eNotarization Markup Language (ENML)
Version 1.0

Public Review Draft 01

6 February 2009

Specification URIs:

This Version:
http://docs.oasis-open.org/legalxml-enotary/enml/v1.0/pr01/ENML-1.0-Specification.html
http://docs.oasis-open.org/legalxml-enotary/enml/v1.0/pr01/ENML-1.0-Specification.odt
http://docs.oasis-open.org/legalxml-enotary/enml/v1.0/pr01/ENML-1.0-Specification.pdf

Previous Version:
None

Latest Version:
http://docs.oasis-open.org/legalxml-enotary/enml/ENML-Specification.html
http://docs.oasis-open.org/legalxml-enotary/enml/ENML-Specification.odt

Technical Committee:
OASIS LegalXML eNotarization TC

Chair(s):
Rolly Chambers, American Bar Association (rolly.chambers@tprr.com)

Editor(s):
Arshad Noor (arshad.noor@strongauth.com)

Related Work:
This specification is related to:

  http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.html
- Symmetric Key Services Markup Language – OASIS Committee Specification
  http://docs.oasis-open.org/ekmi/sksml/v1.0/SKSML-1.0-Specification.html

Declared XML Namespace(s):
http://docs.oasis-open.org/legalxml-enotary/ns/enml/200901
Abstract:
This specification defines the first (1.0) version of the eNotarization Markup Language (ENML), an XML-based messaging protocol, by which applications executing on computing devices may notarize electronic documents for legal purposes. The specification accommodates most business and legal requirements of notarizing electronic documents without being constrained to a specific legal jurisdiction. While the initial specification was driven by notarization requirements of the fifty United States and the District of Columbia, there is foundational support for international jurisdictions and can be easily extended to accommodate specific requirements if needed.

Status:
This document was last revised by the LegalXML eNotarizationTC as of the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=legalxml-enotary.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/legalxml-enotary/ipr.php).

The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=legalxml-enotary/.
Notices

Copyright © OASIS® 2009. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS" and "ENML" are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.
# Table of Contents

1 Introduction .................................................................................................................. 7  
   1.1 Terminology .............................................................................................................. 7  
   1.2 Glossary .................................................................................................................... 7  
   1.3 Normative References .............................................................................................. 8  
   1.4 Non-normative References ...................................................................................... 9  
2 Background (non-normative) ........................................................................................ 10  
   2.1 Requirements (non-normative) ................................................................................. 11  
3 Examples of use of ENML (non-normative) ................................................................. 13  
   3.1 eNotarized document – Single Signer & Notary with NULL Cryptographic Signatures ........................................................................................................... 13  
   3.2 eNotarized document - Multiple Signers & Single Notary with NULL Cryptographic Signatures ........................................................................... 19  
   3.3 eNotarized document - Multiple Signers & Notaries with NULL Cryptographic Signatures ................................................................... 22  
   3.4 eWitnessed document – Single Signer & Notary Public with NULL Cryptographic Signatures ........................................................................... 26  
   3.5 eNotarized document - Multiple Signers, Single Notary & Symmetric Key Signature ........................................................................... 31  
   3.6 eNotarized document - Multiple Signers, Multiple Notaries & Symmetric Key Signatures ........................................................................... 41  
   3.7 eWitnessed document - Single Signer, Single Notary & Symmetric Key Signature ........................................................................... 48  
   3.8 eNotarized document - Single Signer, Single Notary & Digital Signature ........................................................................... 51  
   3.9 eNotarized document - Multiple Signers, Multiple Notaries & Digital Signatures ........................................................................... 55  
   3.10 eWitnessed document - Multiple Signers, Single Notary & Digital Signature ........................................................................... 58  
   3.11 eApostillized document with a Digital Signature ........................................................................... 60  
   3.12 The use of XPath in <ds:Signature> ........................................................................ 65  
4 Specification .................................................................................................................. 66  
   4.1 Element <NotarizedDocument> ................................................................................ 66  
   4.2 Element <WitnessedDocument> ................................................................................ 68  
   4.3 Element <ApostillizedDocument> .............................................................................. 70  
   4.4 Element <SignedDocuments> & <SignedDocument> .................................................. 73  
   4.5 Element <DocumentSigners> & <DocumentSigner> .................................................. 76  
   4.6 Element <NotaryCertificates> & <NotaryCertificate> ........................................... 82  
   4.7 Element <NotarySignatures> & <NotarySignature> ................................................ 85  
   4.8 Element <NotariesPublic> & <NotaryPublic> ........................................................... 90  
   4.9 Element <Document> ............................................................................................... 95  
   4.10 Element <DocumentMIMEType> & <SignedDocumentMIMEType> ......................... 97  
   4.11 Element <DocumentMIMETypeComments> ............................................................ 99  
   4.12 Element <DocumentComments> ............................................................................. 100  
   4.13 Element <SignerName> .......................................................................................... 101  
   4.14 Element <SignerTitle> ............................................................................................ 105  
   4.15 Element <SignerInternationalAddress> ................................................................. 106  
   4.16 Element <SignerUSAddress> .................................................................................. 110  
   4.17 Element <SignerIdentificationMethod> ................................................................. 113  
   4.18 Element <SignerSignature> .................................................................................... 114
4.19 Element <CertificateContent>............................................................... 115
4.20 Element <NotaryName>.......................................................................... 118
4.21 Element <NotaryCommissionNumber>..................................................... 122
4.22 Element <NotaryCommissionExpiryDate>................................................ 123
4.23 Element <NotaryInternationalJurisdiction>............................................. 124
4.24 Element <NotaryUSJurisdiction>.............................................................. 127
4.25 Element <NotaryBondNumber>............................................................... 130
4.26 Element <PersonPrefixName>................................................................. 131
4.27 Element <PersonFirstName> & <PersonGivenName>............................... 132
4.28 Element <PersonMiddleName>............................................................... 133
4.29 Element <PersonLastName> & <PersonSurName>...................................... 133
4.30 Element <PersonSuffixName>................................................................. 134
4.31 Element <PersonMaidenName>............................................................... 135
4.32 Element <PersonFullName>................................................................... 136
4.33 Element <StreetAddress1>................................................................. 136
4.34 Element <StreetAddress2>................................................................. 137
4.35 Element <City>..................................................................................... 138
4.36 Element <County>.................................................................................. 138
4.37 Element <State>................................................................................... 139
4.38 Element <PostalCode>........................................................................... 140
4.39 Element <Country> (International).......................................................... 140
4.40 Element <USState>............................................................................... 141
4.41 Element <USZipCode>........................................................................... 148
4.42 Element <Country> (USA)...................................................................... 149
4.43 Element <NotarizationType>.................................................................. 150
4.44 Element <NotarizationDate>................................................................. 151
4.45 Element <NotarizationInternationalLocation>........................................ 152
4.46 Element <NotarizationUSLocation>......................................................... 154
4.47 Element <ApostilleContent>................................................................. 155
4.48 Element <IssuingCountry>..................................................................... 161
4.49 Element <ApostilleNumber>................................................................. 162
4.50 Element <ApostilleDate>....................................................................... 163
4.51 Element <IssuedAtInternationalLocation>............................................... 165
4.52 Element <IssuedAtUSLocation>............................................................. 167
4.53 Element <DocumentSigningOfficialName>.............................................. 168
4.54 Element <DocumentSigningOfficialTitle>............................................... 170
4.55 Element <DocumentSigningOfficialSeal>................................................ 172
4.56 Processing Rule for ID attributes within ENML...................................... 176

5 Conformance........................................................................... 181
Appendix A. Acknowledgments........................................................... 182
Appendix B. Revision History............................................................ 183
Appendix C. ENML Security Implications (Non-normative)... 184
Appendix D. ENML Error Codes and Error Messages

Appendix E. Process for requesting a block of ENML Error Codes for Vendor Use
1 Introduction

This document presents the specification for the eNotarization Markup Language (ENML), a protocol by which applications may notarize electronic documents for legal purposes. All text is normative unless otherwise indicated.

1.1 Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119.

1.2 Glossary

3DES – Triple Data Encryption Standard; an advancement of the Data Encryption Standard that triples the size of the cryptographic key

AES – Advanced Encryption Standard; a new encryption standard standardized by NIST

Apostille – A certificate, issued by a competent authority (normally Secretaries of State in the USA for notarized documents), attesting to the validity of the notary's signature in the notarized document.

Base64 – An encoding scheme for representing binary data

Ciphertext – Encrypted data that must be decrypted before it can be used by an application

Conforming Application – A software application that has implemented the ENML protocol and meets the conformance requirements specified within this document

Cryptographic Module – A software library or hardware module dedicated to performing cryptographic operations

Cryptographic Signature – The encryption of a message digest using a symmetric or an asymmetric cryptographic key

DES – Data Encryption Standard; an encryption standard that is deprecated and no longer used in secure environments

Document Signer – An individual who signs the document being notarized

ENML – The eNotarization Markup Language, a specification of the OASIS LegalXML eNotarization Technical Committee

eNotarized Document – An electronic document that has been electronically signed by a Notary Public, in their official capacity, and which includes a notarial certificate

eWitnessed Document – An electronic document that has been electronically signed by a Notary Public, in their official capacity, but which DOES NOT include a notarial certificate

Manifest – An XML element that contains references to other XML elements within the same document

Message Digest – The result of a one-way cryptographic transformation of a document (or input) into a fixed-length value which is deemed to be unique. A message digest cannot be used to reconstruct the original document

Notarial Certificate - Official text, specified by law in some states, providing context for the Notary Public's signature within a notarized document
**Notarization** – The act of a Notary Public signing a document in their official capacity

**NULL Cryptographic Signature** – Legally admissible text typed by a Document Signer and/or Notary Public, indicating their fixation of an electronic signature to an electronic document. The NULL Signature **IS NOT** a Cryptographic Signature and does NOT use any cryptographic algorithms. Some sample NULL Cryptographic Signatures are “Signed by John Doe”, or “/Signed by John Doe”, “XXX John Doe XXX Signed by John Doe on January 27, 2008”, etc.

**Notary Public** – An individual, conferred by authority of a jurisdiction to notarize documents

**PII** – Personally Identifiable Information, such as credit card numbers, social security numbers, bank account numbers, drivers license numbers, etc.

**Plaintext** – Unencrypted data that can readily be processed by an application without any transformation

**Schema or XML Schema** – A specification of the World Wide Consortium, whose constructs can be used to structurally define an XML document. All documents that conform to a specific XML Schema definition, have the same structure

**SHA** – Secure Hashing Algorithm; a specific cryptographic transformation of a document (or input) into a fixed-length value (message digest) which is deemed to be unique

**SHA-1** – Secure Hashing Algorithm with a message digest of size 160-bits

**SHA-256** – Secure Hashing Algorithm with a message digest of size 256-bits

**SHA-384** – Secure Hashing Algorithm with a message digest of size 384-bits

**SHA-512** – Secure Hashing Algorithm with a message digest of size 512-bits

**SKMS** – Symmetric Key Management System; a collection of hardware and software providing symmetric encryption key-management services

**SKS - Symmetric Key Services**; a server that provides symmetric key management services to requesting applications and/or devices

**SKSML** – Symmetric Key Services Markup Language; an XML-based protocol to request and receive symmetric encryption key-management services

**Symkey** - A symmetric encryption key

**XMLEncryption** – Encrypted content represented in eXtensible Markup Language that conforms to the World Wide Web Consortium's XML Encryption standard

**XMLSignature** – A digital signature represented in eXtensible Markup Language that conforms to the World Wide Web Consortium's XML Signature standard

### 1.3 Normative References

- **[AES]** Advanced Encryption Standard

- **[DSS]** OASIS Standard
  Digital Signature Services
  11 April 2007
  [http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.html](http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.html)

- **[HCCH]** Convention of 5th October 1961
  Abolishing the Requirement of Legalisation for Foreign Public Documents
1.4 Non-normative References

[ENMLOverview] eNotarization Markup Language Overview
A PDF presentation of the OASIS ENML 1.0
2 Background (non-normative)

Notarized paper documents are a mainstay of many legal transactions around the world. Requiring a modicum of formality in the US – perhaps more than a modicum in others - it offers a safe and reasonably-priced, sometimes free, procedure for engendering trust in business transactions of significant value. With a notarized document, relying parties have access to the legal framework of their jurisdiction in the event they must seek redress for a transaction gone awry.

Computers have dramatically improved the efficiency of business procedures in many fields, including the legal field. The notarization of electronic documents is a logical extension of developments that continue the improvement. To ensure that electronically notarized ("eNotarized") documents are used and understood consistently and uniformly across applications, a standard protocol for depicting eNotarized documents is needed. The eNotarized Markup Language (ENML) is that protocol.

Using ENML, any application – be it a word-processor, a document-management system, a web-application to manage property recordings, a standalone notarizing application, etc. - can either electronically notarize a document, or verify a notarized document depicted in ENML.

ENML is a language that describes the content of electronically notarized, witnessed or apostillized document. An eNotarized document described by ENML, will carry similar constructs as a paper-notarized document:

- It will contain the semantics and language of some business transactions;
- It will identify and contain the signature(s) of the document-signer(s);
- It will contain one or more notarial certificate(s) with appropriate text for the jurisdiction in which the document was signed; and
- It will identify and contain the signature(s) of one or more notaries public.

ENML is designed to allow applications to speed the processing of business transactions by performing two specific actions:

1. It enables a human Notary Public to use software on a computer to electronically notarize an electronic document and create an "eNotarized" document. The eNotarization can be done instead of, or in addition to, the standard paper-document notarization typically performed today. Note that this process does NOT replace the human Notary Public or the process of having a Document Signer(s) present himself/herself/themselves to a Notary Public to perform the eNotarization. It creates an alternative method of notarizing documents which can be used in conjunction with or without paper-based notarized documents.

2. It enables the faster processing of eNotarized documents by having software perform the verification without interspersing a human in the process. Note that it DOES allow humans to verify the document's eNotarization constructs. However, this must be done with a Conforming Application, and within the context under which the document was eNotarized. The verification of an eNotarized document by a human being in conjunction with a Conforming Application, can be performed instead of, and/or in addition to the automated software verification.

ENML is not an electronic notary. It DOES NOT replace the human Notary Public or the non-technological processes typically involved amongst humans as part of a standard paper-document notarization.

While ENML builds on existing standards, it does not tie the language to any specific type of technology (other than the use of the eXtensible Markup Language (XML). While XML is meant to be a readable language by human beings, ENML is designed to be used mostly with software products that implement ENML’s constructs. It can ONLY be verified by Conforming Applications and cannot be verified by human beings without such tools.
2.1 Requirements (non-normative)

The requirements of the ENML protocol are that:

- It must support the legal requirements of paper-document notarization without being constrained to any specific legal jurisdiction;
- It must support the semantics and common business practices used in paper-document notarization while enabling improvements in the speed of creating, verifying and processing eNotarized documents;
- It must support eNotarized as well as eWitnessed documents;
- It must be platform independent – i.e. it must work on a computer running any modern operating system such as Linux, Microsoft Windows, Apple's OS-X, IBM's OS/400, Sun's Solaris, etc., and work with any programming language: Java, C, C++, PHP, Ruby, BASIC, etc.
- It must be usable by any application that understands the eXtensible Markup Language (XML);
- It must be embeddable in existing XML schemas without requiring drastic changes to existing applications;
- It must support many forms of cryptographic signing capabilities, such as the use of symmetric encryption keys, X509 digital certificates, Digital Signature Services [DSS], XML Signature [XMLSignature] without hindering the possibility of using any new cryptographic signing technology that can be depicted in XML;
- It must support applications that may choose not to use cryptographic signatures in ENML, but indicate consent by Document Signer(s) and Notaries Public to having signed the document(s), by entering relevant text in appropriate places in the document (i.e. it must support NULL Cryptographic Signatures).

ENML meets all the above requirements in the following manner:

- ENML uses elements typically required in notarized paper documents in the United States of America. It is hoped that it might address the needs of international jurisdictions, although there may be exceptions to the rule;
- ENML supports the eNotarization of a document by multiple Signers in multiple locations, notarized by multiple Notaries Public;
- ENML supports the creation and verification of both, eNotarized and eWitnessed documents;
- ENML can be used on any platform, application or programming language that understands XML and supports the required supporting software libraries;
- ENML elements can be embedded in existing XML-aware applications, allowing them to convert their electronic documents into eNotarized documents with minimal effort;
- ENML can encapsulate any legacy document (non-XML document) and eNotarize it, thus allowing XML-unaware applications to also take advantage of ENML;
- ENML allows any signing technology to be used for both the Document Signer and the Notary Public, including symmetric cryptographic keys, Digital Signature Services [DSS] and XML Signature [XMLSignature]. It also allows for the possibility of using newer signing technologies which can be depicted in XML;
- ENML allows for using non-cryptographic signatures – i.e. NULL Cryptographic Signatures – in eNotarized documents;
- While it may not currently be legal in any jurisdiction, ENML is technically capable of supporting the eNotarization of multiple documents with a single notarial certificate and notarization. Should the laws in any jurisdiction allow for such a capability in the future, ENML can take advantage of it.
without any changes to the protocol, although software using ENML may need to be updated to support the legal requirement.

ENML is the result of many notarial, business, legal, technical and security experts coming together to create a standard that works for today’s environment while being flexible to address tomorrow’s needs.
3 Examples of use of ENML (non-normative)

While Section 4 provides the definitive and normative details of ENML, this section provides many examples describing the use of ENML in various use-cases.

IMPORTANT NOTE: The documents, digest values, signatures and other cryptographic elements shown in these examples are NOT cryptographically valid. They are shown here, merely as illustrations of what ENML will look like when created by conforming applications. Conforming applications, however, must produce cryptographically valid elements in eNotarized documents, where required by the standard.

3.1 eNotarized document – Single Signer & Notary with NULL Cryptographic Signatures

The first example shows ENML for a document signed by one DocumentSigner and notarized by one Notary Public, who both use NULL Cryptographic Signatures to indicate their consent to having signed the electronic document.
<enml:DocumentSigners>
  <!- - Only one Signer - - >
  <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222640905-f91b09f20bfa099beb6b1cb493dfff9571c36aace2afdf6d273f7d">
    <enml:SignerName>
      <enml:PersonGivenName>John</enml:PersonGivenName>
      <enml:PersonSurName>Doe</enml:PersonSurName>
    </enml:SignerName>
    <enml:SignerIdentificationMethod>
      Produced Government-issued Identification Document
    </enml:SignerIdentificationMethod>
    <enml:SignerSignature>Signed by John Doe</enml:SignerSignature>
  </enml:DocumentSigner>
</enml:DocumentSigners>

<enml:NotaryCertificates>
  <!- - Information about the NotaryCertificates - - >
  <enml:NotaryCertificate Id="NotaryCertificate-US-CA-1565986-1222640980-0ef2d548e82b637a0568fc1ce69e2eb896acfc083defb82e24a05cab840f54">
    <enml:CertificateContent>
      <enml:NotarizationType>
        Acknowledgment
      </enml:NotarizationType>
      <enml:NotarizationDate>2007-02-07T15:19:17-08:00</enml:NotarizationDate>
      <enml:NotarizationUSLocation>
        <enml:City>Cupertino</enml:City>
        <enml:County>Santa Clara</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:NotarizationUSLocation>
      <enml:StatutoryContent>
        State of California
        County of Santa Clara
        On February 07 2007, before me Arshad Noor, personally appeared John Doe, who proved to me on the basis of satisfactory evidence to be the person whose name is subscribed to the within instrument and acknowledged to me that he executed the same in his authorized capacity, and that by his signature on the instrument the person, or the entity upon behalf of which the person acted, executed the instrument.
        I certify under PENALTY OF PERJURY under the laws of the State of California that the foregoing paragraph is true and correct.
        WITNESS my hand and official seal.
      </enml:StatutoryContent>
    </enml:CertificateContent>
  </enml:NotaryCertificate>
</enml:NotaryCertificates>
[a001] is the start tag of the NotarizedDocument element, the root of this eNotarized document instance.

[a002] indicates that this is an instance – an occurrence - of a document that conforms to a specific XML Schema Definition (XSD).

[a003] identifies the namespace to which this XML instance document conforms.

[a004] - [a005] identifies the name and location of the XML Schema Definition (XSD) file that defines the structure of this XML instance document.

[a006] is an attribute called “ID” whose value is "NotarizedDocument-US-CA-1565986-1226972451-802324532320e30067350d4694e7f79729ad7da392d3a18974d5a2e7bd09ee4". The ID attribute allows this document to be uniquely identified within other XML documents (such as an ApostillizedDocument). The value of the ID attribute is created by the application generating this ENML, and is unique for this instance of a NotarizedDocument.

[a007] - [a011] is a comment within the XML document.

[a012] is the start tag of the SignedDocuments element, which indicates the documents that are being eNotarized. Note: While current laws only allow for a single document to be notarized by a single notarial certificate and notary signature per notarization, the ENML schema allows for eNotarizing multiple documents with a single notarial certificate and notary signature. However, applications using/implementing ENML are expected to contain only a single document within this element for the foreseeable future.

[a013] is the start tag of a SignedDocument element, which has an attribute called “ID” whose value is "SignedDocument-US-CA-1565986-1226972451-802324532320e30067350d4694e7f79729ad7da392d3a18974d5a2e7bd09ee4". This is the only document being eNotarized within this NotarizedDocument instance. The ID attribute
allows this document to be uniquely identified within this instance, and to be referenced and covered by the Notary Public's signature as part of the eNotarization. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a NotarizedDocument.

[a014] is the start tag of the Document element.

[a015] - [a024] contains the Base64-encoded document that is being eNotarized. The Base64-encoding allows any type of document – text, graphics, multimedia, etc. - to be eNotarized and transported from machine to machine over networks, and rendered consistently.

[a025] declares the closing tag of the Document element.

[a026] indicates the Multipurpose Internet Mail Extension (MIME)-type of the document in the DocumentMIMETYPE element. This element identifies the type of document that is being eNotarized: a word-processing document created by OpenOffice Writer or Microsoft Word, a Portable Network Graphics (PNG) or Graphics Image File (GIF) file, etc. Receiving computers can use this element to determine which application to use to display the eNotarized document.

[a027] is the start of the DocumentComments tag.

[a028] is a free-form comment about the document being eNotarized. If there is no standard MIME-type for the document being eNotarized, this field can be used to provide hints to the receiving computer about which application to use to display the document.


[a030] declares the closing tag of the SignedDocument element.

[a031] declares the closing tag of the SignedDocuments element.

[a032] - [a036] is a comment within the XML document.

[a037] is the start tag of the DocumentSigners element. This element contains information about all document-signers of the eNotarized document, which does not include the Notary Public.

[a038] is a comment within the XML document.

[a039] is the start tag of a DocumentSigner element, which has an attribute called "ID" whose value is "DocumentSigner-US-CA-1565986-1222649095-f91b09f20bfa099be6b1cbb493fdd9571c3edac5c23fa36aace2af6d273fd7". This is the only signer of the document being eNotarized within this NotarizedDocument instance. The ID attribute allows this signer to be uniquely identified within this instance, and to be referenced and covered by the Notary Public's signature as part of the eNotarization. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a NotarizedDocument.

[a040] is the start tag of the SignerName element. This element contains information about the document-signer's name.

[a041] indicates the DocumentSigner's given name (first name) in the PersonGivenName element.

[a042] indicates the DocumentSigner's surname (last name) in the PersonSurName element.

[a043] declares the closing tag of the SignerName element.

[a044] is the start tag of the SignerIdentificationMethod element. This element contains information about how the document-signer identified himself/herself to the Notary Public. For privacy reasons, this element should not show any information considered sensitive, such as the Social Security Number, a Drivers License Number, a Passport or Identification Card Number, etc., but just indicate what type of identification was provided.

[a045] is free-form text that indicates that this signer provided a “Government-issued Identification Document”. It is assumed that the Notary Public has performed his/her duties with diligence, as required by law, in verifying the authenticity of the identification information.
is the closing tag of the **SignerIdentificationMethod** element.

is a **SignerSignature** element containing a **NULL Cryptographic Signature**. This implies that the document's signer's signature does not use a cryptographic mechanism to create an electronic signature. Please note the informative appendix on "ENML Security Implications" at the end of this document for an understanding of security issues within ENML.

is the closing tag of the **DocumentSigner** element.

is the closing tag of the **DocumentSigners** element.

- is a comment within the XML document.

is the start tag of the **NotaryCertificates** element. This element contains one or more notarial certificates for eNotarized documents.

is the start tag of the **NotaryCertificate** element. This element has an attribute called “ID” (with the value "NotaryCertificate-US-CA-15659856-1222640980-0ef2d548eb2b637a0568fc14e69e2eb896cfd83bef82e24a055554c9840f54" in this example) that allows this notarial certificate to be uniquely identified within this instance, and to be referenced and covered by the Notary Public's signature as part of the eNotarization. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a **NotarizedDocument**.

is the start tag of the **CertificateContent** element. This element contains information about the eNotarization act.

is the start tag of the **NotarizationType** element. This element identifies the type of eNotarization being performed.

identifies the type of eNotarization being performed; in this case, it is an Acknowledgment.

is the closing tag of the **NotarizationType** element.

is the start tag of the **NotarizationDate** element. This element identifies the date, time and time-zone on which the eNotarization was performed.

identifies the date, time and time-zone of eNotarization; in this case, it is February 7th, 2007 at 3:19PM in the Pacific Time Zone (- 08:00 from Coordinated Universal Time or UTC).

is the closing tag of the **NotarizationDate** element.

is the start tag of the **NotarizationUSLocation** element. This element identifies the political jurisdiction in the United States of America where the eNotarization was performed.

identifies the **City** where the eNotarization was performed; in this case, it is Cupertino.

identifies the **County** where the eNotarization was performed; in this case, it is Santa Clara.

identifies the **USState** where the eNotarization was performed; in this case, it is an CA (for California).

identifies the **Country** where the eNotarization was performed; in this case, it is in the USA.

is the closing tag of the **NotarizationUSLocation** element.

is the start tag of the **StatutoryContent** element. This element provides the statutory content that must be provided in a notarial certificate for an eNotarized document in the United States of America.

identifies the statutory content for this **NotarizedDocument**, in this case, it is free-form text for an Acknowledgment as specified by the State of California.

is the closing tag of the **StatutoryContent** element.
[a089] is the closing tag of the CertificateContent element.

[a090] is a blank line within the XML document (for readability).

[a091] is a comment within the XML document.

[a092] is the start tag of the NotaryPublic element. This element provides details about the Notary Public who eNotarized this NotarizedDocument.

[a093] is the start tag of the NotaryName element. This element contains information about the Notary Public's name.

[a094] indicates the Notary Public's given name (first name) in the PersonGivenName element.

[a095] indicates the Notary Public's surname (last name) in the PersonSurName element.

[a096] declares the closing tag of the NotaryName element.

[a097] is the start tag of the NotaryCommissionNumber element. This element identifies the identification number, if any, given to a Notary Public within a political jurisdiction.

[a098] identifies the Notary Public's commission number; in this case, it is 1565986.

[a099] is the closing tag of the NotaryCommissionNumber element.

[a100] is the start tag of the NotaryCommissionExpiryDate element. This element identifies the date and time on which the Notary Public's authorized commission expires.

[a101] identifies the expiry date, time and time-zone of the Notary Public's commission; in this case, it is April 29, 2009 at the 59th second of 11:59 PM in the Pacific time zone (-08:00 from UTC).

[a102] is the closing tag of the NotaryCommissionExpiryDate element.

[a103] is the start tag of the NotaryUSJurisdiction element. This element identifies the political jurisdiction in the United States of America where the Notary Public is authorized to perform his/her official duties.

[a104] identifies the County where the Notary Public performs his/her official duties; in this case, it is Santa Clara.

[a105] identifies the USState where the Notary Public performs his/her official duties; in this case, it is an CA (for California).

[a106] identifies the Country where the Notary Public performs his/her official duties; in this case, it is in the USA.

[a107] is the closing tag of the NotaryUSJurisdiction element.

[a108] is the closing tag of the NotaryPublic element.

[a109] is the closing tag of the NotaryCertificate element.

[a110] is the closing tag of the NotaryCertificates element.

[a111] - [a115] is a comment within the XML document.

[a116] is the start tag of the NotarySignatures element. This element contains information the notarial signatures of the Notaries Public.

[a117] is a NotarySignature element containing a NULL Cryptographic Signature. This implies that the Notary Public's signature was not created using any cryptographic mechanism. Please note the informative appendix on “ENML Security Implications” at the end of this document for an understanding of security issues within ENML.
3.2 eNotarized document - Multiple Signers & Single Notary with NULL Cryptographic Signatures

In this example of an eNotarized document, we show how two document signers are stored in one instance of a NotarizedDocument. It shows ENML for a document signed by multiple document-signers and notarized by one: Notary Public, all of whom use NULL Cryptographic Signatures within the ENML.

We dispense with showing the SignedDocuments, the NotaryCertificates and the NotarySignatures elements, and assume them to be similar to the ones shown in the previous example in Section 3.1, for brevity.

```xml
<enml:NotarizedDocument
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    ...>

    <!- - Information about the DocumentSigners
    -------------------------------------------->

    <enml:DocumentSigners>
        <!- - First Signer -->
        <enml:DocumentSigner Id="DocumentSigner- US-CA-1565986-1222640905-f91b09f20bfa099beb6b1cb493fdd9571c3edac5c23fa36aace2afed6d273fd7">
            <enml:SignerName>
                <enml:PersonFirstName>John</enml:PersonFirstName>
                <enml:PersonLastName>Doe</enml:PersonLastName>
            </enml:SignerName>
            <enml:SignerUSAddress>
                <enml:StreetAddress1>123 Main Street</enml:StreetAddress1>
                <enml:City>Sunnyvale</enml:City>
                <enml:County>Santa Clara</enml:County>
                <enml:USState>CA</enml:USState>
                <enml:USZipCode>94085</enml:USZipCode>
                <enml:Country>USA</enml:Country>
            </enml:SignerUSAddress>
            <enml:SignerIdentificationMethod>
                Personally Known to Notary Public
            </enml:SignerIdentificationMethod>
            <enml:SignerSignature>Signed by John Doe</enml:SignerSignature>
        </enml:DocumentSigner>

        <!- - Second Signer -->
        <enml:DocumentSigner Id="DocumentSigner- US-CA-1565986-1222643179-ca44f9fccc8950e9ad42959c725f2e31073f20b7058b777b2aa2d21f183006dc">
            <enml:SignerName>
                <enml:PersonFirstName>Jane</enml:PersonFirstName>
                <enml:PersonLastName>Doe</enml:PersonLastName>
            </enml:SignerName>
            <enml:SignerUSAddress>
                <enml:StreetAddress1>123 Main Street</enml:StreetAddress1>
                <enml:City>Sunnyvale</enml:City>
                <enml:County>Santa Clara</enml:County>
            </enml:SignerUSAddress>
        </enml:DocumentSigner>
    </enml:DocumentSigners>
</enml:NotarizedDocument>
```
[b001] is the start tag of the **NotarizedDocument** element, the root of this eNotarized document instance.

[b002] indicates that this is an instance – an occurrence - of a document that conforms to a specific XML Schema Definition (XSD).

...

[b031] - [b035] is a comment within the XML document.

[b036] is the start tag of the **DocumentSigners** element. This element contains information about all document-signers of the eNotarized document, which does not include the Notary Public.

[b037] is a comment, in this case referring to the first document-signer.

[b038] is the start tag of the first **DocumentSigner** element, which has an attribute called “ID” (whose value is "DocumentSigner-US-CA-1565986-122640905-f91b092000b099beb6b1cb493d9fd9571c3edac5c23fa3eace2ad8d273f67 " in this example). This is the first of multiple signers of the document being eNotarized within this **NotarizedDocument** instance. The ID attribute allows this signer to be uniquely identified within this instance, and to be referenced and covered by the Notary Public’s signature as part of the eNotarization. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a **NotarizedDocument**.

[b039] is the start tag of the **SignerName** element. This element contains information about the document-signer's name.

[b040] indicates the DocumentSigner's first name (given name) in the **PersonFirstName** element.

[b041] indicates the DocumentSigner's last name (surname) in the **PersonLastName** element.

[b042] declares the closing tag of the **SignerName** element.

[b043] is the start tag of the **SignerUSAddress** element. This element contains the address of the DocumentSigner in the United States of America.

[b044] indicates the first line of the street address of the DocumentSigner in the **StreetAddress1** element. In this example it is "123 Main Street".

[b045] identifies the **City** of the DocumentSigner's US address; in this case, it is Sunnyvale.

[b046] identifies the **County** of the DocumentSigner's US address; in this case, it is Santa Clara.

[b047] identifies the **USState** of the DocumentSigner's US address; in this case, it is CA (for California).
[b048] identifies the **USZipCode** of the DocumentSigner's US address; in this case, it is 94085.

[b049] identifies the **Country** of the DocumentSigner's address; in this case, it is in the USA.

[b050] is the closing tag of the **SignerUSAddress** element.

[b051] is the start tag of the **SignerIdentificationMethod** element. This element contains information about how the document-signer identified himself/herself to the Notary Public.

[b052] is free-form text that indicates that this signer was personally known to the Notary Public.

[b053] is the closing tag of the **SignerIdentificationMethod** element.

[b054] is a **SignerSignature** element using the **NULL Cryptographic Signature**.

[b055] is the closing tag of the **DocumentSigner** element.

[b056] is a blank line, for readability, within the XML document.

[b057] is a comment, in this case referring to the second document-signer.

[b058] is the start tag of the second **DocumentSigner** element, which has an attribute called “ID” (whose value is "DocumentSigner-US-CA-1565986-1222643179-ca44f9fcb8950e9ad42959c725f2e31073f20b705b777b2aa2d2f1f1830006dc " in this example). This is the second of multiple signers of this **NotarizedDocument** instance.

[b059] is the start tag of the **SignerName** element. This element contains information about the document-signer's name.

[b060] indicates the DocumentSigner's first name (given name) in the **PersonFirstName** element.

[b061] indicates the DocumentSigner's last name (surname) in the **PersonLastName** element.

[b062] declares the closing tag of the **SignerName** element.

[b063] is the start tag of the **SignerUSAddress** element. This element contains the address of the DocumentSigner in the United States of America.

[b064] indicates the first line of the street address of the DocumentSigner in the **StreetAddress1** element. In this example it is “123 Main Street”.

[b065] identifies the **City** of the DocumentSigner's US address; in this case, it is Sunnyvale.

[b066] identifies the **County** of the DocumentSigner's US address; in this case, it is Santa Clara.

[b067] identifies the **USState** of the DocumentSigner's US address; in this case, it is CA (for California).

[b068] identifies the **USZipCode** of the DocumentSigner's US address; in this case, it is 94085.

[b069] identifies the **Country** of the DocumentSigner's address; in this case, it is the USA.

[b070] is the closing tag of the **SignerUSAddress** element.

[b071] is the start tag of the **SignerIdentificationMethod** element. This element contains information about how the document-signer identified himself/herself to the Notary Public.

[b072] is free-form text that indicates that this signer was personally known to the Notary Public.

[b073] is the closing tag of the **SignerIdentificationMethod** element.

[b074] is a **SignerSignature** element using the **NULL Cryptographic Signature**.

[b075] is the closing tag of the **DocumentSigner** element.
is the closing tag of the DocumentSigners element.

... - [b106] is a comment within the XML document.

is the closing tag of the NotarizedDocument element, signifying the end of this XML instance.

3.3 eNotarized document - Multiple Signers & Notaries with NULL Cryptographic Signatures

In this example of an eNotarized document, we show two document signers in two different locations, signing a document at different times, and notarized by two different Notaries Public at their respective locations. The business use-case requires both document signers to have signed the document and their signings notarized, before the document is accepted by a Relying Party. ENML supports this use-case, as shown below. This example, like the earlier ones, depicts the use of NULL Cryptographic Signatures.

We dispense with showing the SignedDocuments element and assume it to be similar to the ones shown in the previous example in Sections 3.1.

```xml
[...]

<!- - Information about the DocumentSigners - - >
<enml:DocumentSigners>
  <!- -  First Signer - - >
  <enml:DocumentSigner Id="DocumentSigner- US-CA-1565986-1222643713-8b62fbc3df242c5832169d6514f6eb2799b3759833bdc3a6df68718a0d1a66250">
    <enml:SignerName>
      <enml:PersonFirstName>James</enml:PersonFirstName>
      <enml:PersonLastName>Doe</enml:PersonLastName>
    </enml:SignerName>
    <enml:SignerUSAddress>
      <enml:StreetAddress1>123 California Ave</enml:StreetAddress1>
      <enml:City>Buffalo</enml:City>
      <enml:County>Erie</enml:County>
      <enml:USState>NY</enml:USState>
      <enml:USZipCode>14202</enml:USZipCode>
      <enml:Country>USA</enml:Country>
    </enml:SignerUSAddress>
    <enml:SignerIdentificationMethod>
      Personally Known to Notary Public
    </enml:SignerIdentificationMethod>
    <enml:SignerSignature>Signed by James Doe</enml:SignerSignature>
  </enml:DocumentSigner>

  <!- -  Second Signer - - >
  <enml:DocumentSigner Id="DocumentSigner- US-CA-1565986-1222640905-f91b0f20bfa099be6b1cb493dadb9571c3edac5c23fa36aace2af6d6d273f7">
    <enml:SignerName>
      <enml:PersonFirstName>John</enml:PersonFirstName>
      <enml:PersonLastName>Doe</enml:PersonLastName>
    </enml:SignerName>
    <enml:SignerUSAddress>
      <enml:StreetAddress1>123 Main Street</enml:StreetAddress1>
    </enml:SignerUSAddress>
    <enml:SignerSignature>Signed by John Doe</enml:SignerSignature>
  </enml:DocumentSigner>
</enml:DocumentSigners>
```
<enml:DocumentSigners>
  <enml:DocumentSigner>
    <enml:SignerUSAddress>
      <enml:City>Sunnyvale</enml:City>
      <enml:County>Santa Clara</enml:County>
      <enml:USState>CA</enml:USState>
      <enml:USZipCode>94085</enml:USZipCode>
      <enml:Country>USA</enml:Country>
    </enml:SignerUSAddress>
    <enml:SignerIdentificationMethod>
      Personally Known to Notary Public
    </enml:SignerIdentificationMethod>
    <enml:SignerSignature>Signed by John Doe</enml:SignerSignature>
  </enml:DocumentSigner>
</enml:DocumentSigners>

<!- - Information about the NotaryCertificates - - >
<enml:NotaryCertificates>
  <!- - First notary certificate - - >
  <enml:NotaryCertificate Id="NotaryCertificate- US-CA-1565986-1222643804-a56e5f27bea87a8e19d5652771d939291baca2ec4bfb9148a3b0af374b2af79">
    <enml:CertificateContent>
      <enml:NotarizationType>Acknowledgment</enml:NotarizationType>
      <enml:NotarizationDate>2007-02-07T15:19:17-08:00</enml:NotarizationDate>
      <enml:NotarizationUSLocation>
        <enml:City>Buffalo</enml:City>
        <enml:County>Erie</enml:County>
        <enml:USState>NY</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:NotarizationUSLocation>
      <enml:StatutoryContent>
        State of New York
        County of Erie
        On the 7th day of February in the year 2007 before me, the undersigned, personally appeared James Doe personally known to me or proved to me on the basis of satisfactory evidence to be the individual whose name is subscribed to the within instrument and acknowledged to me that he executed the same in his capacity, and that by his signature on the instrument, the individual, or the person upon behalf of which the individual acted, executed the instrument.
      </enml:StatutoryContent>
    </enml:CertificateContent>

<!- - First notary public who provided this certificate - - >
  <enml:NotaryPublic>
    <enml:NotaryName>
      <enml:PersonGivenName>Howard</enml:PersonGivenName>
      <enml:PersonSurName>Jones</enml:PersonSurName>
    </enml:NotaryName>
    <enml:NotaryCommissionNumber>N1234567</enml:NotaryCommissionNumber>
    <enml:NotaryCommissionExpiryDate></enml:NotaryCommissionExpiryDate>
  </enml:NotaryPublic>
</enml:NotaryCertificate>
</enml:NotaryCertificates>
On February 07 2007, before me Arshad Noor, personally appeared John Doe, who proved to me on the basis of satisfactory evidence to be the person whose name is subscribed to the within instrument and acknowledged to me that he executed the same in his authorized capacity, and that by his signature on the instrument the person, or the entity upon behalf of which the person acted, executed the instrument.

I certify under PENALTY OF PERJURY under the laws of the State of California that the foregoing paragraph is true and correct.

WITNESS my hand and official seal.
[c184] <enml:County>Santa Clara</enml:County>
[c185] <enml:USState>CA</enml:USState>
[c186] <enml:Country>USA</enml:Country>
[c187] </enml:NotaryUSJurisdiction>
[c188] </enml:NotaryPublic>
[c189] </enml:NotaryCertificates>
[c190] <![CDATA[Information about the NotarySignatures]]>
[c191] -*>
[c192] <enml:NotarySignature>Signed by Howard Jones</enml:NotarySignature>
[c193] <enml:NotarySignature>Signed by Arshad Noor</enml:NotarySignature>
[c194] </enml:NotarySignatures>
[c195] <![CDATA[End of XML document]]>
[c196] -*>
[c197] </enml:NotarizedDocument>

[c001] is the start tag of the NotarizedDocument element, the root of this eNotarized document instance.

[c002] indicates that this is an instance – an occurrence - of a document that conforms to a specific XML Schema Definition (XSD).

... 

[c031] - [c035] is a comment within the XML document.

[c036] is is the start tag of the DocumentSigners element. This element contains information about all document-signers of the eNotarized document, which does not include the Notary Public.

[c037] is a comment, in this case referring to the first document-signer.

[c038] – [c055] identifies the first of two DocumentSigner elements. These lines in this XML example refer to a DocumentSigner, James Doe, in Buffalo city of Erie county in New York state, in the USA.

[c056] is a blank line, for readability, within the XML document.

[b057] is a comment, in this case referring to the second document-signer.

[c058] – [c075] identifies the second of two DocumentSigner elements. These lines in this XML example refer to a DocumentSigner, John Doe, in Sunnyvale city of Santa Clara county in the state of California.

[c076] is the closing tag of the DocumentSigners element.

[c077] - [c081] is a comment within the XML document.

[c082] is the start tag of the NotaryCertificates element. This element, in this example, contains two notarial certificates, one for each DocumentSigner.

[c083] - [c133] is the first of the two NotaryCertificate elements. This element contains information about the NotaryCertificate and the NotaryPublic, which in this example, refers to Howard Jones in Buffalo city in Erie county in New York state.

[c134] is a blank line, for readability, within the XML document.
[c135] - [c189] is the second of the two NotaryCertificate elements. This element contains information about the NotaryCertificate and the NotaryPublic, which in this example, refers to Arshad Noor in Cupertino city of Santa Clara county in the state of California.

[c190] is the closing tag of the NotaryCertificates element.

[c191] - [c195] is a comment within the XML document.

[c196] is the start tag of the NotarySignatures element. This element contains information the notarial signatures of the Notaries Public.

[c197] – [c198] are two NotarySignature elements with NULL Cryptographic Signatures used by the two Notaries.

[c199] is the closing tag of the NotarySignatures element.

[c200] - [c204] is a comment within the XML document.

[c205] is the closing tag of the NotarizedDocument element, signifying the end of this XML instance.

3.4 eWitnessed document – Single Signer & Notary Public with NULL Cryptographic Signatures

An eWitnessed document is one, in which the signing of the document by the DocumentSigner(s) is witnessed by the Notary Public, and the document is notarized, but does not contain a Notarial Certificate. The following ENML shows a sample eWitnessed document that contains NULL Cryptographic Signatures:

```xml
<enml:WitnessedDocument>
    <enml:SignedDocuments>
        <enml:SignedDocument Id="SignedDocument-US-CA-1565986-1222644436-42051bbfad827707d07e362511e44bc200636e045c4dd1f1b1ed30590ac3a74">
            <enml:DocumentMIMEType>image/gif</enml:DocumentMIMEType>
        </enml:SignedDocument>
    </enml:SignedDocuments>
</enml:WitnessedDocument>
```
Information about the DocumentSigners

- - >

<enml:DocumentSigners>

<!-- Only one Signer -->

<enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222644495-f91b09f20bfa099be6b6b1cb493dfdd9571c3edac5c23fa36aace2af6d6d273f87f">

<enml:SignerName>
  <enml:PersonGivenName>John</enml:PersonGivenName>
  <enml:PersonSurName>Doe</enml:PersonSurName>
</enml:SignerName>

<enml:SignerIdentificationMethod>
  Produced Government-issued Identification Document
</enml:SignerIdentificationMethod>

<enml:SignerSignature>Signed by John Doe</enml:SignerSignature>
</enml:DocumentSigner>
</enml:DocumentSigners>

Information about the NotaryPublic

- - >

<enml:NotariesPublic>

<!-- NotaryPublic who witnessed this signing -->

<enml:NotaryPublic Id="NotaryPublic-US-CA-1565986-1222644533-4c87e7e6f4e6864ca8641a6790d7504b6d6c69014d9bd624">

<enml:NotaryName>
  <enml:PersonGivenName>Arshad</enml:PersonGivenName>
  <enml:PersonSurName>Noor</enml:PersonSurName>
</enml:NotaryName>

<enml:NotaryCommissionNumber>1565986</enml:NotaryCommissionNumber>

<enml:NotaryCommissionNumber>2009-04-29T23:59:59-08:00</enml:NotaryCommissionNumber>

<enml:NotaryUSJurisdiction>
  <enml:County>Santa Clara</enml:County>
  <enml:USState>CA</enml:USState>
  <enml:Country>United States of America</enml:Country>
</enml:NotaryUSJurisdiction>
</enml:NotaryPublic>
</enml:NotariesPublic>

Information about the NotarySignatures

- - >

<enml:NotarySignatures>

<enml:NotarySignature>Signed by Arshad Noor</enml:NotarySignature>
</enml:NotarySignatures>
</enml:NotarySignatures>

End of XML document

============================================
is the start tag of the **WitnessedDocument** element, the root of this eWitnessed document instance.

indicates that this is an instance – an occurrence - of a document that conforms to a specific XML Schema Definition (XSD).  

identifies the namespace to which this XML instance document conforms.  

- identifies the name and location of the XML Schema Definition (XSD) file that defines the structure of this XML instance document.  

is an attribute called “ID” whose value is "WitnessedDocument-US-CA-1565986-1226972451-8023243532a0e30067350d469c4e77f9729ad7da392d3a18974d5a2e7bd09ee4". The ID attribute allows this document to be uniquely identified within other XML documents (such as an **ApostillizedDocument**) . The value of the ID attribute is created by the application generating this ENML, and is unique for this instance of a **WitnessedDocument**.  

is a comment within the XML document.  

is the start tag of the **SignedDocuments** element, which indicates the documents that are being eWitnessed. Note: While current laws only allow for a single document to be witnessed by a notary signature per eWitnessing, the ENML schema allows for eWitnessing multiple documents with a single notary signature. However, applications using/implementing ENML are expected to contain only a single document within this element for the foreseeable future.  

is the start tag of a **SignedDocument** element, which has an attribute called "ID" whose value is “SignedDocument-US-CA-1565986-1222644436-42051bbfad827707d07e3825f1fe44bc200036e045c4dd01f1f1ed30590ac3a74” in this example. This is the only document being eWitnessed within this **WitnessedDocument** instance. The ID attribute allows this document to be uniquely identified within this instance, and to be referenced and covered by the Notary Public’s signature as part of the eWitnessing. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a **WitnessedDocument**.  

is the start tag of the **Document** element.  

contains the Base64-encoded document being eWitnessed. The Base64-encoding allows any type of document – text, graphics, multimedia, etc. - to be eWitnessed and transported from machine to machine over networks, and rendered consistently.  

declares the closing tag of the **Document** element.  

indicates the Multipurpose Internet Mail Extension (MIME)-type of the document in the **DocumentMIMEType** element. This element identifies the type of document being eWitnessed: a word-processing document created by OpenOffice Writer or Microsoft Word, a Portable Network Graphics (PNG) or Graphics Image File (GIF) file, etc. Receiving computers can use this element to determine which application to use to display the eWitnessed document.  

is the start of the **DocumentComments** tag.  

is a free-form comment about the document being eNotarized. If there is no standard MIME-type for the document being eWitnessed, this field can be used to provide hints to the receiving computer about which application to use to display the document.  

declares the closing tag of the **DocumentComments** element.  

declares the closing tag of the **SignedDocument** element.
[d031] declares the closing tag of the SignedDocuments element.

[d032] - [d036] is a comment within the XML document.

[d037] is the start tag of the DocumentSigners element. This element contains information about all document-signers of this eWitnessed document, which does not include the Notary Public.

[d038] is a comment within the XML document.

[d039] is the start tag of a DocumentSigner element, which has an attribute called "ID" whose value is "DocumentSigner-US-CA-1565986-1222644495-f91b09f20bfa099bebe6b1c1b493fdd9571c3edac5c23fa36aace2af6d273f7d" in this example. This is the only signer of the document being eWitnessed within this WitnessedDocument instance. The ID attribute allows this signer to be uniquely identified within this instance, and to be referenced and covered by the Notary Public's signature as part of the eWitnessing. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a WitnessedDocument.

[d040] is the start tag of the SignerName element. This element contains information about the document-signer's name.

[d041] indicates the DocumentSigner's given name (first name) in the PersonGivenName element.

[d042] indicates the DocumentSigner's surname (last name) in the PersonSurName element.

[d043] declares the closing tag of the SignerName element.

[d044] is the start tag of the SignerIdentificationMethod element. This element contains information about how the document-signer identified himself/herself to the Notary Public. For privacy reasons, this element should not show any information considered sensitive, such as the Social Security Number, a Drivers License Number, a Passport or Identification Card Number, etc., but just indicate what type of identification was provided.

[d045] is free-form text indicating that this signer provided a "Government-issued Identification Document". It is assumed that the Notary Public has performed his/her duties with diligence, as required by law, in verifying the authenticity of the identification information.

[d046] is the closing tag of the SignerIdentificationMethod element.

[d047] is a SignerSignature element containing a NULL Cryptographic Signature. Please note the informative appendix on "ENML Security Implications" at the end of this document for an understanding of ENML security.

[d048] is the closing tag of the DocumentSigner element.

[d049] is the closing tag of the DocumentSigners element.

[d050] - [d054] is a comment within the XML document.

[d055] is the start tag of the NotariesPublic element. This element provides information about all Notaries who witnessed this document-signing.

[d056] is an XML comment.

[d057] is the start tag of the NotaryPublic element. This element provides details about the Notary Public who eWitnessed this WitnessedDocument. It has an attribute called "ID" (with the value "NotaryPublic-US-CA-1565986-12226444533-4c8e1a0ef954439f6e46b694ca8641a6416790d7504b6dccc69014d9b624" in this example) that allows this Notary Public to be uniquely identified within this instance, and to be referenced and covered by the Notary Public's signature as part of the eWitnessing. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a WitnessedDocument.
[d058] is the start tag of the **NotaryName** element. This element contains information about the Notary Public's name.

[d059] indicates the Notary Public's given name (first name) in the **PersonGivenName** element.

[d060] indicates the Notary Public's surname (last name) in the **PersonSurName** element.

[d061] declares the closing tag of the **NotaryName** element.

[d062] is the start tag of the **NotaryCommissionNumber** element. This element identifies the identification number, if any, given to a Notary Public within a political jurisdiction.

[d063] identifies the Notary Public's commission number; in this case, it is 1565986.

[d064] is the closing tag of the **NotaryCommissionNumber** element.

[d065] is the start tag of the **NotaryCommissionExpiryDate** element. This element contains the date and time at which the Notary Public's authorized commission expires.

[d066] defines the expiry date of the Notary Public's commission; in this case, it is April 29, 2009.

[d067] is the closing tag of the **NotaryCommissionExpiryDate** element.

[d068] is the start tag of the **NotaryUSJurisdiction** element. This element identifies the political jurisdiction in the United States of America where the Notary Public is authorized to perform his/her official duties.

[d069] identifies the **County** where the Notary Public performs his/her official duties; in this case, it is Santa Clara.

[d070] identifies the **USState** where the Notary Public performs his/her official duties; in this case, it is CA (for California).

[d071] identifies the **Country** where the Notary Public performs his/her official duties; in this case, it is the United States of America.

[d072] is the closing tag of the **NotaryUSJurisdiction** element.

[d073] is the closing tag of the **NotaryPublic** element.

[d074] is the closing tag of the **NotariesPublic** element.

[d075] - [d079] is a comment within the XML document.

[d080] is the start tag of the **NotarySignatures** element. This element contains information the notarial signatures of the Notaries Public.

[d081] is a **NotarySignature** element with a NULL **Cryptographic Signature**. Please note the informative appendix on "ENML Security Implications" at the end of this document for an understanding of ENML security.

[a082] is the closing tag of the **NotarySignatures** element.

[d083] - [d087] is a comment within the XML document.

[d088] is the closing tag of the **WitnessedDocument** element, signifying the end of this XML instance.
3.5 eNotarized document - Multiple Signers, Single Notary & Symmetric Key Signature

In this example of an eNotarized document, we show two document signers, signing a document with NULL Cryptographic Signatures and notarized by one Notary Public using a symmetric cryptographic key to encrypt the message digest of the Signature's Manifest, thus creating a Cryptographic Signature.

Please note the informative appendix on “ENML Security Implications” at the end of this document for an understanding of ENML security and symmetric encryption keys.

While all the elements of the eNotarized document are shown, for the sake of some brevity, much of the Document element's content is not shown below.
Personally Known to Credible Witnesses

Signed by John Doe

Second Signer

Signed by Jane Doe

Information about the NotaryCertificate

On January 27 2007, before me Arshad Noor, personally appeared John Doe and Jane Doe, who proved to me on the basis of satisfactory evidence to be the persons whose names are subscribed to the within instrument and acknowledged to me that they executed the same in their authorized capacity, and that by their signatures on the instrument the persons, or the entity upon behalf of which the persons acted, executed the instrument.

I certify under PENALTY OF PERJURY under the laws of the State of California that the foregoing paragraph is true and correct.

WITNESS my hand and official seal.

NotaryPublic who provided this certificate
<enml:NotaryName>
  <enml:PersonGivenName>Arshad</enml:PersonGivenName>
  <enml:PersonSurName>Noor</enml:PersonSurName>
</enml:NotaryName>

<enml:NotaryCommissionNumber>1565986</enml:NotaryCommissionNumber>
<enml:NotaryCommissionExpiryDate>2009-04-29T23:59:59-08:00</enml:NotaryCommissionExpiryDate>
<enml:NotaryUSJurisdiction>
  <enml:County>Santa Clara</enml:County>
  <enml:USState>CA</enml:USState>
  <enml:Country>USA</enml:Country>
</enml:NotaryUSJurisdiction>
<enml:NotarySignatures>
  <ds:Signature>
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
      <ds:SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
      <ds:Reference Type="http://www.w3.org/2000/09/xmldsig#Manifest"
        URI="#Manifest-US-CA-1565986-1222641281"/>
      <ds:Transforms>
        <ds:Transform
          Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </ds:Transforms>
      <ds:DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>
        MtGcQfervxf/b3xvrPEstt0h1fg=
      </ds:DigestValue>
    </ds:SignedInfo>
    <ds:SignatureValue>09fbf143326bb09beefabe8d84094ee14c5eee8c</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:KeyName>10514-1-123</ds:KeyName>
      <ds:RetrievalMethod
        URI="http://skms.somecompany.com/symkeyServlet/getsymkey"/>
    </ds:KeyInfo>
    <ds:Object>
      <ds:Manifest
        Id="Manifest-US-CA-1565986-1222641281"
        URI="#SignedDocument-US-CA-1565986-1222646484-546e0248b7e7f52a4427880d3efb84557746e691556e0d37bb92ef1f2ad7fd0b"/>
      <ds:DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>
        09fbf143326bb09beefabe8d84094ee14c5eee8c
      </ds:DigestValue>
    </ds:Object>
  </ds:Signature>
</enml:NotarySignatures>
<enml:NotarySignatures>
  <ds:Reference URI="#DocumentSigner- US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d273fd7">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>W4DupJioi0mW7aG+N1qiNLtqvsk=</ds:DigestValue>
  </ds:Reference>
  <ds:Reference URI="#DocumentSigner- US-CA-1565986-1222643179-ca44f9fccc8950e9ad42959c725f2e31073f20b7058b777b2aa2d21f1830006dc">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>Yx5JkTS0ZxaMOuEpm/SxmSRSgAw=</ds:DigestValue>
  </ds:Reference>
  <ds:Reference URI="#NotaryCertificate- US-CA-1565986-1222643229-1963a865f3516996f711936045ced816a0fccd4aca57c6807a1c44f9b98a1e5f">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>+w49wI2IZOiUOVSx7L0fQp1nO30=</ds:DigestValue>
  </ds:Reference>
</ds:Manifest>
</ds:Object>
</ds:Signature>
</enml:NotarySignatures>
</enml:NotarizedDocument>

[e001] is the start tag of the **NotarizedDocument** element, the root of this eNotarized document instance.

[e002] indicates that this is an instance – an occurrence - of a document that conforms to a specific XML Schema Definition (XSD).

[e003] identifies the namespace to which this XML instance document conforms.

[e004] - [e005] identifies the name and location of the XML Schema Definition (XSD) file that defines the structure of this XML instance document.

[e006] is an attribute called “ID” whose value is "NotarizedDocument-US-CA-1565986-1226972451-8023243532a0e30063750d469c4e7779729ad7da392d3a18974d9a27bd09ee4". The ID attribute allows this document to be uniquely identified within other XML documents (such as an **ApostillizedDocument**). The value of the ID attribute is created by the application generating this ENML, and is unique for this instance of a **NotarizedDocument**.

[e007] - [e011] is a comment within the XML document.
[e012] is the start tag of the SignedDocuments element, which indicates the documents that are being eNotarized. Note: While current laws only allow for a single document to be notarized by a single notarial certificate and notary signature per notarization, the ENML schema allows for eNotarizing multiple documents with a single notarial certificate and notary signature. However, applications using/implementing ENML are expected to contain only a single document within this element for the foreseeable future.

[e013] is the start tag of a SignedDocument element, which has an attribute called “ID” whose value is "SignedDocument-US-CA-1565986-1222646484-546e0248b7e7f52a4427880d3eefb84557746e691556e0d37bb92fe1f2ad7fd0b". This is the only document being eNotarized within this NotarizedDocument instance. The ID attribute allows this document to be uniquely identified within this instance, and to be referenced and covered by the Notary Public's signature as part of the eNotarization. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a NotarizedDocument.

[e014] is the start tag of the Document element.

[e015] - [e017] contains the Base64-encoded document that is being eNotarized. The Base64-encoding allows any type of document – text, graphics, multimedia, etc. - to be eNotarized and transported from machine to machine over networks, and rendered consistently.


[e019] indicates the Multipurpose Internet Mail Extension (MIME)-type of the document in the DocumentMIMETYPE element. This element identifies the type of document that is being eNotarized: a word-processing document created by OpenOffice Writer or Microsoft Word, a Portable Network Graphics (PNG) or Graphics Image File (GIF) file, etc. Receiving computers can use this element to determine which application to use to display the eNotarized document.

[e020] is the DocumentComments tag which permits document creators to place free-form comments about the document in this element.

[e021] declares the closing tag of the SignedDocument element.

[e022] declares the closing tag of the SignedDocuments element.

[e023] - [e027] is a comment within the XML document.

[e028] is the start tag of the DocumentSigners element. This element contains information about all document-signers of the eNotarized document, which does not include the Notary Public.

[e029] is a comment within the XML document.

[e030] is the start tag of the first DocumentSigner element, which has an attribute called “ID” whose value is "DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d273fd7". This is the first of two signers of this document. The ID attribute allows this signer to be uniquely identified within this instance, and to be referenced and covered by the Notary Public's signature as part of the eNotarization. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a NotarizedDocument.

[e031] is the start tag of the SignerName element. This element contains information about the document-signer's name.

[e032] indicates the DocumentSigner's given name (first name) in the PersonGivenName element.

[e033] indicates the DocumentSigner's surname (last name) in the PersonSurName element.

[e034] declares the closing tag of the SignerName element.

[e035] is the start tag of the SignerUSAddress element.
[e036] provides the SignerUSAddress's City element.

[e037] provides the SignerUSAddress's County element.

[e038] provides the SignerUSAddress's USState element as a two-character abbreviation.

[e039] provides the SignerUSAddress's Country element.

[e040] is the closing tag of the SignerUSAddress element.

[e041] is the start tag of the SignerIdentificationMethod element. This element contains information about how the document-signer identified himself/herself to the Notary Public. For privacy reasons, this element should not show any information considered sensitive, such as the Social Security Number, a Drivers License Number, a Passport or Identification Card Number, etc., but just indicate what type of identification was provided.

[e042] is free-form text that indicates that this signer was “Personally known to credible witnesses”. It is assumed that the Notary Public has performed his/her duties with diligence, as required by law, in verifying the identity of the DocumentSigner.

[e043] is the closing tag of the SignerIdentificationMethod element.

[e044] is the SignerSignature element. The value of this element, in this example, states “Signed by John Doe” indicating the use of a NULL Cryptographic Signature.

[e045] is the closing tag of the DocumentSigner element for the first signer.

[e046] is a blank line for readability.

[e047] is an XML comment.

[e048] is the start tag of the second DocumentSigner element, which has an attribute called “ID” whose value is "DocumentSigner-US-CA-1565986-1222643179-ca44f9cc8950e9ad42959c725f2d31073d2b7058b777b2aa2d21f1830006dc ".

[e049] is the start tag of the SignerName element. This element contains information about the document-signer's name.

[e050] indicates the DocumentSigner's given name (first name) in the PersonGivenName element.

[e051] indicates the DocumentSigner's surname (last name) in the PersonSurName element.

[e052] declares the closing tag of the SignerName element.

[e053] is the start tag of the SignerIdentificationMethod element. This element contains information about how the document-signer identified himself/herself to the Notary Public. For privacy reasons, this element should not show any information considered sensitive, such as the Social Security Number, a Drivers License Number, a Passport or Identification Card Number, etc., but just indicate what type of identification was provided.

[e054] is free-form text that indicates that this signer “Produced Government-issued Identification Document”. It is assumed that the Notary Public has performed his/her duties with diligence, as required by law, in verifying the identity of the DocumentSigner.

[e055] is the closing tag of the SignerIdentificationMethod element.

[e056] is the SignerSignature element, which contains a NULL Cryptographic Signature and the value “Signed by Jane Doe”.

[e057] is the closing tag of the DocumentSigner element for the second signer.

[e058] is the closing tag of the DocumentSigners element.
[e059] - [e063] is a comment within the XML document.

[e064] is the start tag of the NotaryCertificates element. This element contains one or more notarial certificates as required by law for eNotarized documents.

[e065] is the start tag of the NotaryCertificate element. This element has an attribute called "ID" (with the value "NotaryCertificate-US-CA-1565986-1222643229-1963a95f51896f711936045cde816a00fcd4aca57cbb07a1c44f9b98a1e5f" in this example) that allows this notarial certificate to be uniquely identified within this instance, and to be referenced and covered by the Notary Public's signature as part of the eNotarization. The value of the ID attribute is specified by the application creating this ENML, and is unique within this instance of a NotarizedDocument.

[e066] is the start tag of the CertificateContent element. This element contains information about the eNotarization act.

[e067] is the NotarizationType element. This element identifies the type of eNotarization being performed; in this case, it is an Acknowledgment.

[e068] - [e070] is the NotarizationDate element. This element identifies the date on which the eNotarization was performed; in this case, it is January 27, 2007 at 3:23:46PM in the Pacific time-zone (-08:00 from UTC).

[e071] is the start tag of the NotarizationUSLocation element. This element identifies the political jurisdiction in the United States of America where the eNotarization was performed.

[e072] identifies the City where the eNotarization was performed; in this case, it is Cupertino.

[e073] identifies the County where the eNotarization was performed; in this case, it is Santa Clara.

[e074] identifies the USState where the eNotarization was performed; in this case, it is an CA (for California).

[e075] identifies the Country where the eNotarization was performed; in this case, it is the USA.

[e076] is the closing tag of the NotarizationUSLocation element.

[e077] is the start tag of the StatutoryContent element. This element provides the statutory content that must be provided in a notarial certificate for an eNotarized document in the United States of America.

[e078] – [e094] identifies the statutory content for this NotarizedDocument; in this case, it is free-form text for an Acknowledgment as specified by the State of California.

[e095] is the closing tag of the StatutoryContent element.

[e096] is the closing tag of the CertificateContent element.

[e097] is a blank line within the XML document (for readability).

[e098] is a comment within the XML document.

[e099] is the start tag of the NotaryPublic element. This element provides details about the Notary Public who eNotarized this NotarizedDocument.

[e100] is the start tag of the NotaryName element. This element contains information about the Notary Public's name.

[e101] indicates the Notary Public's given name (first name) in the PersonGivenName element.

[e102] indicates the Notary Public's surname (last name) in the PersonSurName element.

[e103] declares the closing tag of the NotaryName element.
[e104] is the start tag of the NotaryCommissionNumber element. This element identifies the identification number, if any, given to a Notary Public within a political jurisdiction.

[e105] identifies the Notary Public's commission number; in this case, it is 1565986.

[e106] is the closing tag of the NotaryCommissionNumber element.

[e107] is the start tag of the NotaryCommissionExpiryDate element. This element identifies the date and time at which the Notary Public's authorized commission expires.

[e108] identifies the expiry date of the Notary Public's commission; in this case, it is April 29, 2009.

[e109] is the closing tag of the NotaryCommissionExpiryDate element.

[e110] is the start tag of the NotaryUSJurisdiction element. This element identifies the political jurisdiction in the United States of America where the Notary Public is authorized to perform his/her official duties.

[e111] identifies the County where the Notary Public performs his/her official duties; in this case, it is Santa Clara.

[e112] identifies the USState where the Notary Public performs his/her official duties; in this case, it is CA (for California).

[e113] identifies the Country where the Notary Public performs his/her official duties; in this case, it is the USA.

[e114] is the closing tag of the NotaryUSJurisdiction element.

[e115] is the closing tag of the NotaryPublic element.

[e116] is the closing tag of the NotaryCertificate element.

[e117] is the closing tag of the NotaryCertificates element.

[e118] - [e122] is a comment within the XML document.

[e123] is the start tag of the NotarySignatures element. This element contains information the notarial signatures of the Notaries Public.

[e124] is the start tag of the ds:Signature element. This element belongs to the [XMLSignature] namespace, and consequently has the ds: prefix in front of the element name. It contains information about the notarial signature of the Notary Public.

[e125] is the start tag of the ds:SignedInfo element. This element provides information about what is signed within this eNotarized document.

[e126] – [e127] describes the ds:CanonicalizationMethod element. To ensure that XML documents and their signatures are calculated consistently, software that complies with the [XMLSignature] standard “normalizes” the XML content based on the method specified in the Algorithm attribute of this element. In this example, the algorithm used to normalize the XML content is indicated by the "http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments" value.

[e128] – [e129] describes the ds:SignatureMethod element. Application generating and verifying signatures using the [XMLSignature] standard, need to know the method by which the XML document is signed. The Algorithm attribute of this element identifies this method. In this example, the algorithm used to sign the XML content is the Hashed Message Authentication Code (HMAC) using the SHA1 method, as indicated by the "http://www.w3.org/2000/09/xmldsig#hmac-sha1" value.

[e130] describes the ds:Reference element, which identifies the element(s) that are signed within this eNotarized document. In this example, the Reference element indicates that the object being signed is a
**ds:Manifest** which can be located by the URI "#Manifest-US-CA-1565986-1222641281" within the same document.

[e131] identifies start of the **ds:Transforms** element. This element identifies what transformations were applied to the element(s) before signing this eNotarized document. This also implies that application software verifying this signature must apply the same transformations to the XML elements before verifying the signature.

[e132] – [e133] describes the **ds:Transform** element. In this instance, the algorithm used to transform the XML content is indicated by the "http://www.w3.org/2000/09/xmldsig#enveloped-signature" value.

[e132] is the closing tag of the **ds:Transforms** element.

[e135] – [e136] describes the **ds:DigestMethod** element. The **Algorithm** attribute of this element identifies how the message-digest of the content in the **ds:Manifest** was calculated. In this instance, the algorithm used to calculate the message-digest of the XML content is indicated by the "http://www.w3.org/2000/09/xmldsig#sha1" value.

[e137] is the start tag of the **ds:DigestValue** element.

[e138] provides the Base64 encoded message-digest value of the contents of the **ds:Manifest** element.

[e139] is the closing tag of the **ds:DigestValue** element.

[e140] is the closing tag of the **ds:Reference** element.

[e141] is the closing tag of the **ds:SignedInfo** element.

[e142] is the start tag of the **ds:SignatureValue** element. This element is the container for the electronic signature of the Notary Public on this instance of the eNotarized document.

[e143] provides the actual electronic signature of the Notary Public on this instance of the eNotarized document. It is a Base64-encoded value.

[e144] is the closing tag of the **ds:SignatureValue** element.

[e145] is the start tag of the **ds:KeyInfo** element. This element is the container for the cryptographic symmetric-key, or information about the cryptographic key, used by the Notary Public to sign this instance of the eNotarized document.

[e146] defines the **ds:KeyName** element. The value in this element is, typically, an identifier to a cryptographic key that may be accessed at some location. In this example, the identifier follows the format of a **GlobalKeyId** as defined in [ENML].

[e147] – [e148] defines the **ds:RetrievalMethod** element. The **URI** attribute in this element identifies a location where the cryptographic key referenced in **ds:KeyName** can be accessed. In this example, the URI "http://skms.somecompany.com/symkeyServlet/getsymkey" follows the format defined in [ENML].

[e149] is the closing tag of the **ds:KeyInfo** element.

[e150] is a blank line for readability.

[e151] is the start tag of the **ds:Object** element.

[e150] is the start tag of the **ds:Manifest** element. This element carries information about the elements in the eNotarized document that are covered by the Notary's signature. The unique identifier of this element is shown in this example, in the ID attribute as "Manifest-US-CA-1565986-1222641281".

[e152] is a blank line for readability.

[e154] describes the first **ds:Reference** element within this **Manifest**. The **URI** attribute of this element identifies the content in this instance of an eNotarized document that is covered by the Notary Public's
signature. In this instance, the URI points to the SignedDocument element that has "SignedDocument-US-CA-1565986-1222646484-546e0248b7e7f52a4427880d3efb84557746e691556e0d37bb92fe1f2ad7fd0f" in its ID attribute.

[e155] – [e156] describes the ds:DigestMethod element. The Algorithm attribute identifies the method used to calculate the message-digest of the XML content referenced in the above-mentioned ds:Reference. In this example, it is indicated by the "http://www.w3.org/2000/09/xmldsig#sha1" value.

[e157] is the start tag of the ds:DigestValue element.

[e158] provides the Base64 encoded message-digest value of the contents of the element(s) referenced by this ds:Reference URI.

[e159] is the closing tag of the ds:DigestValue element.

[e160] is the closing tag of the ds:Reference element.

[e161] is a blank line for readability.

[e160] describes the second ds:Reference element within this Manifest. The URI attribute of this element identifies the content in this instance of an eNotarized document that is covered by the Notary Public's signature. In this instance, the URI points to the DocumentSigner element that has "DocumentSigner-US-CA-1565986-1222643107-f91b09f20ba099be6b1c493fdd9571c3edac5c23fa36aace2af6d273fd7" in its ID attribute.

[e162] – [e164] describes the ds:DigestMethod element. The Algorithm attribute identifies the method used to calculate the message-digest of the XML content referenced in the above-mentioned ds:Reference. In this example, it is indicated by the "http://www.w3.org/2000/09/xmldsig#sha1" value.

[e165] is the start tag of the ds:DigestValue element.

[e166] provides the Base64 encoded message-digest value of the contents of the element(s) referenced by this ds:Reference URI.

[e167] is the closing tag of the ds:DigestValue element.

[e168] is the closing tag of the ds:Reference element.

[e169] is a blank line for readability.

[e170] describes the third ds:Reference element within this Manifest. The URI attribute of this element identifies the content in this instance of an eNotarized document that is covered by the Notary Public's signature. In this instance, the URI points to a second DocumentSigner element that has "DocumentSigner-US-CA-1565986-1222643179-ca44f9fccc8950e9ad42959c725f2e31073f20b7058b777b2aa2d21f1830006d" in its ID attribute.

[e171] – [e172] describes the ds:DigestMethod element. The Algorithm attribute identifies the method used to calculate the message-digest of the XML content referenced in the above-mentioned ds:Reference. In this example, it is indicated by the "http://www.w3.org/2000/09/xmldsig#sha1" value.

[e173] is the start tag of the ds:DigestValue element.

[e174] provides the Base64 encoded message-digest value of the contents of the element(s) referenced by this ds:Reference URI.

[e175] is the closing tag of the ds:DigestValue element.

[e176] is the closing tag of the ds:Reference element.

[e177] is a blank line for readability.
[e178] describes the last ds:Reference element within this Manifest. The URI attribute of this element identifies the content in this instance of an eNotarized document that is covered by the Notary Public's signature. In this instance, the URI points to the NotaryCertificate element that has "NotaryCertificate-US-CA-1565986-1222643229-1963a865f3516996f711936045ced816a0fecd4aca57c6807a1c44f9b98a1e5f" in its ID attribute.

[e179] – [e180] describes the ds:DigestMethod element. In this instance, the algorithm used to calculate the message-digest of the XML content referenced in the above-mentioned ds:Reference is indicated by the "http://www.w3.org/2000/09/xmldsig#sha1" value.

[e181] is the start tag of the ds:DigestValue element.

[e182] provides the Base64 encoded message-digest value of the contents of the element(s) referenced by this ds:Reference URI.

[e183] is the closing tag of the ds:DigestValue element.

[e184] is the closing tag of the ds:Reference element. This is the last Reference included in this Manifest.

[e185] is the closing tag of the ds:Manifest element.

[e186] is the closing tag of the ds:Object element.

[e187] is a blank line for readability.

[e188] is the closing tag of the ds:Signature element.

[e189] is the closing tag of the NotarySignatures element.

[e190] - [e194] is a comment within the XML document.

[e195] is the closing tag of the NotarizedDocument element, signifying the end of this XML instance.

3.6 eNotarized document - Multiple Signers, Multiple Notaries & Symmetric Key Signatures

In this example of an eNotarized document, we show three document signers, two signing a document with NULL Cryptographic Signatures and the third signing with a symmetric key to create a Cryptographic Signature, separately; and two Notaries Public using different symmetric cryptographic keys to encrypt the message digest of the Signature's Manifest separately, thus creating two distinct Cryptographic Signatures. Please note the informative appendix on “ENML Security Implications” at the end of this document for an understanding of ENML security and symmetric encryption keys.

For the sake of brevity, many elements content are not repeated here, and only relevant content distinct from the previous example is shown below.

[f001] <enml:NotarizedDocument
[f002] xmlns:xsi= 'http://www.w3.org/2001/XMLSchema-instance'
[f003] xmlns:enml="http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901"
[f004] xsi:schemaLocation=
[f005] 'http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901.xsd'
[f006] Id="NotarizedDocument-US-CA-1565986-1226972451-8023243532a0e30067350d469c4e7f79729ad7da392d3a18974d5a2e7bd09ee4">
[f007] <!--
[f008] =============================================
[f009] Information about the SignedDocuments
[f010] =============================================

[f011]  <!--
[f012]  <enml:SignedDocuments>
[f013]  <enml:SignedDocument Id="SignedDocument-US-CA-1565986-1222646484-546e0248b7e7f52a4a27880d3eefb845577466e91556e0d37bb92fe1f2ad7fd0b">
[f014]  ...  
[f015]  </enml:SignedDocument>
[f016]  ...  
[f017]  </enml:SignedDocuments>
[f018]  <!--
[f019]  <enml:DocumentSigners>
[f020]  <!- - First Signer - - >
[f021]  <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099be66b1cb493dfdd9571c3edac5c23fa36aace2af6d2737d7">
[f022]  <enml:PersonFirstName>John</enml:PersonFirstName>
[f023]  <enml:PersonLastName>Doe</enml:PersonLastName>
[f024]  ...  
[f025]  </enml:DocumentSigner>
[f026]  ...  
[f027]  <!- - Second Signer - - >
[f028]  <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643179-c44f9fcca8950e9ad42959c725f2e31073f20b7058b777b2aa2d21f1830006dc">
[f029]  <enml:PersonFirstName>Jane</enml:PersonFirstName>
[f030]  <enml:PersonLastName>Doe</enml:PersonLastName>
[f031]  ...  
[f032]  </enml:DocumentSigner>
[f033]  ...  
[f034]  <!- - Third Signer - - >
[f035]  <enml:DocumentSigner Id="DocumentSigner-US-NJ-1234567-1223663478-849d55746eeec2d91189e0f7f02f374c0f5b53767ed99794b3a7e92a1abcfe6">
[f036]  <enml:SignedName>Jeff</enml:SignedName>
[f037]  <enml:PersonFirstName>Jeff</enml:PersonFirstName>
[f038]  <enml:PersonLastName>Doe</enml:PersonLastName>
[f039]  ...  
[f040]  </enml:DocumentSigner>
[f041]  ...  
[f042]  <enml:SignerIdentificationMethod>
[f043]  Personally Known to Credible Witnesses
[f044]  </enml:SignerIdentificationMethod>
[f045]  <enml:SignerSignature>
[f047]  </enml:SignerSignature>
[f048]  </enml:DocumentSigners>
Information about the two Notary Certificates

- - >
<enml:NotaryCertificates>
  
  <enml:NotaryCertificate Id="NotaryCertificate- US-CA-1565986-1222643229-1963a865f3516996f711936045ced816a0fccd4aca57c6b07a1c44f9b98a1e5f">

  ...

  </enml:NotaryCertificate>

- - >
<enml:NotaryCertificates>
  
  <enml:NotaryCertificate Id="NotaryCertificate- US-NJ-4665986-1223683212-c4e7f79729ad7da392d3a1f8023643532a0e3067350d4698974d5a2e7bd09e4">

  ...

  </enml:NotaryCertificate>

Information about the SECOND Notarial Certificate

- - >
<enml:NotaryCertificates>
  
  <enml:NotaryCertificate Id="NotaryCertificate- US-NJ-4665986-1223683212-c4e7f79729ad7da392d3a1f8023643532a0e3067350d4698974d5a2e7bd09e4">

  
  <enml:CertificateContent>
    
    <enml:NotarizationType>Acknowledgment</enml:NotarizationType>
    
    <enml:NotarizationDate>2008-02-12</enml:NotarizationDate>
    
    <enml:NotarizationUSLocation>
      
      <enml:City>Basking Ridge</enml:City>
      
      <enml:County>Middlesex</enml:County>
      
      <enml:USState>NJ</enml:USState>
      
      <enml:Country>USA</enml:Country>
      
    </enml:NotarizationUSLocation>
    
    <enml:StatutoryContent>
      
      State of New Jersey
      
      County of Middlesex
      
      On February 12 2008, before me Abraham Lincoln, Notary Public in and for said county, personally appeared Jeff Doe, who has satisfactorily identified himself as the signer or witness to the above-referenced document.
      

    </enml:StatutoryContent>

  </enml:CertificateContent>

  
  <!- - NotaryPublic who provided this certificate -- - >
  
  <enml:NotaryPublic>
    
    <enml:NotaryName>
      
      <enml:PersonGivenName>Abraham</enml:PersonGivenName>
      
      <enml:PersonSurName>Lincoln</enml:PersonSurName>
      
    </enml:NotaryName>
    
    <enml:NotaryCommissionNumber>
      
      NJ4665986
      
    </enml:NotaryCommissionNumber>
    
    <enml:NotaryCommissionExpiryDate>
      
      2009-04-29T23:59:59-08:00
      
    </enml:NotaryCommissionExpiryDate>
    
    <enml:NotaryUSJurisdiction>
      
      <enml:County>Middlesex</enml:County>
      
      <enml:USState>NJ</enml:USState>
      
      <enml:Country>USA</enml:Country>
      
    </enml:NotaryUSJurisdiction>
    
  </enml:NotaryPublic>

  </enml:NotaryCertificate>

</enml:NotaryCertificates>

...
<!DOCTYPE enml PUBLIC "-//OASIS//EN//ENML-1.0-Specification//EN//EN//ENML-1.0-Specification"

[enml:NotaryCertificates>

<enml:NotarySignatures>

<ds:Signature Id="Signature- US-CA-1565986-1222641290-e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855">

<ds:SignedInfo>

<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>

<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>


<ds:Transforms>

<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>

<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestValue>

MtGcQfervxf/b3xvrPEstt0h1fg=

</ds:DigestValue>

</ds:Transforms>

<ds:Reference URI="#SignedDocument-US-CA-1565986-1222646484-546e0248b7e7f52a4427880d3efb84557746e691556e0d37bb92f1f2ad7fd0b" />

<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestMethod/>

<ds:DigestValue>

W4DupJioi0mW7aG+N1qiNLtqvsk=

</ds:DigestValue>

</ds:Reference>

<ds:Reference URI="#DocumentSigner- US-CA-1565986-1222643107-f91b09f20bfa099beeb6b1c493fd99571c3edac5c23fa36aace2afd6d273fd7"/>

<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestMethod/>

<ds:DigestValue>

Yx5JkTS0ZxaM0uEpml/SxmSRSGAw=

</ds:DigestValue>

</ds:Reference>

</ds:Reference>

</ds:SignedInfo>

<ds:SignatureValue>

09fbf143326bb09beefabe8d84094ee14c5ee6e8c

</ds:SignatureValue>

<ds:KeyInfo>

<ds:KeyName>10514-1-123</ds:KeyName>

<ds:RetrievalMethod URI="http://skms.somecompany.com/symkeyServlet/getsymkey"/>

</ds:KeyInfo>

<ds:Object>

<ds:Manifest Id="Manifest- US-CA-1565986-1222641281"/>

<ds:Reference URI="#SignedDocument-US-CA-1565986-1222646484-546e0248b7e7f52a4427880d3efb84557746e691556e0d37bb92f1f2ad7fd0b" />

<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestMethod/>

<ds:DigestValue>

W4DupJioi0mW7aG+N1qiNLtqvsk=

</ds:DigestValue>

</ds:Reference>

<ds:Reference URI="#DocumentSigner- US-CA-1565986-1222643107-f91b09f20bfa099beeb6b1c493fd99571c3edac5c23fa36aace2afd6d273fd7"/>

<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestMethod/>

<ds:DigestValue>

Yx5JkTS0ZxaM0uEpml/SxmSRSGAw=

</ds:DigestValue>

</ds:Reference>

<ds:Reference URI="#DocumentSigner- US-CA-1565986-1222643179-ca44f9fcc8950e9ad42959c725f2e31073f20b7058b777b2aa2d21f1830006dc"/>

<ds:DigestMethod>
<ds:Signature Id="Signature- US-NJ-4665986-1223683785-897d00adf82aaa5ae07add9974407f62779807db73af12dcf6b3adf87e603">
    <ds:SignedInfo>
        <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
        <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
        <ds:Reference Type="http://www.w3.org/2000/09/xmldsig#Manifest"
            URI="#Manifest- US-NJ-4665986-1223683785">
            <ds:Transforms>
                <ds:Transform
                    Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </ds:Transforms>
            <ds:DigestMethod
                Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>
                xtGxfc78fervgfvb3xvrzAEstt0h1fg=
            </ds:DigestValue>
        </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
        c6fbf14dd26bb09beefabe8d84235ee14c5abc8c
    </ds:SignatureValue>
    <ds:KeyInfo>
        <ds:KeyName>10514-2-651</ds:KeyName>
        <ds:RetrievalMethod
            URI="http://skms.somecompany.com/symkeyServlet/getsymkey"/>
    </ds:KeyInfo>
    <ds:Object>
        <ds:Manifest Id="Manifest- US-NJ-4665986-1223683785">
            <ds:Reference URI="#SignedDocument-US-CA-1565986-1222646484-546e0248b7e7f52a4427880d3efb84557746e691556e0d37bb92fe1f2ad7fd0b">
                <ds:DigestMethod
                    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            </ds:Reference>
        </ds:Manifest>
    </ds:Object>
</ds:Signature>
<ds:DigestValue>

</ds:DigestValue>

</ds:Reference>

<ds:Reference URI="#DocumentSigner- US-NJ-1234567-1223663478-849d55746eec2d91189e0ff0f2ff374c0f5b33767ecdf7974b3a7e92a1abcfe6">

<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestValue>

</ds:DigestValue>

</ds:Reference>

<ds:Reference URI="#NotaryCertificate- US-NJ-4665986-1223683212-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4">

<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestValue>

</ds:DigestValue>

</ds:Reference>

</ds:Manifest>

</ds:Object>

</ds:Signature>

<!- -

============================================

End of XML document

============================================

- - >

</enml:NotarizedDocument>

[f001] – [f006] is the usual preamble to an XML instance document as discussed in the previous examples.

[f007] - [f011] is a comment within the XML document.

[f012] - [f041] is the SignedDocuments element, which indicates the documents that are being eNotarized. [f012] provides the identifier of the single SignedDocument in this section: "SignedDocument-US-CA-1565986-1222646484-546e0248b7e7f52a4427880d3e9b84557746e691556e0d37bb92e1f2ad7fd0b". This identifier is referenced in the two Notary signatures section of the eNotarized document.

[f042] - [f046] is a comment within the XML document.

[f047] is the start of the DocumentSigners element.

[f048] - [f060] represents information about the first DocumentSigner, in this case "John Doe" and carrying the following value in its ID attribute: “DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aaace2af6d6d275d47".

[f061] is a blank line for readability.


[f081] is a blank line for readability.
represents information about the third DocumentSigner, in this case “Jeff Doe”; this element has the following value in its ID attribute: “DocumentSigner-US-NJ-1234567-1223663478-849d55746eecc2d91189e0f7f02ff740cf5b53767eced9794b3a7e92a1abcfe6”. A notable fact about the third DocumentSigner element is that this signer does not use the NULL Cryptographic Signature for signing the document, but uses a Cryptographic Signature created by a symmetric key. While the SignerSignature element can contain any content, this example shows the use of some elements from the [XMLSignature] schema.

is a comment within the XML document.

[112] is the start of the NotaryCertificates element.

contains information about the first NotaryCertificate element, with the ID attribute value of: NotaryCertificate-US-CA-1565986-1222643229-1963a865f3516996f711936045ced816a0fcd4aca57c6807a1c44f9b98a1e5f This element refers to the Notary Public from the state of California in this example.

is a comment within the XML document.

contains information about the second NotaryCertificate element, with the ID attribute value of: NotaryCertificate-US-NJ-4665986-1223683212-c4e7ff79729ad7da392d3a18023243532a0e30067350d4698974d5a2e77b09ee4 This element refers to the notary public from the state of New Jersey where the second notarization of this document takes place.

signals the end of the NotaryCertificates element.

is a comment within the XML document.

is the start tag of the NotarySignatures element. This element contains information the notarial signatures of the two Notaries Public.

contains information about the first Signature element. This element, identified by the ID attribute value of "Signature-US-CA-1565986-1222641290-e3b0c44298fc1c149afbf4c8996f92427ae41e4649b934ca495991b7852b855", would have been created by the notary public in California. The Manifest it refers to on [261] is indicated by the URI value “Manifest-US-CA-1565986-1222641281”. This manifest is within the Object element between lines [282] and [309]. As can be noticed, the manifest refers to the single SignedDocument, the two signers from California and the notarial certificate created in California.

is a blank line for readability.

is a comment within the XML document.

contains information about the second Signature element. This element, identified by the ID attribute value of "Signature-US-NJ-4665986-1223683793-897d00ad82aa8aee7add997440f762779807db73af12d6b3ad0f87e603", would have been created by the notary public in New Jersey. The Manifest it refers to on [190] (indicating the elements of this document covered by the signature of the first notary public) is indicated by the URI value “Manifest-US-CA-1565986-1222641281”. This manifest is within the Object element between lines [282] and [309]. As can be noticed, the manifest refers to the same SignedDocument as for the first notary. However, only one DocumentSigner’s signature is referred to in this manifest, and secondly, the NotaryCertificate it refers to, is the one created in New Jersey. This implies that this Signature element covers only the second eNotarization act and does so without altering any of the content related to the first eNotarization performed by the California notary.

signals the end of the NotarySignatures element.

is a comment within the XML document.
3.7 eWitnessed document - Single Signer, Single Notary & Symmetric Key Signature

The ENML is capable of “witnessing” the signing of documents, in addition to notarizing them. An eWitnessed document is identical to an eNotarized document in all respects, save one: it does not contain a notarial certificate. The following example shows an eWitnessed document.

```xml
<enml:WitnessedDocument
    xmlns:xsi= 'http://www.w3.org/2001/XMLSchema-instance'
    xmlns:enml= "http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901"
    xsi:schemaLocation=
        'http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901.xsd'
    Id="WitnessedDocument-US-CA-1565986-1226972451-8023243532a0e3067350d469c4e7f779729ad7da392d3a18974d5a2e7bd09ee4">
    <!- -
    =============================================
    Information about the SignedDocuments
    =============================================
    - - >
    <enml:SignedDocuments>
        <enml:SignedDocument Id="SignedDocument-US-CA-1565986-1222646532-849d55746eece2d91189e0f7f02ff374c0f5b53767eddf7974b3ae7e92a1abcfe6">
            ...
        </enml:SignedDocument>
    </enml:SignedDocuments>
    <!- ->
    -------------------------------
    Information about the DocumentSigners
    -------------------------------------
    - - >
    <enml:DocumentSigners>
        <!- - Only Signer - - >
        <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643107-f91b09f208bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d273f7d">
            <enml:PersonFirstName>John</enml:PersonFirstName>
            <enml:PersonLastName>Doe</enml:PersonLastName>
        </enml:DocumentSigner>
    </enml:DocumentSigners>
    <!- ->
    --------------------
    Information about the Notary Public
    ----------------------------------
    - - >
    <enml:NotariesPublic>
        <enml:NotaryPublic Id="NotaryPublic-US-CA-1565986-1222643107-b269051b713ef56b17b3439e9db96c266e8dc14ae5449deecb3e0bb9b007ddd">
            <enml:NotaryName>
                <enml:PersonGivenName>Arshad</enml:PersonGivenName>
                <enml:PersonSurName>Noor</enml:PersonSurName>
            </enml:NotaryName>
            <enml:NotaryCommissionNumber>1565986</enml:NotaryCommissionNumber>
        </enml:NotaryPublic>
    </enml:NotariesPublic>
    </enml:WitnessedDocument>
```
<enml:NotaryCommissionExpiryDate>2009-04-29T23:59:59-08:00</enml:NotaryCommissionExpiryDate>
<enml:NotaryUSJurisdiction>
  <enml:County>Santa Clara</enml:County>
  <enml:USState>CA</enml:USState>
  <enml:Country>United States of America</enml:Country>
</enml:NotaryUSJurisdiction>
</enml:NotaryPublic>

<!- -
- - >
============================================
Information about the Notary Signature
============================================
- - >
<enml:NotarySignatures>
  <ds:Signature Id="Signature-US-CA-1565986-1222645055-5d2960331a078f80a8504dc6ae0273ef93ff0a4cd962f1c2e0bf4f9223bf98">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
      <ds:Reference URI="#SignedDocument-US-CA-1565986-122264532-849d55746eeec2d91189e0f7f0f2f374c0f5b53767edf9794b3a7e92a1abcfe6">
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>94973de9331f7446d15601328d5b990bf3048448</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dffe9571c3edac5c23fa36aace2af6d673f7d">
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>94973de9331f7446d15601328d5b990bf3048448</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#NotaryPublic-US-CA-1565986-1222643107-26905b173ef56b17b3439ecdbb96c266e8d1c4ae5449deccb3e06b9b007dca">
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>685ded60801600b0dd38d2e57ef3cd98ee5b58de5</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
  </ds:SignatureValue>
  <ds:KeyInfo>
    <ds:KeyName>Key-US-CA-1565986-1222646532-01</ds:KeyName>
    <ds:RetrievalMethod URI="http://krs.someserviceprovider.com"/>
  </ds:KeyInfo>
</enml:NotarySignatures>
is the usual preamble to an XML instance document as discussed in the previous examples.

[g007] - [g010] is a comment within the XML document.

[g012] - [g041] is the SignedDocuments element, which indicates the documents that are being eWitnessed. [g012] provides the identifier of the single SignedDocument in this section: “SignedDocument-US-CA-1565986-1222646532-849d55764ec2d9118e0f7f02ff374c05b53767ecdf9794b3ade9e2a1abcfe6”. This identifier is referenced in the Notary signature section of the eWitnessed document.

[g042] - [g046] is a comment within the XML document.

[g047] is the start of the DocumentSigners element.

[g048] is a comment about the only document signer in the document.

[g049] - [g060] represents information about the first DocumentSigner, in this case “John Doe” and carrying the following value in its ID attribute: “DocumentSigner-US-CA-1565986-1222643107-f91b0f20bfa099beb6b1cb493ddf9571c3edac5c23fa36aace2af6273dfe7”.

[g061] signals the end of the DocumentSigners element.

[g062] - [g066] is a comment within the XML document.

[g067] is the start of the NotariesPublic element.

[g068] - [g084] represents information about the Notary Public that witnessed the signing of this document.

[g085] is the end of the NotariesPublic element.

[g086] - [g090] is a comment within the XML document.

[g091] is the start of the NotarySignatures element.

[g092] is the start of the Signature element. This element and all its sub-elements conform to the [XMLSignature] schema. The ID attribute in this example carries the value “Signature-US-CA-1565986-1222645055-5d2962f1c2e0bf4f9223bf98”.

[g093] is the start tag of the ds:Signature element. This element provides information about what is signed within this eWitnessed document.

[g094] – [g095] describes the ds:CanonicalizationMethod element. To ensure that XML documents and their signatures are calculated consistently, software that complies with the [XMLSignature] standard “normalizes” the XML content based on the method specified in the Algorithm attribute of this element. In this example, the algorithm used to normalize the XML content is indicated by the “http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments” value.

[g096] - [g097] describes the ds:SignatureMethod element. Application generating and verifying signatures using the [XMLSignature] standard, need to know the method by which the XML document is signed. The Algorithm attribute of this element identifies this method. In this example, the algorithm
used to sign the XML content is the Hashed Message Authentication Code (HMAC) using the SHA1 method, as indicated by the "http://www.w3.org/2000/09/xmldsig#hmac-sha1" value.

[g098] is a blank line for readability.

[g099] – [g105] describes the first ds:Reference element, which identifies the document being signed as identified by the ID attribute: " SignedDocument-US-CA-1565986-1222646532-849d55746ee2d91189e0f7f02ff374c0f5b53767ecd9794b3a7e92a1abcfe6".

[g106] is a blank line for readability.

[g107] – [g113] describes the second ds:Reference element, which identifies the signer of the document as identified by the ID attribute: "DocumentSigner-US-CA-1565986-1222643107-f91b09f20ba099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d273fd7".

[g114] is a blank line for readability.

[g115] – [g121] describes the third and last ds:Reference element, which identifies the Notary Public that witnessed the signing of the document, as identified by the ID attribute: "NotaryPublic-US-CA-1565986-1222643107-b269051b713ef56b17b3439ecb6c266e8d1c4ae5449deccb3e0bb9b007ddd".

[g122] is the closing tag of the ds:SignedInfo element.

[g123] is a blank line for readability.

[g124] – [g126] describes the ds:SignatureValue element. This element is the container for the electronic signature of the Notary Public on this instance of the eWitnessed document.

[g127] is a blank line for readability.

[g128] is the start tag of the ds:KeyInfo element. This element is the container for the cryptographic symmetric-key, or information about the cryptographic key, used by the Notary Public to sign this instance of the eNotarized document.

[g129] defines the ds:KeyName element. The value in this element is, typically, an identifier to a cryptographic key that may be accessed at some location.

[g130] defines the ds:RetrievalMethod element. The URI attribute in this element identifies a location where the cryptographic key referenced in ds:KeyName can be accessed.

[g131] is the closing tag of the ds:KeyInfo element.

[g132] is the closing tag of the ds:Signature element.

[g133] is the closing tag of the NotarySignatures element.

[g134] - [g138] is a comment within the XML document.

[g139] is the closing tag of the WitnessedDocument element.

3.8 eNotarized document - Single Signer, Single Notary & Digital Signature

The following example shows an eNotarized document, with a true digital signature generated by a Private Key corresponding to the digital certificate of a Notary Public. The document's signer is assumed to have signed the document with a NULL Cryptographic Signature. It is possible for the document signer to have signed with a cryptographic key too, but this example assumes a NULL Cryptographic Signature for the document's signer and a digital signature for the Notary Public.
Please note the informative appendix on “ENML Security Implications” at the end of this document for an understanding of ENML security and asymmetric cryptographic keys and X.509-compliant digital certificates.

```xml
<enml:NotarizedDocument
   xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns:enml='http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901'
   xsi:schemaLocation='http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901.xsd'
   Id='NotarizedDocument-US-CA-1565986-1226972451-
   8023243532a030067350d469c4e7f79729ad7da392d3a18974d5a2e7bd09ee4'>

   <!--
   Information about the SignedDocuments
   -->
   <enml:SignedDocuments>
      <enml:SignedDocument Id='SignedDocument-US-CA-1565986-1222646651-
      c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4'>

      ...

   </enml:SignedDocument>

   ...

   </enml:SignedDocuments>

   <!--
   Information about the DocumentSigners
   -->
   <enml:DocumentSigners>
      <!-- Only Signer -->
      <enml:DocumentSigner Id='DocumentSigner-US-CA-1565986-1222643107-
      91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d6d737d7'>

      <enml:PersonFirstName>John</enml:PersonFirstName>
      <enml:PersonLastName>Doe</enml:PersonLastName>

      ...

   </enml:DocumentSigner>

   ...

   </enml:DocumentSigners>

   <!--
   Information about the NotaryCertificates
   -->
   <enml:NotaryCertificates>
      <enml:NotaryCertificate Id='NotaryCertificate-US-NJ-4665986-
      1223683212-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4'>

      ...

   </enml:NotaryCertificate>

   ...

   </enml:NotaryCertificates>

   <!--
   Information about the NotarySignatures
   -->
   <enml:NotarySignatures>

      <ds:Signature Id='Signature-US-CA-1565986-1222649290-
      e3b0c44298fc1c149afbf14c8996fb92427ae41e4649b934ca495991b7852b855'>

      <ds:SignedInfo>

         ...

      </ds:SignedInfo>

   </enml:NotarySignatures>

   ...

</enml:NotarizedDocument>
```

ENML-1.0-Specification
Copyright © OASIS® 2009. All Rights Reserved.
Page 52 of 195
<ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
<ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<!- -  The DOCUMENT  - - >
<ds:Reference URI="SignedDocument-US-CA-1565986-1222646651-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4">
<ds:Transforms>
<ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
</ds:Transforms>
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>3b3Bd7TGR/niVS+d3WbnLReFK+g="</ds:DigestValue>
</ds:Reference>

<!- -  The SIGNER  - - >
<ds:Reference URI="#DocumentSigner- US-CA-1565986-1222643107-f91b09720bfa099beb6b1cb493df6d9571c3edac5c23fa36aace2a0f62d736d7">
<ds:Transforms>
<ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
</ds:Transforms>
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>275ded60801633b0dd38d2e57ef3cd98eeb58de5</ds:DigestValue>
</ds:Reference>

<!- -  The NOTARIAL CERTIFICATE  - - >
<ds:Reference URI="#NotaryCertificate- US-CA-1565986-1222643229-1963a865f3516996f711936045ced816a0fcccdaaca57c6807a1c44f9b98a1e5f">
<ds:Transforms>
<ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
</ds:Transforms>
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>275ded60801633b0dd38d2e57ef3cd98eeb58de5</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>

<!- -  The SIGNATURE VALUE  - - >
<ds:SignatureValue>
RVlpQofvWPKZkm3mlnL1mQN5oxagmcINu6MuAKA71qa/
Nhlu9i1N6TnAN7EwN07TrFpc5CBFbje4M7sbcddzzF1A==
</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<!- -  The DIGITAL CERTIFICATE  - - >
<ds:X509Certificate>
MIIDfDCCAmSgAwIBAgIlaIeAv/1Gi3AwDQYJKoZIhvcnaQELBQAwZzEmMCQGA1UEAxMxDUI3Ryb25n
S2V51EFTU8gU3Vib3Jkaw5hdG9yQDEEXJDAiBgsNBAsTGOZvc1BTJvXVbmdLZk REVNTy BVc2Ujg
T25seTEYMBQGA1UECwNWDWpRb25nQXVoACBjJbmvWWcMhNcMVwNz1lMTcweMDMwMh
WNCwMzI1MTcyc
</ds:X509Certificate>
contains XML elements that have already been described in previous examples.

has the first real difference; the SignatureMethod element in this example uses an algorithm not previously presented in these examples. Specifically, it uses the asymmetric-key based RSA cryptographic algorithm with the SHA-1 message digest algorithm for generating the digital signature, as denoted by the URL: http://www.w3.org/2000/09/xmldsig#rsa-sha1.

is a blank line for readability.

identifies the first Reference element that is covered under the Notary's signature in this document. In this example, the Reference points to the SignedDocument instance denoted by the identifier: SignedDocument-US-CA-1565986-1222650676-2f4d9df83bb1e50d5b0fd4e4797082c151a2c7ff6985a28f5f10d6c52e7bd03d44b2df59f1f7e19636b2b52c1f53d91f7398c87866c9e350d

is a blank line for readability.

identifies the second Reference element covered by the Notary's signature in this document. In this example, the Reference points to the DocumentSigner instance denoted by the identifier: DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099b6b1cb493dfdd9571c3edac5c23fa36aace2af6d273fd7.

is a blank line for readability.

identifies the final Reference element covered by the Notary's signature. In this example, the Reference points to the NotaryCertificate denoted by the identifier: NotaryCertificate-US-CA-1565986-1222643229-1963a865f35169667f11936045ced816a0fccd4aca57c6807a1c4f9f98a1e5f.

signals the end of the SignedInfo element which identifies the information signed in this document.

is a blank line for readability.

defines the actual SignatureValue of the digital signature in this document.
is the start of the **KeyInfo** element which provides information about the cryptographic key related to the cryptographic signature in this document.

is the start of the International Standards Organization's (ISO) X.509-related data about the digital certificate related to the cryptographic signature in this document.

- is a blank line and a comment for better readability of the XML data.

contains the X.509 Base64-encoded digital certificate which must be used to verify the digital signature of this document.

contain the end-tags of the element-hierarchy in this document, which signifies the end of this XML eNotarized document.

### 3.9 eNotarized document - Multiple Signers, Multiple Notaries & Digital Signatures

The following example differs from the previous one in that is show multiple signers for a single document, and multiple Notaries Public signing the document with digital signatures generated by Private Keys corresponding to their digital certificates.

Please note the informative appendix on “ENML Security Implications” at the end of this document for an understanding of ENML security and asymmetric cryptographic keys and X.509-compliant digital certificates.

```
<enml:NotarizedDocument
 xmlns:xsi= 'http://www.w3.org/2001/XMLSchema-instance'
 xmlns:enml= "http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901"
 xsi:schemaLocation="http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901.xsd"
 Id="NotarizedDocument-US-CA-1565986-1226972451-8023243532a0e30067350d469c4e7f79729ad7da392d3a18974d5a2e7bd09ee4">
    <!- -
    =============================================
    Information about the SignedDocuments
    =============================================
    - - >
    <enml:SignedDocuments>
    <enml:SignedDocument Id="SignedDocument-US-CA-1565986-1222646651-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4">
       ...
    </enml:SignedDocument>
    ...
    </enml:SignedDocuments>
    <!-
    Information about the DocumentSigners
    ----------------------------------------
    - - >
    <enml:DocumentSigners>
    <!- -  First Signer - - >
    <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643107-f91b09f20ba099be6b1cb493d9d9571c3edac5c23fa36aace2af6d6d273fd7">
       <enml:PersonFirstName>John</enml:PersonFirstName>
       <enml:PersonLastName>Doe</enml:PersonLastName>
       ...
    </enml:DocumentSigner>
    ...
    </enml:DocumentSigners>
    </enml:NotarizedDocument>
```
[001] – [041] contains the usual preamble and SignedDocuments element that have already been described in previous examples.

[042] – [1067] contains the two DocumentSigner elements, one for the signer in California and the other for the signer in New Jersey. Each element is expected to have been added independent of each other to the eNotarized document.

[1068] – [1095] contains the two NotaryCertificate elements. In this example, once again, each notarization is expected to have been done independent of the other by a distinct Notary Public.

ENML-1.0-Specification
Copyright © OASIS® 2009. All Rights Reserved.
6 February 2009
Page 57 of 195
contains the two NotarySignature elements. Each Notary is expected to have their own X.509 digital certificate and corresponding private key to perform the signing. Each individual verification digital certificate is available in the distinct KeyInfo elements – the first at [i151]-[i152] and the second at [i187]-[i188].

signals the end of this eNotarized document.

3.10 eWitnessed document - Multiple Signers, Single Notary & Digital Signature

As was shown earlier in the example of an eWitnessed document with symmetric-key based cryptographic signature, the ENML is also capable of eWitnessing the eNotarization of documents using X.509 digital certificates and corresponding private keys. (An eWitnessed document is identical to an eNotarized document in all respects, save one: it does not contain a notarial certificate). The following example shows an eWitnessed document with a NULL Cryptographic Signature created by the document's signer and a digital signature created by the Notary Public.

```xml
<enml:WitnessedDocument
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:enml='http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901'
  xsi:schemaLocation='http://docs.oasis-open.org/legalxml-enotary/ns/enml-200901.xsd'
  Id="WitnessedDocument-US-CA-1565986-1226972451-8023243532a0e30067350d469c4e7f797729ad7da392d3a18974d5a2e7bd9ee4">
  <!- -
  Information about the SignedDocuments
  - - >
  <enml:SignedDocuments>
    <enml:SignedDocument Id="SignedDocument-US-CA-1565986-1222646532-849d55746eecc2d9189e0f7f02ff374c0f5b53767ecdf9794b3a7e92a1abcfe6">
      ...
      ...
    </enml:SignedDocument>
  </enml:SignedDocuments>
  <!- -
  Information about the DocumentSigners
  - - >
  <enml:DocumentSigners>
    <!- - Only Signer - - >
    <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dffdd9571c3edac5c23fa36aace2af6d273fd7">
      <enml:PersonFirstName>John</enml:PersonFirstName>
      <enml:PersonLastName>Doe</enml:PersonLastName>
      ...
      ...
    </enml:DocumentSigner>
  </enml:DocumentSigners>
  <!- -
  Information about the Notary Public
  - - >
  <enml:NotariesPublic>
    ...
  </enml:NotariesPublic>
</enml:WitnessedDocument>
```
All the elements in this example have been introduced in earlier examples, so they are not repeated. The only distinction of note from the eWitnessed document presented in Section 3.4 is that this example shows the use of the RSA with SHA1 cryptographic algorithm in line [j096] and the X.509 digital certificate that can be used for verifying the digital signature is shown in lines [j112] – [j113].
3.11 eApostillized document with a Digital Signature

In international transactions, it is sometimes necessary for a notarized document to be certified by the competent authority that has jurisdiction over notaries public, and certify that the signature of the public official in the document is a valid signature. In conformance with the Convention at The Hague in 1961 [HCCH], such a certified document is called an Apostille. (A model of the certificate is shown below):

Model of certificate
The certificate will be in the form of a square with sides at least 9 centimetres long

```
APOSTILLE
(Convention de La Haye du 5 octobre 1961)

1. Country: ..................
   This public document

2. has been signed by ..................

3. acting in the capacity of ..................

4. bears the seal/stamp of ..................

Certified

5. at ................  6. the ..................

7. by ..................

8. N°..................

9. Seal/stamp:  10. Signature: 

................  ................
```

Given that ENML defines a markup language for electronically notarizing electronic documents, it is a natural consequence that ENML-based documents may need to be apostillized in the future. Such an
"eApostillized" document must meet the data-element requirements of the 1961 Convention of the Hague, and match the model certificate shown above.

The ENML specification supports this feature, as is shown in the following example of a NotarizedDocument with a digital signature of the notary public, certified by the Secretary of State in California, whose own digital certificate is shown in the ApostillizedDocument. The data-elements of the model-certificate are contained in the ApostilleContent element.

While this is an XML document, readers must remember that software applications that implement ENML will be able to parse the XML, verify its contents cryptographically, and then display the apostille in the visual convention required by [HCCH].
Information about the NotarySignatures

<!- -

============================================= 

Information about the NotarySignatures 

============================================= 

- - >

<enml:NotarySignatures>

-->

<ds:Signature Id="Signature-US-CA-1565986-1222649290-e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855" >

<!- - The DOCUMENT -->
<ds:Reference URI="SignedDocument-US-CA-1565986-1222646651-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4" >

<!- - The SIGNER -->
<ds:Reference URI="#DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c36d7e23fa36aace2af36d273f1d7" >

<!- - The NOTARIAL CERTIFICATE -->
<ds:Reference URI="#NotaryCertificate-US-CA-1565986-1222643229-1963a865f3516996f711936045ced816a0fecd4aca57c6807a1c44f9b98a1f" >

<!- - The SIGNATURE VALUE-->
<ds:SignatureValue>

RVlpQofvWPK2mhmlLmQM5oxgmc1Nu86MuAKA71gg/
Nkhlu9ilN6TnAN7EwxtNTrFpc5CBFbje4m7sbcdzzF1A==

</ds:SignatureValue>

<ds:KeyInfo>

<ds:X509Data>

<!- - The NOTARY'S DIGITAL CERTIFICATE -->
<ds:X509Certificate>

...</ds:X509Certificate>
</ds:X509Data>

</ds:KeyInfo>

<!- - The X509Certificate -->
<ds:X509Data>

</ds:X509Certificate>
</ds:KeyInfo>

<!- - The Signature -->
<ds:Signature>

</ds:Signature>
</enml:NotarySignatures>

<!- -

============================================= 

End of the NotarizedDocument 

============================================= 

- - >

<enml:NotarizedDocument>

-->

<enml:NotarizedDocument>

-->

<ds:NotaryCertificate_Id="ApostilleContent-US-CA-1565986-1226972923-89e89c11cac335b3d7c1030778fca5527cbd44cc61479fc88526c38e01d912a5" >

<enml:IssuingCountry>United States of America</enml:IssuingCountry>
<enml:ApostilleNumber>
- [k001] - [k157] contains XML elements that have already been described in previous examples for digital signature based eNotarized documents (see Sections 3.8 and 3.9). The only exception is that [k001] identifies this document as being an ApostillizedDocument with an ID attribute, on line [k006], that has a value of “ApostillizedDocument-US-CA-SecretaryOfState-1226972584-c4e7f79729ad7da392d3a1802343323a0e3066735d046897475a27e0bd79ee4”. This value is unique for this apostille.

- [k158] - [k162] is an XML comment for better readability of the XML.

- [k163] identifies the start of the ApostilleContent element with an ID attribute value of “ApostilleContent-US-CA-1565986-1226972923-89e89c11cacc355b3d7c10307f85ca5527cbdc46d1479fc85622638e01d912a5”.

- [k164] contains the element IssuingCountry which identifies the country issuing this eApostille; this example contains the value “United States of America”.

---

ENML-1.0-Specification
Copyright © OASIS® 2009. All Rights Reserved.
Page 64 of 195

6 February 2009
contains the **ApostilleNumber** element which identifies unique number of this eApostille within the country issuing it. In this example, the number is "ApostilleNumber-US-CA-1565986-1226971821-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4". This number is expected to be unique for each eApostille issued by the competent authority in this jurisdiction.

contains the element **ApostilleDate** which identifies the date, time and time-zone this eApostille was issued; this example contains the value "2008-11-17T12:13:14-08:00" indicates November 17, 2008 at 12:13:14PM in the Pacific time-zone (-08:00 hours from UTC).

contains the **IssuedAtUSLocation** element which identifies geographic location this eApostille was issued. In this example, it is in the county of "Sacramento", state of "California" in the "USA".

contains the **SignerTitle** element which identifies official title of the signer of this eApostille. In this example, it is the "Secretary of State" in the "State of California".

contains the **StatutoryContent** element for an **ApostillizedDocument**, in accordance with [HCCH]. This value is always "(Convention de La Haye du 5 octobre 1961)".

contains the **DocumentsigningOfficialName** element, which identifies the full name of the official (the notary public) who signed the document for which this eApostille is being issued. In this example, the **NotarizedDocument** is presumed to have been signed by the notary public, Arshad Noor, in the **NotaryCertificate**, it appears again in this element.

contains the **DocumentsigningOfficialTitle** element, which identifies the official title of the person who signed the **NotarizedDocument**. In this example, the official title is "Notary Public".

contains the **DocumentsigningOfficialSeal** element, which identifies the official seal of the person who signed the **NotarizedDocument**. In this example, the seal defined in this ENML is of a text-string with the words: "Notary Public, State of California".

signals the end of the **ApostilleContent** element.

represents the digital signature of the person signing the eApostille. The elements have been previously described in Sections 3.8 and 3.9, so they will not be repeated. It is important to note that the apostille signer's signature refers to **NotarizedDocument** (in line [k202]) and the **ApostilleContent** (in line [k211]) elements referenced within this **ApostillizedDocument**, thus binding the apostille-signer's signature to the referenced **NotarizedDocument** and **ApostilleContent**.

contain the comment and end-tags of the element-hierarchy in this document, which signifies the end of this eApostillized document.

**3.12 The use of XPath in <ds:Signature>**

This specification does not explicitly show the use of XPath transforms in the any of the examples in this section. This is simply because it has not been tested by the creators of this specification. Implementers of ENML are encouraged to use XPath transforms with caution and to verify all test cases (as given in this non-normative section) before releasing their software. The OASIS LegalXML eNotary Technical Committee will be grateful to implementers for any information they provide with respect to their experience in using XPath in their implementation.
4 Specification

4.1 Element <NotarizedDocument>

The `<NotarizedDocument>` element is the *raison d'être* of the ENML protocol. It defines the standard for an electronically notarized document, and contains the required elements, under US law, for such a document.

The `<NotarizedDocument>` element may be represented, graphically, as follows:

![Diagram of NotarizedDocument element]

**Schema Definition:**

```xml
<xsd:element name="NotarizedDocument" type="enml:NotarizedDocumentType"/>

<xsd:complexType name="NotarizedDocumentType">
  <xsd:sequence>
    <xsd:element name="SignedDocuments" type="enml:SignedDocumentsType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="DocumentSigners" type="enml:DocumentSignersType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="NotaryCertificates" type="enml:NotaryCertificatesType" minOccurs="1" />
    <xsd:element name="NotarySignatures" type="enml:NotarySignaturesType" />
  </xsd:sequence>
</xsd:complexType>
```
The `<NotarizedDocument>` element, is of type `NotarizedDocumentType` defined in the ENML schema, and consists of a sequence of four children elements and one attribute, all of which are required:

1. `<SignedDocuments>` [Required]

   This element of type `SignedDocumentsType`, identifies one or more electronic documents being notarized within an instance of a `<NotarizedDocument>`. There MUST be exactly ONE `<SignedDocuments>` element in a `<NotarizedDocument>` element.

   (Note that US law, currently allows only for a single document to be notarized within a unique `<NotarizedDocument>`. While ENML has the ability to include multiple documents within the `<SignedDocuments>` element, software implementations of ENML are expected to comply with legal requirements of the jurisdiction in which they expect the software to be operated).

   The `<SignedDocuments>` element is specified in Section 4.4.

2. `<DocumentSigners>` [Required]

   This element of type `DocumentSignersType`, identifies one or more signers of the electronic documents being notarized within an instance of a `<NotarizedDocument>`. The document signers are distinct from the Notary Public who will also sign this document, but whose signature will be carried in the `<NotarySignatures>` (discussed later). There MUST be exactly ONE `<DocumentSigners>` element in a `<NotarizedDocument>` element.

   The `<DocumentSigners>` element is specified in Section 4.5.

3. `<NotaryCertificates>` [Required]

   This element of type `NotaryCertificatesType`, identifies one or more notarial certificates created by the Notary Public, and being notarized within an instance of a `<NotarizedDocument>`. There MUST be exactly ONE `<NotaryCertificates>` element in a `<NotarizedDocument>` element.

   The `<NotaryCertificates>` element is specified in Section 4.6.

4. `<NotarySignatures>` [Required]

   This element of type `NotarySignaturesType`, identifies one or more electronic signatures of the Notary Public notarizing an instance of a `<NotarizedDocument>`. There MUST be exactly ONE `<NotarySignatures>` element in a `<NotarizedDocument>` element.

   The `<NotarySignatures>` element is specified in Section 4.7.

5. ID attribute [Required]

   The ID (identifier) attribute must be used to identify this instance of the `<NotarizedDocument>`
uniquely from a collection of notarized documents within some other application's XML. It is required when a `<NotarizedDocument>` is certified within an `<ApostillizedDocument>` (see Section 4.3).

Please see Section 4.56 for a discussion of the Processing Rule for ID attributes within ENML.

Many examples of the `<NotarizedDocument>` element were provided in the non-normative Section 3 of this specification document. Please refer to that section for examples of the `<NotarizedDocument>` element.

### 4.2 Element `<WitnessedDocument>`

The `<WitnessedDocument>` element is the second-most important element of the ENML. Almost identical to the `<NotarizedDocument>`, it differs from the `<NotarizedDocument>` in that it does not carry a `<NotaryCertificates>` element, but carries a `<NotariesPublic>` element in its stead.

The `<WitnessedDocument>` element may be represented, graphically, as follows:

```
WitnessedDocument
  SignedDocuments
  DocumentSigners
  NotariesPublic
  NotarySignatures
  ID
```

**Schema Definition:**

```xml
<xsd:element name="WitnessedDocument" type="enml:WitnessedDocumentType"/>

<xsd:complexType name="WitnessedDocumentType">
  <xsd:sequence>
    <xsd:element name="SignedDocuments" type="enml:SignedDocumentsType" minOccurs="1" maxOccurs="1"/>
  </xsd:element>
  <xsd:element name="DocumentSigners" type="enml:DocumentSignersType" minOccurs="1"/>
  <xsd:element name="NotariesPublic" type="enml:NotariesPublicType" minOccurs="1"/>
  <xsd:element name="NotarySignatures" type="enml:NotarySignaturesType" minOccurs="1"/>
</xsd:complexType>
```

The `<WitnessedDocument>` element is of type `WitnessedDocumentType` defined in the ENML schema, and consists of a sequence of four children elements and one attribute, all of which are required:

1. `<SignedDocuments>` [Required]

   This element of type `SignedDocumentsType`, identifies one or more electronic documents whose signing is being witnessed within an instance of a `<WitnessedDocument>`. There MUST be exactly ONE `<SignedDocuments>` element in a `<WitnessedDocument>` element.

   (Note that US law, currently allows only for a single document to be witnessed within a unique `<WitnessedDocument>`. While ENML has the ability to include multiple documents within the `<SignedDocuments>` element, software implementations of ENML are expected to comply with legal requirements of the jurisdiction in which they expect the software to be operated).

   The `<SignedDocuments>` element is specified in Section 4.4.

2. `<DocumentSigners>` [Required]

   This element of type `DocumentSignersType`, identifies one or more signers of the electronic documents being notarized within an instance of a `<WitnessedDocument>`. The document signers are distinct from the Notary Public who will also sign this document, but whose signature will be carried in the `<NotarySignatures>` (discussed later). There MUST be exactly ONE `<DocumentSigners>` element in a `<WitnessedDocument>` element.

   The `<DocumentSigners>` element is specified in Section 4.5.

3. `<NotariesPublic>` [Required]
The <NotariesPublic> element is a container-element, whose purpose is to organize a collection of <NotaryPublic> elements inside one element. The <NotaryPublic> element, which is of NotaryPublicType, contains information about the Notary Public who witnessed the signing of the documents contained within the above-mentioned <SignedDocuments> element.

There MUST be exactly ONE <NotariesPublic> element in a <WitnessedDocument> element. The <NotariesPublic> element MUST contain at least ONE <NotaryPublic> element, but may contain an unbounded (unlimited) number of <NotaryPublic> elements as needed.

The <NotaryPublic> element is specified in Section 4.8.

4. <NotarySignatures> [Required]

This element of type NotarySignaturesType, identifies one or more electronic signatures of the Notary Public notarizing an instance of a <WitnessedDocument>. There MUST be exactly ONE <NotarySignatures> element in a <WitnessedDocument> element.

The <NotarySignatures> element is specified in Section 4.7.

5. ID attribute [Required]

The ID (identifier) attribute must be used to identify this instance of the <WitnessedDocument> uniquely from a collection of witnessed documents within some other application's XML. It is required when a <WitnessedDocument> is certified within an <ApostillizedDocument> (see Section 4.3).

Please see Section 4.56 for a discussion of the Processing Rule for ID attributes within ENML.

Two examples of the <WitnessedDocument> element were provided in the non-normative Section 3.4 and Section 3.10 of this specification document. Please refer to those sections for examples of the <WitnessedDocument> element.

4.3 Element <ApostillizedDocument>

The <ApostillizedDocument> element is used to certify the signature of the Notary Public who signed the <NotarizedDocument> or the <WitnessedDocument> element, in accordance with conventions defined in [HCCH].

The <ApostillizedDocument> element may be represented, graphically, as follows:
<xsd:element name="ApostillizedDocument" type="enml:ApostillizedDocumentType"/>

<xsd:complexType name="ApostillizedDocumentType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="NotarizedDocument" type="enml:NotarizedDocumentType" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="WitnessedDocument" type="enml:WitnessedDocumentType" minOccurs="1" maxOccurs="1"/>
    </xsd:choice>
    <xsd:element name="ApostilleContent" type="enml:ApostilleContentType" minOccurs="1" maxOccurs="1"/>
    <xsd:choice>
      <xsd:element ref="ds:Signature" minOccurs="1" maxOccurs="1"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
The `<ApostillizedDocument>` element is of type `<ApostillizedDocumentType>` defined in the ENML schema, and consists of a sequence of three children elements and one attribute, all of which are required:

1. `<NotarizedDocument>` or `<WitnessedDocument>` [Required]

   The first element is a choice of either a `<NotarizedDocument>` or a `<WitnessedDocument>` element. This is the document, whose Notary's signature is being certified by the signer of the apostille.

   There MUST be exactly ONE `<NotarizedDocument>` or `<WitnessedDocument>` element in an `<ApostillizedDocument>` element.

   The `<NotarizedDocument>` element is specified in Section 4.1.

   The `<WitnessedDocument>` element is specified in Section 4.2.

2. `<ApostilleContent>` [Required]

   This element of type `<ApostilleContentType>` identifies details of the electronic apostille within an instance of an `<ApostillizedDocument>`. There MUST be exactly ONE `<ApostilleContent>` element in a `<ApostillizedDocument>` element.

   The `<ApostilleContent>` element is specified in Section 4.47.

3. `<ds:Signature>` or `<ApostilleSignature>` [Required]

   The next element represents the electronic signature of the apostille-signer. However, depending on the type of technology chosen, this element must be a choice of either the `<ds:Signature>` or the `<ApostilleSignature>` element. It is the responsibility of the application generating the ENML-marked document to choose the appropriate element.

   There MUST be exactly ONE `<ds:Signature>` or `<ApostilleSignature>` element in a `<ApostillizedDocument>` element.

   This element of type `<ds:SignatureType>` is defined in the [XMLSignature] schema. It provides details about the electronic signature of the document-signer who signed the electronic document.

   The `<ds:Signature>` element is a versatile element that allows for electronically signing documents with a variety of technologies, depicted in the XML. It supports the use of symmetric, as well as asymmetric, cryptographic keys for the generation of electronic signatures, and a variety of cryptographic algorithms to provide most implementers with the flexibility to choose a signing mechanism of their choice (as long as they satisfy the semantic meanings of the elements within the `<ds:Signature>` element).
The `<ds:Signature>` element is specified in [XMLSignature].

Please see Section 3.12 for a note on the use of XPath in `<ds:Signature>` elements within ENML.

The `<Apostille Signature>` element of type `xsd:anyType` are for those situations where the software implementation of ENML chooses NOT to use any specific signing technology, and represents the apostille-signer's signature with simple text (or similar representations).

As the type implies, this element does not mandate what content may be placed in it, but it is strongly recommended that implementers use meaningful text (such as “Signed by John Doe”, or “/S=John Doe” or similar well-known and accepted norms of electronic signatures) to represent the signer's signature.

Please note the informative appendix on “ENML Security Implications” at the end of this document for an understanding of security issues within ENML.

4. ID attribute [Required]

The ID (identifier) attribute must be used to identify this instance of the `<Apostillized Document>` uniquely from a collection of apostillized documents within some other application's XML.

Please see Section 4.56 for a discussion of the Processing Rule for ID attributes within ENML.

An example of the `<Apostillized Document>` element was provided in the non-normative Section 3.11 of this specification document. Please refer to that section for the example.

4.4 Element `<SignedDocuments>` & `<SignedDocument>`

The `<SignedDocuments>` element carries information about the business document that was notarized or whose signing was witnessed by a Notary Public and appears only within a `<NotarizedDocument>` or `<WitnessedDocument>` element. It is a container-element, whose purpose is to organize a collection of `<SignedDocument>` elements inside itself.

The `<SignedDocument>` element is the actual element that carries information about the electronic document signed by the document-signers and notarized (or whose signing was witnessed) by the Notary Public.

Note: Current US law does not permit notarizing or witnessing more than one document per notarization or witnessing. If the law changes to permit the electronic notarization or witnessing of multiple electronic documents within a single notarization or witnessing act, the ENML schema does not need to change. Application developers who implement ENML must take legal requirements into account when developing their software.
Schema Definition:

```xml
<xsd:element
    name="SignedDocuments"
    type="enml:SignedDocumentsType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:complexType name="SignedDocumentsType">
    <xsd:sequence>
        <xsd:element
            name="SignedDocument"
            type="enml:SignedDocumentType"
            minOccurs="1"
            maxOccurs="unbounded">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SignedDocumentType">
    <xsd:sequence>
        <xsd:element
            name="Document"
            type="enml:DocumentType"
            minOccurs="1"
            maxOccurs="1">
        </xsd:element>
        <xsd:element
            name="DocumentMIMETYPE"
            type="enml:SignedDocumentMIMETYPE"
            minOccurs="1"
            maxOccurs="1">
        </xsd:element>
        <xsd:element
            name="DocumentMIMETYPEComments"
            type="enml:SignedDocumentMIMETYPEComments"
            minOccurs="1"
            maxOccurs="1">
        </xsd:element>
        <xsd:element
            name="DocumentComments"
            type="enml:DocumentComments"
            minOccurs="1"
            maxOccurs="1">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
```
The `<SignedDocuments>` element is of type `SignedDocumentsType` defined in the ENML schema, while the `<SignedDocument>` element is of the `SignedDocumentType`.

There MUST be exactly ONE `<SignedDocuments>` element in either a `<NotarizedDocument>` or `<WitnessedDocument>` element. The `<SignedDocuments>` element MUST contain at least ONE `<SignedDocument>` element, but may contain an unbounded (unlimited) number of `<SignedDocument>` elements as needed.

The `<SignedDocument>` element of type `SignedDocumentType`, consists of a sequence of four children elements and one ID attribute:

1. `<Document>` [Required]

   This element of type `DocumentType`, identifies a single electronic document being notarized within an instance of a `<WitnessedDocument>`. There MUST be exactly ONE `<Document>` element in a `<SignedDocument>` element.

   The `<Document>` element is specified in Section 4.9.

2. `<DocumentMIMEType>` [Required]

   This element of type `SignedDocumentMIMEType`, identifies the type of electronic document being notarized or witnessed. There MUST be exactly ONE `<DocumentMIMEType>` element in a `<SignedDocument>` element.

   The `<DocumentMIMEType>` element is specified in Section 4.10.

3. `<DocumentMIMETypeComments>` [Optional]
The optional `<DocumentMIMETypeComments>` element of type `xsd:String` allows applications creating the ENML-marked document, to provide hints to receiving applications about what application was used to create the ENML and what other applications are/maybe compatible with the document-MIME type.

When present, There MUST be exactly ONE `<DocumentMIMETypeComments>` element in a `<SignedDocument>` element. Its value may not be more than 1,024-characters in length including white-space, which is preserved as-is.

The `<DocumentMIMETypeComments>` element is specified in Section 4.11.

4. `<DocumentComments>` [Optional]

The optional `<DocumentComments>` element of type `xsd:String` allows applications creating the ENML-marked document, to provide a comment to receiving applications and/or users about the embedded document itself.

When present, there MUST be exactly ONE `<DocumentComments>` element in a `<SignedDocument>` element. Its value may not be more than 1,024-characters in length including white-space, which is preserved as-is.

The `<DocumentComments>` element is specified in Section 4.11.

5. ID attribute [Required]

The ID (identifier) attribute MUST be used to identify this instance of the `<SignedDocument>` uniquely from other elements within the ENML (or enclosing XML if the ENML is wrapped inside another application's XML).

Please see Section 4.56 for a discussion of the Processing Rule for ID attributes within ENML.

Many examples of the `<SignedDocument>` element were provided in the non-normative Section 3 of this specification document. Please refer to that section for examples of the `<SignedDocument>` element.

4.5 Element `<DocumentSigners>` & `<DocumentSigner>`

The `<DocumentSigners>` element carries information about the individuals who signed the document, and appears only within a `<NotarizedDocument>` or `<WitnessedDocument>` element. It is a container-element, whose purpose is to organize a collection of `<DocumentSigner>` elements inside itself.

Each `<DocumentSigner>` element carries information about each signer of the electronic document that was notarized (or whose signing was witnessed) by the Notary Public.
Schema Definition:

```xml
<xsd:element
    name="DocumentSigners"
    type="enml:DocumentSignersType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:complexType name="DocumentSignersType">
    <xsd:sequence>
        <xsd:element
            name="DocumentSigner"
            type="enml:DocumentSignerType"
            minOccurs="1"
            maxOccurs="unbounded">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DocumentSignerType">
    <xsd:sequence>
        <xsd:element
            name="SignerName"
            type="enml:PersonNameType"
            minOccurs="1"
            maxOccurs="1">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
```
The `<DocumentSigners>` element is of type `DocumentSignersType` defined in the ENML schema, while the `<DocumentSigner>` element is of type `DocumentSignerType`.

There MUST be exactly ONE `<DocumentSigners>` element in either a `<NotarizedDocument>` or `<WitnessedDocument>` element. The `<DocumentSigners>` element MUST contain at least ONE `<DocumentSigner>` element, but may contain an unbounded (unlimited) number of `<DocumentSigner>` elements as needed.

The `<DocumentSigner>` element of type `DocumentSignerType`, consists of a sequence of five children elements and one ID attribute:
1. `<SignerName>` [Required]

   This element of type `PersonNameType`, identifies the full name of a document-signer. There MUST be exactly ONE `<SignerName>` element in a `<DocumentSigner>` element.

   The `<SignerName>` element is specified in Section 4.13.

2. `<SignerTitle>` [Optional]

   The optional element of type `PersonTitleType`, identifies the official title of the person signing the document, when the notarized or witnessed document is being signed in an official capacity. When present, there MUST be exactly ONE `<SignerTitle>` element in a `<DocumentSigner>` element.

   The `<SignerTitle>` element is specified in Section 4.14.

3. `<SignerInternationalAddress>` or `<SignerUSAddress>` [Optional]

   The next sub-element represents the physical address of the document-signer. However, depending on the geographical location – either an international or US-based location - this optional element must be a choice of either the `<SignerInternationalAddress>` or the `<SignerUSAddress>` element, respectively. It is the responsibility of the application generating the ENML-marked document to choose the appropriate element based on the jurisdiction of the signer or signing.

   When present, there MUST be exactly ONE `<SignerInternationalAddress>` or `<SignerUSAddress>` element in a `<DocumentSigner>` element.

   The `<SignerInternationalAddress>` element is specified in Section 4.15.

   The `<SignerUSAddress>` element is specified in Section 4.16.

4. `<SignerIdentificationMethod>` [Optional]

   The optional `<SignerIdentificationMethod>` element of `PersonIdentificationMethodType` indicates how the document-signer identified himself/herself during the notarization or witnessing.

   When present, there MUST be exactly ONE `<SignerIdentificationMethod>` element in a `<DocumentSigner>` element.

   The `<SignerIdentificationMethod>` element is specified in Section 4.17.

5. `<ds:Signature>` or `<SignerSignature>` [Required]

   The next sub-element represents the electronic signature of the document-signer. However, depending on the type of technology chosen, this element must be a choice of either the `<ds:Signature>` or the `<SignerSignature>` element. It is the responsibility of the application generating the ENML-marked document to choose the appropriate element.

   There MUST be exactly ONE `<ds:Signature>` or `<SignerSignature>` element in a `<DocumentSigner>` element.

   This element of type `ds:SignatureType` is defined in the [XMLSignature] schema. It provides details about the electronic signature of the document-signer who signed the electronic document.

   The `<ds:Signature>` element is a versatile element that allows for electronically signing documents with a variety of technologies, depicted in the XML. It supports the use of symmetric, as well as asymmetric, cryptographic keys for the generation of electronic signatures, and a
variety of cryptographic algorithms to provide most implementers with the flexibility to choose a signing mechanism of their choice (as long as they satisfy the semantic meanings of the elements within the `<ds:Signature>` element).

The `<ds:Signature>` element is specified in [XMLSignature].

Please see Section 3.12 for a note on the use of XPath in `<ds:Signature>` elements within ENML.

The `<SignerSignature>` element of type `xsd:anyType` are for those situations where the software implementation of ENML chooses NOT to use any specific signing technology, and represents the signer's signature with simple text (or similar representations).

As the type implies, this element does not mandate what content may be placed in it, but it is strongly recommended that implementers use meaningful text (such as “Signed by John Doe”, or “/S=John Doe” or similar well-known and accepted norms of electronic signatures) to represent the signer’s signature.

The `<SignerSignature>` element is specified in 4.18.

Please note the informative appendix on “ENML Security Implications” at the end of this document for an understanding of security issues within ENML.

6. ID attribute [Required]

The ID (identifier) attribute MUST be used to identify this instance of the `<Signature>` or `<SignerSignature>` uniquely from other elements within the ENML (or enclosing XML if the ENML is wrapped inside another application's XML).

Please see Section 4.56 for a discussion of the Processing Rule for ID attributes within ENML.

Some high-level examples of the `<DocumentSigners>` element are as follows:

**Example 1 – A `<DocumentSigners>` element with one `<DocumentSigner>` using a NULL Cryptographic Signature:**

```xml
<enml:DocumentSigners>
    <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d273fd7">
        <enml:SignerName>
            <enml:PersonFirstName>John</enml:PersonFirstName>
            <enml:PersonLastName>Doe</enml:PersonLastName>
        </enml:SignerName>
        <enml:SignerIdentificationMethod>
            Produced Government-issued Identification Document
        </enml:SignerIdentificationMethod>
        <enml:SignerSignature>
            Signed by John Doe
        </enml:SignerSignature>
    </enml:DocumentSigner>
</enml:DocumentSigners>
```

**Example 2 – A `<DocumentSigners>` element with two `<DocumentSigner>` elements, both using NULL Cryptographic Signatures:**

```xml
<enml:DocumentSigners>
    <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d273fd7">
    <enml:DocumentSigners>
        <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d273fd7">
```
Example 3 – A `<DocumentSigners>` element with one `<DocumentSigner>` element, with an international address and using a symmetric-key based cryptographic signature:

```xml
<enml:DocumentSigner Id="DocumentSigner- NL-DR-123456789-1222643107-c3edac5c36aace2adf6d273fd7f91b09f20bfa099beb61cb493dfdd957">
  <enml:SignerName>
    <enml:PersonFirstName>Jan</enml:PersonFirstName>
    <enml:PersonLastName>Doer</enml:PersonLastName>
  </enml:SignerName>
  <enml:SignerInternationalAddress>
    <enml:City>Utrecht</enml:City>
    <enml:County>Utrecht</enml:County>
    <enml:Country>Netherlands</enml:Country>
  </enml:SignerInternationalAddress>
  <enml:SignerIdentificationMethod>
    Produced Government-issued Identification Document
  </enml:SignerIdentificationMethod>
  <enml:SignerSignature>
    Signed by Jan Doer
  </enml:SignerSignature>
</enml:DocumentSigner>
```
4.6 Element <NotaryCertificates> & <NotaryCertificate>

The <NotaryCertificates> element carries information about the Notarial Certificate and the Notary Public who notarized (or witnessed) the document. It appears only within a <NotarizedDocument> element but not in a <WitnessedDocument> element (since witnessed documents do not carry a notarial certificate).

The <NotaryCertificates> element is a container-element, whose sole purpose is to organize a collection of <NotaryCertificate> elements inside itself.

The <NotaryCertificate> element carries the actual information about the notarial certificate, representing a unique notarization event, and about the Notary Public himself/herself.

Note: The <NotaryCertificates> may also appear within propriety XML documents, as a sub-element of documents that are not of NotarizedDocumentType, but which are deemed eNotarized because they have undergone the requisite legal process of electronic notarization and carry the <NotaryCertificates> and <NotarySignatures> elements in their proprietary XML.

However, there is no assurance that applications that implement only the <NotaryCertificates> and <NotarySignatures> elements within propriety XML are inter-operable, since those applications must also understand the proprietary XML of the enclosing document to verify the eNotarized document.

Applications that implement the OASIS-specified <NotarizedDocument> and <WitnessedDocument> elements, however, are expected to inter-operate, since these two elements are standardized through this specification.

Schema Definition:

```xml
<xsd:element name="NotaryCertificates" type="enml:NotaryCertificatesType"/>

<xsd:complexType name="NotaryCertificatesType">
  <xsd:sequence>
    <xsd:element name="NotaryCertificate" type="enml:NotaryCertificateType"/>
  </xsd:sequence>
</xsd:complexType>
```
The `<NotaryCertificates>` element is of type `NotaryCertificatesType`, defined in the ENML schema, while the `<NotaryCertificate>` element is of the `NotaryCertificateType`.

There MUST be exactly ONE `<NotaryCertificates>` element in a `<NotarizedDocument>` element. The `<NotaryCertificates>` element MUST contain at least ONE `<NotaryCertificate>` element, but may contain an unbounded (unlimited) number of `<NotaryCertificate>` elements as needed.

The `<NotaryCertificate>` element consists of a sequence of two children elements and one ID attribute:

1. `<CertificateContent>` [Required]

   This element of type `NotaryCertificateContentType` provides the legal details of the electronic notarization of the document. It is currently geared toward US-based law, but is anticipated to include support for international law as representation from other countries provide input to the OASIS eNotarization Technical Committee.

   The `<CertificateContent>` element is specified in Section 4.19.

2. `<NotaryPublic>` [Required]

   This element of type `NotaryPublicType` provides details about the Notary Public who notarized the electronic document. Once again, this element is currently geared toward US-based Notaries Public, but is anticipated to include support for international Notaries as representation from other countries provide input to the OASIS eNotarization Technical Committee.

   The `<NotaryPublic>` element is specified in Section 4.8.

3. ID attribute [Required]
The ID (identifier) attribute MUST be used to identify this instance of the `<NotaryCertificate>` uniquely from other elements within the ENML (or enclosing XML if the ENML is wrapped inside another application's XML).

Please see Section 4.56 for a discussion of the Processing Rule for ID attributes within ENML.

Some examples of the `<NotaryCertificates>` element follow:

Example 1 – A `<NotaryCertificates>` element with one `<NotaryCertificate>` element:

```xml
<enml:NotaryCertificates>
  <enml:NotaryCertificate  Id="NotaryCertificate- US-CA-1565986-1222640980-0ef2d548e82b637a0568fc1c4e699e2e8b96acfc083fdefb82e24a05cabol40f54">
    <enml:CertificateContent>
      <enml:NotarizationType>
        Acknowledgment
      </enml:NotarizationType>
      <enml:NotarizationDate>
        2007-02-07T15:19:17-08:00
      </enml:NotarizationDate>
      <enml:NotarizationUSLocation>
        <enml:City>Cupertino</enml:City>
        <enml:County>Santa Clara</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:NotarizationUSLocation>
      <enml:StatutoryContent>
        State of California
        County of Santa Clara
        On February 07 2007, before me Arshad Noor, personally appeared John Doe, who proved to me on the basis of satisfactory evidence to be the person whose name is subscribed to the within instrument and acknowledged to me that he executed the same in his authorized capacity, and that by his signature on the instrument the person, or the entity upon behalf of which the person acted, executed the instrument.
        I certify under PENALTY OF PERJURY under the laws of the State of California that the foregoing paragraph is true and correct.
        WITNESS my hand and official seal.
      </enml:StatutoryContent>
    </enml:CertificateContent>
  </enml:NotaryCertificate>
</enml:NotaryCertificates>
```

Example 2 – A `<NotaryCertificates>` element with multiple `<NotaryCertificate>` elements:

```xml
<enml:NotaryCertificates>
  <enml:NotaryCertificate  Id="NotaryCertificate- US-CA-1565986-1222640980-0ef2d548e82b637a0568fc1c4e699e2e8b96acfc083fdefb82e24a05cabol40f54">
    <!-- NotaryPublic who notarized this document -->
    <enml:NotaryPublic>
      <enml:NotaryName>
        <enml:PersonGivenName>Arshad</enml:PersonGivenName>
        <enml:PersonSurName>Noor</enml:PersonSurName>
      </enml:NotaryName>
    </enml:NotaryPublic>
  </enml:NotaryCertificate>
  <enml:NotaryCertificate  Id="NotaryCertificate- US-CA-1565986-1222640980-0ef2d548e82b637a0568fc1c4e699e2e8b96acfc083fdefb82e24a05cabol40f54">
    <!-- NotaryPublic who notarized this document -->
    <enml:NotaryPublic>
      <enml:NotaryName>
        <enml:PersonGivenName>Arshad</enml:PersonGivenName>
        <enml:PersonSurName>Noor</enml:PersonSurName>
      </enml:NotaryName>
    </enml:NotaryPublic>
  </enml:NotaryCertificate>
</enml:NotaryCertificates>
```
Example 2—A <NotaryCertificates> element with three <NotaryCertificate> elements in it (many details have been removed for brevity). Each <NotaryCertificate> represents a unique notarization event of the same document:

```xml
<enml:NotaryCertificates>
  <!-- FIRST -->
  <enml:NotaryCertificate Id="NotaryCertificate-US-CA-.....">
    <enml:CertificateContent>....</enml:CertificateContent>
    <enml:NotaryPublic>....</enml:NotaryPublic>
  </enml:NotaryCertificate>
  <!-- SECOND -->
  <enml:NotaryCertificate Id="NotaryCertificate-US-CA-.....">
    <enml:CertificateContent>....</enml:CertificateContent>
    <enml:NotaryPublic>....</enml:NotaryPublic>
  </enml:NotaryCertificate>
  <!-- THIRD -->
    <enml:CertificateContent>....</enml:CertificateContent>
    <enml:NotaryPublic>....</enml:NotaryPublic>
  </enml:NotaryCertificate>
</enml:NotaryCertificates>
```

### 4.7 Element <NotarySignatures> & <NotarySignature>

The <NotarySignatures> element carries information about the Notarial Certificate and the Notary Public who notarized (or witnessed) the document.

The <NotarySignatures> element is a container-element, whose sole purpose is to organize a collection of either <ds:Signature> or <NotarySignature> elements, which carry the actual information about the Notary's signature, inside itself. Each signature element represents a unique notarization event.

*Note: The <NotarySignatures> element may also appear within propriety XML documents, as a sub-element of documents that are not of NotarizedDocumentType, but which are deemed eNotarized because they have undergone the requisite legal process of electronic notarization and carry the <NotaryCertificates> and <NotarySignatures> elements in their proprietary XML.*
However, there is no assurance that applications that implement only the the `<NotaryCertificates>` and `<NotarySignatures>` elements within proprietary XML are inter-operable, since those applications must also understand the proprietary XML of the enclosing document to verify the eNotarized document.

Applications that implement the OASIS-specified `<NotarizedDocument>` and `<WitnessedDocument>` elements, however, are expected to inter-operate, since these two elements are standardized through this specification.

The `<NotarySignaturesType>` element is of type `NotarySignaturesType`, defined in the ENML schema.

There MUST be exactly ONE `<NotarySignatures>` element in either a `<NotarizedDocument>` or a `<WitnessedDocument>` element. The `<NotarySignatures>` element MUST contain at least ONE `<ds:Signature>` or `<NotarySignature>` element, but may contain an unbounded (unlimited) number of either element as needed.

The `<NotarySignatures>` element consists of one child elements and one ID attribute:

1. `<ds:Signature>` or `<NotarySignature>` [Required]
This element of type \texttt{ds:SignatureType} is defined in the [XMLSignature] schema. It provides details about the electronic signature of the Notary Public who notarized (or witnessed the signing of) the electronic document.

The \texttt{<ds:Signature>} element is a versatile element that allows for electronically signing documents with a variety of technologies, depicted in the XML. It supports the use of symmetric, as well as asymmetric, cryptographic keys for the generation of electronic signatures, and a variety of cryptographic algorithms to provide most implementers with the flexibility to choose a signing mechanism of their choice (as long as they satisfy the semantic meanings of the elements within the \texttt{<ds:Signature>} element).

The \texttt{<ds:Signature>} element is specified in [XMLSignature].

Please see Section 3.12 for a note on the use of XPath in \texttt{<ds:Signature>} elements within ENML.

The \texttt{<NotarySignature>} element of type \texttt{xsd:anyType} are for those situations where the software implementation of ENML chooses NOT to use any specific signing technology, and represents the Notary's signature with simple text (or similar representations).

As the type implies, this element does not mandate what content may be placed in it, but it is strongly recommended that implementers use meaningful text (such as “Signed by Howard Jones”, or “S=Howard Jones” or similar well-known and accepted norms of electronic signatures) to represent the Notary's signature.

Please note the informative appendix on “ENML Security Implications” at the end of this document for an understanding of security issues within ENML.

2. ID attribute [Required]

The ID (identifier) attribute MUST be used to identify this instance of the \texttt{<ds:Signature>} or \texttt{<NotarySignature>} element uniquely from other elements within the ENML (or enclosing XML if the ENML is wrapped inside another application's XML).

Please see Section 4.56 for a discussion of the Processing Rule for ID attributes within ENML.

Some examples of the \texttt{<NotarySignatures>} element follow:

\textbf{Example 1} – A \texttt{<NotarySignatures>} element with one \texttt{<ds:Signature>} using a symmetric cryptographic key. The signature has signed one Manifest enclosed within the Object element and contains four (4) References pointing to one SignedDocument, two DocumentSigners and one NotaryCertificate. The signature uses the HMAC-SHA1 cryptographic algorithm; the symmetric key is identified with a KeyName of “10514-1-123” and can be retrieved from the URL defined in the RetrievalMethod element:

\begin{verbatim}
<enml:NotarySignatures>
  <ds:Signature>
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        </ds:Transforms>
      </ds:Reference>
    </ds:SignedInfo>
  </ds:Signature>
</enml:NotarySignatures>
\end{verbatim}
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>MtGcQfervxf/b3xvrPEstt0h1fg=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
09fbf143326bb09beefabe8d84094ee14c5eee8c
</ds:SignatureValue>
<ds:KeyInfo>
<ds:KeyName>10514-1-123</ds:KeyName>
<ds:RetrievalMethod
URI="http://skms.somecompany.com/symkeyServlet/getsymkey"/>
</ds:KeyInfo>
<ds:Object>
<ds:Manifest  Id="Manifest- US-CA-1565986-1222641281">
<ds:Reference URI="#SignedDocument-US-CA-1565986-1222646484-
546e0248b7e7f52a44f278803eefb84557746e691556e0d37bb92fe1f2ad7fd0b">
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>W4DupJioi0mA7aG+N1qiNLtqsk=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#DocumentSigner- US-CA-1565986-
1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2af6d6273fd7">
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>Yx5JkTS0ZxaM0uEpm/SxmSRSgAw=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#DocumentSigner- US-CA-1565986-1222643179- 
ca4f9fccc8950e9ad42959c725f2e31073f20b7058b777b2aa2d21f183006dc">
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>M0uEpmYx5JkTS0Zxa/SxmSRSgAw=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#NotaryCertificate- US-CA-1565986-1963a865f3516996f711936045ced816a0fcccd4aca57c6807a1c44f9b98a1e5f">
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>+w49wl2lZOiUOV5x7L0fQp1nO30=</ds:DigestValue>
</ds:Reference>
</ds:Manifest>
</ds:Object>
</enml:NotarySignatures>

Example 2 – A <NotarySignatures> element with one <ds:Signature> using an asymmetric cryptographic key. The signature has References pointing to one SignedDocument, DocumentSigner and NotaryCertificate. The signature uses the RSA-SHA1 cryptographic algorithm; the public-key is identified with the X509 digital certificate:
4.8 Element <NotariesPublic> & <NotaryPublic>

The <NotariesPublic> element carries information about the Notary Public who notarized (or witnessed the signing of) the document. While it appears as a child element of the <NotaryCertificate> element in the <NotarizedDocument> element, it appears as a direct child of the <WitnessedDocument> element (since witnessed documents do not carry a notarial certificate).

The <NotariesPublic> element is a container-element, whose sole purpose is to organize a collection of <NotaryPublic> elements inside itself.

The <NotaryPublic> element carries the actual information about the Notary Public himself/herself who either notarized or witnessed the signing of an electronic document.
Schema Definition:

```xml
<xsd:element name="NotariesPublic" minOccurs="1" maxOccurs="1">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="NotaryPublic" type="enml:NotaryPublicType" minOccurs="1" maxOccurs="unbounded"/>
      <xsd:element name="NotaryName" type="enml:PersonNameType" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="NotaryCommissionNumber" type="enml:NotaryCommissionNumberType" minOccurs="0" maxOccurs="1"/>  
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```xml
<xsd:complexType name="NotaryPublicType">  
  <xsd:sequence>
    <xsd:element name="NotaryName" type="enml:PersonNameType" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="NotaryCommissionNumber" type="enml:NotaryCommissionNumberType" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```
The `<NotariesPublic>` element is an anonymous element defined in the ENML schema, while the `<NotaryPublic>` element is of the `<NotaryPublicType>`.

There MUST be exactly ONE `<NotariesPublic>` element in either a `<NotarizedDocument>` or `<WitnessedDocument>` element. The `<NotariesPublic>` element MUST contain at least ONE `<NotaryPublic>` element, but may contain an unbounded (unlimited) number of `<NotaryPublic>` elements as needed.

The `<NotaryPublic>` element consists of a sequence of six (6) children elements and one ID attribute:

1. `<NotaryName>` [Required]

This element of type `<PersonNameType>` defined in the ENML schema, provides the full name of the Notary Public. There MUST be exactly ONE `<NotaryName>` element inside a `<NotaryPublic>` element.

The `<NotaryName>` element is specified in Section 4.20.
2. `<NotaryCommissionNumber>` [Optional]

This element of type `NotaryCommissionNumberType` defined in the ENML schema, provides the official Commission number issued by the authority of a jurisdiction in which the Notary Public may perform his/her official duties. When present, there MUST be exactly ONE `<NotaryCommissionNumber>` element inside a `<NotaryPublic>` element.

The `<NotaryCommissionNumber>` element is specified in Section 4.21.

3. `<NotaryCommissionExpiryDate>` [Required]

This element of type `DateTimeType` defined in the ENML schema, provides calendar date, time and time-zone on which the Notary’s commission expires within the jurisdiction they are authorized to perform their duties. There MUST be exactly ONE `<NotaryCommissionExpiryDate>` element inside a `<NotaryPublic>` element.

The `<NotaryCommissionExpiryDate>` element is specified in Section 4.22.

4. `<NotaryInternationalJurisdiction>` or `<NotaryUSJurisdiction>` [Required - Choice]

This element provides the geographical address within the jurisdiction where the Notary Public is authorized to perform his/her duties. Depending on the location, the application implementing ENML must choose either the `<NotaryInternationalJurisdiction>` or `<NotaryUSJurisdiction>` element. There MUST be exactly ONE of these two jurisdiction elements inside a `<NotaryPublic>` element.

The `<NotaryInternationalJurisdiction>` element is of the type `InternationalAddressType` defined in the ENML schema, and is specified in Section 4.23.

The `<NotaryUSJurisdiction>` element is of the type `USAddressType` defined in the ENML schema, and is specified in Section 4.24.

5. `<NotaryBondNumber>` [Optional]

This optional element of type `NotaryBondNumberType` defined in the ENML schema, provides information about the bond posted by the Notary Public, within the jurisdiction where they perform their duties. When present, there MUST be exactly ONE `<NotaryBondNumber>` element inside a `<NotaryPublic>` element.

*Note: The posting of a bond by a Notary Public, before they may perform their official duties, is a requirement in some US states. Since it is not a mandatory requirement, this element is declared to be optional.*

The `<NotaryBondNumber>` element is specified in Section 4.25.

6. `<NotaryVerificationURI>` [Optional]

This optional element of type `xsd:anyURI` defined in the XML Schema, provides a network location where information about the Notary Public who notarized (or witnessed the signing of) a document, can be verified programmatically by software. There MAY be any number of `<NotaryVerificationURI>` elements inside a `<NotaryPublic>` element, each specifying a unique network location in the form of a Uniform Resource Identifier (typically a Uniform Resource Locator (URL) that can be reached through a web-service).

*Note: Currently there are no known locations in the US where such information can be verified programmatically and securely. However, it is anticipated that as eNotarized documents get*
adopted, these services will become necessary and common, to verify the authenticity of a Notary Public signing a Notarized or Witnessed document.

7. ID attribute [Optional]

The ID (identifier) attribute MAY be used to identify this instance of the `<NotaryPublic>` uniquely from other elements within the ENML (or enclosing XML if the ENML is wrapped inside another application's XML).

Please see Section 4.56 for a discussion of the Processing Rule for ID attributes within ENML.

Some examples of the `<NotaryPublic>` element follow:

Example 1 – A `<NotariesPublic>` element with one `<NotaryPublic>` element, as defined within a `<WitnessedDocument>` (<NotarizedDocument> elements only have the `<NotaryPublic>` element and do not carry the `<NotariesPublic>` element):

```
<enml:WitnessedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotariesPublic>
    <enml:NotaryPublic Id="NotaryPublic- US-CA-1565986-1222644533-4c8e1a0eff9544439fe46b8694ca8641ae64a16790d7504b6d6c69014d9bd624">
      <enml:NotaryName>
        <enml:PersonGivenName>Arshad</enml:PersonGivenName>
        <enml:PersonSurName>Noor</enml:PersonSurName>
      </enml:NotaryName>
      <enml:NotaryCommissionNumber>1565986</enml:NotaryCommissionNumber>
      <enml:NotaryCommissionExpiryDate>2009-04-29T23:59:59-08:00</enml:NotaryCommissionExpiryDate>
      <enml:NotaryUSJurisdiction>
        <enml:County>Santa Clara</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:NotaryUSJurisdiction>
    </enml:NotaryPublic>
  </enml:NotariesPublic>
  <enml:NotarySignatures>....</enml:NotarySignatures>
</enml:WitnessedDocument>
```

Example 2 – A `<NotaryPublic>` element, defined within a `<NotaryCertificate>` element of a `<NotarizedDocument>`:

```
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:NotaryCertificate>
      <enml:CertificateContent>...</enml:CertificateContent>
      <enml:NotaryPublic Id="NotaryPublic- US-CA-1565986-1222644533-4c8e1a0eff9544439fe46b8694ca8641ae64a16790d7504b6d6c69014d9bd624">
        <enml:NotaryName>
          <enml:PersonGivenName>Arshad</enml:PersonGivenName>
          <enml:PersonSurName>Noor</enml:PersonSurName>
        </enml:NotaryName>
        <enml:NotaryCommissionNumber>1565986</enml:NotaryCommissionNumber>
        <enml:NotaryCommissionExpiryDate>2009-04-29T23:59:59-08:00</enml:NotaryCommissionExpiryDate>
      </enml:NotaryPublic>
    </enml:NotaryCertificate>
  </enml:NotaryCertificates>
</enml:NotarizedDocument>
```
4.9 Element <Document>

The <Document> element of type DocumentType from the ENML schema, is used to carry the contents of the notarized, or witnessed for signature, document. The <Document> element is always embedded inside a <SignedDocument> element, and does not stand on its own.

Schema Definition:

```xml
<xsd:element name="Document"
    type="enml:DocumentType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:complexType name="DocumentType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:base64Binary">
            <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

There MUST be exactly ONE <Document> element inside a <SignedDocument> element. However, the <Document> element may itself contain as many pages as XML technology will permit.

The contents of the <Document> element MUST always be base64Binary encoded. This permits any type of document to be notarized: graphics images, scanned-images of printed documents, word-processing documents, reports generated from application software, ZIP files and even XML-encoded documents.

Some software products, such as OpenOffice, Microsoft Office and similar products, create rich-content documents, but store them in XML format using a specific XML schema of their choice. A single rich-content document may contain many files, all stored in XML format. Such related XML files are archived into a ZIP archive and given a product-specific extension, such as *.odf, *.xlsx, etc.

If an application expects to notarize or witness the signing of such rich-content documents in ENML, the application must base64-encode the binary archived file first, and embed the resultant text into the <Document> element, to be compliant with this standard.

Two examples of the <Document> element are shown below:

**Example 1 – A base64-encoded document in a <NotarizedDocument>:**

```xml
<enml:NotarizedDocument>
    <enml:SignedDocuments>
```
Example 2—A base64-encoded document in a <WitnessedDocument>. Note that the only difference between examples 1 and 2 is that example 2 carries a <NotariesPublic> element instead of the <NotaryCertificates> element.
4.10 Element <DocumentMIMEType> & <SignedDocumentMIMEType>

The <DocumentMIMEType> element of type SignedDocumentMIMEType from the ENML schema, is used to describe the contents of the notarized, or witnessed for signature, document. The <DocumentMIMEType> element is always embedded inside a <SignedDocument> element, and does not stand on its own.

The <SignedDocumentMIMEType> element is of the xsd:token type from XML Schema, and consists of a list of enumerated tokens that describe the most popular file-content types. This element provides a hint to receiving applications on the type of content that was notarized and how to process it upon parsing the ENML.

Schema Definition:

```xml
<xsd:element name="DocumentMIMEType"
    type="enml:SignedDocumentMIMEType"
    minOccurs="1"
    maxOccurs="1">
  ...
</xsd:element>

<xsd:simpleType name="SignedDocumentMIMEType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="application/other"/>
        <xsd:enumeration value="application/pdf"/>
        <xsd:enumeration value="application/postscript"/>
        <xsd:enumeration value="application/rtf"/>
        <xsd:enumeration value="application/vnd.oasis.opendocument.chart"/>
        <xsd:enumeration value="application/vnd.oasis.opendocument.formula"/>
        <xsd:enumeration value="application/vnd.oasis.opendocument.graphics"/>
        <xsd:enumeration value="application/vnd.oasis.opendocument.image"/>
        <xsd:enumeration value="application/vnd.oasis.opendocument.presentation"/>
        <xsd:enumeration value="application/vnd.oasis.opendocument.spreadsheet"/>
        <xsd:enumeration value="application/vnd.oasis.opendocument.text"/>
        <xsd:enumeration value="application/xml"/>
        <xsd:enumeration value="application/zip"/>
        <xsd:enumeration value="image/gif"/>
        <xsd:enumeration value="image/jpeg"/>
        <xsd:enumeration value="image/png"/>
        <xsd:enumeration value="text/plain"/>
        <xsd:enumeration value="text/xml"/>
        <xsd:enumeration value="video/mpeg"/>
        <xsd:enumeration value="video/mp4"/>
    </xsd:restriction>
</xsd:simpleType>
```

There MUST be exactly ONE <DocumentMIMEType> element inside a <SignedDocument> element.

The content of the <DocumentMIMEType> element MUST be from the list of enumerated tokens described in the <SignedDocumentMIMEType> element. The following MIME-types are currently recognized for the <SignedDocumentMIMEType> element, by this specification:

<table>
<thead>
<tr>
<th>MIME-type</th>
<th>Document Content Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>application/other</td>
<td>Any content not defined in this enumerated list</td>
</tr>
</tbody>
</table>

ENML-1.0-Specification
Copyright © OASIS® 2009. All Rights Reserved. 6 February 2009  Page 97 of 195
<table>
<thead>
<tr>
<th>MIME-type</th>
<th>Document Content Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>application/pdf</td>
<td>Portable Document Format</td>
</tr>
<tr>
<td>application/postscript</td>
<td>PostScript</td>
</tr>
<tr>
<td>application/rtf</td>
<td>Rich Text Format</td>
</tr>
<tr>
<td>application/vnd.oasis.opendocument.chart</td>
<td>OASIS OpenDocument Chart document</td>
</tr>
<tr>
<td>application/vnd.oasis.opendocument.formula</td>
<td>OASIS OpenDocument Formula document</td>
</tr>
<tr>
<td>application/vnd.oasis.opendocument.graphics</td>
<td>OASIS OpenDocument Graphics document</td>
</tr>
<tr>
<td>application/vnd.oasis.opendocument.image</td>
<td>OASIS OpenDocument Image document</td>
</tr>
<tr>
<td>application/vnd.oasis.opendocument.presentation</td>
<td>OASIS OpenDocument Presentation document</td>
</tr>
<tr>
<td>application/vnd.oasis.opendocument.spreadsheet</td>
<td>OASIS OpenDocument Spreadsheet document</td>
</tr>
<tr>
<td>application/vnd.oasis.opendocument.text</td>
<td>OASIS OpenDocument Text document</td>
</tr>
<tr>
<td>application/xml</td>
<td>eXtensible Markup Language in an application-specific format</td>
</tr>
<tr>
<td>application/zip</td>
<td>ZIP Archive</td>
</tr>
<tr>
<td>image/gif</td>
<td>Graphics Interchange Format image file</td>
</tr>
<tr>
<td>image/jpeg</td>
<td>Joint Photographic Expert Group image file</td>
</tr>
<tr>
<td>image/png</td>
<td>Portable Network Graphics image file</td>
</tr>
<tr>
<td>text/plan</td>
<td>Plain ASCII Text</td>
</tr>
<tr>
<td>text/xml</td>
<td>eXtensible Markup Language in plain ASCII text format</td>
</tr>
<tr>
<td>video/mpeg</td>
<td>Motion video in the Moving Picture Experts Group format</td>
</tr>
<tr>
<td>video/mp4</td>
<td>Motion video in the MPEG-4 format</td>
</tr>
</tbody>
</table>

If a document's content can be specified in more than one MIME-type, the application MUST use the most specific MIME-type that will render the contents in their true form. For example, a document created by OpenOffice Writer uses the MIME-type “application/vnd.oasis.opendocument.text”. However, the document, which is really a collection of smaller files containing content, style, settings, etc. and “zipped” into a ZIP archive, can also be characterized with the MIME-type “application/zip”.

While the document can be “read” by a ZIP-archive utility, and subsequently, by many XML tools and word-processors, the application that most closely renders the content in its “true form” is OpenOffice. As such, the ENML must use the MIME-type “application/vnd.oasis.opendocument.text” in the <DocumentMIMETYPE> element. This provides the receiver with the most specific instructions on how to render the contents correctly with the most appropriate tool or application.

Some examples of the <DocumentMIMETYPE> element are shown below:

**Example 1 – A <WitnessedDocument> with “application/pdf” type content in the <Document>:**

```xml
<enml:WitnessedDocument>
  <enml:SignedDocuments>
      <enml:Document>
```
4.11 Element <DocumentMIMETypeComments>

The element <DocumentMIMETypeComments> of type xsd:String, is used to provide a hint to the receiving application, on how to process the document content if the <DocumentMIMEType> is not explicit (as when this element might contain the MIME-type “application/other”).

Schema Definition:

```xml
<xsd:element name="DocumentMIMETypeComments" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:string/>
  </xsd:simpleType>
</xsd:element>
```
The `<DocumentMIMETypeComments>` element is an optional element; however, when used, there SHALL be only one `<DocumentMIMETypeComments>` element within a `<Document>` element.

The `<DocumentMIMETypeComments>` element may contain alphanumeric text, but is restricted to a length of 1,024-characters including white-space. The white-space must be preserved by XML-parsers and applications during processing.

An example of the `<DocumentMIMETypeComments>` element is as follows:

Example 1 – A `<NotarizedDocument>` with a `<DocumentMIMETypeComments>` element in the `<Document>`, indicating that the content is a Bank Draft created with an application called XYZ, version 1.2.3:

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>
      <enml:Document>
        PD94bWwgd mVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KCjwhLS0KICBM
        ZwdhbFhNTCBiTm90YXQpemFOaWuIaDUEuMCBTcGVjaWZpY2F0aW9uLgogIDA5IERl
        LgogIEhvd2V2ZXIsIHRoaXMgZG9jdW1lbnQgaXRzZWxmIG1heSBub3QgYmUgbW9k
        aWQgaW4gYW55IHdheSwgc3VjaCBhcyBieQogIHJlbW92aW5nIHRoZSBjb3B5B5
        ICAgIzA1Lz4KPC94c2Q6c2NoZW1hPgo=
      </enml:Document>
      <enml:DocumentMIMEType>application/other</enml:DocumentMIMEType>
      <enml:DocumentMIMETypeComments>
        Bank Draft created with application XYZ Version 1.2.3
      </enml:DocumentMIMETypeComments>
    </enml:SignedDocument>
    ....</enml:SignedDocuments>
  </enml:SignedDocuments>
  ....</enml:NotarizedDocument>
```

4.12 Element `<DocumentComments>`

The element `<DocumentComments>` of type `xsd:String`, is used to provide any human-readable comments to Relying Parties about the notarized or witnessed document. Applications implementing ENML, and that possess a User Interface (UI) that might display ENML content to them, are responsible for display the contents of the `<DocumentComments>` element.

Schema Definition:

```xml
<xsd:restriction base="xsd:string">
  <xsd:maxLength value="1024"/>
  <xsd:whiteSpace value="preserve"/>
</xsd:restriction>
</xsd:simpleType>
```
The <DocumentComments> element is an optional element; however, when used, there SHALL be only one <DocumentComments> element within a <Document> element.

The <DocumentComments> element may contain alphanumeric text, but is restricted to a length of 1,024-characters including white-space. The white-space must be preserved by XML-parsers and applications during processing.

An example of the <DocumentComments> element is as follows:

Example 1 – A <NotarizedDocument> with a <DocumentComments> element in the <Document>:

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>
      <enml:Document>
        PD94bWwgd mVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KCjwhLS0KICBMZwdhbFhNTCBiTm90YXJpemF0aW9uIDEuMCBTcGVjaWZpY2F0aW9uCgogIDA5IERlLgogIEhvd2V2ZXIsIHRoaXMgZG9jdW1lbnQgaXRzZWxmIG1heSBucmVjdCBhcyBieQogIHJlbW92aW5nIHRRoZSBpbmF0aW9uWD94c2Q6c2NoZW1hPgo=
      </enml:Document>
      <enml:DocumentMIMEType>application/other</enml:DocumentMIMEType>
      <enml:DocumentMIMETypeComments>
        Bank Draft created with application XYZ Version 1.2.3
      </enml:DocumentMIMETypeComments>
      <enml:DocumentComments>
        While the bank issued the Draft, there is a legitimate question about the viability of the issuing bank.
      </enml:DocumentComments>
    </enml:SignedDocument>
  </enml:SignedDocuments>
</enml:NotarizedDocument>
```

4.13 Element <SignerName>

The <SignerName> element, of the type PersonNameType defined in the ENML schema, carries the full name of a document signer. If there are multiple document signers, there would be multiple <DocumentSigner> elements, and each one would have its own <SignerName> element to identify the signer.
Schema Definition:

```xml
<xsd:element name="SignerName"
    type="enml:PersonNameType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:complexType name="PersonNameType">
    <xsd:sequence>
        <xsd:element name="PersonPrefixName"
            type="enml:PersonPrefixNameType"
            minOccurs="0"
            maxOccurs="1"/>
    </xsd:sequence>
    <xsd:choice>
        <xsd:sequence>
            <xsd:element name="PersonFirstName"
                type="enml:PersonGivenNameType"
                minOccurs="1"
                maxOccurs="1"/>
            <xsd:element name="PersonMiddleName"
                type="enml:PersonMiddleNameType"
                minOccurs="0"
                maxOccurs="1"/>
            <xsd:element name="PersonLastName"
                type="enml:PersonSurNameType"
                minOccurs="0"
                maxOccurs="1"/>
        </xsd:sequence>
        <xsd:element name="PersonGivenName"
            type="enml:PersonGivenNameType"
            minOccurs="0"
            maxOccurs="1"/>
        <xsd:element name="PersonMiddleName"
            type="enml:PersonMiddleNameType"
            minOccurs="0"
            maxOccurs="1"/>
        <xsd:element name="PersonSurName"
            type="enml:PersonSurNameType"
            minOccurs="0"
            maxOccurs="1"/>
    </xsd:choice>
</xsd:complexType>
```
The `<SignerName>` element consists of the following sub-elements:

1. `<PersonPrefixName>` [Optional]

   This optional element of type `PrefixNameType` defined in the ENML schema, allows the ENML document to carry a prefix to the name of the person signing the document. When present, there MUST be exactly ONE `<PersonPrefixName>` element in the `<SignerName>` element.

   The `<PersonPrefixName>` element is specified in Section 4.26.

2. `<PersonFirstName>`, `<PersonMiddleName>` and `<PersonLastName>` or `<PersonGivenName>`, `<PersonMiddleName>` and `<PersonSurName>` [Required]
This element presents a choice of a sequence of three elements, two of which are required in each choice.

The first choice allows the ENML to carry a required first-name, an optional middle-name and a required last-name of the document signer (which serves US-based systems), while the second choice allows the ENML to carry a required given-name, an optional middle-name and a required surname of the document-signer (which serves International systems).

Whichever choice is made by the application implementing ENML, there MUST be exactly ONE <PersonFirstName>, <PersonMiddleName> and <PersonLastName> or <PersonGivenName>, <PersonMiddleName> and <PersonSurName> sequence of elements in the <SignerName> element.

The <PersonFirstName> and <PersonGivenName> element is specified in Section 4.27.

The <PersonMiddleName> element is specified in Section 4.28.

The <PersonLastName> and <PersonSurName> element is specified in Section 4.29.

3. <PersonSuffixName> [Optional]

This optional element of type PersonSuffixNameType defined in the ENML schema, allows the ENML document to carry a suffix to the name of the person signing the document. When present, there MUST be exactly ONE <PersonSuffixName> element in the <SignerName> element.

The <PersonSuffixName> element is specified in Section 4.30.

4. <PersonMaidenName> [Optional]

This optional element of type PersonMaidenNameType defined in the ENML schema, allows the ENML document to carry the maiden name of the person signing the document. When present, there MUST be exactly ONE <PersonMaidenName> element in the <SignerName> element.

The <PersonMaidenName> element is specified in Section 4.31.

5. <PersonFullName> [Optional]

This optional element of type PersonFullNameType defined in the ENML schema, allows the ENML document to carry the full name of the person signing the document. When present, there MUST be exactly ONE <PersonFullName> element in the <SignerName> element.

The <PersonFullName> element is specified in Section 4.32.

Some examples of the <SignerName> element are as follows:

Example 1 – A <SignerName> using the US-based convention within a <NotarizedDocument>:

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>
    <enml:DocumentSigner>
      <enml:SignerName>
        <enml:PersonFirstName>John</enml:PersonFirstName>
        <enml:PersonMiddleName>Q</enml:PersonMiddleName>
        <enml:PersonLastName>Doe</enml:PersonLastName>
        <enml:PersonFullName>John  Q  Doe</enml:PersonFullName>
      </enml:SignerName>
      <enml:SignerSignature>...</enml:SignerSignature>
    </enml:DocumentSigner>
  </enml:DocumentSigners>
</enml:NotarizedDocument>
```
Example 2 – A <SignerName> using the US-based convention:
<enml:SignerName>
  <enml:PersonPrefixName>Dr.</enml:PersonPrefixName>
  <enml:PersonFirstName>John</enml:PersonFirstName>
  <enml:PersonMiddleName>Q</enml:PersonMiddleName>
  <enml:PersonLastName>Doe</enml:PersonLastName>
  <enml:PersonSuffixName>Jr.</enml:PersonSuffixName>
  <enml:PersonFullName>Dr. John Q. Doe Jr.</enml:PersonFullName>
</enml:SignerName>

Example 3 – A <SignerName> using the International convention:
<enml:SignerName>
  <enml:PersonPrefixName>Mrs.</enml:PersonPrefixName>
  <enml:PersonGivenName>Marie</enml:PersonGivenName>
  <enml:PersonSurName>Antoinette-Givenchy</enml:PersonSurName>
  <enml:PersonMaidenName>Antoinette</enml:PersonMaidenName>
  <enml:PersonFullName>Mrs. Marie Antoinette-Givenchy</enml:PersonFullName>
</enml:SignerName>

4.14 Element <SignerTitle>

The element <SignerTitle>, of type PersonTitleType defined in the ENML schema, allows the ENML to carry the official designation of the document-signer when the document-signer signs the electronic document in his/her official capacity in a business transaction.

Schema Definition:

```xml
<xsd:element
  name="SignerTitle"
  type="enml:PersonTitleType"
  minOccurs="0"
  maxOccurs="1">
</xsd:element>

<xsd:simpleType name="PersonTitleType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="256"/>
    <xsd:whiteSpace value="preserve"/>
  </xsd:restriction>
</xsd:simpleType>
```

When used in a <DocumentSigner> element, this is an optional element; when used in an <ApostilleContent> element, it is required. Regardless of when it is used, there SHALL be only one <SignerTitle> element within the <DocumentSigner> or the <ApostilleContent> element.

The PersonTitleType uses the xsd:String as the base type, and permits a title of up to 256-characters, preserving any white-spaces within the title-content.
Two examples of the <SignerTitle> element are shown below:

**Example 1 – A <SignerTitle> using the US-based convention within a <NotarizedDocument>:**

```
<enml:NotarizedDocument>
  <enml:SignedDocuments>.... </enml:SignedDocuments>
  <enml:DocumentSigners>
    <enml:DocumentSigner ID=...>
      <enml:SignerName>....</enml:SignerName>
      <enml:SignerTitle>Loan Officer</enml:SignerTitle>
      <enml:SignerSignature>....</enml:SignerSignature>
    </enml:DocumentSigner>
  </enml:DocumentSigners>
  <enml:NotaryCertificates>....</enml:NotaryCertificates>
  <enml:NotarySignatures>....</enml:NotarySignatures>
</enml:NotarizedDocument>
```

**Example 2 – An example of an <ApostillizedDocument> with a <SignerTitle> element.**

```
<enml:ApostillizedDocument>
  <enml:NotarizedDocument>.... </enml:NotarizedDocument>
  <enml:ApostilleNumber>
    ApostilleNumber-US-CA-1565986-1226972923-
    89e89e1c1cc35b3d7fcf030f78fca5527cb4cc61479fc88526c38e01d912a5>
  </enml:ApostilleNumber>
  <enml:ApostilleDate>2008-11-17T12:13:14-08:00</enml:ApostilleDate>
  <enml:IssuedAtUSLocation>
    <enml:County>Sacramento</enml:County>
    <enml:USState>CA</enml:USState>
    <enml:Country>USA</enml:Country>
  </enml:IssuedAtUSLocation>
  <enml:SignerTitle>Secretary of State, State of California</enml:SignerTitle>
  <enml:StatutoryContent>
    (Convention de La Haye du 5 octobre 1961)
  </enml:StatutoryContent>
  <enml:DocumentSigningOfficialName>
    <enml:PersonFirstName>Arshad</enml:PersonFirstName>
    <enml:PersonLastName>Noor</enml:PersonLastName>
  </enml:DocumentSigningOfficialName>
  <enml:DocumentSigningOfficialTitle>Notary Public</enml:DocumentSigningOfficialTitle>
  <enml:DocumentSigningOfficialSeal>
    <enml:TextSeal>
      Notary Public, State of California
    </enml:TextSeal>
  </enml:DocumentSigningOfficialSeal>
</enml:ApostillizedDocument>
```

4.15 Element <SignerInternationalAddress>

The element <SignerInternationalAddress>, of type `InternationalAddressType` defined in the ENML schema, is used to carry the physical postal address of a document-signer based outside the USA.
The `<SignerInternationalAddress>` element is an optional one, but when present, there SHALL be only one `<SignerInternationalAddress>` element within the `<DocumentSigner>` element.

```
<xsd:element
    name="SignerInternationalAddress"
    type="enml:InternationalAddressType"
    minOccurs="0"
    maxOccurs="1">
</xsd:element>

<xsd:complexType name="InternationalAddressType">
    <xsd:sequence>
        <xsd:element
            name="StreetAddress1"
            type="enml:StreetAddressType"
            minOccurs="0"
            maxOccurs="1"/>

        <xsd:element
            name="StreetAddress2"
            type="enml:StreetAddressType"
            minOccurs="0"
            maxOccurs="1"/>

        <xsd:element
            name="City"
The `<SignerInternationalAddress>` element is composed of a sequence of the following elements:

1. `<StreetAddress1>` [Optional]

   This optional element of type `StreetAddressType` defined in the ENML schema, carries the first line of the document signer's street address. When present, there MUST be exactly ONE `<StreetAddress1>` element in the `<SignerInternationalAddress>` element.

   The `<StreetAddress1>` element is specified in Section 4.33.

2. `<StreetAddress2>` [Optional]

   This optional element of type `StreetAddressType` defined in the ENML schema, carries the second line of the document signer's street address. When present, there MUST be exactly ONE `<StreetAddress2>` element in the `<SignerInternationalAddress>` element.

   The `<StreetAddress2>` element is specified in Section 4.34.

3. `<City>` [Optional]

   This optional element of type `CityType` defined in the ENML schema, carries the name of the city of the document signer's street address. When present, there MUST be exactly ONE `<City>` element in the `<SignerInternationalAddress>` element.

   The `<City>` element is specified in Section 4.35.

4. `<County>` [Optional]
This optional element of type `CountyType` defined in the ENML schema, carries the name of the county of the document signer's street address. When present, there MUST be exactly ONE `<County>` element in the `<SignerInternationalAddress>` element.

The `<County>` element for international addresses is specified in Section 4.36.

5. `<State>` [Optional]

This optional element of type `InternationalStateType` defined in the ENML schema, carries the name of the state or province of the document signer's street address. When present, there MUST be exactly ONE `<State>` element in the `<SignerInternationalAddress>` element.

The `<State>` element is specified in Section 4.37.

6. `<PostalCode>` [Optional]

This optional element of type `InternationalPostalCodeType` defined in the ENML schema, carries the Postal Code of the document signer's street address. When present, there MUST be exactly ONE `<PostalCode>` element in the `<SignerInternationalAddress>` element.

The `<PostalCode>` element is specified in Section 4.38.

7. `<Country>` [Required]

This is the only required element (currently) in the document-signers international street address. This element of type `CountryType` defined in the ENML schema, carries the name of the country of the document signer's street address. There MUST be exactly ONE `<Country>` element in the `<SignerInternationalAddress>` element.

The `<Country>` element is specified in Section 4.39.

Some examples of the `<SignerInternationalAddress>` element are shown below:

Example 1 – A `<SignerInternationalAddress>` within a `<NotarizedDocument>`:
```
<enml:NotarizedDocument>
  <enml:SignedDocuments>.... </enml:SignedDocuments>
  <enml:DocumentSigners>
    <enml:DocumentSigner>
      <enml:SignerName>....</enml:SignerName>
      <enml:SignerTitle>....</enml:SignerTitle>
      <enml:SignerInternationalAddress>
        <enml:City>Utrecht</enml:City>
        <enml:County>Utrecht</enml:County>
        <enml:Country>Netherlands</enml:Country>
      </enml:SignerInternationalAddress>
      <enml:SignerIdentificationMethod>....</enml:SignerIdentificationMethod>
      <enml:SignerSignature>....</enml:SignerSignature>
    </enml:DocumentSigner>
  </enml:DocumentSigners>
  <enml:NotaryCertificates>....</enml:NotaryCertificates>
  <enml:NotarySignatures>....</enml:NotarySignatures>
</enml:NotarizedDocument>
```

Example 2 – A `<SignerInternationalAddress>`:
```
<enml:SignerInternationalAddress>
  <enml:StreetAddress1>490 Sussex Drive</enml:StreetAddress1>
  <enml:City>Ottawa</enml:City>
  <enml:State>Ontario</enml:State>
```
4.16 Element <SignerUSAddress>

The element <SignerUSAddress>, of type USAddressType defined in the ENML schema, is used to carry the physical postal address of a document-signer based in the USA.

The <SignerUSAddress> element is an optional one, but when present, there SHALL be only one <SignerUSAddress> element within the <DocumentSigner> element.

![Signer Address Diagram]

**Schema Definition:**

```xml
<xsd:element
    name="SignerUSAddress"
    type="enml:USAddressType"
    minOccurs="0"
    maxOccurs="1">
</xsd:element>

<xsd:complexType name="USAddressType">
    <xsd:sequence>
        <xsd:element
            name="StreetAddress1"
            type="enml:StreetAddressType"
            minOccurs="0"
            maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
```
The `<SignerUSAddress>` element is composed of a sequence of the following elements:

1. `<StreetAddress1>` [Optional]
   
   This optional element of type `<StreetAddressType>` defined in the ENML schema, carries the first line of the document signer's street address. When present, there MUST be exactly ONE `<StreetAddress1>` element in the `<SignerUSAddress>` element.

   The `<StreetAddress1>` element is specified in Section 4.33.

2. `<StreetAddress2>` [Optional]
   
   This optional element of type `<StreetAddressType>` defined in the ENML schema, carries the second line of the document signer's street address. When present, there MUST be exactly ONE `<StreetAddress2>` element in the `<SignerUSAddress>` element.

   The `<StreetAddress2>` element is specified in Section 4.34.

3. `<City>` [Optional]
   
   This optional element of type `<CityType>` defined in the ENML schema, carries the name of the city
of the document signer's street address. When present, there MUST be exactly ONE <City> element in the <SignerUSAddress> element.

The <City> element is specified in Section 4.35.

4. <County> [Required]

This element of type CountyType defined in the ENML schema, carries the name of the county of the document signer's street address. There MUST be exactly ONE <County> element in the <SignerUSAddress> element.

The <County> element is specified in Section 4.36.

5. <USState> [Required]

This element of type USStateCodeType defined in the ENML schema, carries the 2-character abbreviation of the US-state or US-territory of the document signer's street address. There MUST be exactly ONE <USState> element in the <SignerUSAddress> element.

The <USState> element is specified in Section 4.40.

6. <USZipCode> [Optional]

This optional element of type USZipCodeType defined in the ENML schema, carries the ZIP Code of the document signer's street address. When present, there MUST be exactly ONE <USZipCode> element in the <SignerUSAddress> element.

The <USZipCode> element is specified in Section 4.41.

7. <Country> [Required]

This element of type CountryTypeUSA defined in the ENML schema, carries the full name or abbreviation of the United States of America. There MUST be exactly ONE <Country> element in the <SignerUSAddress> element.

The <Country> element for US-based addresses is specified in Section 4.42.

Some examples of the <SignerUSAddress> element are shown below:

**Example 1 – A <SignerUSAddress> within a <NotarizedDocument>:**

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>
    <enml:DocumentSigner>
      <enml:SignerName>....</enml:SignerName>
      <enml:SignerTitle>....</enml:SignerTitle>
      <enml:SignerUSAddress>
        <enml:County>Santa Clara</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:SignerUSAddress>
      <enml:SignerIdentificationMethod>....</enml:SignerIdentificationMethod>
      <enml:SignerSignature>....</enml:SignerSignature>
    </enml:DocumentSigner>
  </enml:DocumentSigners>
  <enml:NotaryCertificates>....</enml:NotaryCertificates>
  <enml:NotarySignatures>....</enml:NotarySignatures>
</enml:NotarizedDocument>
```
Example 2 – A `<SignerUSAddress>`:

```xml
<enml:SignerUSAddress>
  <enml:StreetAddress1>23 Autumn Drive</enml:StreetAddress1>
  <enml:City>Basking Ridge</enml:City>
  <enml:County>Middlesex</enml:County>
  <enml:USState>NJ</enml:USState>
  <enml:USZipCode>07920</enml:USZipCode>
  <enml:Country>USA</enml:Country>
</enml:SignerUSAddress>
```

4.17 Element `<SignerIdentificationMethod>`

The element `<SignerIdentificationMethod>`, of type `PersonIdentificationMethodType` defined in the ENML schema, is used to describe how the document-signer identified himself/herself to the Notary Public at the time of the notarization or witnessing act. The element does not specify what specific document was produced or any identification number of the document (for privacy reasons), but merely indicates what method of identification was used to assure the Notary Public of the signer’s identity.

**Schema Definition:**

```xml
<xsd:element
  name="SignerIdentificationMethod"
  type="enml:PersonIdentificationMethodType"
  minOccurs="0"
  maxOccurs="1">
</xsd:element>
```

```xml
<xsd:simpleType name="PersonIdentificationMethodType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Produced Government-issued Identification Document"/>
    <xsd:enumeration value="Personally Known to Credible Witnesses"/>
    <xsd:enumeration value="Personally Known to Notary Public"/>
  </xsd:restriction>
</xsd:simpleType>
```

The `<SignerIdentificationMethod>` element is an optional element, but when present, there SHALL be only one `<SignerIdentificationMethod>` element within the `<DocumentSigner>` element.

The `PersonIdentificationMethodType` uses the `xsd:Token` as the base type, and requires that, when present, the content of this element be one of the three specified identification methods:

1. Produced Government-issued Identification Document
2. Personally Known to Credible Witnesses
3. Personally Known to Notary Public

*Note: These choices currently reflect US-based law for notarization of documents. However, it is anticipated to include support for international law as representation from other countries provide input to the OASIS eNotarization Technical Committee.*

An example of the `<SignerIdentificationMethod>` element is shown below:

Example 1 – A `<SignerIdentificationMethod>` within a `<NotarizedDocument>`:
<enml:NotarizedDocument>
  <enml:SignedDocuments>.... </enml:SignedDocuments>
  <enml:DocumentSigners>
    <enml:DocumentSigner>
      <enml:SignerName>....</enml:SignerName>
      <enml:SignerTitle>Loan Officer</enml:SignerTitle>
      <enml:SignerIdentificationMethod>
        Personally Known to Notary Public
      </enml:SignerIdentificationMethod>
      <enml:SignerSignature>....</enml:SignerSignature>
    </enml:DocumentSigner>
    <enml:DocumentSigners>
      <enml:NotaryCertificates>....</enml:NotaryCertificates>
      <enml:NotarySignatures>....</enml:NotarySignatures>
    </enml:DocumentSigners>
  </enml:DocumentSigners>
</enml:NotarizedDocument>

4.18 Element <SignerSignature>

The <SignerSignature> element of type xsd:anyType from the XML Schema definition, allows the ENML to carry a document signer's signature in situations where the software implementation of ENML does NOT use any cryptographic signature, and represents the signer's signature as simple text (or similar representations).

Schema Definition:

```xml
<xsd:element
    name="SignerSignature"
    type="xsd:anyType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>
```

There SHALL be only one <SignerSignature> element within the <DocumentSigner> element. As the type implies, this element does not mandate what content may be placed in it, but it is strongly recommended that implementers use meaningful text (such as “Signed by John Doe”, or “/S=John Doe” or similar well-known and accepted norms of electronic signatures) to represent the signer's signature.

Examples of the <SignerSignature> element are shown below:

Example 1 – A <SignerSignature> within a <NotarizedDocument>:

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>.... </enml:SignedDocuments>
  <enml:DocumentSigners>
    <enml:DocumentSigner>
      <enml:SignerName>....</enml:SignerName>
      <enml:SignerTitle>Loan Officer</enml:SignerTitle>
      <enml:SignerIdentificationMethod>
        Signed by John Doe
      </enml:SignerIdentificationMethod>
      <enml:SignerSignature>....</enml:SignerSignature>
    </enml:DocumentSigner>
    <enml:DocumentSigners>
      <enml:NotaryCertificates>....</enml:NotaryCertificates>
      <enml:NotarySignatures>....</enml:NotarySignatures>
    </enml:DocumentSigners>
  </enml:DocumentSigners>
</enml:NotarizedDocument>
```
Example 2 – A <SignerSignature> within a <NotarizedDocument>:

```
<enml:SignerSignature>
  /S=John Doe
</enml:SignerSignature>
```

### 4.19 Element <CertificateContent>

The `<CertificateContent>` element of the type `NotaryCertificateContentType` from the ENML schema, carries details about the notarization act, such as the type of notarization, the date of notarization, location and the required legal content for the notarial certificate.

It is important to note that the `<WitnessedDocument>` element does not contain a `<NotaryCertificate>` element, and as such, does not contain a `<CertificateContent>` element.

```
<xsd:element
   name="CertificateContent"
   type="enml:NotaryCertificateContentType"
   minOccurs="1"
   maxOccurs="1">
</xsd:element>

<xsd:complexType name="NotaryCertificateContentType">
   <xsd:sequence>
      <xsd:element
         name="NotarizationType"
         type="enml:NotarizationActType"
         minOccurs="1"
         maxOccurs="1">
      </xsd:element>

      <xsd:element
         name="NotarizationDate"
         type="enml:DateTimeType">
      </xsd:element>

      <xsd:element
         name="NotarizationInternationalLocation">
      </xsd:element>

      <xsd:element
         name="NotarizationUSLocation">
      </xsd:element>

      <xsd:element
         name="StatutoryContent">
      </xsd:element>
   </xsd:sequence>
</xsd:complexType>
```
There MUST be exactly ONE `<CertificateContent>` element within the `<NotaryCertificate>` element.

The `<CertificateContent>` contains the following elements:

1. `<NotarizationType>` [Required]

   This element of the *NotarizationActType* from the ENML schema, identifies the type of notarization act executed within an ENML-marked document. There MUST be exactly ONE `<NotarizationType>` element within a `<CertificateContent>` element.

   The `<NotarizationType>` element is specified in 4.43.

2. `<NotarizationDate>` [Required]

   This element of the *DateTimeType* from the ENML schema, identifies the date, time and time-zone of the notarization act executed within an ENML-marked document. There MUST be exactly ONE `<NotarizationDate>` element within a `<CertificateContent>` element.

   The `<NotarizationDate>` element is specified in 4.44.

3. `<NotarizationInternationalLocation>` or `<NotarizationUSLocation>` [Required]

   The next sub-element represents the physical location where the notarization act executed within an ENML-marked document. However, depending on the geographical location – either an international or US-based location – this element must be a choice of either the `<NotarizationInternationalLocation>` or the `<NotarizationUSLocation>` element, respectively. It is the responsibility of the application generating the ENML-marked document to choose the appropriate element based on the jurisdiction of the notarization act.
When present, there MUST be exactly ONE `<NotarizationInternationalLocation>` or `<NotarizationUSLocation>` element in a `<CertificateContent>` element.

The `<NotarizationInternationalLocation>` element is specified in Section 4.45.

The `<NotarizationUSLocation>` element is specified in Section 4.46.

4. `<StatutoryContent>` [Required]

This element of the `xsd:anyType` from the ENML XML Schema Definition, carries details of the legal language required for the notarization act.

Since the element may contain any type of content, it is important to highlight that implementers are encouraged to use standard legal text found in official Notary Public hand-books, as plain text, while preserving white-space. While it is possible to use XML to further qualify the `<StatutoryContent>` element, while XML-parsers may accept it, most applications are not likely to be able to do anything with it other than merely display it as XML-formatted text. Only applications that understand an implementers XML may be able to parse and display the content with meaningful accoutrements.

There MUST be exactly ONE `<StatutoryContent>` element within a `<CertificateContent>` element.

One example of the `<CertificateContent>` element is shown below:

**Example 1 – An example of a `<CertificateContent>` element specifying that the notarization act was an Acknowledgment, that took place on February 07, 2007 in Santa Clara County of California, US. The element also contains the legal content required for an Acknowledgment in the state of California:**

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:CertificateContent>
      <enml:NotarizationType>Acknowledgment</enml:NotarizationType>
      <enml:NotarizationDate>2007-02-07T15:19:17-08:00</enml:NotarizationDate>
      <enml:NotarizationUSLocation>
        <enml:City>Cupertino</enml:City>
        <enml:County>Santa Clara</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:NotarizationUSLocation>
      <enml:StatutoryContent>
        State of California
        County of Santa Clara
        On February 07 2007, before me Arshad Noor, personally appeared John Doe, who proved to me on the basis of satisfactory evidence to be the person whose name is subscribed to the within instrument and acknowledged to me that he executed the same in his authorized capacity, and that by his signature on the instrument the person, or the entity upon behalf of which the person acted, executed the instrument.
    </enml:CertificateContent>
  </enml:NotaryCertificates>
</enml:NotarizedDocument>
```
4.20 Element <NotaryName>

The <NotaryName> element, of the type PersonNameType defined in the ENML schema, carries the full name of a Notary Public who either notarized or witnessed the signing of the ENML-marked document. If there are multiple notarizations performed on the same document, there must be multiple <NotaryPublic> elements, and each one would have its own <NotaryName> element to identify the Notary Public.

```
<xsd:element name="NotaryName"
             type="enml:PersonNameType"
             minOccurs="1"
             minOccurs="1"
           
```

Schema Definition:
<xsd:element name="PersonPrefixName" type="enml:PersonPrefixNameType" minOccurs="0" maxOccurs="1"/>

<xsd:choice>
  <xsd:element name="PersonFirstName" type="enml:PersonGivenNameType" minOccurs="1" maxOccurs="1"/>
  <xsd:element name="PersonMiddleName" type="enml:PersonMiddleNameType" minOccurs="0" maxOccurs="1"/>
  <xsd:element name="PersonLastName" type="enml:PersonSurNameType" minOccurs="0" maxOccurs="1"/>
</xsd:choice>

<xsd:element name="PersonGivenName" type="enml:PersonGivenNameType" minOccurs="1" maxOccurs="1"/>
<xsd:element name="PersonMiddleName" type="enml:PersonMiddleNameType" minOccurs="0" maxOccurs="1"/>
<xsd:element name="PersonSurName" type="enml:PersonSurNameType" minOccurs="0" maxOccurs="1"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="PersonNameType">
  <xsd:sequence>
    <xsd:element name="PersonPrefixName" type="enml:PersonPrefixNameType" minOccurs="0" maxOccurs="1"/>
    <xsd:choice>
      <xsd:element name="PersonFirstName" type="enml:PersonGivenNameType" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="PersonMiddleName" type="enml:PersonMiddleNameType" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="PersonLastName" type="enml:PersonSurNameType" minOccurs="0" maxOccurs="1"/>
    </xsd:choice>
    <xsd:element name="PersonGivenName" type="enml:PersonGivenNameType" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PersonMiddleName" type="enml:PersonMiddleNameType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PersonSurName" type="enml:PersonSurNameType" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
The `<NotaryName>` element consists of the following sub-elements:

1. `<PersonPrefixName>` [Optional]

   This optional element of type `PersonPrefixNameType` defined in the ENML schema, allows the ENML document to carry a prefix to the name of the Notary signing the document. When present, there MUST be exactly ONE `<PersonPrefixName>` element in the `<NotaryName>` element.

   The `<PersonPrefixName>` element is specified in Section 4.26.

2. `<PersonFirstName>`, `<PersonMiddleName>` and `<PersonLastName>` or `<PersonGivenName>`, `<PersonMiddleName>` and `<PersonSurName>` [Required]

   This element presents a choice of a sequence of three elements, two of which are required in each choice.

   The first choice allows the ENML to carry a required first-name, an optional middle-name and a required last-name of the Notary Public signing the document; this convention serves US-based systems, while the second choice allows the ENML to carry a required given-name, an optional middle-name and a required surname of the Notary Public signing the document, which serves International systems.

   Whichever choice is made by the application implementing ENML, there MUST be exactly ONE `<PersonFirstName>`, `<PersonMiddleName>` and `<PersonLastName>` or `<PersonGivenName>`, `<PersonMiddleName>` and `<PersonSurName>` sequence of elements in the `<NotaryName>` element.

   The `<PersonFirstName>` and `<PersonGivenName>` element is specified in Section 4.27.

   The `<PersonMiddleName>` element is specified in Section 4.28.

   The `<PersonLastName>` and `<PersonSurName>` element is specified in Section 4.29.

3. `<PersonSuffixName>` [Optional]

   This optional element of type `PersonSuffixNameType` defined in the ENML schema, allows the ENML document to carry a suffix to the name of the Notary signing the document. When present, there MUST be exactly ONE `<PersonSuffixName>` element in the `<NotaryName>` element.

   The `<PersonSuffixName>` element is specified in Section 4.30.

4. `<PersonMaidenName>` [Optional]

   This optional element of type `PersonMaidenNameType` defined in the ENML schema, allows the ENML document to carry the maiden name of the Notary signing the document. When present, there MUST be exactly ONE `<PersonMaidenName>` element in the `<NotaryName>` element.
The `<PersonMaidenName>` element is specified in Section 4.31.

5. `<PersonFullName>` [Optional]

This optional element of type `PersonFullNameType` defined in the ENML schema, allows the ENML document to carry the full name of the Notary signing the document. When present, there MUST be exactly ONE `<PersonFullName>` element in the `<NotaryName>` element.

The `<PersonFullName>` element is specified in Section 4.32.

Some examples of the `<NotaryName>` element are as follows:

Example 1 – A `<NotaryName>` using the US-based convention within a `<NotarizedDocument>`:

```
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:NotaryCertificate>
      <enml:CertificateContent>....</enml:CertificateContent>
      <enml:NotaryPublic>
        <enml:NotaryName>
          <enml:PersonFirstName>Arshad</enml:PersonFirstName>
          <enml:PersonLastName>Noor</enml:PersonLastName>
          <enml:PersonFullName>Arshad Noor</enml:PersonFullName>
        </enml:NotaryName>
        <enml:NotaryCommissionExpiryDate>2009-04-29T23:59:59-08:00</enml:NotaryCommissionExpiryDate>
        <enml:NotaryUSJurisdiction>
          <enml:County>Santa Clara</enml:County>
          <enml:USState>CA</enml:USState>
          <enml:Country>USA</enml:Country>
        </enml:NotaryUSJurisdiction>
      </enml:NotaryPublic>
    </enml:NotaryCertificate>
  </enml:NotaryCertificates>
  <enml:NotarySignatures>....</enml:NotarySignatures>
</enml:NotarizedDocument>
```

Example 2 – A `<NotaryName>` using the US-based convention:

```
<enml:NotaryName>
  <enml:PersonPrefixName>Mr.</enml:PersonPrefixName>
  <enml:PersonFirstName>John</enml:PersonFirstName>
  <enml:PersonMiddleName>Q</enml:PersonMiddleName>
  <enml:PersonLastName>Doe</enml:PersonLastName>
  <enml:PersonSuffixName>Esq.</enml:PersonSuffixName>
  <enml:PersonFullName>Mr. John Q. Doe Esq.</enml:PersonFullName>
</enml:NotaryName>
```

Example 3 – A `<NotaryName>` using the International convention:

```
<enml:NotaryName>
  <enml:PersonPrefixName>Mrs.</enml:PersonPrefixName>
  <enml:PersonGivenName>Marie</enml:PersonGivenName>
  <enml:PersonSurName>Antoinette- Givenchy</enml:PersonSurName>
  <enml:PersonFullName>Mrs. Marie Antoinette-
```
Givenchy</enml:PersonFullName>
<enml:NotaryName>

4.21 Element <NotaryCommissionNumber>

The <NotaryCommissionNumber> element of the type NotaryCommissionNumberType from the ENML schema, provides the official Commission number issued by the authority of a jurisdiction in which the Notary Public may perform his/her official duties. However, it should be noted that some US states do not use an official Commission number to identify a unique Notary Public within their state.

Schema Definition:

```
<xsd:element
    name="NotaryCommissionNumber"
    type="enml:NotaryCommissionNumberType"
    minOccurs="0"
    maxOccurs="1">
</xsd:element>

<xsd:simpleType name="NotaryCommissionNumberType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="255"/>
        <xsd:WhiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```

The optional <NotaryCommissionNumber> element permits the use of any alphanumeric character, to a maximum length of 255 characters, preserving white-space characters in its representation. When present, there MUST be exactly ONE <NotaryCommissionNumber> element inside a <NotaryPublic> element.

An example of the <NotaryCommissionNumber> element follows:

Example 1 – An example of a <NotaryCommissionNumber>:

```
<enml:NotarizedDocument>
    <enml:SignedDocuments>....</enml:SignedDocuments>
    <enml:DocumentSigners>....</enml:DocumentSigners>
    <enml:NotaryCertificates>
        <enml:CertificateContent>....</enml:CertificateContent>
        <enml:NotaryPublic>
            <enml:NotaryName>....</enml:NotaryName>
            <enml:NotaryCommissionNumber>1565986</enml:NotaryCommissionNumber>
            <enml:NotaryCommissionExpiryDate>2009-04-29T23:59:59-08:00</enml:NotaryCommissionExpiryDate>
            <enml:NotaryUSJurisdiction>
                <enml:County>Santa Clara</enml:County>
                <enml:USState>CA</enml:USState>
                <enml:Country>USA</enml:Country>
            </enml:NotaryUSJurisdiction>
        </enml:NotaryPublic>
    </enml:NotaryCertificates>
</enml:NotarizedDocument>
```
4.22 Element <NotaryCommissionExpiryDate>

The <NotaryCommissionExpiryDate> element is used to identify the date and time when the official commission of the Notary Public who notarized, or witnessed the signing of, the ENML-marked document, expires. It also specifies the time-zone in which this time falls.

Schema Definition:

```xml
<xsd:element
    name="NotaryCommissionExpiryDate"
    type="enml:DateTimeType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:simpleType name="DateTimeType">
    <xsd:restriction base="xsd:dateTime"/>
</xsd:simpleType>
```

The <NotaryCommissionExpiryDate> element of the type DateTimeType from the ENML schema, requires a date in the following format:

```
CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm]
```

where CCYY denotes the 4-digit calendar year, MM denotes the 2-digit month of the year and DD denotes the 2-digit day-of-month; the "T" is a fixed value; hh denotes the hour of day in 24-hour format, mm denotes the minute of the hour and ss denotes the second within the minute. The time-zone may be specified either with a fixed ‘Z’ which then implies that the preceding time is specified in Coordinated Universal Time (UTC), or it may be specified with the number of hours and minutes ahead or behind UTC, with a + or – sign respectively. There MUST be exactly ONE <NotaryCommissionExpiryDate> element within a <NotaryPublic> element.

Since the expiration date of a notary’s commission is usually specified by date, the time value of this element is always assumed to be 23:59:59 – the time at the last second of the specified date in its time-zone.

An example of the <NotaryCommissionExpiryDate> element follows:

**Example 1 – An example of a <NotaryCommissionExpiryDate> element:**

```xml
<enml:NotarizedDocument>
    <enml:NotarizedDocument>....</enml:NotarizedDocument>
    <enml:SignedDocuments>....</enml:SignedDocuments>
    <enml:DocumentSigners>....</enml:DocumentSigners>
    <enml:NotaryCertificates>
        <enml:NotaryCertificate>
            <enml:CertificateContent>....</enml:CertificateContent>
            <enml:NotaryPublic>
                <enml:NotaryName>....</enml:NotaryName>
                <enml:NotaryCommissionNumber>....</enml:NotaryCommissionNumber>
                <enml:NotaryCommissionExpiryDate>
                    2007-02-27T23:59:59-08:00
                </enml:NotaryCommissionExpiryDate>
            </enml:NotaryPublic>
        </enml:NotaryCertificate>
    </enml:NotaryCertificates>
</enml:NotarizedDocument>
```
4.23 Element <NotaryInternationalJurisdiction>

The element <NotaryInternationalJurisdiction>, of type `InternationalAddressType` defined in the ENML schema, is used to specify the legal jurisdiction in which the Notary Public, based outside the USA, is authorized to carry out his/her official duties. For Notaries who perform their official duties within the USA, the <NotaryUSJurisdiction> element must be used (see the next section). Software implementing ENML must choose the appropriate element based on the physical jurisdiction of the Notary Public and the notarization.

There MUST be exactly one <NotaryInternationalJurisdiction> element within the <NotaryPublic> element when it is chosen.

```
<internationalAddressType>
  <streetAddress1/>
  <streetAddress2/>
  <city/>
  <county/>
  <state/>
  <postalcode/>
  <country/>
</internationalAddressType>
```

Schema Definition:

```
<xsd:element name="NotaryInternationalJurisdiction"
  type="enml:InternationalAddressType"
  minOccurs="1"
  maxOccurs="1">
  </xsd:element>

<xsd:complexType name="InternationalAddressType">
  <xsd:sequence>
    ...<xsd:element name="streetAddress1" type="enml:streetAddressType"/>
    ...</xsd:sequence>
  </xsd:complexType>
```
The `<NotaryInternationalJurisdiction>` element is composed of a sequence of the following elements:

1. `<StreetAddress1>` [Optional]

   This optional element of type `StreetAddressType` defined in the ENML schema, carries the first line of the Notary's street address. When present, there MUST be exactly ONE `<StreetAddress1>` element in the `<NotaryInternationalJurisdiction>` element.

   The `<StreetAddress1>` element is specified in Section 4.33.

2. `<StreetAddress2>` [Optional]

   This optional element of type `StreetAddressType` defined in the ENML schema, carries the
second line of the Notary's street address. When present, there MUST be exactly ONE
<StreetAddress2> element in the <NotaryInternationalJurisdiction> element.

The <StreetAddress2> element is specified in Section 4.34.

3. <City> [Optional]

This optional element of type CityType defined in the ENML schema, carries the name of the city of the Notary's street address. When present, there MUST be exactly ONE <City> element in the <NotaryInternationalJurisdiction> element.

The <City> element is specified in Section 4.35.

4. <County> [Optional]

This optional element of type CountyType defined in the ENML schema, carries the name of the county of the Notary's street address. When present, there MUST be exactly ONE <County> element in the <NotaryInternationalJurisdiction> element.

The <County> element for international addresses is specified in Section 4.36.

5. <State> [Optional]

This optional element of type InternationalStateType defined in the ENML schema, carries the name of the state or province of the Notary's street address. When present, there MUST be exactly ONE <State> element in the <NotaryInternationalJurisdiction> element.

The <State> element is specified in Section 4.37.

6. <PostalCode> [Optional]

This optional element of type InternationalPostalCodeType defined in the ENML schema, carries the Postal Code of the Notary's street address. When present, there MUST be exactly ONE <PostalCode> element in the <NotaryInternationalJurisdiction> element.

The <PostalCode> element is specified in Section 4.38.

7. <Country> [Required]

This is the only required element (currently) in the Notary's international street address. This element of type CountryType defined in the ENML schema, carries the name of the country of the Notary's street address. There MUST be exactly ONE <Country> element in the <NotaryInternationalJurisdiction> element.

The <Country> element is specified in Section 4.39.

An example of the <NotaryInternationalJurisdiction> element is shown below:

Example 1 – A <NotaryInternationalJurisdiction> within a <NotarizedDocument>:

<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:NotaryCertificate>
      <enml:CertificateContent>....</enml:CertificateContent>
      <enml:NotaryPublic>
        <enml:NotaryName>....</enml:NotaryName>
        <enml:NotaryCommissionNumber>....</enml:NotaryCommissionNumber>
4.24 Element <NotaryUSJurisdiction>

The element <NotaryUSJurisdiction>, of type USAddressType defined in the ENML schema, is used to specify the legal jurisdiction in which the Notary Public, based inside the USA, is authorized to carry out his/her official duties. For Notaries who perform their official duties outside the USA, the <NotaryInternationalJurisdiction> element must be used (see the previous section). Software implementing ENML must choose the appropriate element based on the physical jurisdiction of the Notary Public and the notarization.

There MUST be exactly one <NotaryUSJurisdiction> element within the <NotaryPublic> element when it is chosen.

```
<xs:element name="NotaryUSJurisdiction" type="enml:USAddressType"
minOccurs="1"
maxOccurs="1">
</xs:element>
```
The `<NotaryUSJurisdiction>` element is composed of a sequence of the following elements:

1. `<StreetAddress1>` [Optional]
   
   This optional element of type `StreetAddressType` defined in the ENML schema, carries the first line of the Notary's street address. When present, there MUST be exactly ONE `<StreetAddress1>` element in the `<NotaryUSJurisdiction>` element.
   
   The `<StreetAddress1>` element is specified in Section 4.33.

2. `<StreetAddress2>` [Optional]
This optional element of type `StreetAddressType` defined in the ENML schema, carries the second line of the Notary's street address. When present, there MUST be exactly ONE `<StreetAddress2>` element in the `<NotaryUSJurisdiction>` element.

The `<StreetAddress2>` element is specified in Section 4.34.

3. `<City>` [Optional]

This optional element of type `CityType` defined in the ENML schema, carries the name of the city of the Notary's street address. When present, there MUST be exactly ONE `<City>` element in the `<NotaryUSJurisdiction>` element.

The `<City>` element is specified in Section 4.35.

4. `<County>` [Required]

This element of type `CountyType` defined in the ENML schema, carries the name of the county of the Notary's street address. There MUST be exactly ONE `<County>` element in the `<NotaryUSJurisdiction>` element.

The `<County>` element is specified in Section 4.36.

5. `<USState>` [Required]

This element of type `USStateCodeType` defined in the ENML schema, carries the 2-character abbreviation of the US-state or US-territory of the Notary's street address. There MUST be exactly ONE `<USState>` element in the `<NotaryUSJurisdiction>` element.

The `<USState>` element is specified in Section 4.40.

6. `<USZipCode>` [Optional]

This optional element of type `USZipCodeType` defined in the ENML schema, carries the ZIP Code of the Notary's street address. When present, there MUST be exactly ONE `<USZipCode>` element in the `<NotaryUSJurisdiction>` element.

The `<USZipCode>` element is specified in Section 4.41.

7. `<Country>` [Required]

This element of type `CountryTypeUSA` defined in the ENML schema, carries the full name or abbreviation of the United States of America. There MUST be exactly ONE `<Country>` element in the `<NotaryUSJurisdiction>` element.

The `<Country>` element for US-based addresses is specified in Section 4.42.

Some examples of the `<NotaryUSJurisdiction>` element are shown below:

**Example 1 – A `<NotaryUSJurisdiction>` within a `<NotarizedDocument>`:**

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:NotaryCertificate>
      <enml:CertificateContent>....</enml:CertificateContent>
      <enml:NotaryPublic>
        <enml:NotaryName>....</enml:NotaryName>
    </enml:NotaryCertificate>
  </enml:NotaryCertificates>
</enml:NotarizedDocument>
```
Example 2 – Another <NotaryUSJurisdiction> example:

<enml:NotaryUSJurisdiction>
  <enml:StreetAddress1>10846 Via San Marino</enml:StreetAddress1>
  <enml:City>Cupertino</enml:City>
  <enml:County>Santa Clara</enml:County>
  <enml:USState>CA</enml:USState>
  <enml:Country>United States of America</enml:Country>
</enml:NotaryUSJurisdiction>

4.25 Element <NotaryBondNumber>

The <NotaryBondNumber> element of the type NotaryBondNumberType from the ENML schema, provides the serial number of the Bond posted by the Notary Public before starting to perform his/her official duties. Such a bond is required in many US states and is public information. This number is typically issued by the insurance company that underwrites the Bond. However, since not all jurisdictions require a bond, this element is optional.

Schema Definition:

```xml
<xsd:element name="NotaryBondNumber" type="enml:NotaryBondNumberType" minOccurs="0" maxOccurs="1"/>
```

The optional <NotaryBondNumber> element permits the use of any alphanumeric character, to a maximum length of 255 characters, preserving white-space characters in its representation. When present, there MUST be exactly ONE <NotaryBondNumber> element inside a <NotaryPublic> element.
An example of the `<NotaryBondNumber>` element is shown below:

**Example 1 – An example of a `<NotaryBondNumber>`**

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:NotaryCertificate>
      <enml:CertificateContent>....</enml:CertificateContent>
    </enml:NotaryCertificate>
    <enml:NotaryPublic>
      <enml:NotaryName>....</enml:NotaryName> 
    </enml:NotaryPublic>
    <enml:NotaryCommissionNumber>....</enml:NotaryCommissionNumber>
    <enml:NotaryCommissionExpiryDate>2009-04-29T23:59:59-08:00</enml:NotaryCommissionExpiryDate>
    <enml:NotaryUSJurisdiction>
      <enml:County>Santa Clara</enml:County>
      <enml:USState>CA</enml:USState>
      <enml:Country>USA</enml:Country>
    </enml:NotaryUSJurisdiction>
    <enml:NotaryBondNumber>ABC-12345</enml:NotaryBondNumber>
  </enml:NotaryCertificates>
  <enml:NotarySignatures>....</enml:NotarySignatures>
</enml:NotarizedDocument>
```

### 4.26 Element `<PersonPrefixName>`

The `<PersonPrefixName>` element allows the ENML to carry the prefix of a document-signer's or Notary's name. This is not the same as the document-signer's or Notary's title, which has its own element.

The optional `<PersonPrefixName>` element of the `PersonPrefixNameType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 32-characters for the element, while preserving all white-space characters within its content.

**Schema Definition:**

```xml
<xsd:element
   name="PersonPrefixName"
   type="enml:PersonPrefixNameType"
   minOccurs="0"
   maxOccurs="1">
</xsd:element>

<xsd:simpleType name="PersonPrefixNameType">
<xsd:restriction base="xsd:string">
  <xsd:maxLength value="32"/>
  <xsd:whiteSpace value="preserve"/>
</xsd:simpleType>
```
Examples of the `<PersonPrefixName>` element are as follows.

**Example 1:**

```
<enml:PersonPrefixName>Mr.</enml:PersonPrefixName>
```

**Example 2:**

```
<enml:PersonPrefixName>Judge</enml:PersonPrefixName>
```

**Example 3:**

```
<enml:PersonPrefixName>Dr.</enml:PersonPrefixName>
```

### 4.27 Element `<PersonFirstName>` & `<PersonGivenName>`

The `<PersonFirstName>` and the `<PersonGivenName>` elements carry the first- or given-name (depending on the local convention where the notarization is being performed) of a document-signer's or Notary's name. Only one of these elements MUST be chosen by the software implementing ENML.

These required elements of the `PersonGivenNameType` from the ENML schema, use the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 32-characters for the element, while preserving all white-space characters within their content.

**Schema Definition:**

```xml
<xsd:element
    name="PersonFirstName"
    type="enml:PersonGivenNameType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:element
    name="PersonGivenName"
    type="enml:PersonGivenNameType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:simpleType name="PersonGivenNameType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="32"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```

Examples of the `<PersonFirstName>` and `<PersonGivenName>` element are as follows.

**Example 1:**

```
<enml:PersonFirstName>John</enml:PersonFirstName>
```
Example 2:

<enml:PersonFirstName>Arshad</enml:PersonFirstName>

Example 3:

<enml:PersonGivenName>Klaus</enml:PersonGivenName>

4.28 Element <PersonMiddleName>

The <PersonMiddleName> element allows the ENML to carry the middle-name or middle-initial of a document-signer's or Notary's name.

The optional <PersonMiddleName> element of the PersonMiddleNameType from the ENML schema, uses the xsd:string type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 32-characters for the element, while preserving all white-space characters within its content.

Schema Definition:

```xml
<xsd:element name="PersonMiddleName" type="enml:PersonMiddleNameType" minOccurs="0" maxOccurs="1">
</xsd:element>

<xsd:simpleType name="PersonMiddleNameType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="32"/>
    <xsd:whiteSpace value="preserve"/>
  </xsd:restriction>
</xsd:simpleType>
```

Examples of the <PersonMiddleName> element are as follows.

Example 1:

<enml:PersonMiddleName>M.</enml:PersonMiddleName>

Example 2:

<enml:PersonMiddleName>Anthony</enml:PersonMiddleName>

Example 3:

<enml:PersonMiddleName>Ibn</enml:PersonMiddleName>

4.29 Element <PersonLastName> & <PersonSurName>

The <PersonLastName> and the <PersonSurName> elements carry the last- or surname (depending on the local convention where the notarization is being performed) of a document-signer's or Notary's name. Only one of these elements MUST be chosen by the software implementing ENML.

These required elements of the PersonSurNameType from the ENML schema, use the xsd:string type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-
characters for the element, while preserving all white-space characters within their content.

**Schema Definition:**

```xml
<xs:element name="PersonLastName"
    type="enml:PersonSurNameType"
    minOccurs="1"
    maxOccurs="1">
</xs:element>

<xs:element name="PersonSurName"
    type="enml:PersonSurNameType"
    minOccurs="1"
    maxOccurs="1">
</xs:element>

<xs:simpleType name="PersonSurNameType">
    <xs:restriction base="xsd:string">
        <xs:maxLength value="64"/>
        <xs:whiteSpace value="preserve"/>
    </xs:restriction>
</xs:simpleType>
```

Examples of the `<PersonLastName>` and `<PersonSurName>` element are as follows.

**Example 1:**

```xml
<enml:PersonLastName>Doe</enml:PersonLastName>
```

**Example 2:**

```xml
<enml:PersonLastName>Noor</enml:PersonLastName>
```

**Example 3:**

```xml
<enml:PersonSurName>Weinerholst</enml:PersonSurName>
```

### 4.30 Element `<PersonSuffixName>`

The `<PersonSuffixName>` element allows the ENML to carry the suffix of a document-signer's or Notary's name. This is not the same as the document-signer's or Notary's title, which has its own element.

The optional `<PersonSuffixName>` element of the PersonSuffixNameType from the ENML schema, uses the xsd:string type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 32-characters for the element, while preserving all white-space characters within its content.

**Schema Definition:**

```xml
<xs:element name="PersonSuffixName"
```
Examples of the `<PersonSuffixName>` element are as follows.

Example 1:

<enml:PersonSuffixName>Jr.</enml:PersonSuffixName>

Example 2:

<enml:PersonSuffixName>III</enml:PersonSuffixName>

4.31 Element `<PersonMaidenName>`

The `<PersonMaidenName>` element allows the ENML to carry the maiden-name (unmarried last-name or surname) of a document-signer's or Notary's name, where necessary.

Note: While this author does not know of any requirement for this within US Notary laws, given that maiden-names may be used in some parts of the world as a custom, this optional element is made available in the schema.

The optional `<PersonMaidenName>` element of the `PersonMaidenNameType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.

Schema Definition:

```xml
<xsd:element
    name="PersonMaidenName"
    type="enml:PersonMaidenNameType"
    minOccurs="0"
    maxOccurs="1">
  </xsd:element>

<xsd:simpleType name="PersonMaidenNameType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="64"/>
    <xsd:whiteSpace value="preserve"/>
  </xsd:restriction>
</xsd:simpleType>
```

Examples of the `<PersonMaidenName>` element are as follows.
Example 1:
<enml:PersonMaidenName>Hepburn</enml:PersonMaidenName>

Example 2:
<enml:PersonMaidenName>Antoinette</enml:PersonMaidenName>

4.32 Element <PersonFullName>

The `<PersonFullName>` element allows the ENML to carry the full name – which includes the prefix, the first- or given-name, middle-name, last- or surname and suffix - of a document-signer's or Notary's name in a single element.

While it is possible for software to construct the full name of the document-signer or the Notary Public using the component elements described earlier, carrying the full-name of the person in ENML can save many software products a fair amount of processing time during verifications, for display on screens and for printing on reports. It is provided as a convenience rather than a requirement.

The optional `<PersonFullName>` element of the `PersonFullNameType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 256-characters for the element, while preserving all white-space characters within its content.

Schema Definition:

```xml
<xs:element name="PersonFullName"
    type="enml:PersonFullNameType"
    minOccurs="0"
    maxOccurs="1">
</xs:element>

<xs:simpleType name="PersonFullNameType">
    <xs:restriction base="xsd:string">
        <xs:maxLength value="256"/>
        <xs:whiteSpace value="preserve"/>
    </xs:restriction>
</xs:simpleType>
```

Examples of the `<PersonFullName>` element are as follows.

Example 1:
<enml:PersonFullName>Ms. Audrey Hepburn, Jr.</enml:PersonFullName>

Example 2:
<enml:PersonFullName>Mr. John Q. Doe III</enml:PersonFullName>

4.33 Element <StreetAddress1>

The `<StreetAddress1>` element allows the ENML to carry the first line of the street address of a physical location for a document-signer or a Notary Public.
The optional `<StreetAddress1>` element of the `StreetAddressType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.

**Schema Definition:**

```xml
<xsd:element name="StreetAddress1"
    type="enml:StreetAddressType"
    minOccurs="0"
    maxOccurs="1"/>
</xsd:element>

<xsd:simpleType name="StreetAddressType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="64"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```

Examples of the `<StreetAddress1>` element are as follows.

**Example 1:**

```xml
<enml:StreetAddress1>123 Main Street</enml:StreetAddress1>
```

**Example 2:**

```xml
<enml:StreetAddress1>Via Palazzio Real 23</enml:StreetAddress1>
```

### 4.34 Element `<StreetAddress2>`

The `<StreetAddress2>` element allows the ENML to carry the second line of the street address of a physical location for a document-signer or a Notary Public.

The optional `<StreetAddress2>` element, also of the `StreetAddressType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.

**Schema Definition:**

```xml
<xsd:element name="StreetAddress2"
    type="enml:StreetAddressType"
    minOccurs="0"
    maxOccurs="1"/>
</xsd:element>

<xsd:simpleType name="StreetAddressType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="64"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```
An example of the `<StreetAddress2>` element follows.

**Example 1:**

```enml
<enml:StreetAddress2>Suite No. 10</enml:StreetAddress2>
```

### 4.35 Element `<City>`

The `<City>` element allows the ENML to carry the name of the city of the street address for a document-signer or a Notary Public.

The optional `<City>` element, also of the `CityType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.

**Schema Definition:**

```xml
<xsd:element name="City"
    type="enml:CityType"
    minOccurs="0"
    maxOccurs="1"/>
</xsd:element>

<xsd:simpleType name="CityType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="64"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```

An example of the `<City>` element is:

**Example 1:**

```enml
<enml:City>Cupertino</enml:City>
```

### 4.36 Element `<County>`

The `<County>` element allows the ENML to carry the name of the county or municipality of the street address for a document-signer or a Notary Public. While the `<County>` element is optional within an international address, it is a required element for notarized or witnessed documents with US-based addresses. This is because Notaries in the US are commissioned by county within a State. While they may perform their official duties outside their county within their state, they must indicate the name of their county in notarized, or witnessed for signature, documents.

The `<County>` element, also of the `CountyType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.
An example of the `<County>` element is:

Example 1:

```xml
<enml:County>Santa Clara</enml:County>
```

### 4.37 Element `<State>`

The `<State>` element allows the ENML to carry the name of the state or province of the street address for a document-signer or a Notary Public in an international location; for US-based locations, the appropriate element is `<USState>`. While the `<State>` element is optional within an international address, it is a required element for notorized or witnessed documents with US-based addresses.

The `<State>` element, also of the `InternationalStateType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.
Examples of the `<State>` element are:

**Example 1 – A `<State>` in Canada:**

    <enml:State>Ontario</enml:State>

**Example 2 – A `<State>` (province) in the Netherlands:**

    <enml:State>Utrecht</enml:State>

### 4.38 Element `<PostalCode>`

The `<PostalCode>` element allows the ENML to carry the postal code of the street address for a document-signer or a Notary Public in an international location; for US-based locations, the appropriate element is `<USZipCode>`.

The optional `<PostalCode>` element, of the `InternationalPostalCodeType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.

**Schema Definition (for International addresses/jurisdictions):**

```xml
<xsd:element name="PostalCode"
    type="enml:InternationalPostalCodeType"
    minOccurs="0"
    maxOccurs="1"/>
</xsd:element>

<xsd:simpleType name="InternationalPostalCodeType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="64"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```

An example of the `<PostalCode>` element is:

**Example 1 – A `<PostalCode>` in Canada:**


### 4.39 Element `<Country>` (International)

The `<Country>` element allows the ENML to carry the name of an international country, of the street address for a document-signer or a Notary Public.

The required `<Country>` element is of the `CountryType` from the ENML schema for an international location, and of the `CountryTypeUSA` for a US-based location. The `CountryType` uses the `xsd:string`
type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-
characters for the element, while preserving all white-space characters within its content.

Schema Definition:

```xml
<xsd:element name="Country"
    type="enml:CountryType"
    minOccurs="1"
    maxOccurs="1"/>
</xsd:element>

<xsd:simpleType name="CountryType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="64"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```

Examples of the `<Country>` element are:

**Example 1:**

```xml
<enml:Country>Canada</enml:Country>
```

**Example 2:**

```xml
<enml:Country>United Kingdom</enml:Country>
```

### 4.40 Element `<USState>`

The `<USState>` element allows the ENML to carry the name of the state or territory of the street
address for a document-signer or a Notary Public in a US-based location. For international locations, the
appropriate element is `<State>`.

The required `<USState>` element, of the `USStateCodeType` from the ENML schema, uses the
`xsd:token` type from the XML Schema Definition as the base type, with a restriction to choose one of
restricted set of enumerations. The enumerations are highlighted in the Schema Definition below; the
annotations from the schema-definition has been left in-place to avoid duplicating the explanation.

There MUST be exactly ONE `<USState>` element with any element that uses a US-based address in
ENML.

Schema Definition:

```xml
<xsd:element name="USState"
    type="enml:USStateCodeType"
    minOccurs="1"
    maxOccurs="1"/>
</xsd:element>

<xsd:simpleType name="USStateCodeType">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">
```

ENML-1.0-Specification
Copyright © OASIS® 2009. All Rights Reserved.
A code list that enumerates the states of the USA.

<xsd:annotation>
  <xsd:documentation xml:lang="en-US">Armed Forces Americas (except Canada)
</xsd:documentation>
</xsd:annotation>

<xsd:enumeration value="AA">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">Armed Forces Americas (except Canada)
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="AE">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">Armed Forces Africa, Canada, Europe, Middle East
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="AK">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">ALASKA
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="AL">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">ALABAMA
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="AP">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">Armed Forces Pacific
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="AR">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">ARKANSAS
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="AS">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">AMERICAN SAMOA
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="AZ">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">ARIZONA
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="CA">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">CALIFORNIA
</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CO">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">COLORADO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CT">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">CONNECTICUT</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DC">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">DISTRICT OF COLUMBIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DE">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">DELAWARE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FL">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">FLORIDA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FM">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">FEDERATED STATES OF MICRONESIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GA">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">GEORGIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GU">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">GUAM</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HI">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">HAWAI'I</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IA">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">IOWA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ID">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">IDAHO</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IL">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">ILLINOIS</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IN">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">INDIANA</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KS">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">KANSAS</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KY">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">KENTUCKY</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LA">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">LOUISIANA</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MA">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">MASSACHUSETTS</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MD">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">MARYLAND</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ME">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">MAINE</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MH">
    <xsd:annotation>
        <xsd:documentation xml:lang="en-US">MARSHALL ISLANDS</xsd:documentation>
    </xsd:annotation>
</xsd:enumeration>
<table>
<thead>
<tr>
<th>State Code</th>
<th>State Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>MICHIGAN</td>
</tr>
<tr>
<td>MN</td>
<td>MINNESOTA</td>
</tr>
<tr>
<td>MO</td>
<td>MISSOURI</td>
</tr>
<tr>
<td>MP</td>
<td>NORTHERN MARIANA ISLANDS</td>
</tr>
<tr>
<td>MS</td>
<td>MISSISSIPPI</td>
</tr>
<tr>
<td>MT</td>
<td>MONTANA</td>
</tr>
<tr>
<td>NC</td>
<td>NORTH CAROLINA</td>
</tr>
<tr>
<td>ND</td>
<td>NORTH DAKOTA</td>
</tr>
<tr>
<td>NE</td>
<td>NEBRASKA</td>
</tr>
<tr>
<td>NH</td>
<td>NEW HAMPSHIRE</td>
</tr>
</tbody>
</table>
<xsd:enumeration value="RI">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">RHODE ISLAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="SC">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">SOUTH CAROLINA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="SD">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">SOUTH DAKOTA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="TN">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">TENNESSEE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="TX">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">TEXAS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="UT">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">UTAH</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="VA">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">VIRGINIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="VI">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">VIRGIN ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="VT">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">VERMONT</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>

<xsd:enumeration value="WA">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-US">WASHINGTON</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
4.41 Element `<USZipCode>`

The `<USZipCode>` element allows the ENML to carry the postal code of the street address for a document-signer or a Notary Public in a US-based location; for international locations, the appropriate element is `<PostalCode>`.

The optional `<USZipCode>` element, of the `USZipCodeType` from the ENML schema, uses the `xsd:string` type from the XML Schema Definition as the base type, with a restriction that indicates that the zip-code may either be one of the following:

1. A 5-digit numeral; or
2. A 5-digit numeral followed by a literal hyphen (“-”) and a 4-digit numeral.

Schema Definition:

```xml
<xsd:element
    name="USZipCode"
    type="enml:USZipCodeType"
    minOccurs="0"
    maxOccurs="1"/>
```

```xml
<xsd:simpleType name="USZipCodeType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="\d{5}|-\d{4}"/>
    </xsd:restriction>
</xsd:simpleType>
```

Examples of the `<USZipCode>` element are:

**Example 1** – A 5-digit `<USZipCode>`:

```xml
<USZipCode>12345</USZipCode>
```
Example 2 – A 9-digit <USZipCode>:

<enml:USZipCode>95014-6333</enml:USZipCode>

### 4.42 Element <Country> (USA)

The <Country> element allows the ENML to carry the name of the country of the street address for a document-signer or a Notary Public.

In the case of the USA, the required <Country> element is of the CountryTypeUSA from the ENML schema; for an international location it is of the CountryType. The CountryTypeUSA uses the xsd:token type from the XML Schema Definition as the base type, with a restriction to choose one of two restricted enumerations. The enumerations are highlighted in the Schema Definition below.

There MUST be exactly ONE <Country> element within an element that uses a US-based address in ENML.

**Schema Definition (for US addresses/jurisdictions):**

```xml
<xsd:element name="Country"
    type="enml:CountryTypeUSA"
    minOccurs="1"
    maxOccurs="1"/>

<xsd:simpleType name="CountryTypeUSA">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="USA"/>
        <xsd:enumeration value="United States of America"/>
    </xsd:restriction>
</xsd:simpleType>
```

The two valid examples of the use of the <Country> element for US-based addresses (within ENML) are:

**Example 1:**

<enml:Country>USA</enml:Country>

**Example 2:**

<enml:Country>United States of America</enml:Country>
4.43 Element <NotarizationType>

The `<NotarizationType>` element, of the type `NotarizationActType` defined in the ENML schema, identifies the type of notarization acts that is described within the `<NotarizedDocument>`.

Schema Definition:

```xml
<xsd:element name="NotarizationType"
    type="enml:NotarizationActType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:simpleType name="NotarizationActType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="Acknowledgment"/>
        <xsd:enumeration value="Attestation"/>
        <xsd:enumeration value="CertifiedCopy"/>
        <xsd:enumeration value="Jurat"/>
        <xsd:enumeration value="Other"/>
    </xsd:restriction>
</xsd:simpleType>
```

The `NotarizationActType` is an enumerated list of `xsd:token` types from the XML Schema Definition, and consists of a choice of one the following tokens:

- Acknowledgment
- Attestation
- CertifiedCopy
- Jurat
- Other

There MUST be exactly ONE `<NotarizationType>` element within a `<CertificateContent>` element.
Example 1 – An example of a <NotarizationType> element specifying that the notarization act was an Acknowledgment:

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:CertificateContent>
      <enml:NotarizationType>Accreditation</enml:NotarizationType>
      <enml:NotarizationDate>2007-02-07T15:19:17-08:00</enml:NotarizationDate>
      <enml:NotarizationUSLocation>
        <enml:City>Cupertino</enml:City>
        <enml:County>Santa Clara</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:NotarizationUSLocation>
      <enml:StatutoryContent>
        State of California
        County of Santa Clara
        On February 07 2007, before me Arshad Noor, personally
        appeared John Doe, who proved to me on the basis of
        satisfactory evidence to be the person whose name is
        subscribed to the within instrument and acknowledged
        to
        me that he executed the same in his authorized
        capacity,
        and that by his signature on the instrument the
        person,
        or the entity upon behalf of which the person acted,
        executed the instrument.
        I certify under PENALTY OF PERJURY under the laws of
        the
        the State of California that the foregoing paragraph is
        true
        and correct.
        WITNESS my hand and official seal.
      </enml:StatutoryContent>
    </enml:CertificateContent>
  </enml:NotaryCertificates>
</enml:NotarizedDocument>
```

4.44 Element <NotarizationDate>

The <NotarizationDate> element is used to identify the date and time of the notarization or witnessing act. It is the date and time the <NotarizedDocument> or <WitnessedDocument> was issued.

Schema Definition:
The `<NotarizationDate>` element of the type `DateTimeType` from the ENML schema, requires a date in the following format:

CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm]

where CCYY denotes the 4-digit calendar year, MM denotes the 2-digit month of the year and DD denotes the 2-digit day-of-month; the "T" is a fixed value; hh denotes the hour of day in 24-hour format, mm denotes the minute of the hour and ss denotes the second within the minute. The time-zone may be specified either with a fixed "Z" which then implies that the preceding time is specified in Coordinated Universal Time (UTC), or it may be specified with the number of hours and minutes ahead or behind UTC, with a + or – sign respectively.

There MUST be exactly ONE `<NotarizationDate>` element within a `<CertificateContent>` element.

Example 1 – An example of a `<NotarizationDate>` element:

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
  <enml:DocumentSigners>....</enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:CertificateContent>
      <enml:NotarizationType>Acknowledgment</enml:NotarizationType>
      <enml:NotarizationDate>2007-01-27T13:01:05-08:00</enml:NotarizationDate>
      <enml:NotarizationUSLocation>
        <enml:City>Cupertino</enml:City>
        <enml:County>Santa Clara</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:NotarizationUSLocation>
      <enml:StatutoryContent>...</enml:StatutoryContent>
      <enml:CertificateContent>
        <enml:NotaryPublic>....</enml:NotaryPublic>
      </enml:CertificateContent>
    </enml:CertificateContent>
    <enml:NotarySignatures>....</enml:NotarySignatures>
  </enml:NotaryCertificates>
</enml:NotarizedDocument>
```

4.45 Element `<NotarizationInternationalLocation>`

The element `<NotarizationInternationalLocation>`, of type `InternationalAddressType` defined in the ENML schema, is used to carry the name of the physical location outside the USA where the `<NotarizedDocument>` was issued.

When used, there SHALL be only one `<NotarizationInternationalLocation>` element within the `<NotarizedDocument>` element. While issuers of `<NotarizedDocument>` are permitted to use as many
elements as required from the *InternationalAddressType*, only the `<Country>` element is mandatory.

```
NotarizationInternationalLocation
   StreetAddress1
   StreetAddress2
   City
   County
   State
   PostalCode
   Country
```

**Schema Definition:**

```xml
<xsd:element
   name="NotarizationInternationalLocation"
   type="enml:InternationalAddressType"
   minOccurs="1"
   maxOccurs="1">
</xsd:element>
```

Note: The elements of *InternationalAddressType* are defined in Section 4.15.

**Example 1 – An example of a `<NotarizationInternationalLocation>` element:**

```xml
<enml:NotarizedDocument>
   <enml:SignedDocuments>.... </enml:SignedDocuments>
   <enml:DocumentSigners>.... </enml:DocumentSigners>
   <enml:NotaryCertificates>
      <enml:CertificateContent>
         <enml:NotarizationType>Acknowledgment</enml:NotarizationType>
         <enml:NotarizationDate>2007-02-07T15:19:17-08:00</enml:NotarizationDate>
         <enml:NotarizationInternationalLocation>
            <enml:StreetAddress1>490 Sussex Drive</enml:StreetAddress1>
            <enml:City>Ottawa</enml:City>
            <enml:State>Ontario</enml:State>
            <enml:Country>Canada</enml:Country>
         </enml:NotarizationInternationalLocation>
         <enml:StatutoryContent>...</enml:StatutoryContent>
         <enml:CertificateContent>
         <enml:NotaryPublic>....</enml:NotaryPublic>
      </enml:CertificateContent>
   </enml:NotaryCertificates>
</enml:NotarizedDocument>
```
4.46 Element <NotarizationUSLocation>

The element <NotarizationUSLocation>, of type USAddressType defined in the ENML schema, is used to identify the name of the physical location inside the USA where the <NotarizedDocument> was issued.

When used, there SHALL be only one <NotarizationUSLocation> element within the <NotarizedDocument> element. While issuers of <NotarizedDocument> are permitted to use as many elements as required from the USAddressType, only the <County>, <State> and <Country> elements are mandatory.

Schema Definition:

```xml
<xsd:element
  name="NotarizationUSLocation"
  type="enml:USAddressType"
  minOccurs="1"
  maxOccurs="1">
</xsd:element>
```

Note: The elements of USAddressType are defined in Section 4.16.

Example 1 – An example of a <NotarizationUSLocation> element that shows additional elements from the USAddressType in addition to the required sub-elements:

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>....</enml:SignedDocuments>
</enml:NotarizedDocument>
```

IssuedAtUSLocation

- StreetAddress1
- StreetAddress2
- City
- County
- US State
- US ZipCode
- Country
Example 2 – An example of a <NotarizationUSLocation> element with minimum required sub-elements:

```xml
<enml:NotarizedDocument>
  <enml:SignedDocuments>.... </enml:SignedDocuments>
  <enml:DocumentSigners>.... </enml:DocumentSigners>
  <enml:NotaryCertificates>
    <enml:CertificateContent>
      <enml:NotarizationType>Acknowledgment</enml:NotarizationType>
      <enml:NotarizationDate>2007-02-07T15:19:17-08:00</enml:NotarizationDate>
      <enml:NotarizationUSLocation>
        <enml:StreetAddress1>10846 Via San Marino</enml:StreetAddress1>
        <enml:City>Cupertino</enml:City>
        <enml:County>Santa Clara</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
      </enml:NotarizationUSLocation>
      <enml:StatutoryContent>...</enml:StatutoryContent>
    </enml:CertificateContent>
  </enml:NotaryCertificates>
  <enml:NotaryPublic>.... </enml:NotaryPublic>
  <enml:NotarySignatures>.... </enml:NotarySignatures>
</enml:NotarizedDocument>
```

4.47 Element <ApostilleContent>

The <ApostilleContent> element of the ApostilleContentType, carries information about the apostille – the certified document issued by a competent authority for a jurisdiction, certifying the signature of the public official who signed the embedded document: either a <NotarizedDocument> or <WitnessedDocument>.

The <ApostilleContent> element can be graphically represented as follows:
<xsd:element name="ApostilleContent"
    type="enml:ApostilleContentType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:complexType name="ApostilleContentType">
    <xsd:sequence>
        <xsd:element name="IssuingCountry"
            type="enml:CountryType"
            minOccurs="1"
            maxOccurs="1">
        </xsd:element>
        <xsd:element name="ApostilleNumber"
            type="enml:ApostilleNumberType"
            minOccurs="1"
            maxOccurs="1">
        </xsd:element>
        <xsd:element name="ApostilleDate"
            type="enml:DateTimeType"
            minOccurs="1"
            maxOccurs="1">
        </xsd:element>
        <xsd:element name="IssuedAtInternationalLocation">
        </xsd:element>
        <xsd:element name="IssuedAtUSLocation">
        </xsd:element>
        <xsd:element name="SignerTitle">
        </xsd:element>
        <xsd:element name="StatutoryContent">
        </xsd:element>
        <xsd:element name="DocumentSigningOfficialName">
        </xsd:element>
        <xsd:element name="DocumentSigningOfficialTitle">
        </xsd:element>
        <xsd:element name="DocumentSigningOfficialSeal">
        </xsd:element>
        <xsd:element name="ID">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="IssuedAtInternationalLocation"
    type="enml:InternationalAddressType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:element name="IssuedAtUSLocation"
    type="enml:USAddressType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:element name="SignerTitle"
    type="enml:PersonTitleType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:element name="StatutoryContent"
    minOccurs="1"
    maxOccurs="1">
    <xsd:simpleType>
        <xsd:restriction base="xsd:token">
            <xsd:enumeration
                value="(Convention de La Haye du 5 octobre 1961)"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>

<xsd:element name="DocumentSigningOfficialName"
    type="enml:PersonNameType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:element name="DocumentSigningOfficialTitle"
    type="enml:PersonTitleType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:element name="DocumentSigningOfficialSeal"
    type="enml:ApostilleSealType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

</xsd:complexType>
There MUST be exactly ONE <ApostilleContent> element in an <ApostillizedDocument> element.

The <ApostilleContent> element consists of a sequence of nine children elements and one ID attribute, all of which are required.

1. <IssuingCountry> [Required]
   
   This element of type CountryType, identifies the full name of the country issuing the <ApostillizedDocument>. There MUST be exactly ONE <IssuingCountry> element in an <ApostilleContent> element.

   The <IssuingCountry> element is specified in Section 4.48.

2. <ApostilleNumber> [Required]
   
   This element of type ApostilleNumberType, identifies the unique number of the apostille from the issuing-country. This number is, typically, determined by the competent authority of the jurisdiction issuing the apostille.

   While each country is permitted to use numbers of its choice, the ENML specification recommends using the Processing Rule for ENML ID-types (Section 4.x) for this element. The advantage of doing so ensures a unique <ApostilleNumber> without the need to reference any other element. If this Processing Rule is not used, applications must qualify the <ApostilleNumber> with the <IssuingCountry> element to ensure uniqueness. There MUST be exactly ONE <ApostilleNumber> element in an <ApostilleContent> element.

   The <ApostilleNumber> element is specified in Section 4.49.

3. <ApostilleDate> [Required]
   
   This element of type DateTimeType, identifies the date, time and time-zone of the apostille from the issuing-country. There MUST be exactly ONE <ApostilleDate> element in an <ApostilleContent> element.

   The <ApostilleDate> element is specified in Section 4.50.

4. <IssuedAtInternationalLocation> or <IssuedAtUSLocation> [Required]
   
   This element represents the physical location where the apostille was issued. Depending on the geographical location – either an international or US-based location - this element must be a choice of either the <IssuedAtInternationalLocation> or the <IssuedAtUSLocation> element, respectively. It is the responsibility of the application generating the ENML-marked document to choose the appropriate element based on the jurisdiction of the certificate-issuer.

   There MUST be exactly ONE <IssuedAtInternationalLocation> or <IssuedAtUSLocation> element in an <ApostilleContent> element.

   The <IssuedAtInternationalLocation> element is specified in Section 4.51.

   The <IssuedAtUSLocation> element is specified in Section 4.52.

5. <SignerTitle> [Required]
   
   The <SignerTitle> element of PersonTitleType indicates who signed the apostille, thus certifying the signature of the public-official in the embedded <NotarizedDocument> or <WitnessedDocument>.
There MUST be exactly ONE `<SignerTitle>` element in a `<ApostilleContent>` element. The `<SignerTitle>` element is specified in Section 4.14.

6. `<StatutoryContent>` [Required]

This element is restricted to an `xsd:token` from the XML Schema Definition, and allows only a single enumeration with the required language necessary for an apostille. This requirement token is:

(Convention de La Haye du 5 octobre 1961)

There MUST be exactly ONE `<StatutoryContent>` element within an `<ApostilleContent>` element.

7. `<DocumentSigningOfficialName>` [Required]

This element of the `PersonNameType` carries the full name of the Notary Public who notarized or witnessed the signing of the embedded document within the `<ApostillizedDocument>`.

There MUST be exactly ONE `<DocumentSigningOfficialName>` within an `<ApostilleContent>` element.

The `<DocumentSigningOfficialName>` element is specified in Section 4.53.

8. `<DocumentSigningOfficialTitle>` [Required]

This element of the `PersonTitleType` carries the official title of the Notary Public who notarized or witnessed the signing of the embedded document within the `<ApostillizedDocument>`.

There MUST be exactly ONE `<DocumentSigningOfficialTitle>` within an `<ApostilleContent>` element.

The `<DocumentSigningOfficialTitle>` element is specified in Section 4.54.


This element of the `ApostilleSealType` carries the official seal of the Notary Public who notarized or witnessed the signing of the embedded document within the `<ApostillizedDocument>`.

The seal information may be textual, a graphic image or an X509 digital certificate. The choice is left up to the application creating the ENML for the `<ApostillizedDocument>`.

There MUST be exactly ONE `<DocumentSigningOfficialSeal>` within an `<ApostilleContent>` element.

The `<DocumentSigningOfficialSeal>` element is specified in Section 4.55.

10. ID attribute [Required]

The ID (identifier) attribute MUST be used to identify the `<ApostilleContent>` uniquely from other elements within the ENML. This is necessary to ensure that the signature of the apostille-signer covers the contents of the `<ApostilleContent>` element.

Please see Section 4.53 for a discussion of the Processing Rule for ID attributes within ENML.

Example 1 – An example of an `<ApostillizedDocument>` element issued in the USA:
Example 2 – An example of an <ApostillizedDocument> element issued in Canada:

<enml:ApostillizedDocument>
  <enml:NotarizedDocument>....</enml:NotarizedDocument>
  <enml:ApostilleContent Id="ApostilleContent-CA-ON-XVL65986-1226972923-89e89c11ca0335b37c1030f7e85fca5527c0d4cc61479fc88526c38e01d912a5"/>
  <enml:IssuingCountry>Canada</enml:IssuingCountry>
  <enml:ApostilleNumber>CA-ON-XVL65986-1226971821-c4e7f79729ad7da392d3a18023243532ea30067350d4698974d5a2e7bd09ee4</enml:ApostilleNumber>
  <enml:ApostilleDate>2008-11-17T12:13:14-08:00</enml:ApostilleDate>
  <enml:IssuedAtInternationalLocation>
    <enml:City>Ottawa</enml:City>
    <enml:State>Ontario</enml:State>
    <enml:Country>Canada</enml:Country>
  </enml:IssuedAtInternationalLocation>
  <enml:SignerTitle>Secretary of State, State of Ontario</enml:SignerTitle>
  <enml:StatutoryContent>(Convention de La Haye du 5 octobre 1961)</enml:StatutoryContent>
</enml:ApostillizedDocument>
4.48 Element <IssuingCountry>

The <IssuingCountry> element carries the name of the country issuing an <ApostillizedDocument>.

The required <IssuingCountry> element is of the CountryType from the ENML schema. The CountryType uses the xsd:string type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.

Schema Definition:

```xml
<xsd:element name="IssuingCountry"
             type="enml:CountryType"
             minOccurs="1"
             maxOccurs="1"/>
</xsd:element>

<xsd:simpleType name="CountryType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="64"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```

Example 1 – An example of an <ApostillizedDocument> with an <IssuingCountry> element.

```xml
<enml:ApostillizedDocument>
    <enml:NotarizedDocument>....</enml:NotarizedDocument>
    <enml:ApostilleContent Id="ApostilleContent-US-CA-1565986-1226972923-89e89c11cac335b3d7c1030f78ffca5527c6d4cc61479fc88526c38e01d912a5">
        <enml:IssuingCountry>United States of America</enml:IssuingCountry>
        <enml:ApostilleNumber>US-CA-1565986-1226971821-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4</enml:ApostilleNumber>
        <enml:ApostilleDate>2008-11-17T12:13:14-08:00</enml:ApostilleDate>
    </enml:ApostilleContent>
</enml:ApostillizedDocument>
```
4.49 Element <ApostilleNumber>

The <ApostilleNumber> element carries the unique certificate number of the <ApostilledDocument>.

The required <ApostilleNumber> element is of the <ApostilleNumberType> from the ENML schema. The <ApostilleNumberType> uses the <xsd:string> type from the XML Schema Definition as the base type, with a restriction to allow up a maximum of 64-characters for the element, while preserving all white-space characters within its content.

Schema Definition:

```xml
<xsd:element name="ApostilleNumber"
    type="enml:ApostilleNumberType"
    minOccurs="1"
    maxOccurs="1"/>
</xsd:element>

<xsd:simpleType name="ApostilleNumberType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="1024"/>
        <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
</xsd:simpleType>
```
Example 1 – An example of an <ApostillizedDocument> with an <ApostilleNumber> element. This example uses the ENML Processing Rule derived value (highly recommended):

```xml
<enml:ApostillizedDocument>
  <enml:NotarizedDocument>....</enml:NotarizedDocument>
  <enml:ApostilleContent Id="ApostilleContent-US-CA-1565986-1226972923-89e89c11cac335b3d7c1030f78fca5527cbd4cc61479fc88526c38e01d912a5">
    <enml:IssuingCountry>United States of America</enml:IssuingCountry>
    <enml:ApostilleNumber>ApostilleNumber-US-CA-1565986-1226971821-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4</enml:ApostilleNumber>
    <enml:ApostilleDate>2008-11-17T12:13:14-08:00</enml:ApostilleDate>
    <enml:IssuedAtUSLocation>
      <enml:County>Sacramento</enml:County>
      <enml:USState>CA</enml:USState>
      <enml:Country>USA</enml:Country>
    </enml:IssuedAtUSLocation>
    <enml:SignerTitle>Secretary of State, State of California</enml:SignerTitle>
    <enml:StatutoryContent>(Convention de La Haye du 5 octobre 1961)</enml:StatutoryContent>
    <enml:DocumentSigningOfficialName>
      <enml:PersonFirstName>Arshad</enml:PersonFirstName>
      <enml:PersonLastName>Noor</enml:PersonLastName>
    </enml:DocumentSigningOfficialName>
    <enml:DocumentSigningOfficialTitle>Notary Public</enml:DocumentSigningOfficialTitle>
    <enml:DocumentSigningOfficialSeal>
      <enml:TextSeal>Notary Public, State of California</enml:TextSeal>
    </enml:DocumentSigningOfficialSeal>
  </enml:ApostilleContent>
</enml:ApostillizedDocument>
```

4.50 Element <ApostilleDate>

The <ApostilleDate> element is used to identify the date and time when the electronic <ApostillizedDocument> was issued. It also specifies the time-zone where the apostille is signed.

The <ApostilleDate> element of the type Date-Time-Type from the ENML schema, requires a date in the following format:

```
CCYY-MM-DDThh:mm:ssZ[+|-]hh:mm]
```

where CCYY denotes the 4-digit calendar year, MM denotes the 2-digit month of the year and DD denotes the 2-digit day-of-month; the “T” is a fixed value; hh denotes the hour of day in 24-hour format, mm denotes the minute of the hour and ss denotes the second within the minute. The time-zone may be specified either with a fixed ‘Z’ which then implies that the preceding time is specified in Coordinated
Universal Time (UTC), or it may be specified with the number of hours and minutes ahead or behind UTC, with a + or – sign respectively. There MUST be exactly ONE <ApostilleDate> element within a <ApostillizedDocument> element.

**Schema Definition:**

```xml
<xsd:element
    name="ApostilleDate"
    type="enml:DateTimeType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:simpleType name="DateTimeType">
    <xsd:restriction base="xsd:dateTime"/>
</xsd:simpleType>
```

**Example 1 – An example of an <ApostillizedDocument> with an <ApostilleDate> element.**

```xml
<enml:ApostillizedDocument>
    <enml:NotarizedDocument>.... </enml:NotarizedDocument>
    <enml:ApostilleContent Id="ApostilleContent- US-CA-1565986-1226972923-89e8c1cac335b3d7c1030f78ffca5527c883626c38e01d912a5"/>
    <enml:IssuingCountry>United States of America</enml:IssuingCountry>
    <enml:ApostilleNumber>ApptistleNumber-US-CA-1565986-1226971821-c4e7f79729ad7da392d3a180232343532a0e30067350d4698974d5a2e7bd09ee4</enml:ApostilleNumber>
    <enml:ApostilleDate>2008-11-17T12:36:45-08:00</enml:ApostilleDate>
    <enml:IssuedAtUSLocation>
        <enml:County>Sacramento</enml:County>
        <enml:USState>CA</enml:USState>
        <enml:Country>USA</enml:Country>
    </enml:IssuedAtUSLocation>
    <enml:SignerTitle>
        Secretary of State, State of California
    </enml:SignerTitle>
    <enml:StatutoryContent>
        (Convention de La Haye du 5 octobre 1961)
    </enml:StatutoryContent>
    <enml:DocumentSigningOfficialName>
        <enml:PersonFirstName>Arshad</enml:PersonFirstName>
        <enml:PersonLastName>Noor</enml:PersonLastName>
    </enml:DocumentSigningOfficialName>
    <enml:DocumentSigningOfficialTitle>
        Notary Public
    </enml:DocumentSigningOfficialTitle>
    <enml:DocumentSigningOfficialSeal>
        <enml:TextSeal>
            Notary Public, State of California
        </enml:TextSeal>
    </enml:DocumentSigningOfficialSeal>
</enml:ApostillizedDocument>
```
4.51 Element <IssuedAtInternationalLocation>

The element <IssuedAtInternationalLocation>, of type InternationalAddressType defined in the ENML schema, is used to carry the name of the physical location outside the USA where the <ApostillizedDocument> was issued.

When used, there SHALL be only one <IssuedAtInternationalLocation> element within the <ApostillizedDocument> element. While issuers of <ApostillizedDocument> are permitted to use as many elements as required from the InternationalAddressType, only the <Country> element is mandatory. Based on current conventions, this is sufficient for an <ApostillizedDocument>.

---

**Schema Definition:**

```xml
<xsd:element
    name="IssuedAtInternationalLocation"
    type="enml:InternationalAddressType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>
```

Note: The elements of InternationalAddressType are defined in Section 4.15.

---

**Example 1** – An example of an <ApostillizedDocument> with an <IssuedAtInternationalLocation> element. The <City> and <State> elements are not mandatory for an <ApostillizedDocument>, but are shown here as an example of additional elements that can be included in the <IssuedAtInternationalLocation> element:

```xml
<enml:ApostillizedDocument>
    <enml:NotarizedDocument>....  </enml:NotarizedDocument>
</enml:ApostillizedDocument>
```
4.52 Element <IssuedAtUSLocation>

The element <IssuedAtUSLocation>, of type USAddressType defined in the ENML schema, is used to carry the name of the physical location inside the USA where the <ApostillizedDocument> was issued.

When used, there SHALL be only one <IssuedAtUSLocation> element within the <ApostillizedDocument> element. While issuers of <ApostillizedDocument> are permitted to use as many elements as required from the USAddressType, only the <City>, <State> and <Country> elements are mandatory. Based on current conventions, this is more than sufficient for an US-issued <ApostillizedDocument>.

Schema Definition:

```
<xsd:element
    name="IssuedAtUSLocation"
    type="enml:USAddressType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>
```

Note: The elements of USAddressType are defined in Section 4.16.

Example 1 – An example of an <ApostillizedDocument> with an <IssuedAtUSLocation> element. The <County> and <State> elements are not mandatory for an <ApostillizedDocument>, but are necessary for an USAddressType and are included here:

```
<enml:ApostillizedDocument>
    <enml:NotarizedDocument>....  </enml:NotarizedDocument>
    <enml:ApostilleContent Id="ApostilleContent-US-CA-1565986-1226972923-89e89c11cac335b3d7c1030f78ffca5527cbd4cc61479fc88526c38e01d912a5">
```

```
    <enml:IssuedAtUSLocation>
        <enml:StreetAddress1>
            <enml:StreetAddress2>
                <enml:City>
                    <enml:County>
                        <enml:USState>
                            <enml:USZipCode>
                                <enml:Country>
```

```
    </enml:Country>
</enml:USZipCode>
</enml:USState>
</enml:County>
</enml:USState>
</enml:City>
</enml:StreetAddress2>
</enml:StreetAddress1>
```
<enml:IssuingCountry>United States of America</enml:IssuingCountry>
<enml:ApostilleNumber>
    ApostilleNumber-US-CA-1565986-1226971821-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4
</enml:ApostilleNumber>
<enml:ApostilleDate>2008-11-17T09:23:45-08:00</enml:ApostilleDate>
<enml:IssuedAtUSLocation>
    <enml:County>Sacramento</enml:County>
    <enml:USState>CA</enml:USState>
    <enml:Country>USA</enml:Country>
</enml:IssuedAtUSLocation>
<enml:SignerTitle>
    Secretary of State, State of California
</enml:SignerTitle>
<enml:StatutoryContent>
    (Convention de La Haye du 5 octobre 1961)
</enml:StatutoryContent>
<enml:DocumentSigningOfficialName>
    <enml:PersonFirstName>Arshad</enml:PersonFirstName>
    <enml:PersonLastName>Noor</enml:PersonLastName>
</enml:DocumentSigningOfficialName>
<enml:DocumentSigningOfficialTitle>
    Notary Public
</enml:DocumentSigningOfficialTitle>
<enml:DocumentSigningOfficialSeal>
    Notary Public, State of California
</enml:DocumentSigningOfficialSeal>
</enml:ApostilleContent>
<ds:Signature>....</ds:Signature>
</enml:ApostillizedDocument>

4.53 Element <DocumentSigningOfficialName>

The <DocumentSigningOfficialName> element, of the type PersonNameType defined in the ENML schema, carries the full name of a Notary Public who either notarized or witnessed the signing of the embedded document within the <ApostillizedDocument>.
Schema Definition:

```xml
<xsd:element
    name="DocumentSigningOfficialName"
    type="enml:PersonNameType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

Note: The PersonNameType is defined in Section 4.x.
```

Example 1 – An example of an <ApostillizedDocument> with an <DocumentSigningOfficialName> element.

```xml
<enml:ApostillizedDocument>
    <enml:NotarizedDocument>.... </enml:NotarizedDocument>
    <enml:ApostilleContent Id="ApostilleContent-US-CA-1565986-1226972923-89e89c11c35b3d7c1030f78ffca5527cbd4cc61479fc88526c38e01d912a5"
                          IssuingCountry="United States of America"
                          ApostilleNumber="ApostilleNumber-US-CA-1565986-1226971821-c4e7f79729ad7da392d3aa30b23243532a0e30067350d4698974d5a2e7bd09ee4"
                          ApostilleDate="2008-11-17T15:45:28-08:00"></enml:ApostillizedDocument>
```
4.54 Element <DocumentSigningOfficialTitle>

The <DocumentSigningOfficialTitle> element, of the type PersonTitleType defined in the ENML schema, carries the official title of the Notary Public who either notarized or witnessed the signing of the embedded document within the <ApostillizedDocument>.

Schema Definition:

```
<xsd:element
  name="DocumentSigningOfficialTitle"
  type="enml:PersonTitleType"
  minOccurs="1"
  maxOccurs="1">
</xsd:element>

<xsd:simpleType name="PersonTitleType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="256"/>
    <xsd:WhiteSpace value="preserve"/>
  </xsd:restriction>
</xsd:simpleType>
```

Example 1 – An example of an <ApostillizedDocument> with an <DocumentSigningOfficialTitle> element.
4.55 Element <DocumentSigningOfficialSeal>

The `<DocumentSigningOfficialSeal>` element, of the type `ApostilleSealType` defined in the ENML schema, carries the official seal of the Notary Public who either notarized or witnessed the signing of the embedded document within the `<ApostilzedDocument>`.

**Schema Definition:**

```
<xsd:element
    name="DocumentSigningOfficialSeal"
    type="enml:ApostilleSealType"
    minOccurs="1"
    maxOccurs="1">
</xsd:element>

<xsd:complexType name="ApostilleSealType">
    <xsd:choice>
        <xsd:element name="TextSeal"
            type="xsd:string">
        </xsd:element>
        <xsd:element name="GIFImageSeal"
            type="xsd:base64Binary">
        </xsd:element>
        <xsd:element name="JPEGImageSeal"
            type="xsd:base64Binary">
        </xsd:element>
        <xsd:element name="PNGImageSeal"
            type="xsd:base64Binary">
        </xsd:element>
        <xsd:element name="X509CertificateSeal"
            type="xsd:base64Binary">
        </xsd:element>
    </xsd:choice>
</xsd:complexType>
```
The `<DocumentSigningOfficialSeal>` element may consist of a choice of one of the following elements:

- `<TextSeal>` [Required]
  This element carries a text-based seal of the public-official who signed the embedded `<NotarizedDocument>` or `<WitnessedDocument>`. The text content in this element MUST be 1,024 characters or less. If used, there MUST be exactly ONE `<TextSeal>` element in an `<DocumentSigningOfficialSeal>` element.

- `<GIFImageSeal>` [Required]
  This element carries an electronic image of the seal of the public-official who signed the embedded `<NotarizedDocument>` or `<WitnessedDocument>`. The content in this element MUST be Base64-encoded GIF seal image. If used, there MUST be exactly ONE `<GIFImageSeal>` element in an `<DocumentSigningOfficialSeal>` element.

- `<JPEGImageSeal>` [Required]
  This element carries an electronic image of the seal of the public-official who signed the embedded `<NotarizedDocument>` or `<WitnessedDocument>`. The content in this element MUST be Base64-encoded JPEG seal image. If used, there MUST be exactly ONE `<JPEGImageSeal>` element in an `<DocumentSigningOfficialSeal>` element.

- `<PNGImageSeal>` [Required]
  This element carries an electronic image of the seal of the public-official who signed the embedded `<NotarizedDocument>` or `<WitnessedDocument>`. The content in this element MUST be Base64-encoded PNG seal image. If used, there MUST be exactly ONE `<PNGImageSeal>` element in an `<DocumentSigningOfficialSeal>` element.

- `<X509CertificateSeal>` [Required]
  This element carries the Base64-encoded X509-compliant digital certificate seal of the public-official who signed the embedded `<NotarizedDocument>` or `<WitnessedDocument>`. If used, there MUST be exactly ONE `<X509CertificateSeal>` element in an `<DocumentSigningOfficialSeal>` element.

There MUST be exactly ONE `<DocumentSigningOfficialSeal>` element within an `<ApostilleContent>` element.

**Example 1** – An example of an `<ApostillizedDocument>` with a `<TextSeal>` in an `<DocumentSigningOfficialSeal>` element.

```xml
<enml:ApostillizedDocument>
  <enml:NotarizedDocument>....</enml:NotarizedDocument>
  <enml:ApostilleContent Id="ApostilleContent-US-CA-1565986-1226972923-89e89c11cac335b3d7c1030f78ffca5527c6b4cc61479fc88526c38e01d912a5">
    <enml:IssuingCountry>United States of
```

<enml:ApostillizedDocument>
  <enml:NotarizedDocument>....</enml:NotarizedDocument>
  <enml:ApostilleContent Id="ApostilleContent-US-CA-1565986-1226972923-89e89c11cac335b3d7c1030f78ffca5527cb2d4cc61479fc88526c38e01d912a5">
  
  <enml:IssuingCountry>United States of America</enml:IssuingCountry>
  <enml:ApostilleNumber-US-CA-1565986-1226971821-c4e7f79729ad7da392da18023243532a0e30067350d4698974d5a2e7bd079ee4</enml:ApostilleNumber>
  <enml:ApostilleDate>2008-11-17T15:45:28-08:00</enml:ApostilleDate>
  <enml:IssuedAtUSLocation>
    <enml:County>Sacramento</enml:County>
    <enml:USState>CA</enml:USState>
    <enml:Country>USA</enml:Country>
  </enml:IssuedAtUSLocation>
  <enml:SignerTitle>
    Secretary of State, State of California
  </enml:SignerTitle>
  <enml:StatutoryContent>
    (Convention de La Haye du 5 octobre 1961)
  </enml:StatutoryContent>
  <enml:DocumentSigningOfficialName>
    <enml:PersonFirstName>Arshad</enml:PersonFirstName>
    <enml:PersonLastName>Noor</enml:PersonLastName>
  </enml:DocumentSigningOfficialName>
  <enml:DocumentSigningOfficialTitle>Notary Public</enml:DocumentSigningOfficialTitle>
  <enml:DocumentSigningOfficialSeal>
    <enml:TextSeal>
      Notary Public, State of California
    </enml:TextSeal>
  </enml:DocumentSigningOfficialSeal>
  </enml:ApostilleContent>
</enml:ApostillizedDocument>
Secretary of State, State of California
</enml:SignerTitle>

(enml:StatutoryContent)

(Convention de La Haye du 5 octobre 1961)
</enml:StatutoryContent>

(enml:DocumentSigningOfficialName)

<enml:PersonFirstName>Arshad</enml:PersonFirstName>
<enml:PersonLastName>Noor</enml:PersonLastName>
</enml:DocumentSigningOfficialName>

(enml:DocumentSigningOfficialTitle)

Notary Public
</enml:DocumentSigningOfficialTitle>

(enml:DocumentSigningOfficialSeal)

MIIDfCCAmGwIBAgIAa/AvliGc3AwDQYJKoZIhlv4sNAQELBQAzwZzEwMCQGA1UEAxUdU3Yb25nS2V51ERFTU8gU3Vib3JkIa5hd9GLjQEXoJDA1BqMVB8aTGOZviBTdhbmdLZXkgREVNTyBVc2UgT25seTE projection of X509CertificateSeal>

DS:Signature>....</ds:Signature>
</enml:ApostilledDocument>
4.56 Processing Rule for ID attributes within ENML

To ensure that the <Signature>, <SignerSignature>, <NotarySignature> and <ApostilleSignature> elements refer to the precise elements being signed within the ENML-based document, ENML has chosen to explicitly identify those elements being signed through <Reference> elements in the different signature elements. Additionally, since the <NotarizedDocument> and the <WitnessedDocument> elements may appear within an <ApostillizedDocument>, it is necessary for these elements to also carry a unique ID value.

The identification is accomplished through the mandatory use of the ID attribute (of the xsd:id type from the XML Schema Definition) for the following required elements in ENML-based documents:

a. <NotarizedDocument>

b. <WitnessedDocument>

c. <ApostillizedDocument>

d. <SignedDocument>

e. <DocumentSigner>

f. <NotaryCertificate> (for notarized documents)

g. <NotaryPublic> (for witnessed documents)

h. <ApostilleNumber>

Given that there may be millions of eNotarized, eWitnessed and eApostillized documents, it is critical that documents not duplicate the value of the ID attribute for these elements.

ENML, therefore, requires that conforming implementations MUST generate the value of the ID attribute using the following algorithm for seven (7) of the above-mentioned eight (8) elements. It is only recommended for the <ApostilleNumber> element, but not mandated.

The ID value may be derived as follows:

1. Begin with the name of the element;
2. Concatenate a literal hyphen ("-"),
3. Concatenate the two (2) character country code as defined by [ISO-3166-1];
4. Concatenate a literal hyphen ("-"),
5. Concatenate a two (2) character state code to uniquely identify the state within the country defined in step 3; if the country has no state-abbreviations, use the first two (2) characters of the city or county where the notarization/witnessing is being performed. If the notarization is being performed in a rural area that has no city/county name, use the fixed abbreviation "XX" (two capitalized instances of the alphabet letter X)
6. Concatenate a literal hyphen ("-"),
7. Concatenate the unique Notary Commission number assigned to a Notary Public within the legal jurisdiction; if the jurisdiction does not assign Commission numbers to Notaries, use the name of the Notary Public by which he/she is commissioned to perform his/her official duties, concatenated to each other without any spaces and with the first letter of each part of the name capitalized;
8. Concatenate a literal hyphen ("-"),
9. Concatenate the “epoch” time at the time this ID value is being generated; the “epoch” time is the number of seconds elapsed since 00:00:00 Coordinated Universal Time (UTC) January 01, 1970 (not counting leap seconds).

10. Concatenate a literal hyphen (“-”);

11. Concatenate the SHA-256 message-digest of the element whose ID value is being assembled, **WITHOUT** the ID value itself.

While the above algorithm cannot absolutely guarantee that the value generated for the ID attribute will be unique, it is the contention of this TC that the probability of values being identical for different eNotarized, eWitnessed or eApostillized document are extremely low.

**Example 1** – This example shows a `<SignedDocument>` that has the following ID attribute value. In this example, the:

- **SignedDocument** refers to the name of the element (step 1);
- **US** refers to the 2-character [ISO-3166-1] country code for the USA (step 3);
- **CA** refers to the 2-character state abbreviation for California within the US (step 5);
- **1565986** refers to the unique Commission number of the Notary Public notarizing this document (step 7);
- **1222646651** refers to the epoch time at the time of generating this value (step 9);
- **c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4** refers to the SHA-256 message digest of the `<SignedDocument>` element (step 11);

**Note:** The SHA-256 digest is not necessarily cryptographically valid for this element; it is shown here as an illustration of what the ID value might look like applying this processing rule.

```xml
<enml:SignedDocuments>
  <enml:SignedDocument Id="SignedDocument-US-CA-1565986-1222646651-c4e7f79729ad7da392d3a18023243532a0e30067350d4698974d5a2e7bd09ee4"
    xmlns:enml="http://www.oasis-open.org/committees/enml"
    <enml:Document>
      PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KCjwhLS0KI
      CBM ZwdhbFhNTCBiTm90YXJpemF0aW9uIDEuMCBT cGVjaWZpY2F0aW9u
      ERl Y2VtYmVyIiwIDm9wdGluZ3RvcmFnZSBAZGl0IGl0
      Wu dCBhbmQgdHJhbnNsYXRpb25zIG9mIGl0IG1heSBiZSBjb3BpZWQgYm9ke
      m5p c2hlZCB0byBvdGhlcnMsIAogIGFub29t
      XRz bWVudCBvbibVcicnNc2Jzd29t
      ICAgICATLT4KICAgICAKPC94c2Q6c2NoZW1hPgo=
    </enml:Document>
  </enml:SignedDocument>
</enml:SignedDocuments>
```

**Example 2** – This example shows a `<DocumentSigner>` that has the following ID attribute value. In this example, the:

```xml
<enml:DocumentSigner>
  <enml:DocumentMIMEType>application/xml</enml:DocumentMIMEType>
  <enml:DocumentComments>Property Deed</enml:DocumentComments>
</enml:DocumentSigner>
```
- **DocumentSigner** refers to the name of the element (step 1);
- **US** refers to the 2-character [ISO-3166-1] country code for the USA (step 3);
- **CA** refers to the 2-character state abbreviation for California within the US (step 5);
- **1565986** refers to the unique Commission number of the Notary Public notarizing this document (step 7);
- **1222643107** refers to the epoch time at the time of generating this value (step 9);
- **f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2afd6d273fd7** refers to the SHA-256 message digest of the `<DocumentSigner>` element (step 11);

Note: The SHA-256 digest is not necessarily cryptographically valid for this element; it is shown here as an illustration of what the ID value might look like applying this processing rule.

```xml
<enml:DocumentSigners>
  <enml:DocumentSigner Id="DocumentSigner-US-CA-1565986-1222643107-f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2afd6d273fd7">
    <enml:SignerName>
      <enml:PersonFirstName>John</enml:PersonFirstName>
      <enml:PersonLastName>Doe</enml:PersonLastName>
    </enml:SignerName>
    <enml:SignerIdentificationMethod>
      Produced Government-issued Identification Document
    </enml:SignerIdentificationMethod>
    <enml:SignerSignature>Signed by John Doe</enml:SignerSignature>
  </enml:DocumentSigner>
</enml:DocumentSigners>
```

**Example 3** – This example shows a `<NotaryCertificate>` that has the following ID attribute value. In this example, the:

- **NotaryCertificate** refers to the name of the element (step 1);
- **US** refers to the 2-character [ISO-3166-1] country code for the USA (step 3);
- **CA** refers to the 2-character state abbreviation for California within the US (step 5);
- **1565986** refers to the unique Commission number of the Notary Public notarizing this document (step 7);
- **1222643229** refers to the epoch time at the time of generating this value (step 9);
- **1963a865f3516996f711936045ced816a0fccd4aca57c6807a1c44f9b98a1e5f** refers to the SHA-256 message digest of the `<DocumentSigner>` element (step 11);

Note: The SHA-256 digest is not necessarily cryptographically valid for this element; it is shown here as an illustration of what the ID value might look like applying this processing rule.

```xml
<enml:NotaryCertificates>
  <enml:NotaryCertificate Id="NotaryCertificate-US-CA-1565986-1222643229-1963a865f3516996f711936045ced816a0fccd4aca57c6807a1c44f9b98a1e5f">
    <enml:CertificateContent>
      <enml:NotarizationType>Acknowledgment</enml:NotarizationType>
      <enml:NotarizationDate>2007-01-27T15:45:28-08:00</enml:NotarizationDate>
      <enml:NotarizationUSLocation>
        <enml:City>Cupertino</enml:City>
        <enml:County>Santa Clara</enml:County>
      </enml:NotarizationUSLocation>
    </enml:CertificateContent>
  </enml:NotaryCertificate>
</enml:NotaryCertificates>
```
<enml:USState>CA</enml:USState>
<enml:Country>USA</enml:Country>
</enml:NotarizationUSLocation>
<enml:StatutoryContent>
State of California
County of Santa Clara

On January 27 2008, before me Arshad Noor, personally appeared John Doe and Jane Doe, who proved to me on the basis of satisfactory evidence to be the persons whose names are subscribed to the within instrument and acknowledged to me that they executed the same in their authorized capacity, and that by their signature on the instrument the persons, or the entity upon behalf of which the persons acted, executed the instrument.

I certify under PENALTY OF PERJURY under the laws of the State of California that the foregoing paragraph is true and correct.

WITNESS my hand and official seal.
</enml:StatutoryContent>
</enml:CertificateContent>

<enml:NotaryPublic>
<enml:NotaryName>
<enml:PersonGivenName>Arshad</enml:PersonGivenName>
<enml:PersonSurName>Noor</enml:PersonSurName>
</enml:NotaryName>
<enml:NotaryCommissionNumber>1565986</enml:NotaryCommissionNumber>
<enml:NotaryCommissionExpiryDate>2009-04-29T23:59:59-08:00</enml:NotaryCommissionExpiryDate>
<enml:NotaryUSJurisdiction>
<enml:County>Santa Clara</enml:County>
<enml:USState>CA</enml:USState>
<enml:Country>USA</enml:Country>
</enml:NotaryUSJurisdiction>
</enml:NotaryPublic>
</enml:NotaryCertificates>

Example 4 – This example shows a <DocumentSigner> that has the following ID attribute value. In this example, the:
- DocumentSigner refers to the name of the element (step 1);
- BB refers to the 2-character [ISO-3166-1] country code for the Barbados (step 3);
- KI refers to the first 2-characters of the city name (Kingston) when there is no state abbreviation available (step 5);
- HowardQJones refers to the full-name of the NotaryPublic, concatenated with no spaces between the name, and with the first letter of each part of the name capitalized, when there is no
unique Commission number of the Notary Public notarizing this document is available (step 7);
- **1222643107** refers to the epoch time at the time of generating this value (step 9);
- **f91b09f20bfa099beb6b1cb493dfdd9571c3edac5c23fa36aace2afd6d273fd7** refers to the SHA-256 message digest of the <DocumentSigner> element (step 11);

Note: The SHA-256 digest is not necessarily cryptographically valid for this element; it is shown here as an illustration of what the ID value might look like applying this processing rule.
5 Conformance

An implementation conforms to this specification if it satisfies all of the MUST, REQUIRED or SHALL level requirements defined within this specification. An ENML Node MUST NOT use the XML namespace identifier for this specification (listed in the Title section under Declared Namespace(s)) unless it is compliant with this specification.

This specification references a number of other specifications (see the table above). In order to comply with this specification, an implementation MUST implement the portions of referenced specifications necessary to comply with the required provisions of this specification. Additionally, the implementation of the portions of the referenced specifications that are specifically cited in this specification MUST comply with the rules for those portions as established in the referenced specification.

Additionally normative text within this specification takes precedence over normative outlines, which in turn take precedence over the XML Schema [XML Schema Part 1, Part 2] descriptions. That is, the normative text in this specification further constrains the schema part of this specification; and this specification contains further constraints on the elements defined in referenced schemas.

If an OPTIONAL message is not supported, then the implementation SHOULD Fault just as it would for any other unrecognized/unsupported message. If an OPTIONAL message is supported, then the implementation MUST satisfy all of the MUST and REQUIRED sections of the message.
Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:
- Rolly Chambers, American Bar Association
- John Messing, American Bar Association
- Laurence Leff, Individual, Member
- Marc Aronson, Pennsylvania Assoc. of Notaries
- Eric Cohen, PricewaterhouseCoopers LLP
- David Ewan, Property Records Industry Assn.
- Mark Ladd, Property Records Industry Assn.
- Arshad Noor, Associate Member
# Appendix B. Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAFT 1</td>
<td>November 2, 2008</td>
<td>Arshad Noor</td>
<td>Initial version</td>
</tr>
<tr>
<td>DRAFT 2</td>
<td>December 1, 2008</td>
<td>Arshad Noor</td>
<td>Includes ApostillizedDocument and supporting elements, Appendices C, E and F</td>
</tr>
<tr>
<td>DRAFT 2.1</td>
<td>December 8, 2008</td>
<td>Arshad Noor</td>
<td>Minor corrections to Glossary to correct typographic errors and some descriptions</td>
</tr>
<tr>
<td>DRAFT 2.2</td>
<td>December 16, 2008</td>
<td>Arshad Noor</td>
<td>Removed Appendix D related to XPath</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Removed section 4.57 related to XPath</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added section 3.12 with a cautionary (non-normative) note on the use of XPath within ENML</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Minor correction to the description of ENML in the Introduction.</td>
</tr>
</tbody>
</table>
Appendix C. ENML Security Implications (Non-normative)
Appendix D. ENML Error Codes and Error Messages

The OASIS LegalXML eNotary TC has determined that it is useful to standardize on the structure and content of error and informational messages within ENML so that implementations and their users are clear about the problem they are dealing with eNotarized documents.

Structure

The following structure is proposed for ENML-based messages. The message will consist of a three-part string, with each part separated from the others by a hyphen (“-”). The 3-part message consists of the following:

- The first part is the fixed string “ENML”
- The second part is a fixed string consisting of one of the following choices:
  - ERR
  - MSG
- The third, and last, part is a 5-digit integer identifying the message

Thus, an ENML error message might look like the following: ENML-ERR-NNNNN; and an ENML informational message might look like the following: ENML-MSG-NNNNN

Five-digit Codes

The 5-digit integer is divided into the following groups to ensure consistency amongst implementations:

- 00001 – 10000 Reserved for OASIS LegalXML eNotary TC use (as described below);
- 10001 – 99999 Reserved for vendor implementations of ENML on a first-come, first-served basis (the process is described below);

ENML Standard Code-ranges

The 5-digit code range reserved for OASIS LegalXML eNotary TC ENML standards use will be reserved as follows:

<table>
<thead>
<tr>
<th>Code-range</th>
<th>Reserved for</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001 - 00099</td>
<td>Authentication related errors and messages</td>
</tr>
<tr>
<td>00100 - 00199</td>
<td>Authorization related errors and messages</td>
</tr>
<tr>
<td>00200 - 00299</td>
<td>Cryptographic-module related errors and messages</td>
</tr>
<tr>
<td>00300 - 00399</td>
<td>Notarized Document related errors and messages</td>
</tr>
<tr>
<td>00400 - 00499</td>
<td>Witnessed Document related errors and messages</td>
</tr>
<tr>
<td>00500 - 00599</td>
<td>Apostillized Document related errors and messages</td>
</tr>
<tr>
<td>00600 - 00699</td>
<td>Signed Documents related errors and messages</td>
</tr>
<tr>
<td>00700 - 00799</td>
<td>Document Signers related errors and messages</td>
</tr>
<tr>
<td>Code-range</td>
<td>Reserved for</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>00800 - 00899</td>
<td>Notary Certificates related errors and messages</td>
</tr>
<tr>
<td>00900 - 00999</td>
<td>Notaries Public related errors and messages</td>
</tr>
<tr>
<td>01000 - 01099</td>
<td>Notary Signatures related errors and messages</td>
</tr>
<tr>
<td>01100 - 01199</td>
<td>Miscellaneous errors and messages</td>
</tr>
<tr>
<td>01200 - 10000</td>
<td>Future OASIS LegalXML eNotary TC use</td>
</tr>
</tbody>
</table>

Note: The {0} symbol at the end of each message is a placeholder for a parameter that can be used by implementations for adding additional information pertaining to the error. The additional information will be useful to administrators and software developers in helping them focus on the part of the system where the underlying problem has manifested itself.

### Authentication ERROR Codes (00001 - 00099)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00001</td>
<td>Authentication failure – invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00002</td>
<td>Authentication failure – invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00003</td>
<td>Authentication failure – unverifiable certificate: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00004</td>
<td>Authentication failure – expired certificate: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00005</td>
<td>Authentication failure – revoked certificate: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00006</td>
<td>Authentication failure – revoked certificate issuer: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00007</td>
<td>Authentication failure – missing certificate: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00008</td>
<td>Authentication failure – missing certificate keyUsage: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00009</td>
<td>Authentication failure – missing certificate crlDistributionPoint: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00010</td>
<td>Authentication failure – missing certificate authorityInfoAccess: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00011</td>
<td>Authentication failure – invalid certificate Subject DN: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00012</td>
<td>Authentication failure – invalid certificate Validity: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00013</td>
<td>Authentication failure – invalid certificate keyUsage: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00014</td>
<td>Authentication failure – invalid certificate crlDistributionPoint: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00015</td>
<td>Authentication failure – invalid certificate authorityInfoAccess: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00016</td>
<td>Authentication failure – unreachable certificate crlDistributionPoint: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00017</td>
<td>Authentication failure – unreachable certificate authorityInfoAccess: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00099</td>
<td>Authentication failure – other authentication error: {0}</td>
</tr>
</tbody>
</table>
Authorization ERROR Codes (00100 - 00199)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00101</td>
<td>Authorization failure – invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00102</td>
<td>Authorization failure – invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00103</td>
<td>Authorization failure – invalid requester: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00104</td>
<td>Authorization failure – invalid request: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00105</td>
<td>Authorization failure – invalid identifier: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00106</td>
<td>Authorization failure – invalid policy: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00107</td>
<td>Authorization failure – invalid role: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00108</td>
<td>Authorization failure – invalid application: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00109</td>
<td>Authorization failure – invalid date: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00110</td>
<td>Authorization failure – invalid day: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00111</td>
<td>Authorization failure – invalid time: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00112</td>
<td>Authorization failure – invalid use: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00113</td>
<td>Authorization failure – unauthorized access: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00114</td>
<td>Authorization failure – unauthorized date: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00115</td>
<td>Authorization failure – unauthorized day: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00116</td>
<td>Authorization failure – unauthorized time: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00117</td>
<td>Authorization failure – unauthorized use: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00119</td>
<td>Authorization failure – other authorization error: {0}</td>
</tr>
</tbody>
</table>

Cryptographic-module ERROR Codes (00200 - 00299)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00201</td>
<td>Cryptographic-module failure – invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00202</td>
<td>Cryptographic-module failure – invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00203</td>
<td>Cryptographic-module failure – invalid cryptographic provider: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00204</td>
<td>Cryptographic-module failure – invalid algorithm: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00205</td>
<td>Cryptographic-module failure – invalid initialization vector: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00206</td>
<td>Cryptographic-module failure – invalid padding: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00207</td>
<td>Cryptographic-module failure – invalid key-size: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00208</td>
<td>Cryptographic-module failure – invalid password or PIN: {0}</td>
</tr>
<tr>
<td>Code</td>
<td>Message</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>ENML-ERR-00209</td>
<td>Cryptographic-module failure – missing cryptographic provider: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00210</td>
<td>Cryptographic-module failure – missing cryptographic module: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00211</td>
<td>Cryptographic-module failure – missing password or PIN: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00212</td>
<td>Cryptographic-module failure – missing private key: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00213</td>
<td>Cryptographic-module failure – missing digital certificate: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00214</td>
<td>Cryptographic-module failure – missing certificate chain: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00215</td>
<td>Cryptographic-module failure – failed to sign: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00216</td>
<td>Cryptographic-module failure – failed to verify: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00217</td>
<td>Cryptographic-module failure – failed to encrypt: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00218</td>
<td>Cryptographic-module failure – failed to decrypt: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00219</td>
<td>Cryptographic-module failure – failed to digest (hash): {0}</td>
</tr>
<tr>
<td>ENML-ERR-00220</td>
<td>Cryptographic-module failure – failed to generate key: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00299</td>
<td>Cryptographic-module failure – other cryptographic-module error: {0}</td>
</tr>
</tbody>
</table>

**Notarized Document ERROR Codes (00300 - 00399)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00301</td>
<td>Notarized Document failure – invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00302</td>
<td>Notarized Document failure – invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00303</td>
<td>Notarized Document failure – invalid ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00304</td>
<td>Notarized Document failure – invalid content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00305</td>
<td>Notarized Document failure – missing ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00306</td>
<td>Notarized Document failure – missing content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00399</td>
<td>Notarized Document failure – other Notarized Document error: {0}</td>
</tr>
</tbody>
</table>

**Witnessed Document ERROR Codes (00400 - 00499)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00401</td>
<td>Witnessed Document failure – invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00402</td>
<td>Witnessed Document failure – invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00403</td>
<td>Witnessed Document failure – invalid ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00404</td>
<td>Witnessed Document failure – invalid content: {0}</td>
</tr>
<tr>
<td>Code</td>
<td>Message</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ENML-ERR-00405</td>
<td>Witnessed Document failure – missing ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00406</td>
<td>Witnessed Document failure – missing content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00499</td>
<td>Witnessed Document failure – other Witnessed Document error: {0}</td>
</tr>
</tbody>
</table>

**Apostillized Document ERROR Codes (00500 - 00599)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00501</td>
<td>Apostillized Document error - invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00502</td>
<td>Apostillized Document error - invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00503</td>
<td>Apostillized Document error - invalid ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00504</td>
<td>Apostillized Document error - invalid Notarized Document: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00505</td>
<td>Apostillized Document error - invalid Witnessed Document: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00506</td>
<td>Apostillized Document error - invalid Issuing Country: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00507</td>
<td>Apostillized Document error - invalid Apostille Number: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00508</td>
<td>Apostillized Document error - invalid Apostille Date: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00509</td>
<td>Apostillized Document error - invalid International Location: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00510</td>
<td>Apostillized Document error - invalid US Location: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00511</td>
<td>Apostillized Document error - invalid Signer Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00512</td>
<td>Apostillized Document error - invalid Statutory Content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00513</td>
<td>Apostillized Document error - invalid Document Signing Official Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00514</td>
<td>Apostillized Document error - invalid Document Signing Official Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00515</td>
<td>Apostillized Document error - invalid Document Signing Official Seal: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00516</td>
<td>Apostillized Document error - missing ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00517</td>
<td>Apostillized Document error - missing Notarized Document: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00518</td>
<td>Apostillized Document error - missing Witnessed Document: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00519</td>
<td>Apostillized Document error - missing Issuing Country: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00520</td>
<td>Apostillized Document error - missing Apostille Number: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00521</td>
<td>Apostillized Document error - missing Apostille Date: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00522</td>
<td>Apostillized Document error - missing International Location: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00523</td>
<td>Apostillized Document error - missing US Location: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00524</td>
<td>Apostillized Document error - missing Signer Title: {0}</td>
</tr>
</tbody>
</table>
### Apostillized Document error - missing Statutory Content:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00525</td>
<td>Apostillized Document error - missing Statutory Content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00526</td>
<td>Apostillized Document error - missing Document Signing Official Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00527</td>
<td>Apostillized Document error - missing Document Signing Official Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00528</td>
<td>Apostillized Document error - missing Document Signing Official Seal: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00599</td>
<td>Apostillized Document error - other Apostillized Document error: {0}</td>
</tr>
</tbody>
</table>

### Apostillized Document error - missing Document Signing Official Name:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00525</td>
<td>Apostillized Document error - missing Statutory Content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00526</td>
<td>Apostillized Document error - missing Document Signing Official Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00527</td>
<td>Apostillized Document error - missing Document Signing Official Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00528</td>
<td>Apostillized Document error - missing Document Signing Official Seal: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00599</td>
<td>Apostillized Document error - other Apostillized Document error: {0}</td>
</tr>
</tbody>
</table>

### Apostillized Document error - missing Document Signing Official Title:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00525</td>
<td>Apostillized Document error - missing Statutory Content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00526</td>
<td>Apostillized Document error - missing Document Signing Official Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00527</td>
<td>Apostillized Document error - missing Document Signing Official Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00528</td>
<td>Apostillized Document error - missing Document Signing Official Seal: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00599</td>
<td>Apostillized Document error - other Apostillized Document error: {0}</td>
</tr>
</tbody>
</table>

### Apostillized Document error - missing Document Signing Official Seal:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00525</td>
<td>Apostillized Document error - missing Statutory Content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00526</td>
<td>Apostillized Document error - missing Document Signing Official Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00527</td>
<td>Apostillized Document error - missing Document Signing Official Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00528</td>
<td>Apostillized Document error - missing Document Signing Official Seal: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00599</td>
<td>Apostillized Document error - other Apostillized Document error: {0}</td>
</tr>
</tbody>
</table>

### Apostillized Document error - other Apostillized Document error:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00525</td>
<td>Apostillized Document error - missing Statutory Content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00526</td>
<td>Apostillized Document error - missing Document Signing Official Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00527</td>
<td>Apostillized Document error - missing Document Signing Official Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00528</td>
<td>Apostillized Document error - missing Document Signing Official Seal: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00599</td>
<td>Apostillized Document error - other Apostillized Document error: {0}</td>
</tr>
</tbody>
</table>

### Signed Documents ERROR Codes (00600 - 00699)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00601</td>
<td>Signed Documents error - invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00602</td>
<td>Signed Documents error - invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00603</td>
<td>Signed Documents error - invalid ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00604</td>
<td>Signed Documents error - invalid Document: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00605</td>
<td>Signed Documents error - invalid Document MIME Type: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00606</td>
<td>Signed Documents error - invalid Document MIME Type Comments: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00607</td>
<td>Signed Documents error - invalid Document Comments: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00608</td>
<td>Signed Documents error - missing ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00609</td>
<td>Signed Documents error - missing Document: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00610</td>
<td>Signed Documents error - missing Document MIME Type: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00611</td>
<td>Signed Documents error - missing Document MIME Type Comments: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00612</td>
<td>Signed Documents error - missing Document Comments: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00699</td>
<td>Signed Documents error - other error: {0}</td>
</tr>
</tbody>
</table>

### Document Signers ERROR Codes (00700 - 00799)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00701</td>
<td>Document Signers error - invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00702</td>
<td>Document Signers error - invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00703</td>
<td>Document Signers error - invalid ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00704</td>
<td>Document Signers error - invalid Signer Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00705</td>
<td>Document Signers error - invalid Signer Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00706</td>
<td>Document Signers error - invalid International Address: {0}</td>
</tr>
<tr>
<td>Code</td>
<td>Message</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ENML-ERR-00707</td>
<td>Document Signers error - invalid US Address: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00708</td>
<td>Document Signers error - invalid Identification Method: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00709</td>
<td>Document Signers error - invalid Signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00710</td>
<td>Document Signers error - invalid Signed Info: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00711</td>
<td>Document Signers error - invalid Signature Value: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00712</td>
<td>Document Signers error - invalid Key Info: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00713</td>
<td>Document Signers error - invalid Object: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00714</td>
<td>Document Signers error - missing ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00715</td>
<td>Document Signers error - missing Signer Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00716</td>
<td>Document Signers error - missing Signer Title: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00717</td>
<td>Document Signers error - missing International Address: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00718</td>
<td>Document Signers error - missing US Address: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00719</td>
<td>Document Signers error - missing Identification Method: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00720</td>
<td>Document Signers error - missing Signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00721</td>
<td>Document Signers error - missing Signed Info: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00722</td>
<td>Document Signers error - missing Signature Value: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00723</td>
<td>Document Signers error - missing Key Info: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00724</td>
<td>Document Signers error - missing Object: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00799</td>
<td>Document Signers error - other error: {0}</td>
</tr>
</tbody>
</table>

**Notary Certificates ERROR Codes (00800 - 00899)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00801</td>
<td>Notary Certificates error - invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00802</td>
<td>Notary Certificates error - invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00803</td>
<td>Notary Certificates error - invalid ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00804</td>
<td>Notary Certificates error - invalid Notarization Type: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00805</td>
<td>Notary Certificates error - invalid Notarization Date: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00806</td>
<td>Notary Certificates error - invalid International Location: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00807</td>
<td>Notary Certificates error - invalid US Location: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00808</td>
<td>Notary Certificates error - invalid Statutory Content: {0}</td>
</tr>
<tr>
<td>Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>ENML-ERR-00809</td>
<td>Notary Certificates error - missing ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00810</td>
<td>Notary Certificates error - missing Notarization Type: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00811</td>
<td>Notary Certificates error - missing Notarization Date: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00812</td>
<td>Notary Certificates error – missing International Location: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00813</td>
<td>Notary Certificates error - missing US Location: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00814</td>
<td>Notary Certificates error – missing Statutory Content: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00899</td>
<td>Notary Certificates error - other error: {0}</td>
</tr>
</tbody>
</table>

### Notaries Public ERROR Codes (00900 - 00999)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-00901</td>
<td>Notaries Public error - invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00902</td>
<td>Notaries Public error - invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00903</td>
<td>Notaries Public error - invalid ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00904</td>
<td>Notaries Public error - invalid Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00905</td>
<td>Notaries Public error - invalid Commission Number: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00906</td>
<td>Notaries Public error - invalid Commission Expiry Date: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00907</td>
<td>Notaries Public error - invalid International Jurisdiction: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00908</td>
<td>Notaries Public error - invalid US Jurisdiction: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00909</td>
<td>Notaries Public error - invalid Bond Number: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00910</td>
<td>Notaries Public error - invalid Verification URI: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00911</td>
<td>Notaries Public error - missing ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00912</td>
<td>Notaries Public error - missing Name: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00913</td>
<td>Notaries Public error - missing Commission Number: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00914</td>
<td>Notaries Public error - missing Commission Expiry Date: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00915</td>
<td>Notaries Public error - missing International Jurisdiction: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00916</td>
<td>Notaries Public error - missing US Jurisdiction: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00917</td>
<td>Notaries Public error - missing Bond Number: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00918</td>
<td>Notaries Public error - missing Verification URI: {0}</td>
</tr>
<tr>
<td>ENML-ERR-00999</td>
<td>Notaries Public error - other error: {0}</td>
</tr>
</tbody>
</table>
### Notary Signatures ERROR Codes (01000 - 01099)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-01001</td>
<td>Notary Signatures error - invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01002</td>
<td>Notary Signatures error - invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01003</td>
<td>Notary Signatures error - invalid ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01004</td>
<td>Notary Signatures error - invalid Signed Info: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01005</td>
<td>Notary Signatures error - invalid Signature Value: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01006</td>
<td>Notary Signatures error - invalid Key Info: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01007</td>
<td>Notary Signatures error - invalid Object: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01008</td>
<td>Notary Signatures error - missing ID attribute: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01009</td>
<td>Notary Signatures error - missing Signed Info: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01010</td>
<td>Notary Signatures error - missing Signature Value: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01011</td>
<td>Notary Signatures error - missing Key Info: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01012</td>
<td>Notary Signatures error - missing Object: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01099</td>
<td>Notary Signatures error - other error: {0}</td>
</tr>
</tbody>
</table>

### Miscellaneous ERROR Codes (01100 – 01199)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENML-ERR-01101</td>
<td>Miscellaneous error - invalid signature: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01102</td>
<td>Miscellaneous error - invalid status: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01103</td>
<td>Miscellaneous error – SOAP layer error: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01104</td>
<td>Miscellaneous error - network layer error: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01105</td>
<td>Miscellaneous error – database layer error: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01106</td>
<td>Miscellaneous error – epoch time error: {0}</td>
</tr>
<tr>
<td>ENML-ERR-01199</td>
<td>Miscellaneous error - other miscellaneous error: {0}</td>
</tr>
</tbody>
</table>
Appendix E. Process for requesting a block of ENML Error Codes for Vendor Use

Vendors who choose to implement ENML will be able to apply for a unique block of 1,000 code numbers to be assigned for their exclusive use within their implementation of the ENML protocol. They may choose to use these numbers for messages related to their implementation, in addition to the Standard ENML Codes & Messages.

In order to receive this unique block of code-numbers, the Vendor must use the following process:

1. An authorized representative of the Vendor must send an e-mail to the OASIS LegalXML eNotary TC (using the guidelines in this URL: http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=legalxml-enotary) asserting the following:
   a) That they intend to implement the ENML 1.0 specification within 6-12 months of their dated e-mail;
   b) That they will implement ALL the Standard Codes & Messages as described in this document, in their implementation;
   c) That they will not duplicate ANY Standard Code-message within their assigned private-block of numbers;
   d) That if the TC later chooses to standardize a specific message within the Standard Codes, that may overlap with a Member's private-block message, the Member will use the Standard Code in implementations created subsequent to the standardization of the code/message;
   e) That they will notify this TC of the release date of their product, with the relevant section of their documentation pointing to the use of the Standard Codes in their product
2. The OASIS LegalXML eNotary TC Chair or Secretary will verify that the e-mail contains all assertions;
3. The OASIS LegalXML eNotary TC Chair or Secretary will setup a ballot for the TC to vote to issue the next available block of 1,000 code-numbers to the requesting Vendor. (The first OASIS Member to apply will receive the numbers 10001 – 11000; the next Vendor request will receive 11001 – 12000 and so on).
4. Upon the conclusion of the vote, the TC Chair or Secretary will notify the Vendor of the assigned block of code-numbers (copying the TC);
5. The TC Chair or Secretary will update a web-page on the TC's home-page with the following information:
   a) The name of the Vendor
   b) A link to the e-mail request from the Vendor
   c) A link to the TC ballot authorizing the assignment
   d) The date of assignment and
   e) The assigned block of code-numbers
6. When the Vendor releases the product, they will notify the OASIS TC of the product, a link to the product's web URL and a link to the specific section of their documentation high-lighting the Standard Codes & Messages, as well as the Private Block Codes & Messages;
7. The TC Chair or Secretary will then update the above-mentioned web-page to now include this additional information:
   a) The web-link to the Vendor's implementation
   b) The web-link to the documentation highlighting the Codes and Messages
8. In the event that a Vendor does not implement the protocol within the 6-12 month period, the TC will vote and reclaim the private-block assigned to the original requester. Any use of the private-block of assigned numbers, after the TC has voted to reclaim them, will be a violation of the TC's guidelines for the ENML Specification. OASIS Administration will then be directed by this TC to take whatever action it is legally permitted to address this violation.