



# Web Services Security X.509 Certificate Token Profile

Working Draft 10, 19th August 2003

**Document identifier:**

urn:oasis:names:tc:WSS:1.0:profiles:X509-10

**Location:**

<http://www.oasis-open.org/committees/download.php/2427/WSS-X509-10.pdf>

**Editors:**

Phillip Hallam-Baker, VeriSign  
Chris Kaler, Microsoft  
Ronald Monzillo, Sun  
Anthony Nadalin, IBM

**Contributors:**

**Current voting members of the WSS TC (as of July 1st 2003)**

*Note: It is assumed that we will update this on the day of Committee Spec to be the current list*

Gene	Thurston	AmberPoint
Frank	Siebenlist	Argonne National Lab
Merlin	Hughes	Baltimore Technologies
Irving	Reid	Baltimore Technologies
Peter	Dapkus	BEA
Hal	Lockhart	BEA
Symon	Chang	CommerceOne
Thomas	DeMartini	ContentGuard
Guillermo	Lao	ContentGuard
TJ	Pannu	ContentGuard
Shawn	Sharp	Cyclone Commerce
Ganesh	Vaideeswaran	Documentum
Sam	Wei	Documentum
John	Hughes	Entegrity
Tim	Moses	Entrust
Toshihiro	Nishimura	Fujitsu
Tom	Rutt	Fujitsu
Jason	Rouault	HP
Yutaka	Kudo	Hitachi
Maryann	Hondo	IBM
Kelvin	Lawrence	IBM (co-Chair)
Anthony	Nadalin	IBM
Nataraj	Nagaratnam	IBM
Don	Flinn	Individual
Bob	Morgan	Individual
Paul	Cotton	Microsoft

43	Vijay	Gajjala	Microsoft
44	Chris	Kaler	Microsoft (co-Chair)
45	Chris	Kurt	Microsoft
46	John	Shewchuk	Microsoft
47	Prateek	Mishra	Netegrity
48	Frederick	Hirsch	Nokia
49	Senthil	Sengodan	Nokia
50	Lloyd	Burch	Novell
51	Ed	Reed	Novell
52	Charles	Knouse	Oblix
53	Steve	Anderson	OpenNetwork (Sec)
54	Vipin	Samar	Oracle
55	Jerry	Schwarz	Oracle
56	Eric	Gravengaard	Reactivity
57	Stuart	King	Reed Elsevier
58	Andrew	Nash	RSA Security
59	Rob	Philpott	RSA Security
60	Peter	Rostin	RSA Security
61	Martijn	de Boer	SAP
62	Pete	Wenzel	SeeBeyond
63	Jonathan	Tourzan	Sony
64	Yassir	Elley	Sun Microsystems
65	Jeff	Hodges	Sun Microsystems
66	Ronald	Monzillo	Sun Microsystems
67	Jan	Alexander	Systinet
68	Michael	Nguyen	The IDA of Singapore
69	Don	Adams	TIBCO
70	John	Weiland	US Navy
71	Phillip	Hallam-Baker	VeriSign
72	Morten	Jorgensen	Vordel

**Contributors of input documents (if not already listed above) :**

74	Bob	Blakley	IBM
75	Joel	Farrell	IBM
76	Satoshi	Hada	IBM
77	Hiroshi	Maruyama	IBM
78	David	Melgar	IBM
79	Bob	Atkinson	Microsoft
80	Allen	Brown	Microsoft
81	Giovanni	Della-Libera	Microsoft
82	Johannes	Klein	Microsoft
83	Scott	Konersmann	Microsoft
84	Brian	LaMacchia	Microsoft
85	Paul	Leach	Microsoft
86	John	Manferdell	Microsoft
87	Dan	Simon	Microsoft
88	Hervey	Wilson	Microsoft
89	Hemma	Prafullchandra	VeriSign

**Abstract:**

91 This document describes how to use X.509 Certificates with the [WS-Security](#)  
92 specification.

**Status:**

94 This is an interim draft.

95 Committee members should send comments on this specification to the [wss@lists.oasis-](mailto:wss@lists.oasis-)  
96 [open.org](mailto:wss@lists.oasis-open.org) list. Others should subscribe to and send comments to the [wss-](#)

97  
98

[comment@lists.oasis-open.org](mailto:comment@lists.oasis-open.org) list. To subscribe, visit <http://lists.oasis-open.org/ob/adm.pl>.

99  
100  
101  
102

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 WS-Security TC web page (<http://www.oasis-open.org/committees/wss/ipr.php>).

---

## Table of Contents

104	1	Introduction (Non-Normative).....	5
105	2	Notations and Terminology .....	6
106	2.1	Notational Conventions.....	6
107	2.2	Namespaces.....	6
108	2.3	Terminology .....	6
109	3	Usage.....	7
110	3.1	Token types .....	7
111	3.1.1	wsse:X509v3 Token Type.....	7
112	3.1.2	wsse:X509PKIPathv1 Token Type.....	7
113	3.1.3	wsse:PKCS7 Token Type .....	7
114	3.2	Token References .....	7
115	3.2.1	Reference to a Subject Key Identifier.....	8
116	3.2.2	Reference to a Security Token .....	8
117	3.2.3	Reference to an Issuer and Serial Number .....	8
118	3.3	Signature .....	8
119	3.3.1	Key Identifier .....	9
120	3.3.2	Reference to a Binary Security Token .....	9
121	3.3.3	Reference to an Issuer and Serial Number .....	10
122	3.4	Encryption .....	11
123	3.5	Error Codes .....	12
124	3.6	Threat Model and Countermeasures .....	12
125	4	References.....	13
126		Appendix A: Revision History .....	14
127		Appendix B: Notices.....	15
128			

---

## 129 **1 Introduction (Non-Normative)**

130 This specification describes the use of the X.509 authentication framework with the [Web Services](#)  
131 [Security: SOAP Message Security](#) specification [WS-Security].

132 An X.509 certificate specifies a binding between a public key and a set of attributes that includes  
133 (at least) a subject name, issuer name, serial number and validity interval. This binding may be  
134 subject to subsequent revocation advertised by mechanisms that include issuance of CRLs,  
135 OCSP tokens or mechanisms that are outside the X.509 framework, such as XKMS.

136 An X.509 certificate may be used to validate a public key that may be used to authenticate a WS-  
137 Security-enhanced message or to identify the public key with which a WS-Security-enhanced  
138 message has been encrypted.

---

## 139 2 Notations and Terminology

140 This section specifies the notations, namespaces and terminology used in this specification.

### 141 2.1 Notational Conventions

142 This document uses the notational conventions defined in [SOAP Message Security \[WS-](#)  
143 [Security\]](#).

144 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",  
145 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be  
146 interpreted as described in RFC2119 [KEYWORDS].

### 147 2.2 Namespaces

148 The [XML namespace](#) URIs that MUST be used by implementations of this specification are as  
149 follows (note that elements used in this specification are defined in one or other of these  
150 namespaces):

```
151 http://schemas.xmlsoap.org/ws/2003/06/secext  
152 http://schemas.xmlsoap.org/ws/2003/06/utility
```

153 The following namespace prefixes are used in this document:

Prefix	Namespace
S	<a href="http://www.w3.org/2001/12/soap-envelope">http://www.w3.org/2001/12/soap-envelope</a>
ds	<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>
xenc	<a href="http://www.w3.org/2001/04/xmlenc#">http://www.w3.org/2001/04/xmlenc#</a>
wsse	<a href="http://schemas.xmlsoap.org/ws/2003/06/secext">http://schemas.xmlsoap.org/ws/2003/06/secext</a>
wsu	<a href="http://schemas.xmlsoap.org/ws/2003/06/utility">http://schemas.xmlsoap.org/ws/2003/06/utility</a>

154 *Table 1- Namespace prefixes*

### 155 2.3 Terminology

156 This specification adopts the terminology defined in [SOAP Message Security \[WS-Security\]](#).

157 Readers are presumed to be familiar with the definitions of terms in the [Internet Security Glossary](#)  
158 [Glossary].

---

## 159 3 Usage

160 This specification describes the syntax and processing rules for the use of the X.509  
161 authentication framework with the [Web Services Security: SOAP Message Security](#) specification  
162 [[WS-Security](#)].

### 163 3.1 Token types

164 This profile defines the syntax of, and processing rules for, three types of binary security token  
165 using the QName values specified in Table 2.

166

Token	ValueType QName	Description
Single certificate	wsse:X509v3	An X.509 v3 signature-verification certificate
Certificate Path	wsse:X509PKIPathv1	An ordered list of X.509 certificates packaged in a PKIPath
Set of certificates and CRLs	wsse:PKCS7	A list of X.509 certificates and (optionally) CRLs packaged in a PKCS#7 wrapper

167

*Table 2 – Token types*

#### 168 3.1.1 wsse:X509v3 Token Type

169 The type of the end-entity that is authenticated by a certificate used in this manner is a matter of  
170 policy that is outside the scope of this specification.

#### 171 3.1.2 wsse:X509PKIPathv1 Token Type

172 The wsse:X509PKIPathv1 token type MAY be used to represent a certificate path.

#### 173 3.1.3 wsse:PKCS7 Token Type

174 The wsse:PKCS7 token type MAY be used to represent a certificate path. It is RECOMMENDED  
175 that applications use the PKIPath object for this purpose instead.

176 The order of the certificates in a PKCS#7 data structure is not significant. If an ordered certificate  
177 path is converted to PKCS#7 encoded bytes and then converted back, the order of the  
178 certificates may not be preserved. Processors SHALL NOT assume any significance to the order  
179 of the certificates in the data structure. See [PKCS7] for more information.

## 180 3.2 Token References

181 In order to ensure a consistent processing model across all the token types supported by [WS-](#)  
182 [Security](#), the wsse:SecurityTokenReference element SHALL be used to specify all references to  
183 X.509 token types in signature or encryption elements that comply with this profile.

184 A wsse:SecurityTokenReference MAY reference an X.509 token type by one of the following  
185 means:

186 Reference to a Subject Key Identifier  
187 The wsse:SecurityTokenReference element contains a wsse:KeyIdentifier element that  
188 specifies the token data by means of a X.509 SubjectKeyIdentifier reference.  
189 Reference to a Binary Security Token  
190 The wsse:SecurityTokenReference element contains a wsse:Reference element that  
191 references a local wsse:BinarySecurityToken element or a remote data source that  
192 contains the token data itself.  
193 Reference to an Issuer and Serial Number  
194 The wsse:SecurityTokenReference element contains a ds:X509IssuerSerial element that  
195 uniquely identifies an end entity certificate by its X.509 Issuer and Serial Number.

### 196 **3.2.1 Reference to a Subject Key Identifier**

197 The wsse:KeyIdentifier element is used to specify a reference to an X.509 certificate by means of  
198 a reference to its X.509 SubjectKeyIdentifier attribute.

199 The wsse:SecurityTokenReference from which the reference is made contains the  
200 wsse:KeyIdentifier element. The wsse:KeyIdentifier element MUST have a ValueType attribute  
201 with the value wsse:X509SubjectKeyIdentifier and its contents MUST be the value of the  
202 certificate's X.509 SubjectKeyIdentifier extension, encoded as per the wsse:KeyIdentifier  
203 element's EncodingType attribute. For the purposes of this specification, the value of the  
204 SubjectKeyIdentifier extension is the contents of the KeyIdentifier octet string, excluding the  
205 encoding of the octet string prefix.

### 206 **3.2.2 Reference to a Security Token**

207 The wsse:Reference element is used to reference an X.509 security token value by means of a  
208 URI reference.

209 The URI reference MAY be internal in which case the URI reference SHOULD be a bare name  
210 XPointer reference to a wsse:BinarySecurityToken element contained in a preceding message  
211 header that contains the binary X.509 security token data.

### 212 **3.2.3 Reference to an Issuer and Serial Number**

213 The ds:IssuerSerial element is used to specify a reference to an X.509 security token by means  
214 of the certificate issuer name and serial number.

215 The ds:IssuerSerial element is a direct child of the wsse:SecurityTokenReference element in  
216 which the reference is made.

## 217 **3.3 Signature**

218 Signed data MAY specify the certificate associated with the signature using any of the X.509  
219 security token types and references defined in this specification.

220 An X.509 certificate specifies a binding between a public key and a set of attributes that includes  
221 (at least) a subject name, issuer name, serial number and validity interval. Other attributes MAY  
222 specify constraints on the use of the certificate or affect the recourse that may be open to a  
223 relying party that depends on the certificate. A given public key may be specified in more than  
224 one X.509 certificate; consequently a given public key MAY be bound to two or more distinct sets  
225 of attributes.

226 It is therefore necessary to ensure that a signature created under an X.509 certificate token  
227 uniquely and irrefutably specifies the certificate under which the signature was created.

228 Implementations SHOULD protect against a certificate substitution attack by including either the  
229 certificate itself or an immutable and unambiguous reference to the certificate within the scope of  
230 the signature according to the method used to reference the certificate as follows:



231

### 3.3.1 Key Identifier

232 The wsse:KeyIdentifier element does not guarantee an immutable and unambiguous reference to  
233 the certificate referenced. Consequently implementations that use this form of reference within a  
234 signature SHOULD employ the wsse:SecurityTokenReference deferencing transform within a  
235 core barename XPointer reference to the signature key information in order to ensure that the  
236 referenced certificate is signed, and not just the ambiguous reference.

237 The following example shows a certificate referenced by means of a Key Identifier

```
238 <S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope">
239   <S:Header>
240     <wsse:Security
241       xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
242       xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
243       <wsse:BinarySecurityToken
244         wsu:Id="AlUdAQQ8MDqAEEVs"
245         ValueType="wsse:X509v3"
246         EncodingType="wsse:Base64Binary">
247         MIEZzCCA9CgAwIBAgIQEmtJZc0...
248       </wsse:BinarySecurityToken>
249       <ds:Signature
250         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
251         <ds:SignedInfo>...
252         <ds:Reference URI="#body">...</ds:Reference>
253         <ds:Reference URI="#keyinfo">
254           <ds:Transforms>
255             <ds:Transform
256               Algorithm="http://schemas.xmlsoap.org/2003/06/STR-Transform">
257               <ds:CanonicalizationMethod Algorithm="..." />
258             </ds:Transform>
259           </ds:Transforms>...
260         </ds:Reference>
261         </ds:SignedInfo>
262         <ds:SignatureValue>HFLP...</ds:SignatureValue>
263         <ds:KeyInfo Id="keyinfo">
264           <wsse:SecurityTokenReference>
265             <wsse:KeyIdentifier EncodingType="wsse:Base64Binary"
266               ValueType="wsse:X509SubjectKeyIdentifier">
267               MIGfMa0GCSq..
268             </wsse:KeyIdentifier>
269           </wsse:SecurityTokenReference>
270         </ds:KeyInfo>
271       </ds:Signature>
272     </wsse:Security>
273   </S:Header>
274   <S:Body wsu:Id="body"
275     xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
276     ...
277   </S:Body>
278 </S:Envelope>
```

### 279 3.3.2 Reference to a Binary Security Token

280 The signed data SHOULD contain a core bare name XPointer reference to the  
281 wsse:BinarySecurityToken element that contains the security token referenced, or a core  
282 reference to the external data source containing the security token.

283 The following example shows a certificate embedded in a wsse:BinarySecurityToken element and  
284 referenced by URI within a signature.

```
285 <S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope">
```

```

286 <S:Header>
287   <wsse:Security
288     xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
289     xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
290     <wsse:BinarySecurityToken
291       wsu:Id="binarytoken"
292       ValueType="wsse:X509v3"
293       EncodingType="wsse:Base64Binary">
294       MIEZzCCA9CgAwIBAgIQEmtJZc0...
295     </wsse:BinarySecurityToken>
296     <ds:Signature
297       xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
298       <ds:SignedInfo>...
299         <ds:Reference URI="#body">...</ds:Reference>
300         <ds:Reference URI="#binarytoken">...</ds:Reference>
301       </ds:SignedInfo>
302       <ds:SignatureValue>HFLP...</ds:SignatureValue>
303       <ds:KeyInfo>
304         <wsse:SecurityTokenReference>
305           <wsse:Reference URI="#binarytoken" />
306         </wsse:SecurityTokenReference>
307       </ds:KeyInfo>
308     </ds:Signature>
309   </wsse:Security>
310 </S:Header>
311 <S:Body wsu:Id="body"
312   xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
313   ...
314 </S:Body>
315 </S:Envelope>

```

### 316 3.3.3 Reference to an Issuer and Serial Number

317 The signed data SHOULD contain a core bare name XPointer reference to the ds:KeyInfo  
318 element that contains the security token reference.

319 The following example shows a certificate referenced by means of its issuer name and serial  
320 number.

```

321 <S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope">
322   <S:Header>
323     <wsse:Security
324       xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
325       xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
326     <ds:Signature
327       xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
328       <ds:SignedInfo>...
329         <ds:Reference URI="#body"></ds:Reference>
330         <ds:Reference URI="#keyinfo"></ds:Reference>
331       </ds:SignedInfo>
332       <ds:SignatureValue>HFLP...</ds:SignatureValue>
333       <ds:KeyInfo Id="keyinfo">
334         <wsse:SecurityTokenReference>
335           <ds:X509IssuerSerial>
336             <ds:X509IssuerName>
337               DC=ACMECorp, DC=com
338             </ds:X509IssuerName>
339             <ds:X509SerialNumber>12345678</X509SerialNumber>
340           </ds:X509IssuerSerial>
341         </wsse:SecurityTokenReference>
342       </ds:KeyInfo>
343     </ds:Signature>

```

```

344     </wsse:Security>
345 </S:Header>
346 <S:Body wsu:Id="body"
347     xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
348     ...
349 </S:Body>
350 </S:Envelope>

```

### 351 3.4 Encryption

352 Encrypted keys or data MAY identify a key required for decryption by identifying the  
353 corresponding key used for encryption by means of any of the X.509 security token types or  
354 references specified herein.

355 Since the sole purpose is to identify the decryption key it is not necessary to specify either a trust  
356 path or the specific contents of the certificate itself.

357 It is RECOMMENDED that implementations specify an encryption key by reference to the Issuer  
358 and Serial Number of an X509v3 certificate security token.

359 The following example shows a decryption key referenced by means of the issuer name and  
360 serial number of an associated certificate.

```

361 <S:Envelope
362     xmlns:S="http://www.w3.org/2001/12/soap-envelope"
363     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
364     xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
365     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
366 <S:Header>
367     <wsse:Security>
368         <xenc:EncryptedKey>
369             <xenc:EncryptionMethod Algorithm="..."/>
370             <ds:KeyInfo>
371                 <wsse:SecurityTokenReference>
372                     <ds:X509IssuerSerial>
373                         <ds:X509IssuerName>
374                             DC=ACMECorp, DC=com
375                         </ds:X509IssuerName>
376                         <ds:X509SerialNumber>12345678</X509SerialNumber>
377                     </ds:X509IssuerSerial>
378                 </wsse:SecurityTokenReference>
379             </ds:KeyInfo>
380             <xenc:CipherData>
381                 <xenc:CipherValue>...</xenc:CipherValue>
382             </xenc:CipherData>
383             <xenc:ReferenceList>
384                 <xenc:DataReference URI="#encrypted"/>
385             </xenc:ReferenceList>
386         </xenc:EncryptedKey>
387     </wsse:Security>
388 </S:Header>
389 <S:Body>
390     <xenc:EncryptedData Id="encrypted" Type="...">
391         <xenc:CipherData>
392             <xenc:CipherValue>...</xenc:CipherValue>
393         </xenc:CipherData>
394     </xenc:EncryptedData>
395 </S:Body>
396 </S:Envelope>

```

### 397 **3.5 Error Codes**

398 When using X.509 certificates, the error codes defined in the [SOAP Message Security](#) [WS-  
399 Security] specification MUST be used.

400 If an implementation requires the use of a custom error it is recommended that a sub-code be  
401 defined as an extension of one of the codes defined in the [SOAP Message Security](#) specification.

### 402 **3.6 Threat Model and Countermeasures**

403 The use of X509 certificates with [WS-Security](#) introduces no new threats beyond those identified  
404 in [SOAP Message Security](#).

405 Message alteration and eavesdropping can be addressed by using the integrity and confidentiality  
406 mechanisms described in [SOAP Message Security](#). Replay attacks can be addressed by using  
407 message timestamps and caching, as well as other application-specific tracking mechanisms.

408 For X.509 certificates, identity is authenticated by use of keys, man-in-the-middle attacks are  
409 generally mitigated.

410 It is strongly RECOMMENDED that all relevant and immutable message data be signed.

411 It should be noted that transport-level security MAY be used to protect the message and the  
412 security token as an alternative.

---

## 4 References

- 413
- 414     **[Glossary]**         Informational RFC 2828, "[Internet Security Glossary](#)," May 2000.
- 415     **[KEYWORDS]**        S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels,"  
416     [RFC 2119](#), Harvard University, March 1997
- 417     **[SOAP]**             W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May 2000.
- 418     **[URI]**              T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers  
419     (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C. Irvine, Xerox  
420     Corporation, August 1998.
- 421     **[WS-Security]**     [http://www.oasis-open.org/committees/download.php/1686/WSS-  
422     SOAPMessageSecurity-12-04021.pdf](http://www.oasis-open.org/committees/download.php/1686/WSS-SOAPMessageSecurity-12-04021.pdf)
- 423     **[XML-ns]**            W3C Recommendation, "[Namespaces in XML](#)," 14 January 1999.
- 424     **[XML Signature]**    W3C Recommendation, "[XML Signature Syntax and Processing](#)," 12  
425     February 2002.
- 426     **[PKCS7]**            **TBS** <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html>
- 427     **[X509]**             **TBS**
- 428
- 429

---

## Appendix A: Revision History

Rev	Date	What
01	18-Sep-02	Initial draft based on input documents and editorial review
03	30-Jan-03	Changes in title
04	19-May-03	Added by reference and pkipath modes of cert identification. Added section 1 introduction, changes to formatting etc.
05	6 June 2003	
06	20 June 2003	Included examples showing how tokens must be referenced from signatures and cipher values. Defined how key-agreement keys are to be conveyed in a Security header.
07	4 August 2003	Modifications to KeyIdentifier handling and use of SecurityTokenReference. Changes to the acknowledgements section.
08	6 August 2003	Reorganization of major sections to simplify flow
09	14 August 2003	Editorial corrections raised in off list emails.
10	19 August 2003	Editorial corrections raised in profile teleconference.

---

## Appendix B: Notices

433 OASIS takes no position regarding the validity or scope of any intellectual property or other rights  
434 that might be claimed to pertain to the implementation or use of the technology described in this  
435 document or the extent to which any license under such rights might or might not be available;  
436 neither does it represent that it has made any effort to identify any such rights. Information on  
437 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS  
438 website. Copies of claims of rights made available for publication and any assurances of licenses  
439 to be made available, or the result of an attempt made to obtain a general license or permission  
440 for the use of such proprietary rights by implementors or users of this specification, can be  
441 obtained from the OASIS Executive Director.

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

445 Copyright © OASIS Open 2003. *All Rights Reserved.*

446 This document and translations of it may be copied and furnished to others, and derivative works  
447 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,  
448 published and distributed, in whole or in part, without restriction of any kind, provided that the  
449 above copyright notice and this paragraph are included on all such copies and derivative works.  
450 However, this document itself does not be modified in any way, such as by removing the  
451 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS  
452 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual  
453 Property Rights document must be followed, or as required to translate it into languages other  
454 than English.

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

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

462