

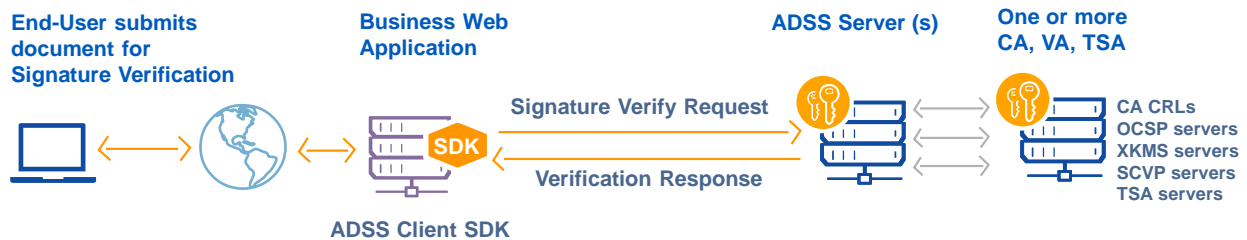
ADSS Server – Exploring Verification Services

ADSS Server is a multi-function server providing digital signature creation and verification web services, as well as supporting infrastructure services such as Time Stamp Authority (TSA) services, OCSP validation services and certification services.

This solution sheet discusses how ADSS Server can be used to verify digital signatures or verify certificates. It also explains how verification profiles are created and how these are used as a means of controlling the way verification services are offered. This approach ensures that only specific trust profiles and associated parameters, such as digital signature keys, are available to authorised business applications. The trust anchors (CA's) and their associated TSAs and OCSP Servers can also be policy controlled.

Verifying Digital Signatures and Validating Certificates

The ADSS Server Verification Service offers a very effective high-trust solution for verifying digitally signed documents and data. It can also validate end-user certificates and the entire certificate chain, including checking the certificate has not expired; is not revoked, is signed by a trusted CA, has the correct key usage, policy OIDs, etc.).



In the example above an end-user has submitted a signed document such as a signed purchase order to a business web-application. Before processing this, the application needs to check the signature on the document is trustworthy. It forwards the document to the ADSS Server for its verification analysis and verdict.

The ADSS Verification Service performs a sequence of verification checks to ensure that the signature has been generated correctly, that the signer's certificate was issued by a recognised and trusted CA, and has not expired, that the key usage and policy OIDs and any trust criteria are met. As part of this signature verification process, ADSS Server also requests certificate status information from the relevant certificate status provider, either in the form of real-time OCSP calls or by checking the relevant CA's CRL. ADSS Server constantly monitors CRLs so that it always has access to the current latest status information and can immediately process this.

The ADSS Verification Service then provides a signature verification response showing the overall status of each signature in the document, i.e. "trusted", "not trusted" or "indeterminate". The Verification Service can also be asked to return a range of details from the signer's certificate and other items if these are specified in the verification request from the client application.

For sensitive documents it is possible for the client application to either extract the signature from the document and send just this and the document hash for signature verification, or perform signature verification itself and to delegate only certificate chain validation to the ADSS Verification Service. The application is then only responsible for hash calculation and matching however ADSS Client SDK makes this process very easy.

Signature Verification Interface

Business Applications access ADSS Verification Service using OASIS DSS and DSS-X web services with XML/SOAP protocols for this communication interface. Ascertia also provide a much faster HTTP/S protocol within its DotNet and Java ADSS Client SDKs. Both protocols are supported within the ADSS Client SDK and set at the flick of a switch.

Verification Profiles

Verification Profiles make it easier for business applications to make verification requests. The application does not need to understand all the options – an appropriately privileged ADSS Server operator can configure a suitable verification profile. The business application now simply references this profile in the request message, thus removing all the overhead and complexity of verification request configuration and checking. Multiple verification profiles can be configured to suit different application needs, e.g. checking against a list of EU Qualified Certificates, checking against AATL certificates, or other global high trust PKIs. The main attributes that can be configured in a verification profile are:

- Verification Profile Identifier
- Trust Anchors
- Signature Types
- Acceptable Certificate Policies and Key Usages
- Signature and Certificate Quality Levels
- Extending a basic signature to a long term format (commanded in detail by API)

Verification Profile Identifier:

Verification profiles are uniquely identified within ADSS Server via a system defined Profile ID:

| Verification Service > Verification Profiles | | | | | |
|---|---|---|---|------------|---|
| Showing page 1 of 1 | | | | | |
| <input type="button" value=" <"/> <input type="button" value="<"/> <input type="button" value=">"/> <input type="button" value=" >"/> | | Order by: | Created At | Descending | <input type="button" value="Clear Search"/> <input type="button" value="Search"/> |
| <input type="radio"/> | adss:verification:profile:004 | Upgrade Default Verification Profile - 04 | SMIME, CMS, CAdES-BES, CAdES-T, CAdES-C, CAdES-X, CAdES-X-L, CAdES-A, XMLDSig, XAdES-BES, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L, XAdES-A, PDF signature (basic), PDF signature with embedded timestamp, PDF signature with embedded timestamp and revocation info, PKCS#7 | Active | |
| <input type="radio"/> | adss:verification:profile:003 | Upgrade Default Verification Profile - 03 | PDF signature (basic), PDF signature with embedded timestamp, PDF signature with embedded timestamp and revocation info, SMIME, XMLDSig, XAdES-BES, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L, XAdES-A, CMS, CAdES-BES, CAdES-T, CAdES-C, CAdES-X, CAdES-X-L, CAdES-A, PKCS#7 | Active | |
| <input type="radio"/> | adss:verification:profile:002 | Upgrade Default Verification Profile - 02 | XMLDSig, XAdES-BES, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L, XAdES-A, CMS, CAdES-BES, CAdES-T, CAdES-C, CAdES-X, CAdES-X-L, CAdES-A, SMIME, PKCS#7, PDF signature (basic), PDF signature with embedded timestamp, PDF signature with embedded timestamp and revocation info | Active | |
| <input type="radio"/> | adss:verification:profile:001 | PDF Verification | PDF signature (basic), PDF signature with embedded timestamp, PDF signature with embedded timestamp and revocation info, CMS, CAdES-BES, CAdES-T, CAdES-C, CAdES-X, CAdES-X-L, CAdES-A, SMIME, PAdES-BES (basic signature - CAdES-BES incorporated), PAdES-BES-T (timestamped signature - CAdES-T incorporated), PAdES Part 4 long term signature with timestamped and validation information only, XMLDSig, XAdES-BES, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L, XAdES-A, PKCS#7 | Active | <input type="button" value="New"/> <input type="button" value="Edit"/> <input type="button" value="Make a Copy"/> <input type="button" value="Delete"/> |

Specifying Trust Anchors:

The administrator can configure which trust anchors should be considered final trust points for each verification profile. This allows each business application to rely on different trust anchors by using different verification profiles held within the ADSS Verification Service:

Verification Service > Verification Profiles > PDF Verification (adss:verification:profile:001)

General | **Trust Anchors Settings** | Signature Settings | Algorithms Settings | Path Discovery Settings | Path Validation Settings | Advanced Settings

Select Trust Anchors

Available Trust Anchors:

>> <<

Selected Trust Anchors:

- Belgium Root CA2
- Belgium Root CA3
- Belgium Root CA4
- Betrusted Production SSP CA A1
- BT Class 2 CA - G2
- Buypass Class 2 CA 1
- Buypass Class 3 CA 1

Automatically trust any new Trust Authorities added to ADSS Trust Manager

[View Certificate](#)

[Next](#) [Save](#) [Cancel](#)

Signature Types:

Each verification profile defines the signature types that can be verified by the profile:

Verification Service > Verification Profiles > PDF Verification (adss:verification:profile:001)

General | Trust Anchors Settings | **Signature Settings** | Algorithms Settings | Path Discovery Settings | Path Validation Settings | Advanced Settings

Supported Signature Types

- PDF (and PAdES profiles)
 - Standard PDF Signature
 - PDF Signature with embedded timestamp
 - PDF Signature with embedded timestamp and revocation information
 - PAdES-BES (Basic PAdES Part 3 Signature)
 - PAdES-T (Basic PAdES Part 3 Signature with embedded timestamp)
 - PAdES-LTV (PAdES Part 4, PAdES-LTV with document timestamp)
 - Verify Explicit Policy Electronic Signature (EPES) attribute
- CMS (and CAdES profiles)
 - CMS (RFC 3852)
 - CAdES-BES (Basic Electronic Signature)
 - CAdES-T (Signature Timestamp)
 - CAdES-C (Certificate and Revocation References)
 - CAdES-X (Signature and References Timestamp, References Only Timestamp)
 - CAdES-X-L (Certificate and Revocation Status Info)
 - CAdES-A (Archive Timestamp)
 - Verify Explicit Policy Electronic Signature (EPES) attribute
- XML Dsig (and XAdES profiles)
 - XML Dsig (W3C)
 - XAdES-BES (Basic Electronic Signature)
 - XAdES-T (Signature Timestamp)
 - XAdES-C (Certificate and Revocation References)
 - XAdES-X (Signature and References Timestamp, References Only Timestamp)
 - XAdES-X-L (Certificate and Revocation Status Info)
 - XAdES-A (Archive Timestamp)
 - Verify Explicit Policy Electronic Signature (EPES) attribute
- MS Office
- PKCS#7
- S/MIME

[Next](#) [Save](#) [Cancel](#)

ADSS Server - Exploring Verification Services

Configuring Acceptable Hash and Signature Algorithms:

The administrator can also configure the list of acceptable hash and signature algorithms permitted:

Verification Service > Verification Profiles > PDF Verification (adss:verification:profile:001)

General | [Trust Anchors Settings](#) | [Signature Settings](#) | **[Algorithms Settings](#)** | [Path Discovery Settings](#) | [Path Validation Settings](#) | [Advanced Settings](#)

Select Hash Algorithms

| | | |
|---|--|---|
| <p style="text-align: center;">Available Hash Algorithms</p> <div style="border: 1px solid gray; height: 80px; width: 100%;"></div> | <input type="button" value=">>"/> <input type="button" value="<<"/> | <p style="text-align: center;">Selected Hash Algorithms</p> <div style="border: 1px solid gray; padding: 5px;"> SHA1 SHA224 SHA256 SHA384 SHA512 RipeMD128 RipeMD160 RipeMD256 MD5 </div> |
|---|--|---|

Note: XML type signature does not support SHA224,RipeMD128,RipeMD256 and MD5 hash algorithm

Select Key Algorithms with size

| | | |
|--|--|--|
| <p style="text-align: center;">Available Key Algorithms with Key Length</p> <div style="border: 1px solid gray; height: 80px; width: 100%;"></div> | <input type="button" value=">>"/> <input type="button" value="<<"/> | <p style="text-align: center;">Selected Key Algorithms with Key Length</p> <div style="border: 1px solid gray; padding: 5px;"> RSA (1024 bits) RSA (2048 bits) RSA (3072 bits) RSA (4096 bits) ECDSA (192 bits) ECDSA (224 bits) ECDSA (256 bits) ECDSA (384 bits) ECDSA (521 bits) </div> |
|--|--|--|

Configuring Path Discovery Options:

A suitably privileged ADSS Server administrator can define the path discovery options:

Verification Service > Verification Profiles > PDF Verification (adss:verification:profile:001)

General | [Trust Anchors Settings](#) | [Signature Settings](#) | [Algorithms Settings](#) | **[Path Discovery Settings](#)** | [Path Validation Settings](#) | [Advanced Settings](#)

Path Discovery Settings

Use basic path discovery

- Build path using certificates registered in ADSS Trust Manager
- Build path using certificates provided in request

Use advanced path discovery

Final Trust Point

- Build and validate certificate path up to any CA registered in ADSS Trust Manager
- Build and validate certificate path up to a self-signed Root CA registered in ADSS Trust Manager

ADSS Server - Exploring Verification Services

Configuring Path Validation Options:

A suitably privileged ADSS Server administrator can use advanced options to also configure the list of acceptable certificate policies and any permitted or excluded certificate subject names required by the calling business applications:

Verification Service > Verification Profiles > Upgrade Default Verification Profile -10 (adss:verification:profile:010)

General | Trust Anchors Settings | Signature Settings | Algorithms Settings | Path Discovery Settings | **Path Validation Settings** | Advanced Settings

Path Validation Settings

Use basic path validation
 Use advanced path validation

PKIX Certificate Validation Settings

Inhibit Policy Mappings
 Require Explicit Policy
 Inhibit anyPolicy
 Acceptable certificate policy OIDs

Enter Acceptable Certificate Policy OID:
List of Accepted Certificate Policy OIDs:

Permitted Subject Names

List of Permitted Subject Names:

Excluded Subject Names

List of Excluded Subject Names:

Key Usages Extension Checks

Full List of Key Usages

digitalSignature
nonRepudiation
keyEncipherment
dataEncipherment
keyAgreement
keyCertSign
cRLSign
encipherOnly
decipherOnly

AND
 OR

Selected Key Usages

Note: Key Usages with OR operator are shown in multiple lines in the Selected Key Usages while the Key Usages with AND operator are shown comma separated in a single line

Extended Key Usages Extension Checks

Full List of Extended Key Usages

serverAuth
clientAuth
codeSigning
emailProtection
timeStamping
OCSPSigning
smartCardLogon
anyPurpose

Selected Extended Key Usages

Note: The "AND" operator is used between the selected Extended Key Usages. If none of the selected Extended Key Usage found in the EE certificate then a failed response is generated.

Key Usage and Extended Key Usage may also be enforced:

Verification Service > Verification Profiles > Upgrade Default Verification Profile -10 (adss:verification:profile:010)

[General](#) | [Trust Anchors Settings](#) | [Signature Settings](#) | [Algorithms Settings](#) | [Path Discovery Settings](#) | **[Path Validation Settings](#)** | [Advanced Settings](#)

Path Validation Settings

Use basic path validation
 Use advanced path validation

Key Usages Extension Checks

Full List of Key Usages

- digitalSignature
- nonRepudiation
- keyEncipherment
- dataEncipherment
- keyAgreement
- keyCertSign
- cRLSign
- encipherOnly
- decipherOnly

Selected Key Usages

AND
 OR

>>
<<

Note: Key Usages with OR operator are shown in multiple lines in the Selected Key Usages while the Key Usages with AND operator are shown comma separated in a single line

Extended Key Usages Extension Checks

Full List of Extended Key Usages

- serverAuth
- clientAuth
- codeSigning
- emailProtection
- timeStamping
- OCSPSigning
- smartCardLogon
- anyPurpose

Selected Extended Key Usages

>>
<<

Note:- The "AND" operator is used between the selected Extended Key Usages. If none of the selected Extended Key Usage found in the EE certificate then a failed response is generated.

Next Save Cancel

Advanced Options:

A range of advanced options are supported as can be seen here

Verification Service > Verification Profiles > Upgrade Default Verification Profile -10 (adsss:verification:profile:010)

[General](#) | [Trust Anchors Settings](#) | [Signature Settings](#) | [Algorithms Settings](#) | [Path Discovery Settings](#) | [Path Validation Settings](#) | [Advanced Settings](#)

Validation Policy for Non-registered CAs

OCSP (configured address)
XKMS (for full certificate validation)

OCSP (using AIA extension)
CRLs (using CDP extension)

OCSP Request Settings

Add Nonce extension
 Add Service Locator extension
 Sign request

Signing Certificate:

Check responder revocation
 Check OCSP responder is authorised by the CA

Hash Algorithm:
Clock Tolerance: (sec)
Response Timeout: (sec)
*Set to 0 if the timeout is unlimited

Accepted Signature & Certificate Quality Levels

Minimum Certificate Quality Level:
Minimum Independent Assurance Level:
Minimum Hash Quality Level:
Minimum Public key Quality Level:

Allow client application to override these parameters in its request message

Historical Validation Settings

Allow historical verification requests

Signature Grace Period Settings

Signature grace period: (min)

Specify Verification Time

If verification time is not specified in the call to ADSS Server then the preference is given to the timestamp token. If the signature is not protected by a timestamp token then following configurations are used to decide the verification time:

Return error if trusted timestamp token is not present in long term signature
 Verify signatures at the current time
 Verify signatures at the time of signing
 Use embedded validation data if present

Signature Enhancement Settings

Allow signature enhancement

Available TSAs:

Selected TSAs:

Note: Each selected TSA will be tried in order to obtain a timestamp. If a timestamp cannot be obtained from any of the selected TSAs then an error will be returned.

Signature and Certificate Quality Levels:

As seen above the PEPOL trust ratings scheme establishes the concept of certificate and signature quality levels to enable business applications to make decisions about the acceptability of particular signatures. Business applications can therefore accept or reject signatures that have insufficient quality, perhaps they are not using qualified certificates; or are using certificates from a CA that does not adequately identify the end-entity, or the algorithms used are now considered weak. This will become important as the world transitions from SHA-1 with RSA1024 to SHA-2 with RSA2048.

ADSS Server enables minimum signature and certificate quality levels to be set per Verification Profile and can be configured to allow the business application to override these quality settings if this is required. Further details can be provided on request.

Authenticating Client Business Applications

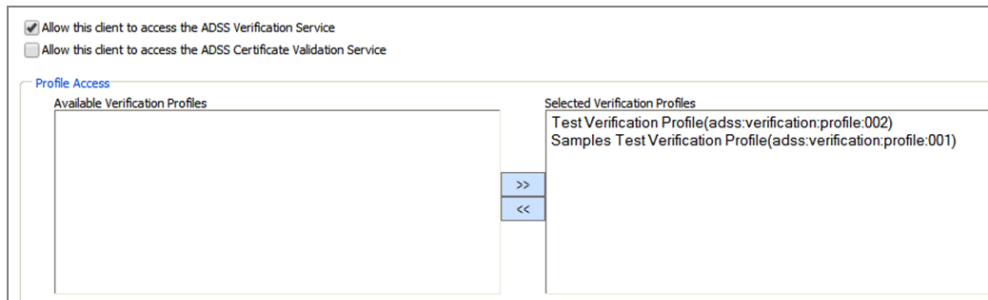
Business applications are clients to the ADSS Server. They can be authenticated using any of the following three techniques depending on the level of security desired:

- Registering a business application within the ADSS Client Manager and assigning it an “Originator ID”. The business application must then use this “Originator ID” in its service request messages in order for it to be authenticated successfully by ADSS Server.
- The service request messages can be sent over an SSL connection with client authentication enabled. The business application’s SSL client certificate must be registered within ADSS Server and the certificate’s Subject Common Name value must match the “Originator ID”.
- The business application can sign the service request message using a request signing certificate. This certificate must be registered within the ADSS Server. The signature format is XML DSig.

All three levels of security can be used together.

Assigning Profiles to Business Applications:

The configured verification profiles are assigned to specific business applications using the Client Manager Service. The two checkboxes define whether the business application can make verification service requests:



The screenshot shows a configuration window with two checkboxes at the top: Allow this client to access the ADSS Verification Service and Allow this client to access the ADSS Certificate Validation Service. Below these is a section titled "Profile Access" containing two panes. The left pane, "Available Verification Profiles", is currently empty. The right pane, "Selected Verification Profiles", contains two entries: "Test Verification Profile(adss:verification:profile.002)" and "Samples Test Verification Profile(adss:verification:profile.001)". Between the panes are two buttons: ">>" and "<<".

Authorised administrators can then simply select an available profile and move it across to the selected list by clicking on the ‘>>’ button, which updates the profiles assigned to the business application.

Signature Enhancement

In some environments the business application may wish to enhance the end-user's basic signature to create a timestamped or long-term signature, so that its validity and date/time of creation can be more easily established in the future. The problem with a basic signature is that the date/time is from the desktop system clock that can be varied at will. Adding a timestamp from a reliable server establishes a new baseline time of signing or signature acceptance.

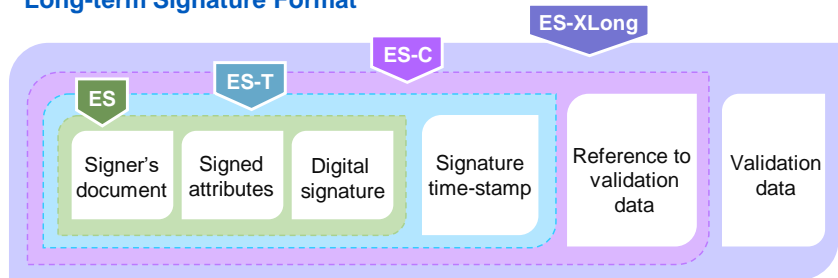
The business application can request this from the ADSS Server as part of the verification request message. During processing ADSS Server verify the signature, including the status of the signer's certificate chain, and if successful then actually embed this evidential information inside the user's signature, for future reference. The ADSS Verification Service will also embed a timestamp from a configured Time Stamp Authority (TSA) within the end-customer's signature, so that a long-term signature is formed. This long-term signature will be returned to the Client Application.

The diagram below shows both the basic and enhanced long term signature formats:

Basic Signature Format



Long-term Signature Format



Business Application Development

Business Applications can be developed for particular projects but there are also a number of pre-prepared applications developed by Ascertia such as Secure Email Server for verifying emails and attachments. Auto File Processor (AFP) has verification functionality on its roadmap to support bulk signature verification of documents and sort verified data into a trusted folder and other data files into an untrusted folder – discuss your requirements if this is of interest.

When a new Business Application is required this can be written in Java or DotNet (C#) as ADSS Client SDKs are provided in both languages. Alternatively, the underlying OASIS DSS XML/SOAP protocols may be implemented directly.

Summary

ADSS Server provides a rich set of features for supporting different signature types, configuring different verification profiles and linking these to business applications. Strong authentication, authorisation and role based controls together with secure logging of operator actions and application request/responses ensure the highest levels of security.

ADSS Server is well proven with hundreds of implementations around the world. Verification services are implicit within many of these deployments.