DSS Use Case Requirements Analysis

Working Draft 23, 23 March 2 April 2003

Document identifier:

dss-requirements-1.0-draft-03TBD

Location:

http://www.oasis-open.org/apps/org/workgroup/dss/TBD

Editor:

Trevor Perrin <trevp@trevp.net>

Contributors:

John Messing <jmsging@law-on-line.com>
Nick Pope <pope@secstan.com>
Krishna Sankar* <ksankar@cisco.com>

Abstract:

This document analyses the use cases given in the OASIS DSS Use Case document and identifies the requirements of Digital Signature Services (DSS) protocols.

Status:

This document is a work in progress. Comments are welcome, and should be sent to the dss@lists.oasis-open.org list.

Copyright © 2002 and 2003 The Organization for the Advancement of Structured Information Standards [OASIS]
# Table of Contents

1. Introduction .......................... 5
2. Particular Features of Use Cases ......................................................... 5
   2.1 Corporate Seal ...................................................... 5
   2.2 SOAP Signing .................................................... 5
   2.3 Identified Requestor ............................................... 5
   2.4 Individual Signatures .............................................. 5
   2.5 Long Term Corporate Signatures ................................ 6
   2.6 Delegated Signature Verification ................................ 6
   2.7 Securing the Transform Chain .................................... 6
   2.8 Non-XML Data Signing ........................................ 7
   2.9 ENotarization in presence of notary ........................ 7
   2.10 Court Filings ...................................................... 8
   2.11 Client-Side Hashing ............................................. 8
   2.12 Time-stamping / Time-marking ................................ 8
3. Requirements ........................................ 9
   3.1 Supported Formats ............................................... 9
      3.1.1 To-Be-Signed Format ..................................... 9
      3.1.2 Signature Format ........................................... 9
   3.2 Signature Contents ............................................... 10
      3.2.1 Requestor Identity .......................................... 10
      3.2.2 Signing Time ................................................ 10
   3.3 Generic Request Requirements ........................................ 11
      3.3.1 General ........................................................ 11
      3.3.2 Input Preparation ........................................... 11
      3.3.3 Input Delivery ................................................ 11
      3.3.4 Authentication ............................................... 12
   3.4 Signing Request Requirements ........................................... 13
      3.4.1 Selective Signing ............................................ 13
      3.4.2 Signature Placement ....................................... 13
      3.4.3 Output Delivery ............................................. 13
      3.4.4 Signature Policy .............................................. 13
   3.5 Explicit Signed/Unsigned Attributes ........................................... 14
3.5 Signing Response Requirements ........................................................................... 1413
  3.5.1 Signature Placement ....................................................................................... 1413
  3.5.2 Output Delivery ............................................................................................. 1413
3.6 Verification Request Requirements ..................................................................... 1413
  3.6.1 Selective Verification ..................................................................................... 1413
  3.6.2 General ......................................................................................................... 1514
3.7 Verification Response Requirements .................................................................... 1514
  3.7.1 Failure Information ......................................................................................... 1514
  3.7.2 Signature Information .................................................................................... 1514
  3.7.3 Transformed Signed Data .............................................................................. 1645
  3.7.4 Time stamp / Time mark ................................................................................ 1645
  3.7.5 Information used in Verification .................................................................... 1645
3.8 Query Requirements ............................................................................................ 1645
  3.8.1 Signing Service ............................................................................................. 1645
  3.8.2 Verification Service ....................................................................................... 1645
3.9 Bindings Requirements ........................................................................................ 1645
Appendix A. Revision History .................................................................................... 1817
Appendix B. Notices ................................................................................................... 1918
1 Introduction ............................................................................................................. 4
2 Particular Features of Use Cases .............................................................................. 4
  2.1 Corporate Seal ................................................................................................. 4
  2.2 SOAP Signing ................................................................................................. 4
  2.3 Identified Requestor ......................................................................................... 4
  2.4 Individual Signatures ....................................................................................... 4
  2.5 Long Term Corporate Signatures ..................................................................... 5
  2.6 Delegated Signature Verification ...................................................................... 5
  2.7 Securing the Transform Chain ......................................................................... 5
  2.8 Non-XML Data Signing ................................................................................... 6
  2.9 eNotarization ................................................................................................. 6
  2.10 Court Filings ................................................................................................. 6
  2.11 Client-Side Hashing ...................................................................................... 6
  2.12 Time stamping / Time marking ...................................................................... 6
3 Requirements ........................................................................................................... 7
  3.1 Supported Formats .......................................................................................... 7
    3.1.1 To Be-Signed Format ............................................................................... 7
    3.1.2 Signature Format ...................................................................................... 7
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2 Signature Contents</td>
<td>7</td>
</tr>
<tr>
<td>3.2.1 Requestor Identity</td>
<td>7</td>
</tr>
<tr>
<td>3.2.2 Signing Time</td>
<td>8</td>
</tr>
<tr>
<td>3.2.3 Other Signed/Unsigned Attributes</td>
<td>8</td>
</tr>
<tr>
<td>3.3 Generic Request Requirements</td>
<td>8</td>
</tr>
<tr>
<td>3.3.1 General</td>
<td>8</td>
</tr>
<tr>
<td>3.3.2 Input Preparation</td>
<td>8</td>
</tr>
<tr>
<td>3.3.3 Input Delivery</td>
<td>9</td>
</tr>
<tr>
<td>3.4 Signing Request Requirements</td>
<td>9</td>
</tr>
<tr>
<td>3.4.1 Selective Signing</td>
<td>9</td>
</tr>
<tr>
<td>3.4.2 Signature Placement</td>
<td>9</td>
</tr>
<tr>
<td>3.4.3 Output Delivery</td>
<td>9</td>
</tr>
<tr>
<td>3.4.4 Signature Policy</td>
<td>9</td>
</tr>
<tr>
<td>3.4.5 Explicit Signed/Unsigned Attributes</td>
<td>10</td>
</tr>
<tr>
<td>3.5 Signing Response Requirements</td>
<td>10</td>
</tr>
<tr>
<td>3.5.1 Signature Placement</td>
<td>10</td>
</tr>
<tr>
<td>3.5.2 Output Delivery</td>
<td>10</td>
</tr>
<tr>
<td>3.6 Verification Request Requirements</td>
<td>10</td>
</tr>
<tr>
<td>3.6.1 Selective Verification</td>
<td>10</td>
</tr>
<tr>
<td>3.6.2 General</td>
<td>10</td>
</tr>
<tr>
<td>3.7 Verification Response Requirements</td>
<td>11</td>
</tr>
<tr>
<td>3.7.1 Failure Information</td>
<td>11</td>
</tr>
<tr>
<td>3.7.2 Signature Information</td>
<td>11</td>
</tr>
<tr>
<td>3.7.3 Transformed Signed Data</td>
<td>11</td>
</tr>
<tr>
<td>3.7.4 Time stamp / Time mark</td>
<td>11</td>
</tr>
<tr>
<td>3.7.5 Information used in Verification</td>
<td>11</td>
</tr>
<tr>
<td>3.8 Query Requirements</td>
<td>12</td>
</tr>
<tr>
<td>3.8.1 Signing Service</td>
<td>12</td>
</tr>
<tr>
<td>3.8.2 Verification Service</td>
<td>12</td>
</tr>
<tr>
<td>3.9 Bindings Requirements</td>
<td>12</td>
</tr>
<tr>
<td>Appendix A. Revision History</td>
<td>13</td>
</tr>
<tr>
<td>Appendix B. Notices</td>
<td>14</td>
</tr>
<tr>
<td>Appendix C. Example</td>
<td>15</td>
</tr>
</tbody>
</table>
1 Introduction

This document analyses the use cases given in the OASIS DSS Use Case document and identifies the requirements of Digital Signature Services (DSS) protocols. This analysis identifies the particular features of each of the use cases, along with other requirements which have been identified as being necessary to support digital signature services in an open environment.

2 Particular Features of Use Cases

2.1 Corporate Seal

Submitted by: Carlisle Adams, Entrust
Signature Request: Server checks requestor as authorized to sign on behalf of the organization
Signature Response: Signature only identifies organization, not requestor
Verification: Uses public key belonging to organization

2.2 SOAP Signing

Submitted by: Carlisle Adams, Entrust
Signature Request: Input is XML, requestor may be gateway rather than originator
Signature Response: XML structure which fits in SOAP structure
Verification: Uses public key belonging to organization

2.3 Identified Requestor

Submitted by: Carlisle Adams, Entrust
Signature Request: Requestor must be authenticated
Signature Response: Identifies the requestor
Verification: Uses public key belonging to organization

2.4 Individual Signatures

Submitted by: Nick Pope
2.5 Long Term Corporate Signatures

Submitted by: Nick Pope


Signature Request: Requestor must be authenticated

Signature Response: Signature created using key unique to requestor

Verification: Uses public key belonging to requestor

2.6 Delegated Signature Verification

Submitted by: Pieter Kasselman, Baltimore


Verification Request: Optional specification of signature verification policy

Verification Response:

- Returns valid / invalid indicator
- Return includes information on reason if invalid
- Return includes signature verification policy applied in validating signature
- Response signed by server trusted by verification requestor

2.7 Securing the Transform Chain

Submitted by: Gregor Karlinger, Austria Federal Ministry for Public Services and Sports


For use cases where the relying party would like to check the relationship between the the 'transforms process input data' (which is the data he wants to operate on) and the 'transforms process output data' (which is the data the signing party has actually signed) all the information used by the signing party to compute the transforms process must be signed.

Most of this information is included in a XMLDSIG signature anyway. However, there are some exceptions, for instance imported
stylesheet referred to in an XSLT transform. Those additional
test, for instance as part of a

dsig:Manifest, Signature Request:

- Signed data may be signed data
- Signed data may be in format viewable by human user (e.g. Text, HTML, XML)
- Extensible list of attributes
- Requestor to indicate which attributes may be signed
- Request includes profile of transforms to be applied to signature before signing
- Support for ds:Manifest
- Signed data included in request or referenced by URI
- Support for the signature types defined in XML-DSIG (enveloping, enveloped, detached)

Verification Request:

- Requestor can provide data to be transformed by verification server
- Verification of signatures using ID Reference
- Verification of signatures based on certificate status in the past
- Specify verification policy / profile Specify verification policy / profile
- Support for dsig:Manifest
- Signed data included in request or referenced by URI
- Support for the signature types defined in XML-DSIG (enveloping, enveloped, detached)

Verification Response:

- Identify information about signer / requestor
- In case of invalid signature, indicate which references are correct / incorrect

2.8 Non-XML Data Signing

Submitted by: Merlin Hughes, Baltimore
Signature Response: S/MIME signatures, signed code (e.g. JAR files), proprietary formats
Verification Request: S/MIME signatures, signed code (e.g. JAR files), proprietary formats

2.9 ENotarization in presence of notary

Submitted by: John Messing
Signature Request:

- Notary requests signature upon jurat (attestation of witnessing signing and identity and voluntariness of party’s signature) and an electronically signed document which is logically associated with the jurat.
- Server authenticates notary as authorized to sign (e.g. RFC 3039)
- Server signs jurat and document or its hash on behalf of notary

Signature Response: Signature created using key of server and logical association of notary with the ultimate requestor

Verification: Uses public key of server with logical association of notary’s authentication as maintained at server

[To be completed]

---

2.10 Court Filings

Submitted by: John Messing


Signature Request: Requestor sends document to server (of court, agency or infomediary), which authenticates filer (e.g. RFC 3039)

Signature Response: Signature created using private key of server PLUS logical association of identity with transaction

Verification: Uses public key of server PLUS logical, stored association of filer’s authentication

[To be completed]

2.11 Client-Side Hashing

Signature Request: Requestor sends list of dsig:References to be signed

Verification Request: Requestor sends list of dsig:References to be verified

2.12 Time-stamping / Time-marking

Signature Response: Time-mark included in signed data or time-stamp applied to signed data

Verification: Time-mark or time-stamp included when verifying data
3 Requirements

3.1 Supported Formats

3.1.1 To-Be-Signed Format

- XML
- Binary
- Extensible to others

A DSS service should be able to apply transforms to, and sign, any sequence of octets. We will consider only XML in particular for now, and assume everything else can be treated as raw “binary” data without any transforms. Encoding and canonicalization transforms for non-XML data types such as HTML, text, multimedia, etc., are beyond the scope of DSS.

3.1.2 Signature Format

- XML-DSIG
- XML Timestamp Tokens
- CMS/PKCS#7 (RFC 2630)
- Extensible to others

We will focus on XML-DSIG signatures applied to XML content. As this is the most flexible case and has the most complications around transforms, references, signature placement, etc., the resulting protocol should easily generalize to simpler formats like CMS (which supports the email and code-signing use cases) or OpenPGP. Other formats that DSS could be extended to in the future include low-level signature schemes like PKCS #1 v1.5, RFC 3161 TimeStampTokens, and even tokens created from an input hash value which aren’t properly signatures at all, such as timestamps using “linking” schemes. We will also define an XML Timestamp Token format, which will be similar to an RFC 3161 TimeStampToken, and can be used for time stamping XML-DSIG signatures.

[QUESTION 1: DO WE WANT TO CONSIDER CMS/PKCS#7 EXPLICITLY, OR JUST GROUP IT UNDER “EXTENSIBLE TO OTHERS”?]

[QUESTIONS TO BE ANSWERED]

[QUESTIONS TO BE ANSWERED]
3.2 Signature Contents

These requirements don’t apply to the DSS protocol per se, but to the signed and unsigned attributes of the signatures that it produces.

3.2.1 Requestor Identity

- None
- String
- SAML Assertion (or a reference to one)

The server may represent the requestor's name or other identifying information in a signed property attribute, as either a string (such as an email address added as a signed attribute to a CMS/PKCS#7 signature), or a SAML Assertion. The Assertion may contain information on how the requestor authenticated to the DSS, so the relying party can determine how much trust to place in the signature. The Assertion may be generated by the DSS server, or by an Authentication Authority, in which case the DSS server would be simply passing it on.

[QUESTION: Is it unduly restrictive to limit permissible methods of representing a Requestor's Identity to none, a string attribute or SAML Assertion?]

3.2.2 Signing Time

- None
- Time mark
- Time-stamp

A time-stamp may be of the following types:

- Simple time stamp via an RFC 3161 TimeStampToken
- Simple time stamp via XML (details still under discussion)
- Linked time stamp via ISO 18014-3 extensions to TimeStampToken (under discussion)
- Linked time stamp via XML (under discussion)

A time-stamp may be applied:

- over one or more objects within the signed data, as an affirmation from an independent party of the time at which the signed data existed, which is on or before the signing time
- over the signature as an affirmation from an independent party that the signature was created on or before the time-stamp time. This can be used to prove that any revocations after the time-stamp time are not applicable.
over the signature, certificates and revocation information signed by an archiving service
in order to protect against weak algorithms, and key compromise.

The Signing Time may be a signed attribute added by the signer (a “time mark”), or an
unsigned attribute generated by a 3rd party (a “time stamp”). The time stamp is best applied
to the signed data as a whole to prove that the signature was applied before the given time,
and hence that any revocations after that time are not applicable.

A time stamp may be retrieved through the use of the DSS protocol, or through some other
protocol such as RFC 3161. XML time stamps and time marks should share the same data
items as far as possible – when these items are included in a signed attribute, the result is a
time mark, and when they are included in a document that also contains a hash, and this
document is signed, the result is a time stamp. Both approaches should satisfy legal and
operational requirements for “time stamping”.

[QUESTION 2: DO WE NEED TO SUPPORT REQUEST TIME AS DISTINCT FROM
SIGNING TIME?]

3.2.3 Other Signed/Unsigned Attributes

- Requestor may request additional attributes
- Server has final say

The requestor may send the DSS server other attributes to include in the signature. The
server may accept or refuse the request, and may add other attributes on its own
initiative. RFC 3126 on “Long term electronic signatures” is the CMS analogue of XAdES
“XML Advanced Electronic Signatures”, and both documents contain examples of attributes
that might be added.

[QUESTION 3: DO WE WANT TO MANDATE SUPPORT FOR XAdES, OR JUST LEAVE IT
AS AN EXAMPLE?]

3.3 Generic Request Requirements

These requirements apply to both the Signature Request and Verification Request messages
sent by clients, and deal with how the data to be signed or verified is transferred from client to
server.

3.3.1 General

- Multiple inputs may be allowed, depending on the signature format
- Each input may be of a different format
- An XML input may include the whole document, or only the part of it that is to be signed
An XML-DSIG signature’s dsig:References can refer to multiple documents, so the client must be able to transfer multiple documents or portions of documents to the server. Each individual document may be of a different type (see 3.1.1), and may be prepared and delivered in different ways (see 3.3.2 and 3.3.3). Also, it must be possible for the client to tell the server about schema/DTD validation information for these documents; otherwise the server cannot know which XML elements have defined ID attributes, and won’t be able to create or verify XML-DSIG signatures that refer to the to-be-signed data via ID references.

### 3.3.2 Input Preparation

- Plain Unaltered
- Transformed
- Hashed

The client may send an input document in unaltered form, or may apply some transforms and send the transformed data, or may even apply the transforms, then hash the transformed data, then send the hash. Transferring the document plain unaltered or transformed allows the server to archive and review what it is signing or verifying. Sending Transferring it hashed is more efficient and preserves client privacy. Applying transforms and hash algorithms on the client side gives the client flexibility to use transforms and hash algorithms the server doesn’t know about. When client-side transforms or hashing are applied, the client must tell the server which transforms and which hash algorithms were applied, so the server can represent these in the signature, for example, within a dsig:Reference.

Client-side hashing requires the client to have knowledge of which hash algorithms the server is capable of signing, so the protocol needs a query exchange where clients can discover these (see 3.8).

### 3.3.3 Input Delivery

- Direct
- Indirect, via URI

### 3.3.4 Authentication

- None (or by other means, such as TLS/SSL, IPsec)
- Directly to the server within the protocol (perhaps using WS-Security?)
- Indirectly via an Authentication Authority (perhaps using SAML?)
- Combinations of above

The protocol should be capable of supporting the above approaches to mutual authentication of the client and server to each other. Bindings will fill in the details and mandate specific techniques. Authentication is performed at a lower layer than the DSS protocol itself. The
details of authentication techniques used to access a DSS server are not within the scope of this specification and there is no attempt by this specification to restrict such methods of authentication.

3.4 Signing Request Requirements

These requirements apply to the message sent by a client to request a signature.

3.4.1 Selective Signing

- Which elements to sign
- Which transforms to apply

The client may specify which particular parts of the documents he sent he wants to sign, and how these should be transformed. The client may need to query the server to determine which transforms are supported (see 3.8.1). The client may refer to a transform profile, which specifies a sequence of transforms, instead of stating each transform individually.

3.4.2 Signature Placement

- Where and how to insert signature

If the client wants an enveloping signature, he must specify what part of which document should be enveloped, and where this signature should be inserted in these documents, if at all. If the client wants an enveloped signature, he must also specify where this signature should be inserted.

3.4.3 Output Delivery

- How to return output
  - Immediate or delayed

The client may specify whether he just wants to receive the signature back, and splice it into (or send it alongside) the documents himself, or whether he wants to receive the document with the signature embedded in it (for enveloped; vice versa for enveloping). The server may return the signature immediately, or may return a transaction identifier, which instructs the client to check back periodically and see if the signature is ready yet.

3.4.4 Signature Policy

- Whether/how requestor identity is included
- Whether/how signing time is included
- What key/certificate is used to sign
- What validation/key info is used to sign
• What policy/commitment identifiers are included
• What other attributes are included
The client may use the server’s default policy, or select a policy (probably through some sort of identifier). The client may query the server to see what policies are supported (see 3.8.1).

3.4.5 Explicit Signed/Unsigned Attributes
• The client may ask the server to insert particular attributes into the signature (see 3.2.3)

3.5 Signing Response Requirements
These requirements apply to the message returned by a DSS server after a client requests it to create a signature.

3.5.1 Signature Placement
• Detached
• Enveloping – Server only returns signature
• Enveloping – Server returns signature with embedded input document
• Enveloped – Server only returns signature
• Enveloped – Server returns input document with embedded signature

3.5.2 Output Delivery
• Direct
  ←Indirect, via URI
If the DSS service is returning a full document along with the signature (enveloped or enveloping), then it can return it directly within the protocol message, or via a URI.
[QUESTION 4: DO WE REALLY WANT TO SUPPORT THIS MUCH FLEXIBILITY?]

3.6 Verification Request Requirements
These requirements apply to the message sent by a client to verify a signature.

3.6.1 Selective Verification
• Which signature to verify
  • Whether to perform signature validation, reference validation, or both
  • Which references to verify if performing reference validation
The client may specify parameters identifying the signed information to verify. An XML-DSIG signed document may have multiple signatures, each with multiple references, and these references may include dsig:Manifests. The client may indicate which signature to verify, whether or not references should be verified (and perhaps the client may choose to verify all or some of these himself), and whether or not dsig:Manifests should be verified.

### 3.6.2 General

- Which policy to use (including trust settings)
- What time in the past to verify signature at (if this is desired)

**[Question 5]:** IF THE SERVER LEARNS ON THE 10th THAT A KEY WAS COMPROMISED ON THE 8th, WHEN A CLIENT ASKS TO VERIFY AS IF IT WAS THE 9th, WHAT SHOULD THE SERVER SAY? WHAT HE SHOULD HAVE SAID THEN, OR WHAT HE WOULD HAVE SAID THEN?

- Explicit key and validation info submitted by client (Certificates, CRLS, OCSP responses)
- Whether/how transformed signed data should be returned

### 3.7 Verification Response Requirements

These requirements apply to the message returned by a DSS server after a client requests it to verify a signature.

#### 3.7.1 Failure Information

- Reason code
- Which references/manifests failed, if appropriate

The reason codes will differentiate between failures in reference validation, signature validation, an untrusted or revoked signing key, or something unknown/unsupported (a signature format, hash algorithm, transform, verification policy, etc.).

#### 3.7.2 Signature Information

- Requestor / Signer identity
- Signing time
- Signature policy
- Signature validity interval
- Others?
The server may process the attributes and keying information associated with the signature, and return information about it to the client, to save the client the difficulty of parsing X.509 certificates, SAML Assertions, XAdES attributes, and so on.

3.7.3 Transformed Signed Data

- Direct
- Indirect, via URI

If requested by the client, the server may return the transformed data that was signed (i.e. the to-be-signed data after the transforms have been applied), to save the client the difficulty of applying the transforms itself.

3.7.4 Time stamp / Time mark

The verification service may return a time stamp or time mark indicating when the verification was carried out. If this is close to the time when the data was signed this may be used as evidence that the signature was valid near the signing time. If the verification was carried out for some time in the past (see 3.6.2, bullet 2), a time stamp or time mark for that time should be returned as well.

3.7.5 Information used in Verification

The verification service may also return information used in verification (CRLs, OCSP responses, certificates).

3.8 Query Requirements

3.8.1 Signing Service

- What hash algorithms are supported (see 3.3.2)
- What transforms and transform profiles are supported (see 3.4.1)
- What policies are supported (see 3.4.4)

3.8.2 Verification Service

- What signature algorithms are supported
- What transforms are supported
- What policies are supported (see 3.6.2)

3.9 Bindings Requirements

- Each binding must define a transport
• Each binding must define confidentiality and integrity means
• Each binding must define authentication means
• Each binding must define which to-be-signed and signature formats it supports
• Each binding may otherwise profile or restrict DSS. Nonetheless, there will be some common minimum requirements that must be fulfilled by every DSS.
## Appendix A. Revision History

<table>
<thead>
<tr>
<th>Rev</th>
<th>Date</th>
<th>By Whom</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft-001</td>
<td>2003-3-22</td>
<td>Trevor Perrin</td>
<td>Initial version based on Nick’s draft</td>
</tr>
<tr>
<td>Draft-012</td>
<td>2003-3-23</td>
<td>Trevor Perrin</td>
<td>Nick’s clarifications on time-stamping</td>
</tr>
<tr>
<td>Draft-03</td>
<td>2003-4-2</td>
<td>Trevor Perrin</td>
<td>Incorporating feedback from list</td>
</tr>
</tbody>
</table>
Appendix B. Notices

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's procedures with respect to rights in OASIS specifications can be found at 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 implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

Copyright © The Organization for the Advancement of Structured Information Standards [OASIS] 2002 and 2003. All Rights Reserved.

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 paragraph are included on all such copies and derivative works. However, this document itself does not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document 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 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.