Miscellaneous

Table of Contents

- How to configure ADSS Server to produce PAdES-LTV signatures that are verifiable in Adobe Reader?
- What are the ADSS Server limitations when PSS padding scheme is used?
- What are the ADSS Server limitations when ECDSA key algorithm is is used?
- How to trust a Root CA certificate in the Adobe Reader?

How to configure ADSS Server to produce PAdES-LTV signatures that are verifiable in Adobe Reader?

ADSS Signing Server, the Verification service and the Go>Sign service all support PAdES LTV signatures - however the appropriate configurations must be made in these services and the utility modules.

In order to ensure digital certificates are trusted by ADSS Server, the PDF signer certificate Issuer CA must be registered within Trust Manager. The validation policy of the Issuer CA must be defined, i.e. whether it is using CRLs or OCSP?

Using CRLs:

The CRL Monitor module checks the certificate revocation information for registered CAs when the validation policy is set to "Local CRL Cache". It is not used when the policy is set to dynamically fetch a CRL from the CDP location or if OCSP only is configured. When using a local CRL cache ensure that

- The CRL retrieval policy is configured correctly for the CAs within the ADSS Trust Manager > Configure CRL Settings.
- Make sure that the latest CRLs for all the registered CAs are available within CRL Monitor > CRL Details screen with the status CURRENT.

> If AKI & SKI extensions are being used in the target certificate hierarchy then ensure that the SKI of the target CA must match with the AKI extension in the relevant CRL.

Using OCSP:

The OCSP responses must be signed using a certificate issued by the same CA that certified the signer certificate. The OCSP response signing certificate must have an Extended Key Usage of “OCSP Signing”.

- To force a check that the OCSP responder is authorized go to Trust Manager > Validation Policy screen and enable the check box **Check OCSP responder is authorized by the CA** under OCSP settings section.
- If the OCSP responder certificate does not contain the NoCheck extension then you must configure OCSP responder certificate status checking. To do this go to Trust Manager > Validation Policy screen and enable the check box **Enable certificate status checking for responder's certificate** under **OCSP Settings** section. It is usual to see OCSP responder certificates with a NoCheck extension so that the OCSP validation authority is trusted.

What are the ADSS Server limitations when PSS padding scheme is used?

1. When signing with the SHA512 hash algorithm, the minimum key length must be 2048 bits, otherwise it will throw the encoding exception: **Encoding error: emLen (128) shorter than hashLen + saltLen + 2!**
2. Due to the limitations in some of the third party libraries used by ADSS Server, PSS padding scheme could not be supported as yet, i.e.:
    a. When using some of the hardware tokens (Safenet, Utimaco)
    b. When doing client-side signing by using Go>Sign Desktop, as MS-CAPI doesn’t support PSS padding scheme.
3. ADSS Server supports Microsoft Office Word/ Excel native signing and verification. However Microsoft Office doesn’t support PSS padding scheme. Therefore for Microsoft Office native signing, ADSS Server will fallback to the PKCS1.5 padding scheme for server-side signing even if PSS padding scheme is configured in **Global Settings > Advanced Settings** for signing service.
4. While creating a signing profile for PKCS1 signatures, **Compute hash at signing time** option must be enabled in the **Advanced Settings** under Hash Signing Settings.

What are the ADSS Server limitations when ECDSA key algorithm is is used?

1. ADSS Server supports Microsoft Office Word/ Excel native signing and verification. However Microsoft Office doesn’t support ECDSA.
2. ECDSA is not supported for PKCS7 signatures as per RFC limitations.
3. ECDSA is not supported for client side hahsing:

a. While using .NET Test Tool
b. While generating XAdES signatures

4. ECDSA is not supported with PSS padding scheme.

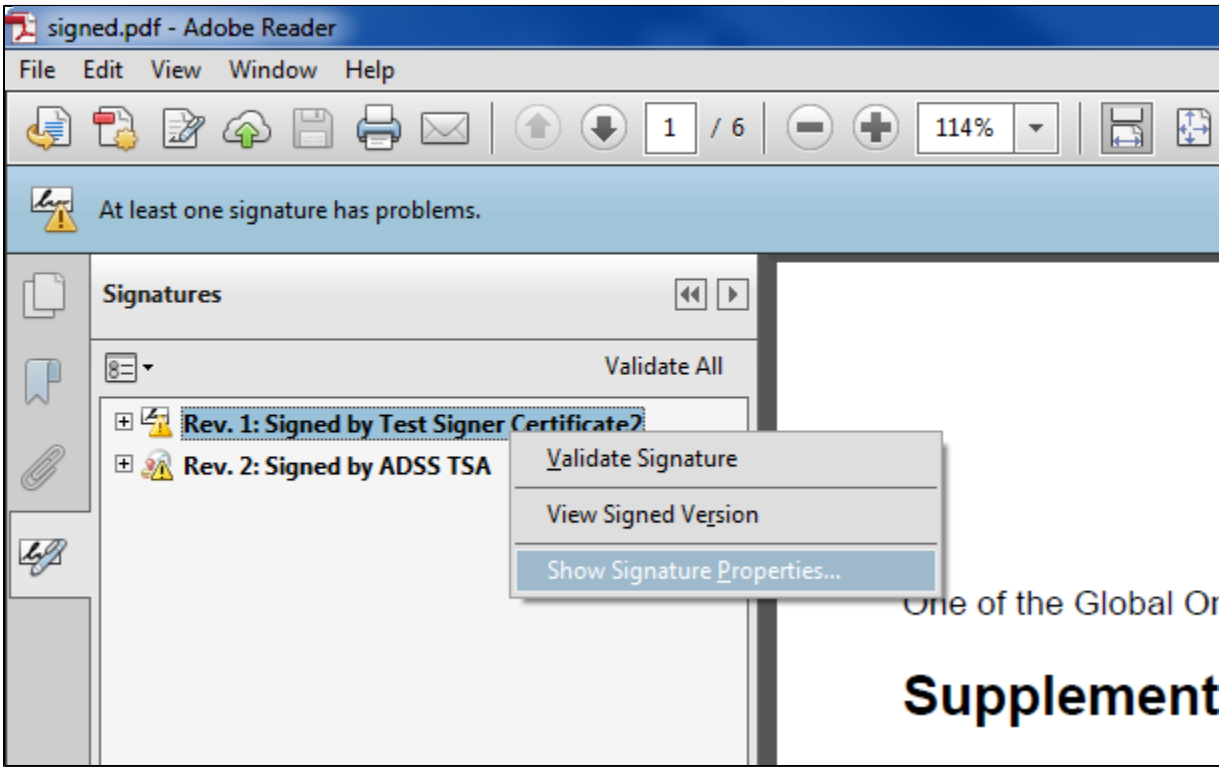
How to trust a Root CA certificate in the Adobe Reader?

If the signatures produced by the ADSS Server are reported as not trusted by Adobe Reader, the reason is that the issuer CAs are not set as trusted in the Adobe Reader trust store.
If the issuer of your PDF signing certificate is not already trusted by Adobe Reader then you will face this issue.

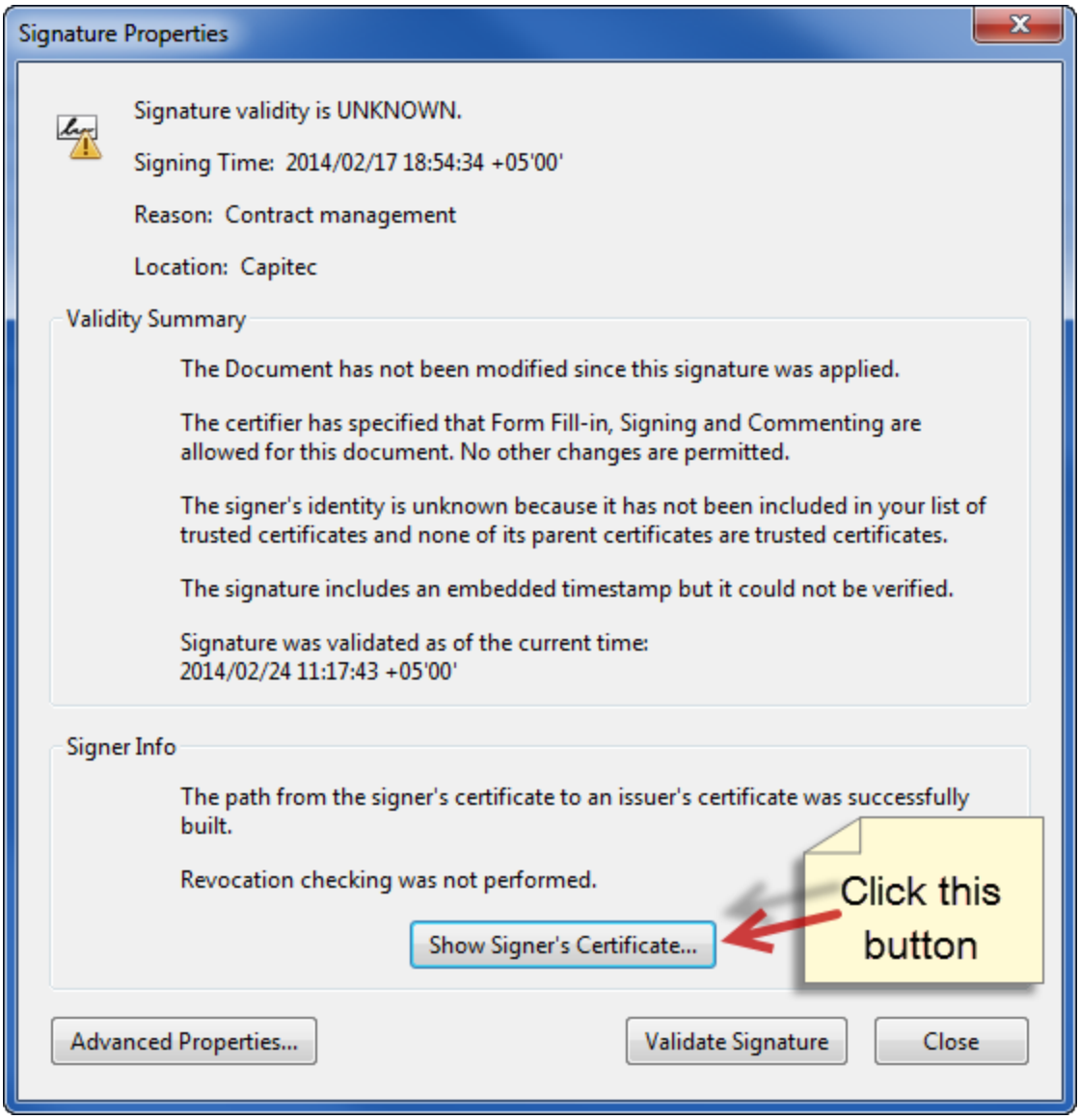
There are two possible solutions to resolve this issue:

1. Use a certificate from a CA which is pre-embedded in the Adobe Reader so that the signatures are immediately trusted, without any manual update of the Adobe Reader trust store. If you are interested in this option then we do have partnerships with Adobe CDS/AATL CAs and can guide you further on this. Email your inquiries to sales@ascertia.com.

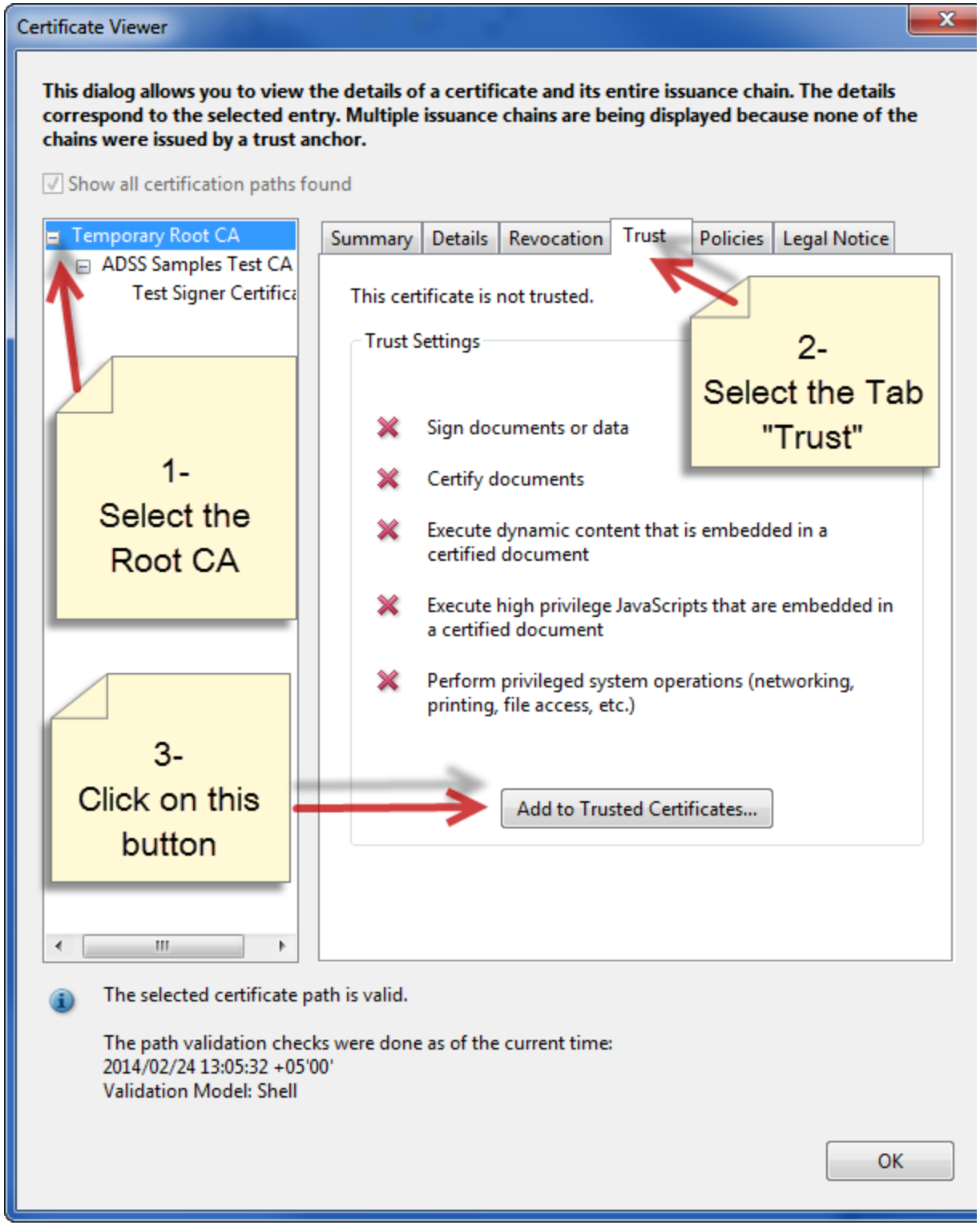
2. Import your Root CA certificate into Adobe Reader trust store. Please note that it's a one-time action which all users who verify signatures will need to perform in their Adobe Reader instances. Follow these steps to import the Root CA in Adobe Reader trust store:
   a. Open the signed PDF file in Adobe Reader
   b. Right click on the signature and select the option "Show Signature properties"



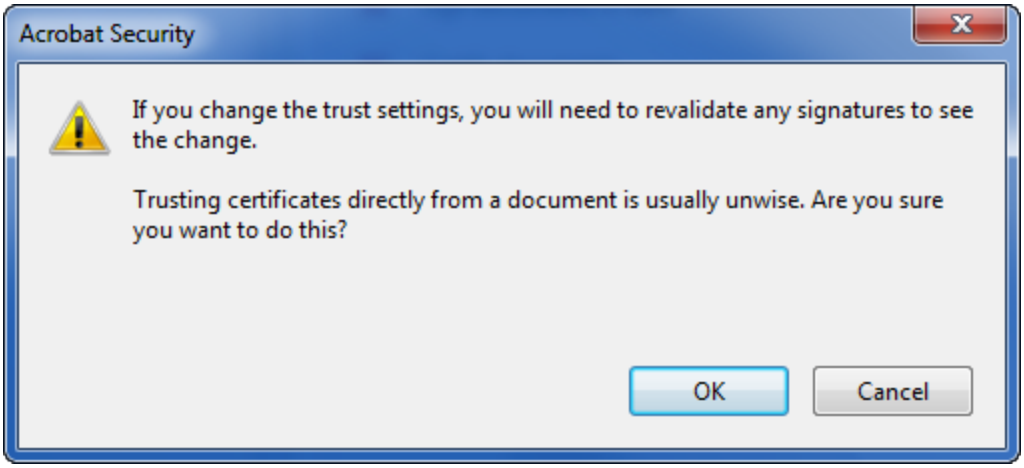
c. The following signature properties dialogue will be shown:



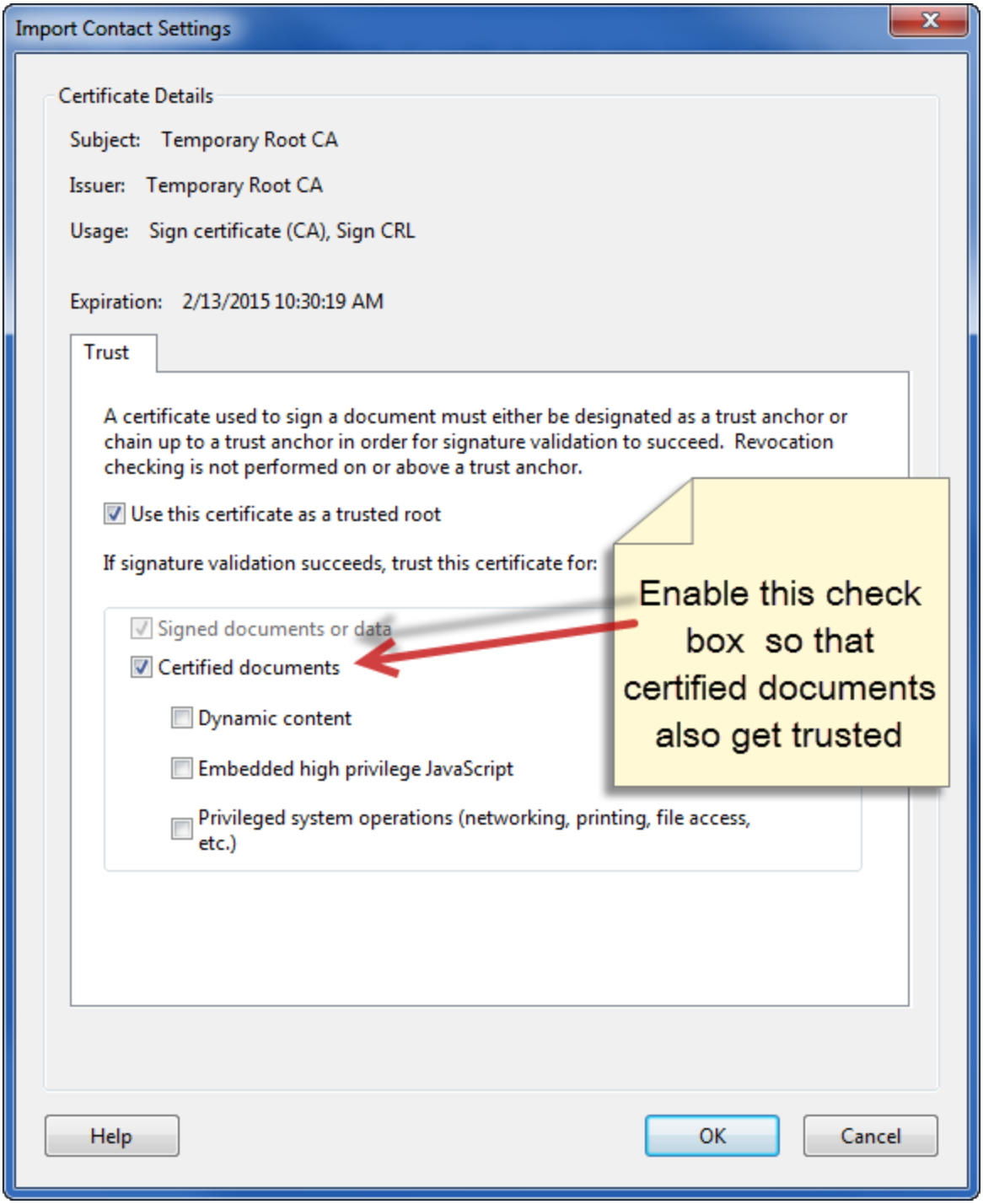
d. Clicking on the “Show Signer’s Certificate” button will show the following popup window:



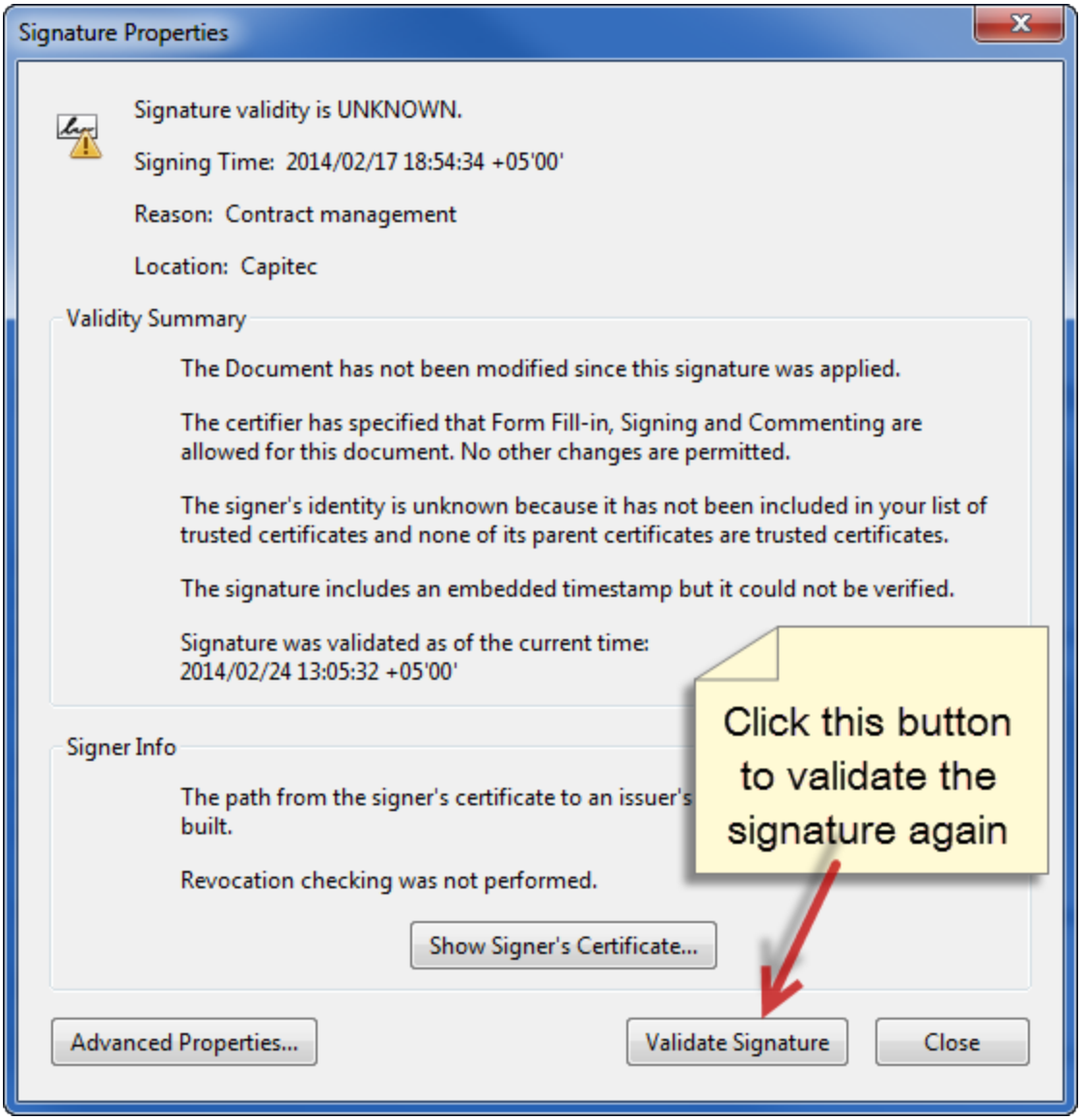
e. Clicking on the “Add to Trusted Certificates” button will show the following warning message:



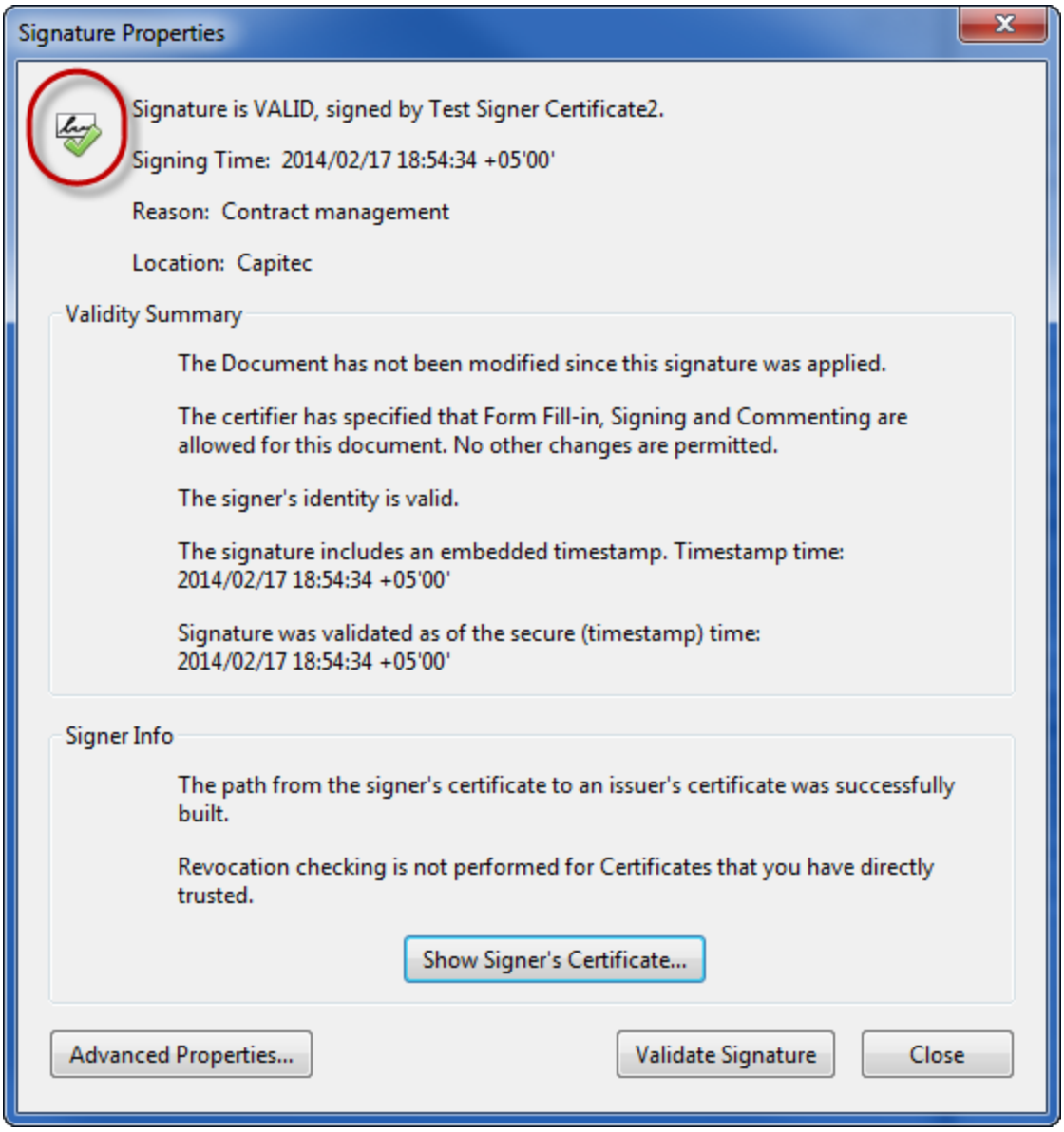
f. Clicking on the "OK" button will show the following window:



g. Enable the check box "Certified documents" and click on the "OK" button
h. Again click on the "OK" button to close the popup window
i. Click on the "Validate Signature" button



j. Now the signatures produced by Root CA issued certificates will be shown as valid and trusted by Adobe Reader


Signature Properties
Signature is VALID, signed by Test Signer Certificate2.
Signing Time: 2014/02/17 18:54:34 +05'00'
Reason: Contract management
Location: Capitec
Validity Summary
The Document has not been modified since this signature was applied.
The certifier has specified that Form Fill-in, Signing and Commenting are allowed for this document. No other changes are permitted.
The signer's identity is valid.
The signature includes an embedded timestamp. Timestamp time: 2014/02/17 18:54:34 +05'00'
Signature was validated as of the secure (timestamp) time: 2014/02/17 18:54:34 +05'00'
Signer Info
The path from the signer's certificate to an issuer's certificate was successfully built.
Revocation checking is not performed for Certificates that you have directly trusted.
Show Signer's Certificate...
Advanced Properties...
Validate Signature
Close