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.

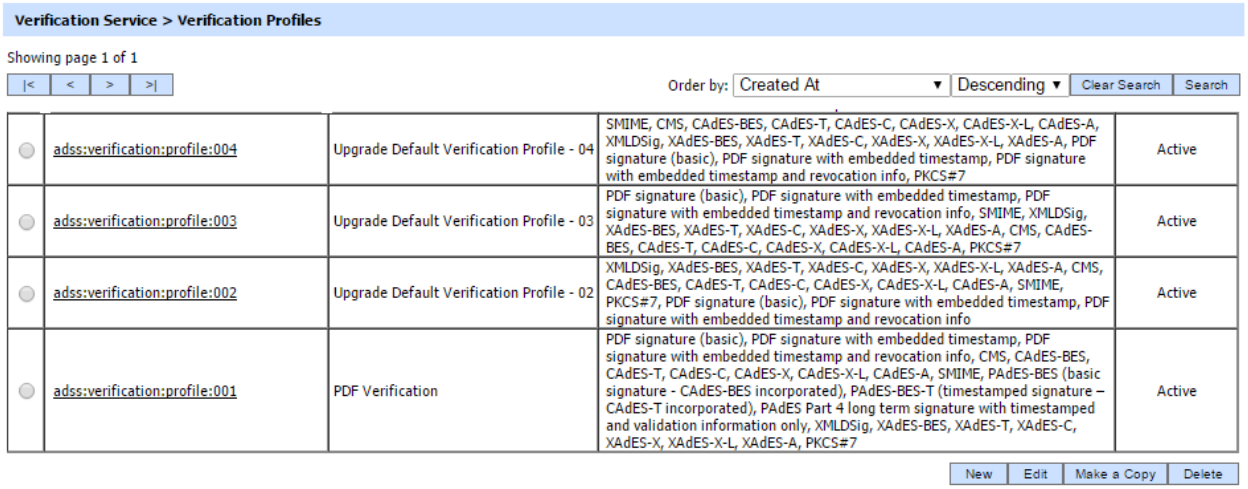
Verification Profiles

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)

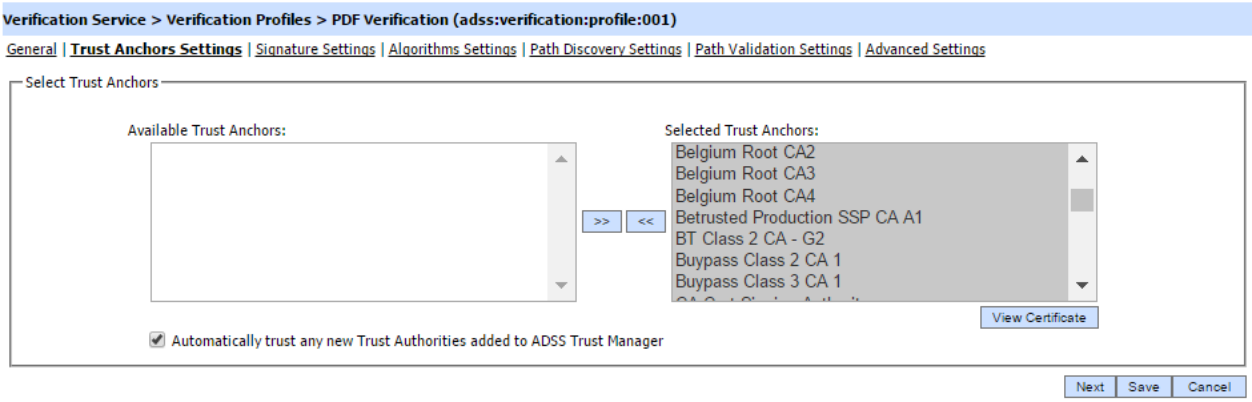
Verification Profile Identifier:

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



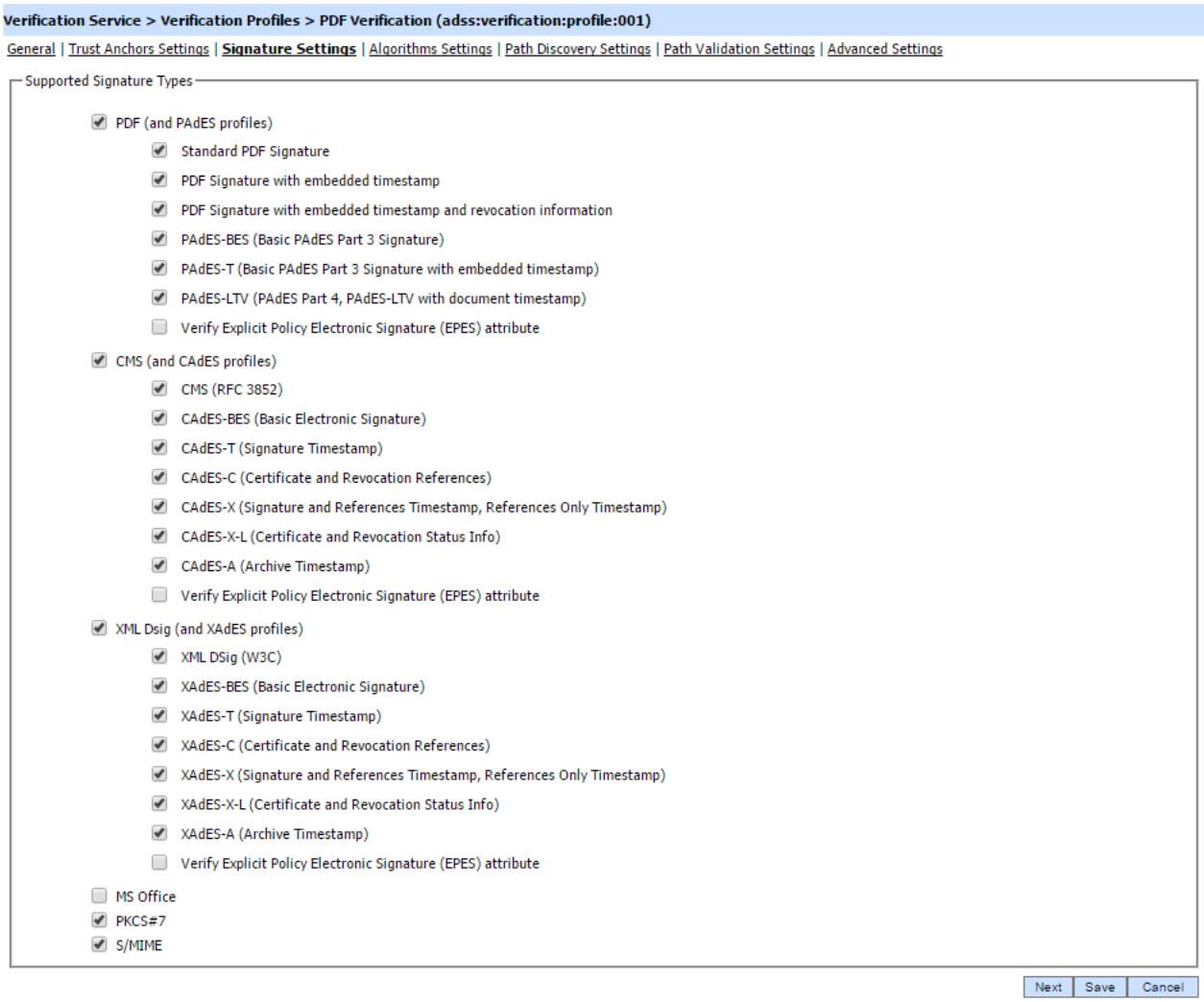
Specifying Trust Anchors:

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:



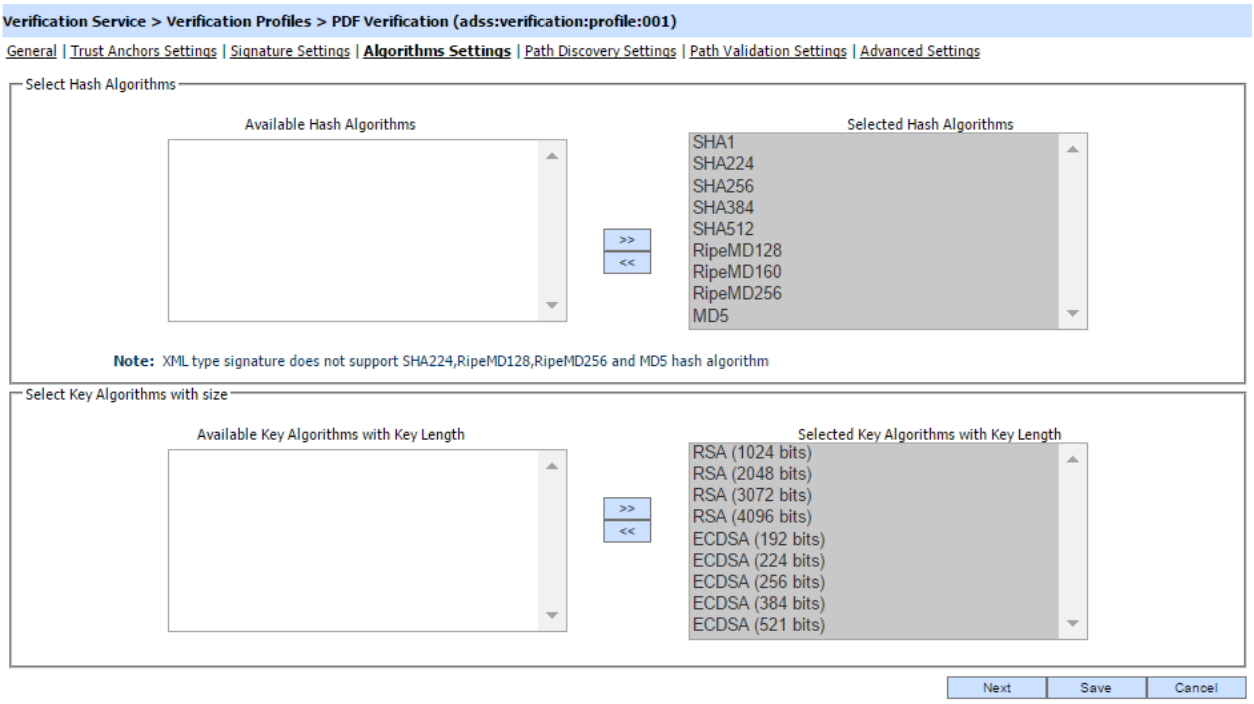
Signature Types:

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



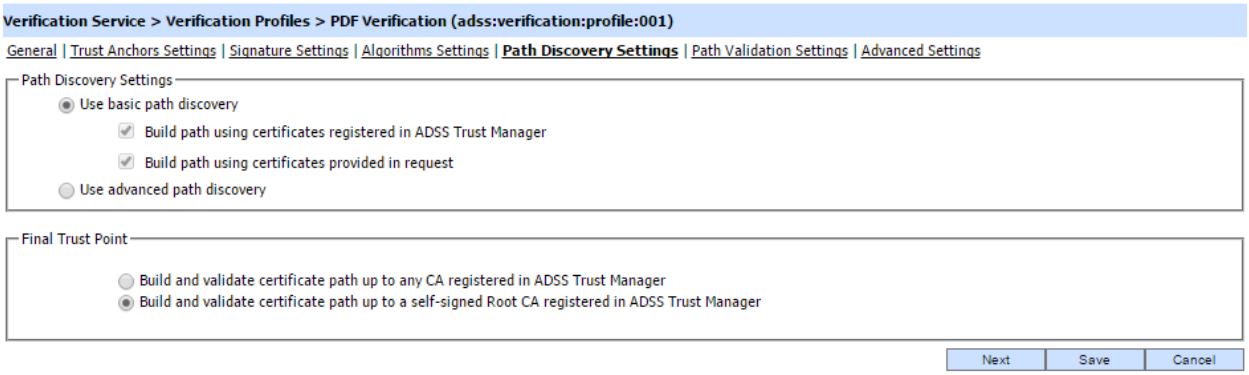
Configuring Acceptable Hash and Signature Algorithms:

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



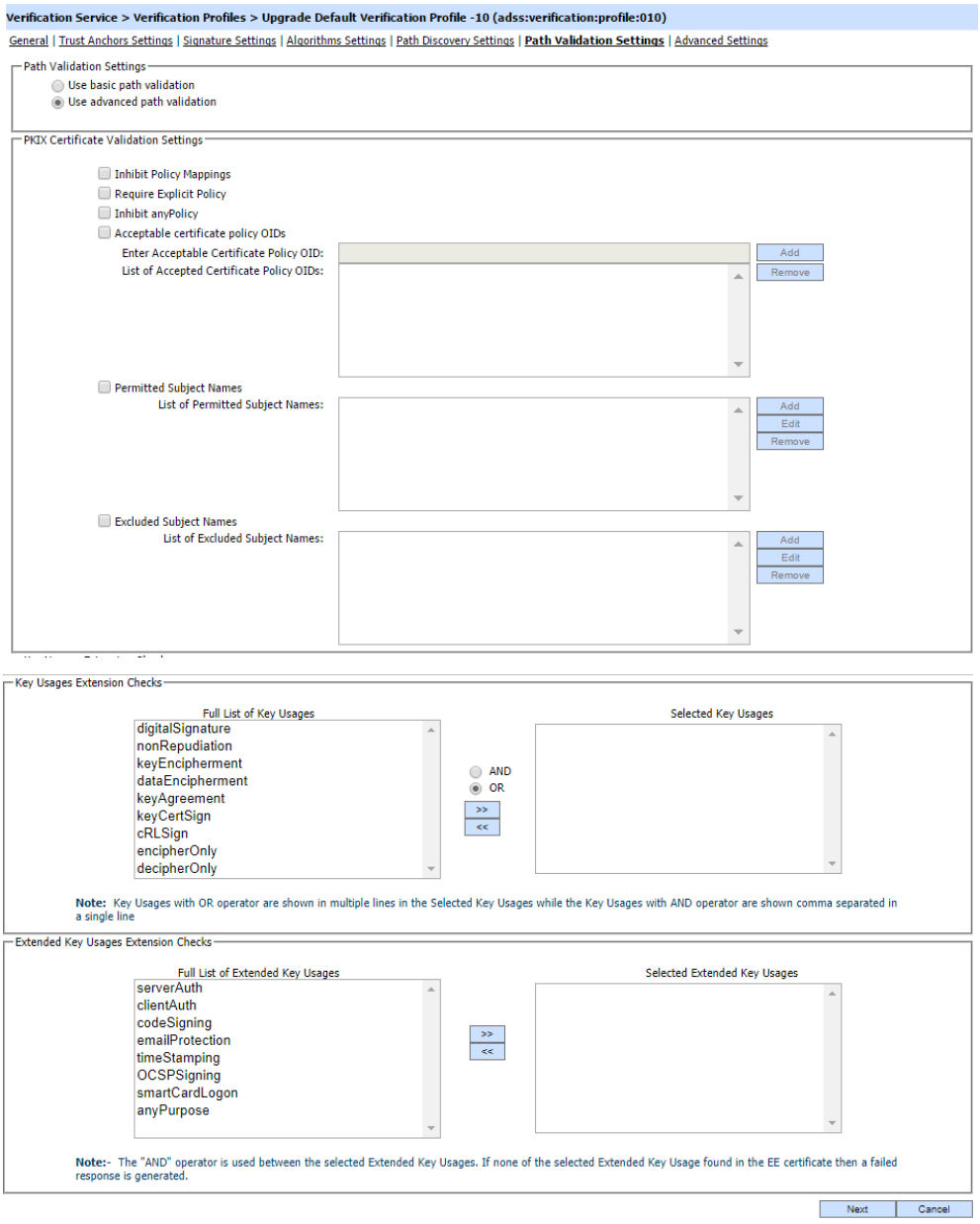
Configuring Path Discovery Options:

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

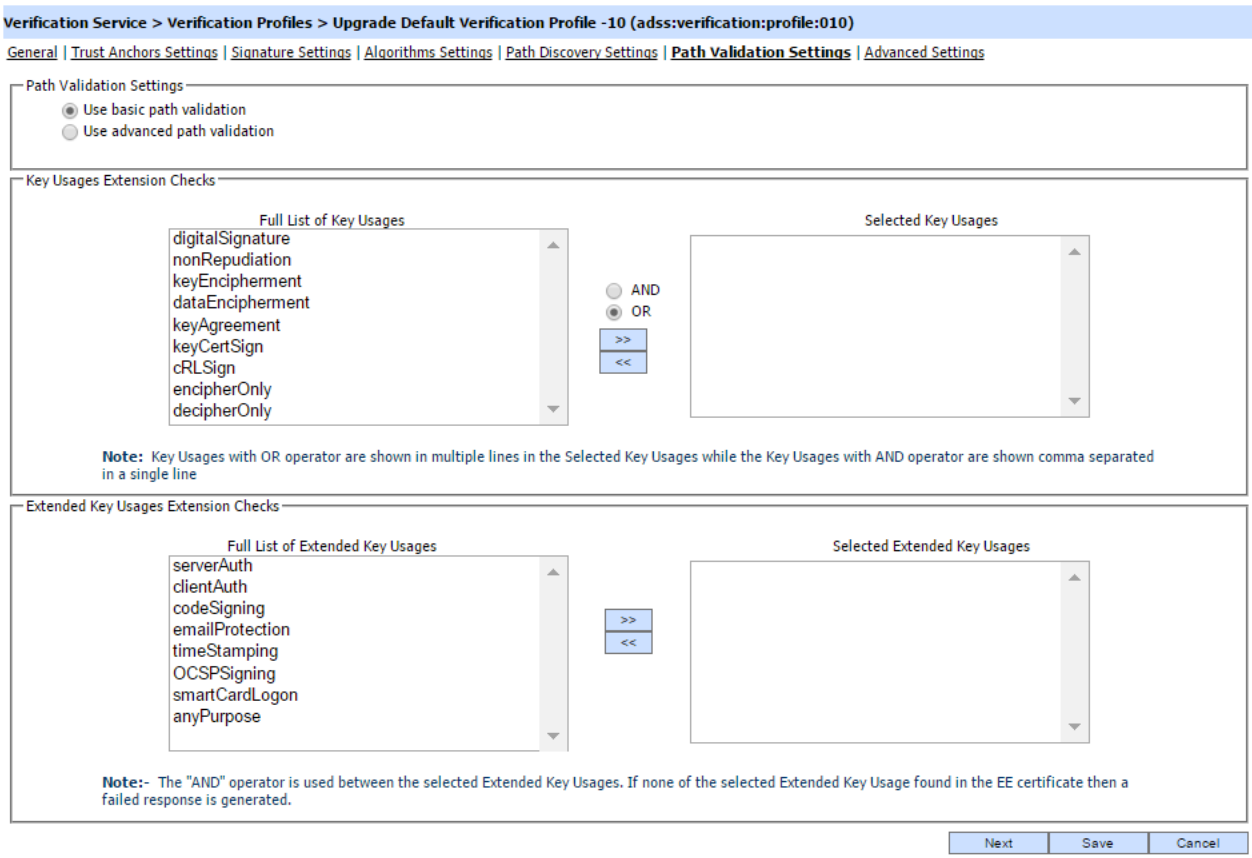


Configuring Path Validation Options:

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:

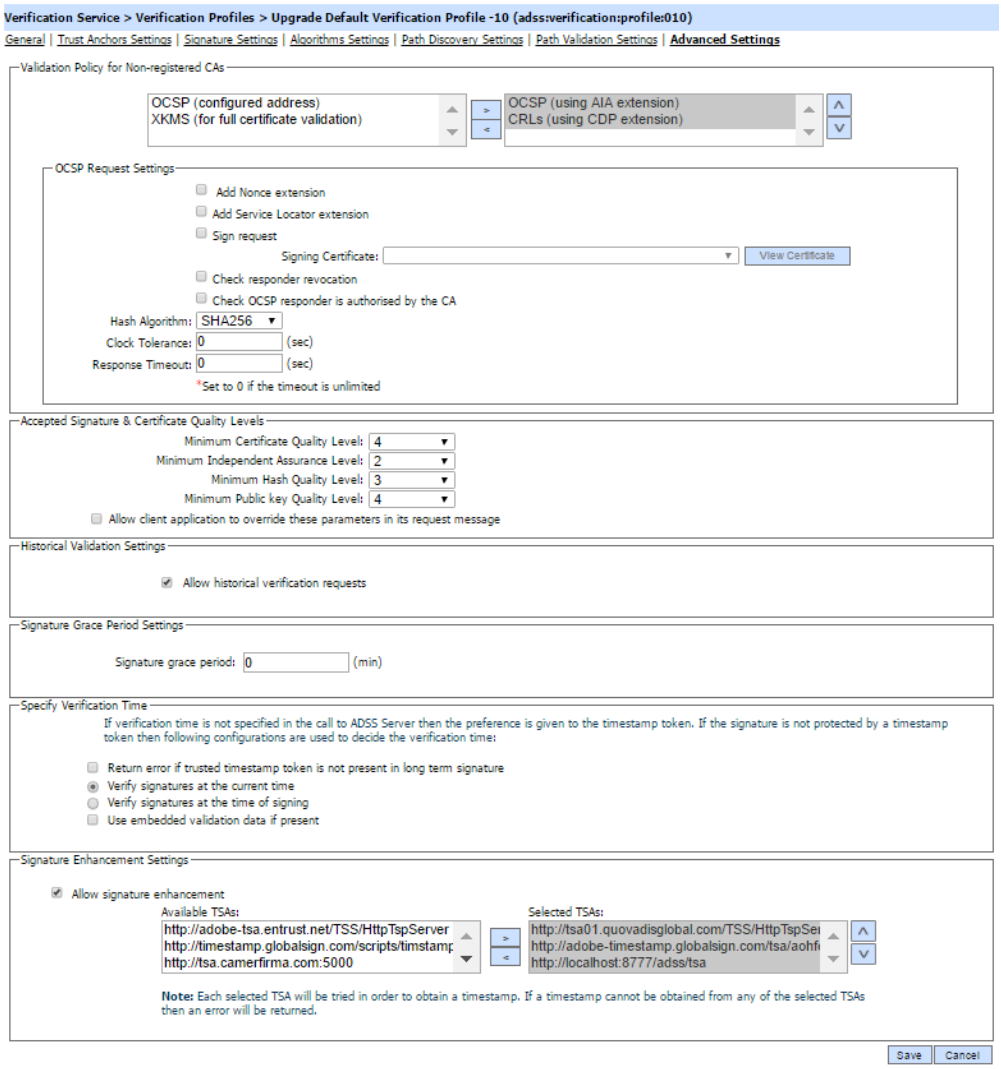


Key Usage and Extended Key Usage may also be enforced:



Advanced Options:

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



Signature and Certificate Quality Levels:

As seen above the PEPPOL 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

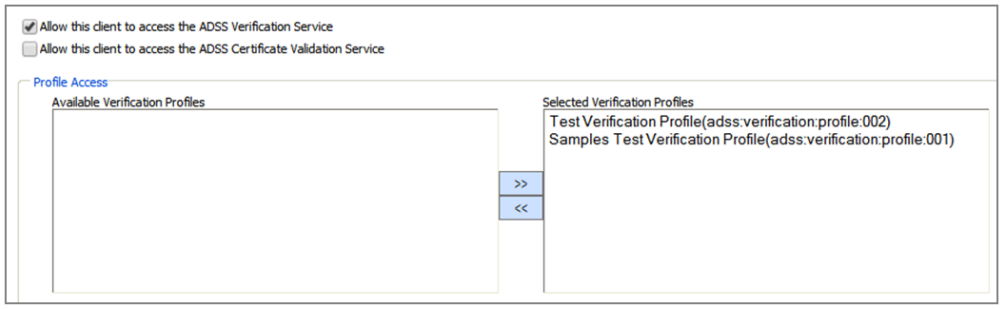
Business applications are clients to the ADSS Server. They can be authenticated using any of the following three techniques depending on the level of security desired:

- 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:



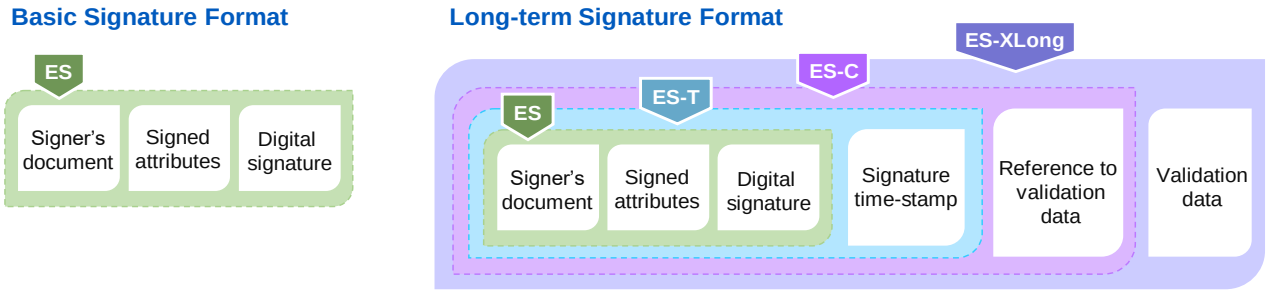
Authorised administrators can then simply select an available profile and move it across to the selected list by clicking on the '>>' button, which updates the profiles assigned to the business application.

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:



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.