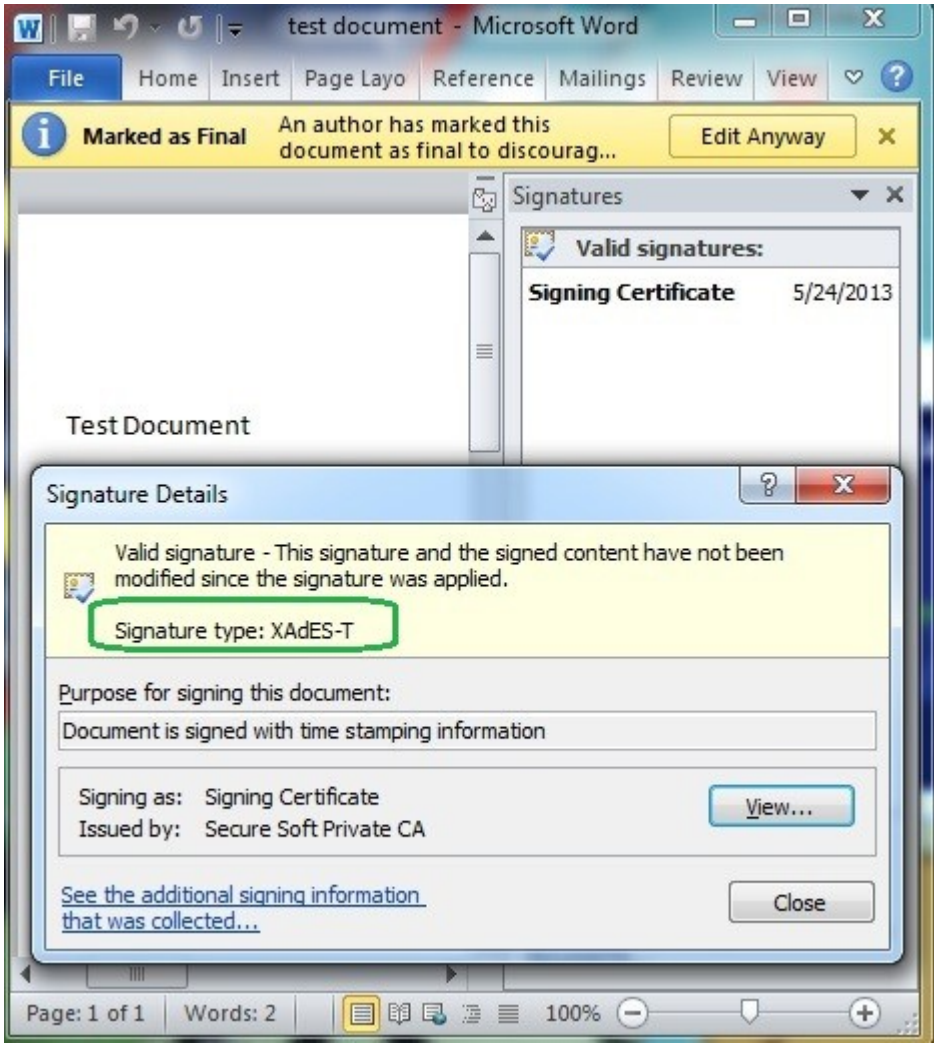
Add a Digital Signature in Office

On the next step, select the signing certificate that will be used to digitally sign the document and, optionally, enter the purpose for signing.

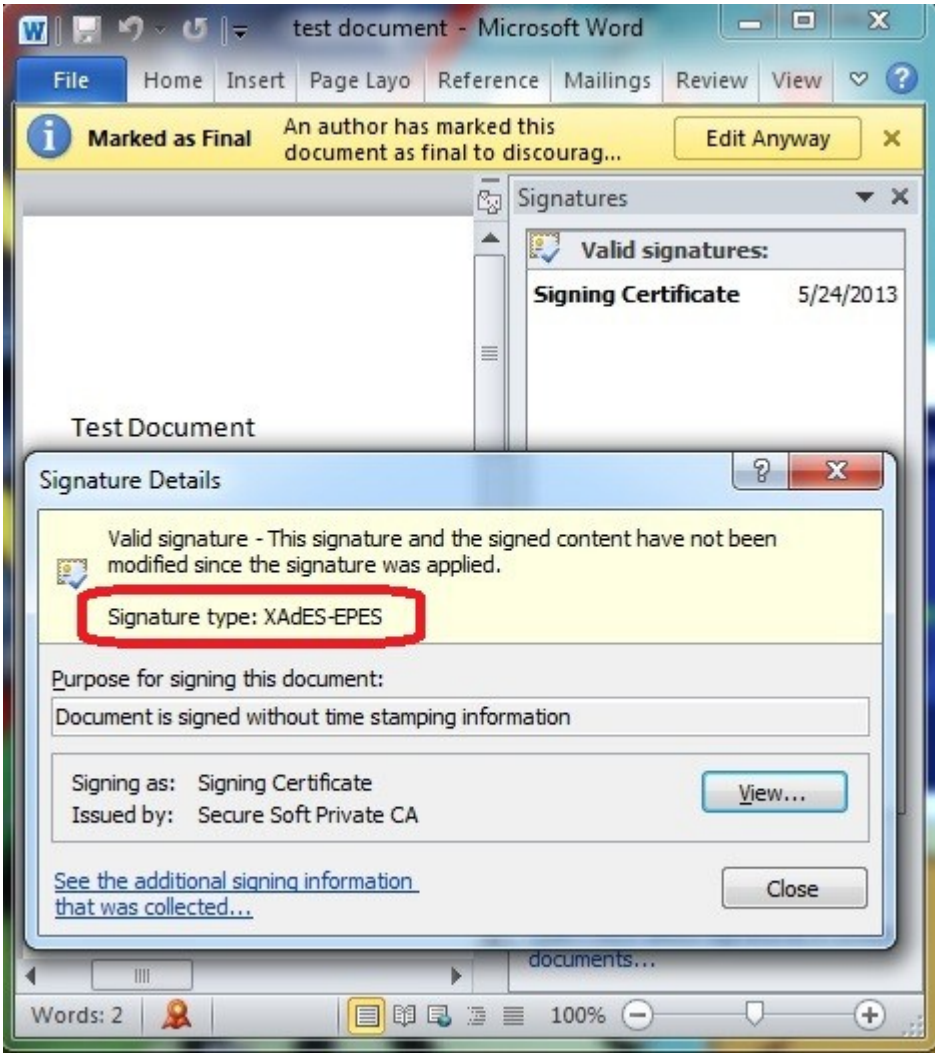
After the digital signature is created, a proper message will appear.

XAdES-T Signature Verification

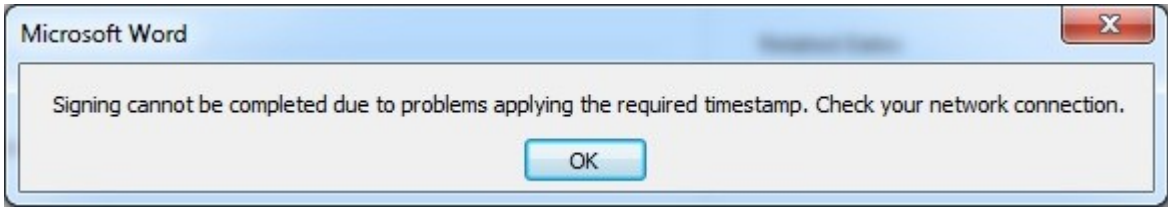
If the XadES-T signature is created successfully, it will look like this:



If the time stamp server is not available, the signature mode will be *XAdES-EPES* and will look at below:

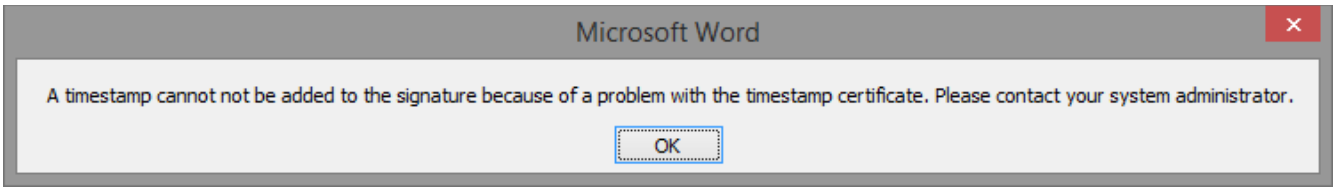


If the time stamp server is not available and the *MinXAdESLevel* registry entry is set to 2 (XadES-T), an error message will appear:



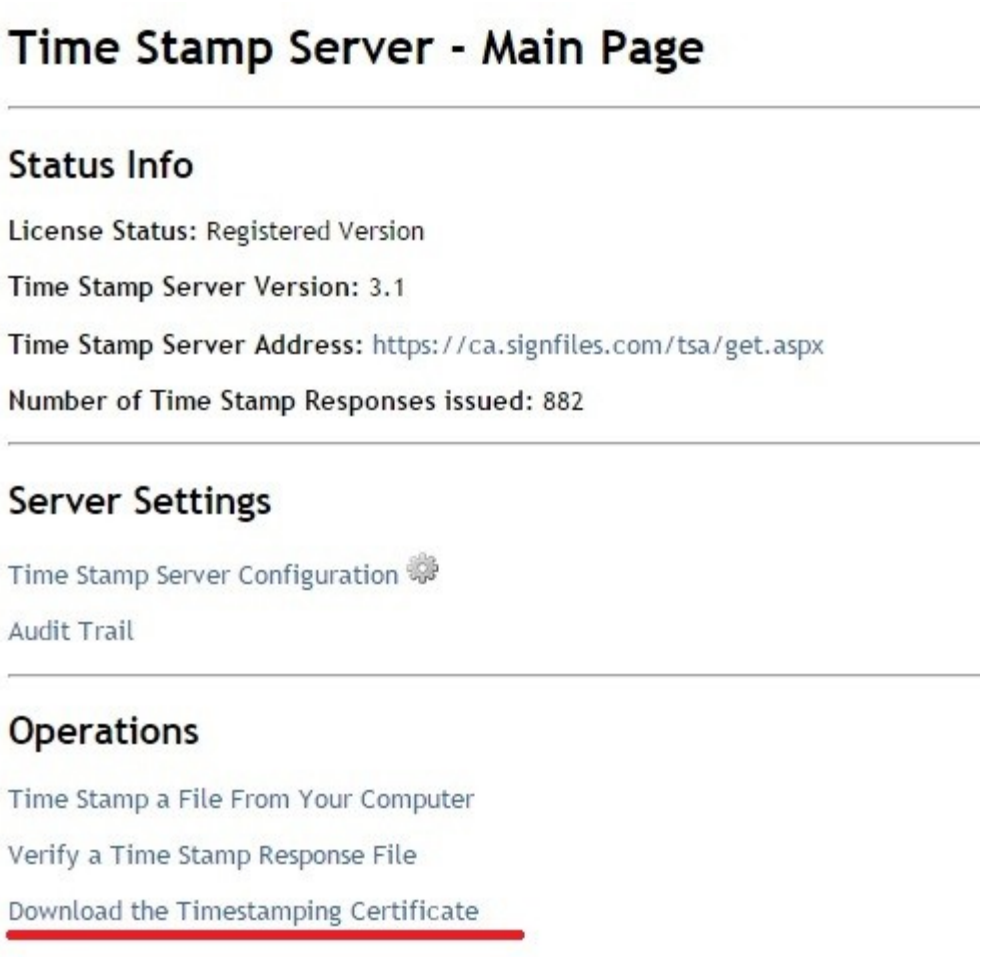
Trusting the Timestamp Certificate

Microsoft Office requires that the Timestamping certificate to be trusterd in order to be used for timestamping. If the certificate used by the Time Stamp Server is not trusted, when the signature is applied, a warning message will appear, as below:

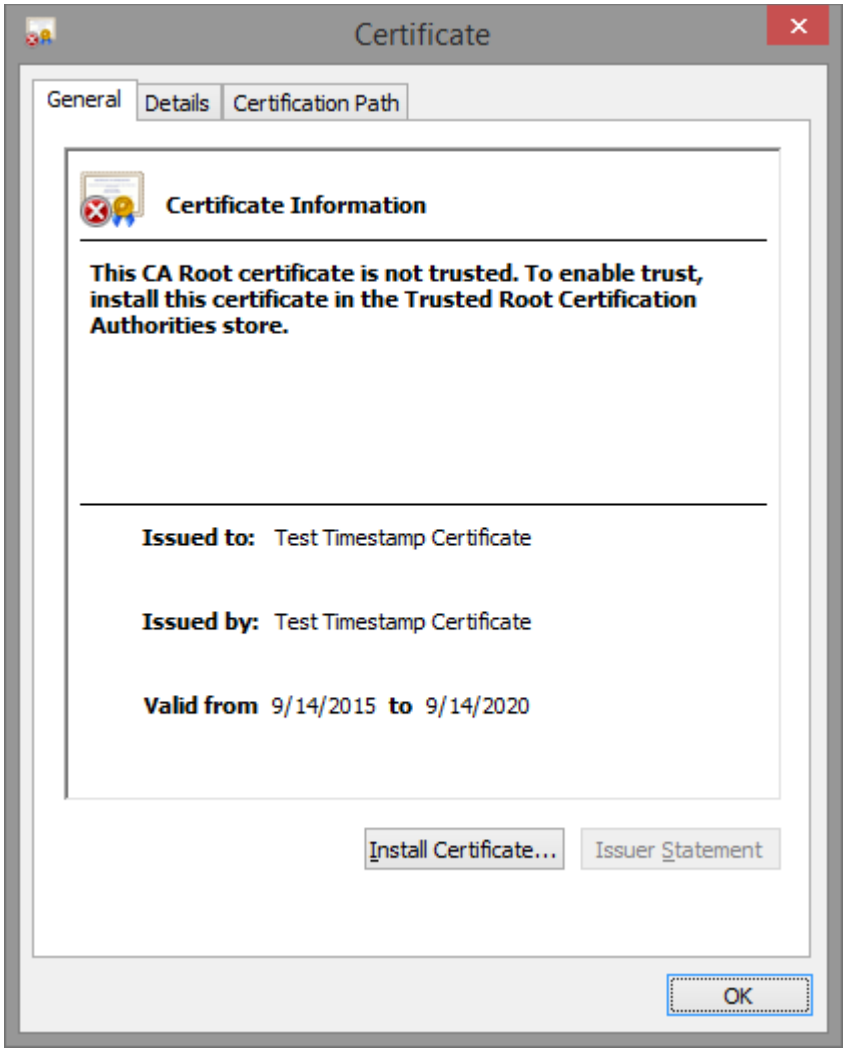


In order to trust the Timestamping certificate, follow the steps below:

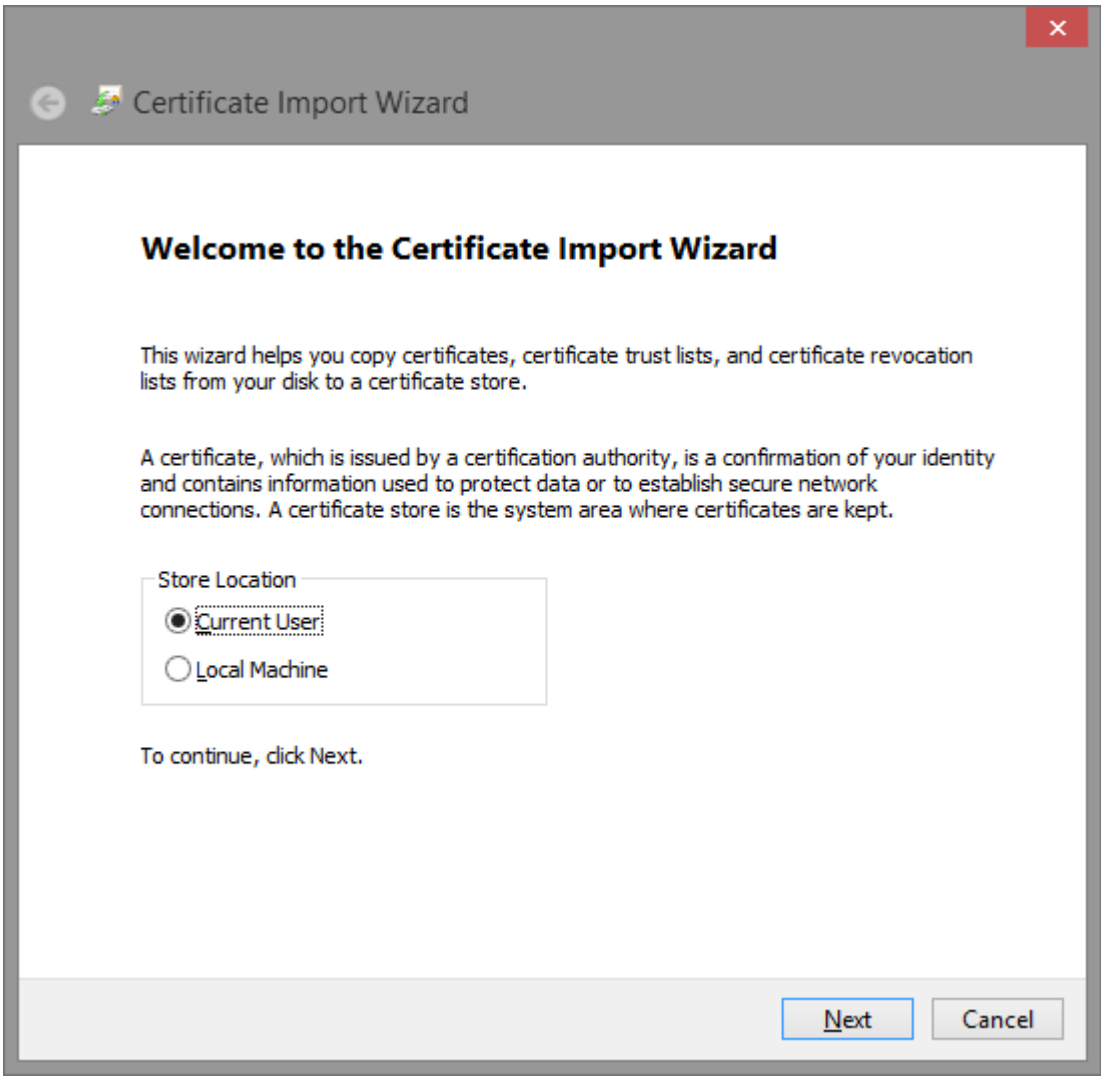
1. Go to the Time Stmap Server main page and download the Timestamping Certificate:



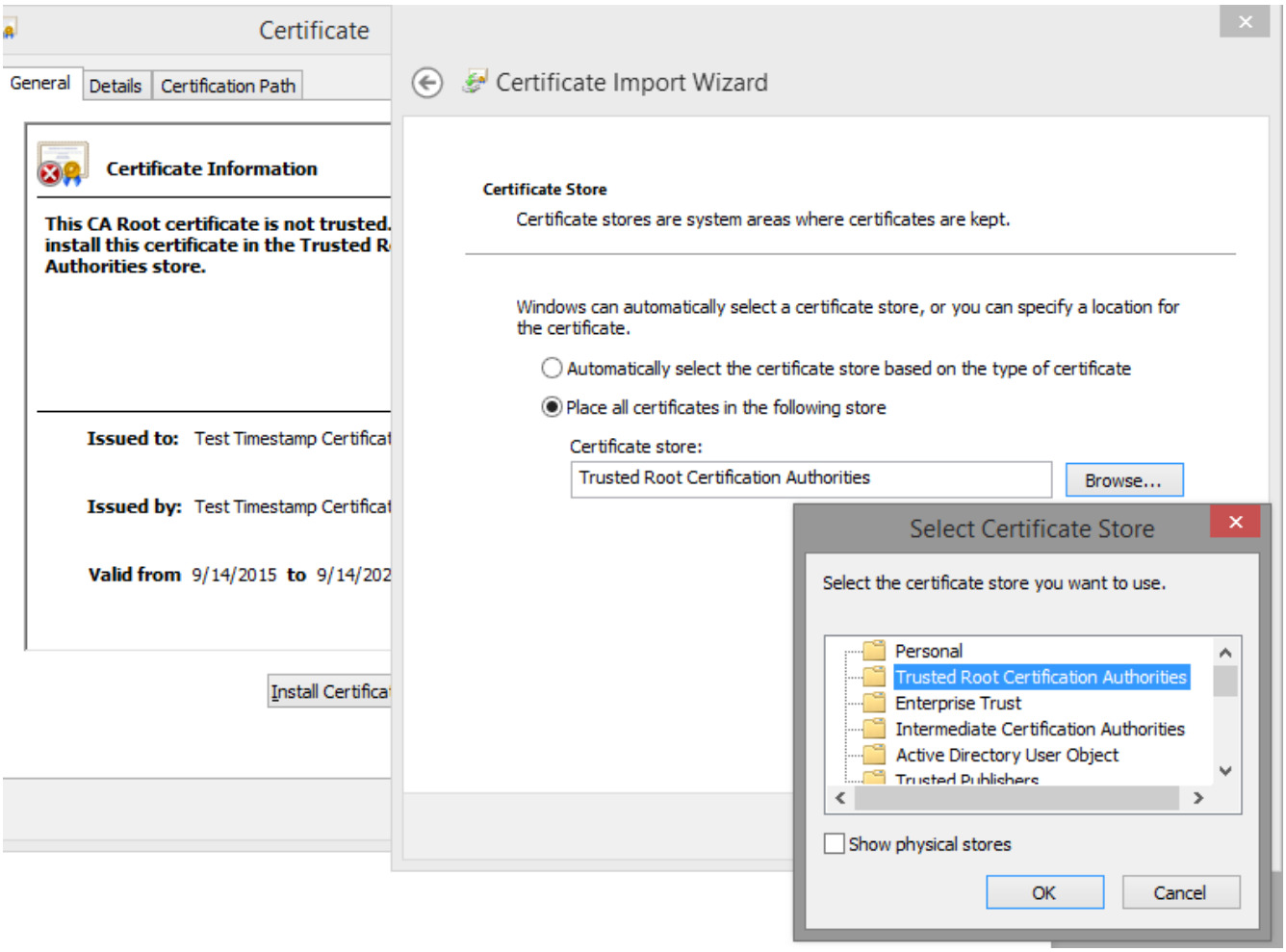
2. Open the certificate and click *Install Certificate*, as below:



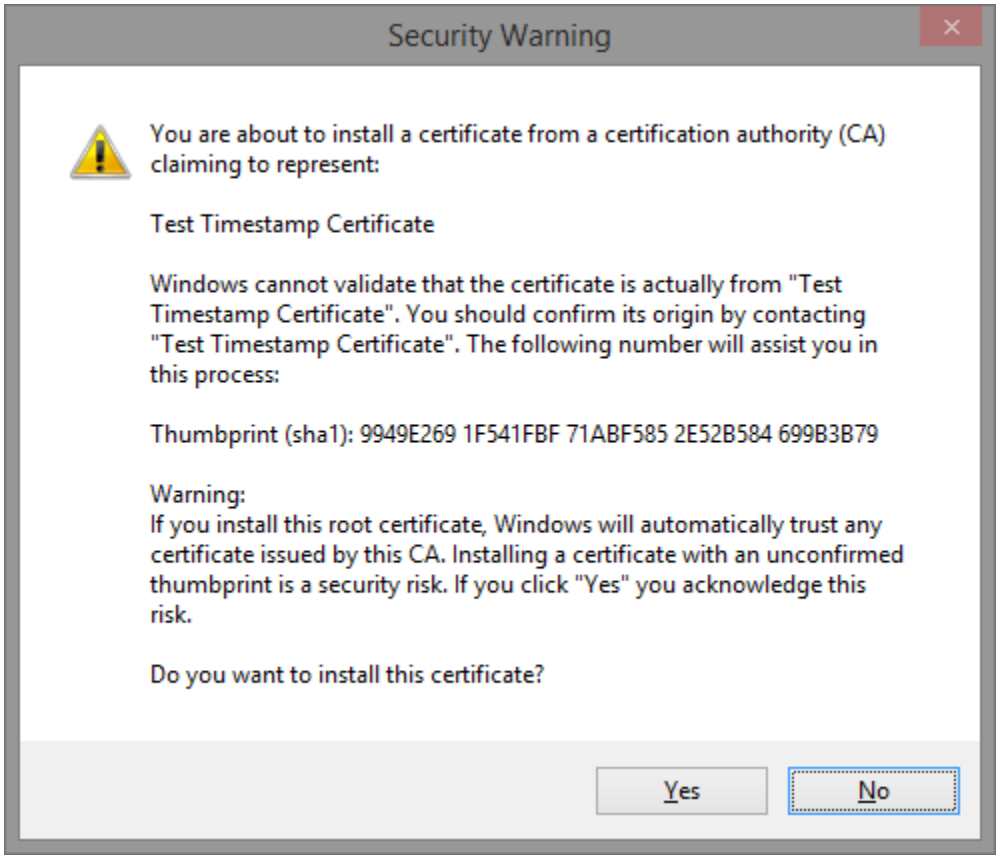
3. Select *Current User* store location and click *Next*:



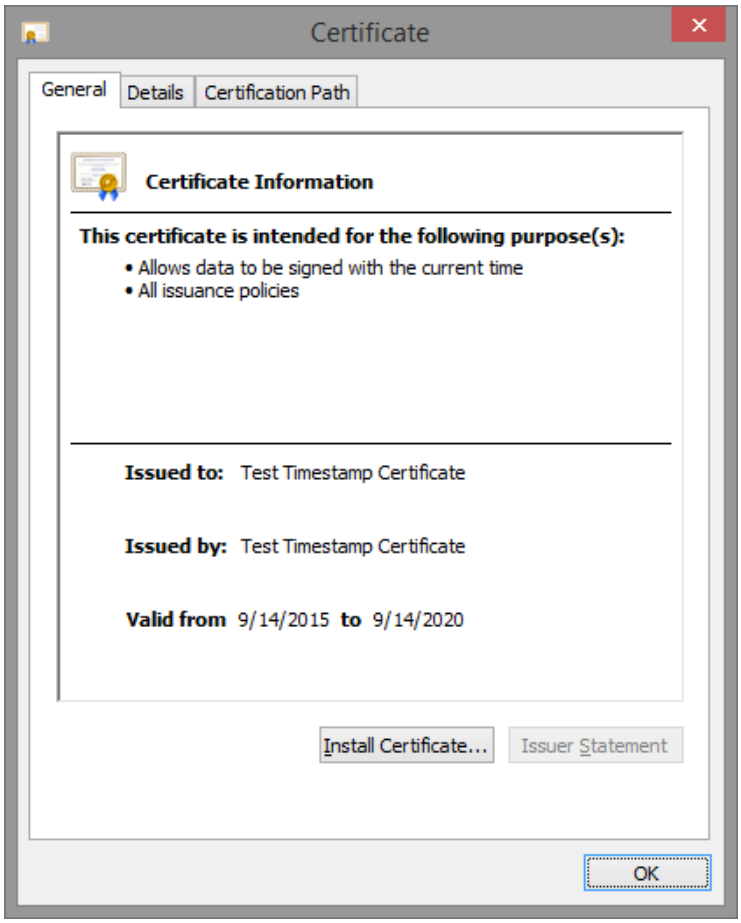
4. Select *Trusted Root Certification Authorities* store and click OK:



5. After the operation is confirmed, a dialog box will appear. After the buttom *Yes* is clicked, the Timestamping certificate is considered trusted.



6. After the last step, the certificate is now trusted and it can be used for timestamp Microsoft documents.

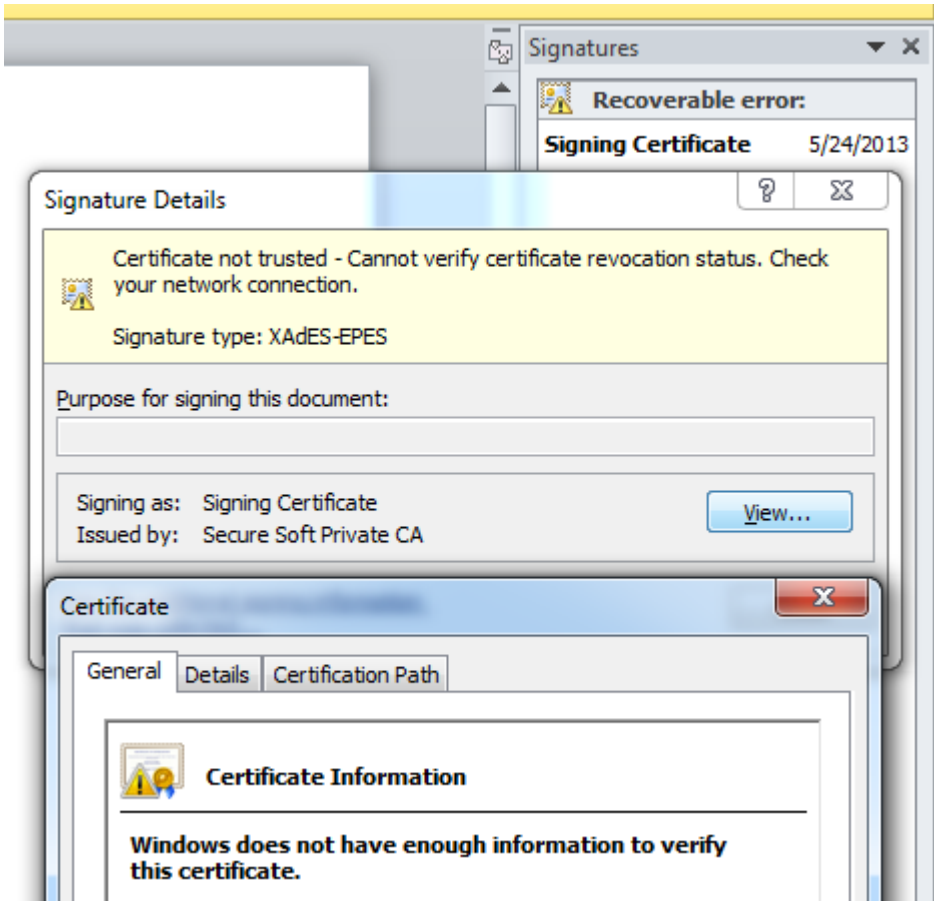


A trusted digital certificate

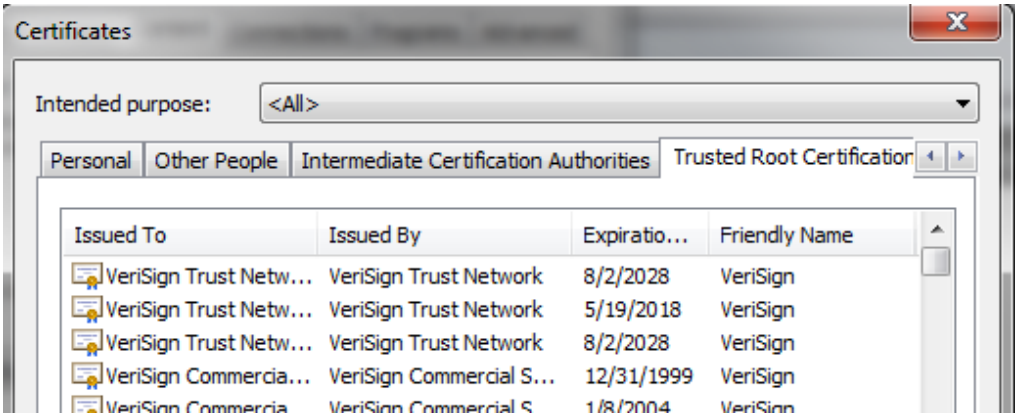
The digital signature can be perfomed

Microsoft Certificate Store

When the Root of the signing certificate is not trusted, it must be imported on *Microsoft Store – Trusted Root Certification Authorities*.



All digital certificates installed on the system appears in Microsoft Certificate Store.

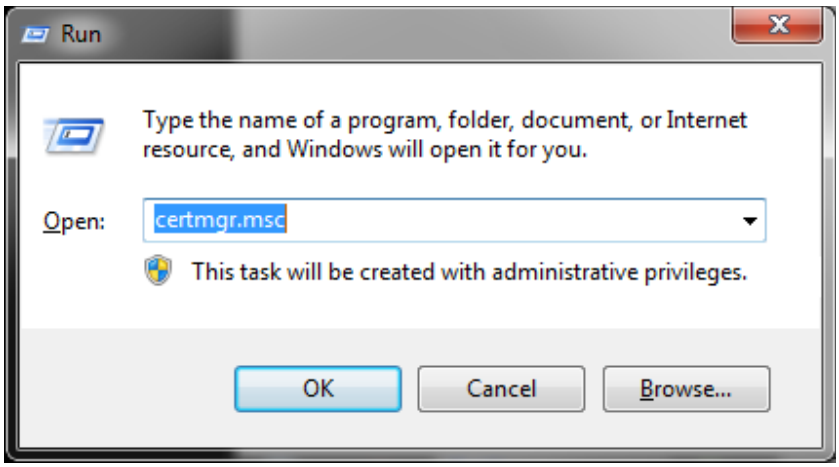


Microsoft Certificate Store

How to Access Microsoft Certificate Store

- start Internet Explorer
- go to *Tools* menu – *Internet Options* – *Content* tab – *Certificates* button
- on *Certificates* window your personal certificates appears in *Personal* tab.
- The Root certificates appears in *Trusted Root Certification Authorities* tab.

Also, the Microsoft Store can be accessed by running *certmgr.msc* on Run Command.



Accessing Microsoft Certificate Store

Export the Root Certificate from Microsoft Store

- Go to Microsoft Store
- Select *Trusted Root Certification Authorities* tab
- Select the Root Certificate that you want to export
- Click *Export* button and *Next*
- Select the path and filename for your exported certificate
- Click *Finish*.

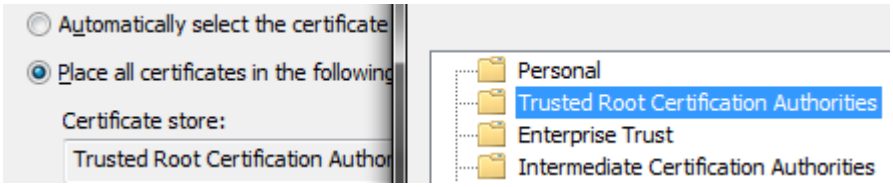
The Root Certificate is exported as *.cer* file. This file can be imported on the computers where you want to validate your certificate.

Note that if you digitally sign a file or send a digitally sign an email message to a computer that not have the Root Certificate installed, an warning message can appear.

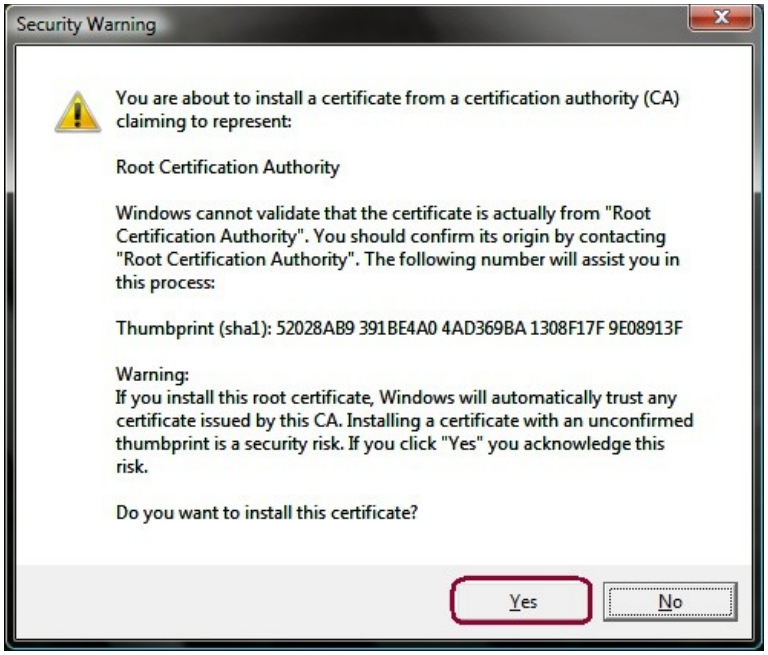
Import the Root Certificate on Microsoft Store

- Copy the exported .cer file obtained above (*Export the Root Certificate from Microsoft Store*) on the target computer
- Right click on the imported .cer file and select *Install Certificate*

- Click *Next* and select *Place all certificates in the following store*
- Click *Browse* and select *Trusted Root Certification Authorities*



– Click *Finish*
– press Yes when the message below appears.



After the Root Certificate is imported in Microsoft Store, the certificates issued by that Root Certification Authority will be considered valid on the machine where the Root Certificate was imported.