



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

Bindings for the Service Provisioning Markup Language (SPML) Version 1.0

OASIS Committee Specification, June 3, 2003

Document identifier:

cs-pstc-spml-bindings-1.0.doc

Location:

<http://www.oasis-open.org/committees/provision/docs/>

Editor:

Jeff Bohren, OpenNetwork Technologies

Contributors:

Steve Anderson, OpenNetwork Technologies
Darran Rolls, Waveset Technologies

Abstract:

This specification defines protocol bindings and profiles for the use of SPML request-response elements in communications protocols and applications.

30 **Table of Contents**

31	1	Protocol Bindings.....	3
32	1.1	SOAP/HTTP Binding	3
33	1.1.1	Basic Operations	3
34	1.1.2	Authentication.....	4
35	1.1.3	Message Integrity	4
36	1.1.4	Message Confidentiality.....	4
37	1.1.5	Error Reporting	4
38	1.1.6	Declaring Supported SPML Operations	5
39	1.2	File Binding	5
40	1.2.1	Bulk Processing Files	5
41	1.2.2	Error Reporting	6
42	1.2.3	Example Bulk Processing Files.....	6
43	1.2.4	Provisioning Schema Files	7
44	1.2.5	Example Provisioning Schema File.....	7
45		Appendix A. Sample WSDL Definition	8
46		Appendix B. References	12
47		Appendix C. Revision history	13
48		Appendix D. Notices.....	14
49			

50 1 Protocol Bindings

51 The following sections define SPML protocol bindings sanctioned by the OASIS PS TC. Two bindings, the
52 SOAP/HTTP binding and the file binding, are defined.

53 1.1 SOAP/HTTP Binding

54 SOAP (Simple Object Access Protocol) 1.1 **[SOAP1.1]** is a specification for RPC-like interactions and
55 message communications using XML and HTTP. It has three main parts. One is a message format that
56 uses an envelope and body metaphor to wrap XML data for transmission between parties. The second is
57 a restricted definition of XML data for making strict RPC-like calls through SOAP, without using a
58 predefined XML schema. Finally, it provides a binding for SOAP messages to HTTP and extended HTTP.

59 This binding is limited to SOAP/HTTP and SOAP/HTTPS. SPML using SOAP over a non-HTTP protocol
60 may behave significantly different and is considered outside of the scope of this binding. SOAP/HTTPS is
61 considered to be within the scope of this binding as an HTTP variant.

62 For the purposes of this document, the SPML/SOAP/HTTP(S) Server will be referred to as the “SPML
63 Service” and the SPML/SOAP/HTTP(S) Client will be referred to as the “SPML Client”.

64 1.1.1 Basic Operations

65 SOAP messages consist of three elements: an envelope, header data, and a message body. SPML
66 request-response protocol elements **MUST** be enclosed within the SOAP message body.

67 The following table designates what are valid responses to client requests:

SPML Client Sends	SPML Service MUST Respond With
batchRequest	batchResponse
addRequest	addResponse
modifyRequest	modifyResponse
deleteRequest	deleteResponse
searchRequest	searchResponse
schemaRequest	schemaResponse
statusRequest	statusResponse
cancelRequest	cancelResponse

68 For more information about these SPML operations, see the SPML Operations section of **[SPML]**.

69 An SPML Service is considered conformant if it supports all or a subset of SPML operations. See the
70 Conformance section of **[SPML]** for more details. If the SPML client sends a request that the SPML
71 Service does not support, the SPML Service **MUST** return the corresponding response element as
72 defined in the previous table with a result code of `urn:oasis:names:tc:SPML:1.0#failure` and an
73 error code of `urn:oasis:names:tc:SPML:1.0#unsupportedOperation`.

74 An SPML client **MAY** specify requests as synchronous or asynchronous. If a client specifies a request as
75 synchronous, the service **MUST** process the request in a synchronous fashion or return an error. If a
76 client specifies a request as asynchronous, the service **MAY** process the request synchronously and
77 return the results in the response.

78

79 **1.1.2 Authentication**

80 Authentication of the SPML Client to the SPML Service SHOULD be accomplished using one of the
81 following methods:

- 82 1. HTTP basic client authentication [RFC2617] with and without SSL 3.0 or TLS 1.0 [RFC2246]
- 83 2. HTTP over SSL 3.0 or TLS 1.0 client authentication with a client-side certificate [RFC2246]
- 84 3. WSS Authentication of the SPML request as specified in WS-Security [WSS]

85 Authentication of the SPML Service to the SPML Client SHOULD be accomplished using one of the
86 following methods:

- 87 1. HTTP over SSL 3.0 or TLS 1.0 server authentication with a server-side certificate [RFC2246]
- 88 2. WSS Authentication of the SPML response as specified in WS-Security [WSS]

89 If an SPML Service uses SSL 3.0 or TLS 1.0, it MUST use a server-side certificate.

90 **1.1.3 Message Integrity**

91 SPML Message integrity SHOULD be accomplished using one of the following methods:

- 92 1. Through the use of HTTP over SSL 3.0 or TLS 1.0
- 93 2. Through the use of XML Encryption as specified in WS-Security [WSS]
- 94 3. Through the use of XML Digital Signatures as specified in WS-Security [WSS]

95 If an SPML Service uses SSL 3.0 or TLS 1.0, it MUST use a server-side certificate.

96 1.1.4 Peter Cordery (solutions center).

97 **1.1.4 Message Confidentiality**

98 SPML message confidentiality SHOULD be accomplished using one of the following methods:

- 99 1. Through the use of HTTP over SSL 3.0 or TLS 1.0
- 100 2. Through the use of XML Encryption as specified in WS-Security [WSS]

101 If an SPML Service uses SSL 3.0 or TLS 1.0, it MUST use a server-side certificate.

102 **1.1.5 Error Reporting**

103 An SPML Service that refuses to perform a message exchange with the SPML Client SHOULD return a
104 "403 Forbidden" response. In this case, the content of the HTTP body is not significant.

105 As described in [SOAP1.1], in the case of a SOAP error while processing a SOAP request, the SOAP
106 HTTP server MUST return a "500 Internal Server Error" response and include a SOAP
107 message in the response with a SOAP fault element. This type of error SHOULD be returned for SOAP-
108 related errors detected before control is passed to the SPML processor, or when the SOAP processor
109 reports an internal error.

110 In the case of a SPML processing error, the SOAP HTTP server MUST respond with "200 OK". The
111 SPML should include an result code of `urn:oasis:names:tc:SPML:1.0#failure` and an
112 appropriate error code.

113 Additional human-readable error text may optionally be returned in the error message sub-element of the
114 response. To specify the locale of the error message, the `xml:lang` attribute on the error message
115 element MAY be set to the error message locale.

116 For more information about the specifics of SPML error handling, see the SPML Operations section of
117 **[SPML]**.

118 **1.1.6 Declaring Supported SPML Operations**

119 An SPML Service is considered conformant if it supports all or a subset of SPML operations. See the
120 Conformance section of **[SPML]** for more details. An SPML Service SHOULD document all its supported
121 SPML operations via WSDL.

122 For example if an SPML Service only supported the SPML Batch Request, the following WSDL could
123 define the service:

```
124 <wsdl:message name="SPMLProvRequestMessage">  
125 <wsdl:part name="body" element="spml:batchRequest" />  
126 </wsdl:message>  
127  
128 <wsdl:message name="SPMLProvResponseMessage">  
129 <wsdl:part name="body" element="spml:batchResponse" />  
130 </wsdl:message>  
131 <wsdl:portType name="SPMLProvRequestPortType">  
132  
133 <wsdl:operation name="SPMLProvRequest">  
134 <wsdl:input name="SPMLProvInput"  
135 message="tns:SPMLProvRequestMessage"/>  
136 <wsdl:output name="SPMLProvOutput"  
137 message="tns:SPMLProvResponseMessage"/>  
138 </wsdl:operation>  
139  
140 </wsdl:portType>  
141 <wsdl:binding name="SPMLProvSoapBinding" type="tns:SPMLProvRequestPortType">  
142 <soap:binding style="document"  
143 transport="http://schemas.xmlsoap.org/soap/http"/>  
144  
145 <wsdl:operation name="SPMLProvRequest">  
146 <soap:operation style="document"  
147 soapAction="urn:oasis:names:tc:SPML:1:0:req/BatchRequest" />  
148 <wsdl:input name="SPMLProvInput" >  
149 <soap:body use="literal" />  
150 </wsdl:input>  
151 <wsdl:output name="SPMLProvOutput" >  
152 <soap:body use="literal" />  
153 </wsdl:output>  
154 </wsdl:operation>  
155  
156 </wsdl:binding>
```

157 A complete WSDL definition that defines an SPML Service that supports all SPML operations is shown in
158 Appendix A.

159 **1.2 File Binding**

160 This binding refers to using SPML elements in a file, typically for the purposes of bulk processing
161 provisioning data and provisioning schema documentation.

162 **1.2.1 Bulk Processing Files**

163 When using the SPML File binding for bulk processing of provisioning operations, the input file MUST
164 consist of a single SPML request element. An SPML Application MAY output a file containing an SPML
165 response element that corresponds to the request element. If it exists, the resulting output file MUST
166 consist of a single SPML response element.

167 The following table designates what are valid requests that may be used in an input file and the
168 corresponding responses that may be used in an output file:

Input File MAY Contain	Output File MUST Contain (if it exists)
batchRequest	batchResponse
addRequest	addResponse
modifyRequest	modifyResponse
deleteRequest	deleteRequest
searchRequest	searchResponse
schemaRequest	schemaResponse
statusRequest	statusResponse
cancelRequest	cancelResponse

169 For more information about these SPML operations, see the SPML Operations section of **[SPML]**.

170 An SPML Application is considered conformant if it supports all or a subset of SPML operations listed in
171 the previous table. See the Conformance section of **[SPML]** for more details. If the input file contains a
172 request that the SPML Application does not support, the output file MUST contain the corresponding
173 response element as defined in the previous table with a result code of
174 urn:oasis:names:tc:SPML:1.0#failure and an error code of
175 urn:oasis:names:tc:SPML:1.0#unsupportedOperation.

176 The batchRequest element MAY specify processing as either sequential or parallel by setting the
177 processing attribute on the batchRequest to urn:oasis:names:tc:SPML:1.0#sequential or
178 urn:oasis:names:tc:SPML:1.0#parallel. The execution attribute on the batchRequest
179 element SHOULD not be defined, as this attribute is not relevant to the SPML File Binding. If the
180 execution attribute is defined is SHOULD be ignored, rather than generating an error.

181 The batchRequest element MAY specify a request identifier. If the request identifier is specified, the
182 corresponding batchResponse element MUST specify the same identifier.

183 1.2.2 Error Reporting

184 In the case of a SPML processing error, the SPML batchResponse should include an result code of
185 urn:oasis:names:tc:SPML:1.0#failure.

186 Additional human-readable error text may optionally be returned in the error message sub-element of the
187 response. To specify the locale of the error message, the xml:lang attribute on the error message
188 element MAY be set to the error message locale.

189 For more information about the specifics of SPML error handling, see the SPML Operations section of
190 **[SPML]**.

191 1.2.3 Example Bulk Processing Files

192 The following is an example of an SPML Bulk Processing File using the SPML File Binding:

```

193 <?xml version="1.0" encoding="utf-8"?>
194 <batchRequest xmlns="urn:oasis:names:tc:SPML:1:0:req"
195 xmlns:spml="urn:oasis:names:tc:SPML:1:0:core">
196   <spml:addRequest>
197     <spml:identifier type="urn:oasis:names:tc:SPML:1:0:core#GUID">
198       <spml:id>584D268K</spml:id>
199     </spml:identifier>
200     <spml:attributes>
201       <attr name="objectclass" xmlns="urn:oasis:names:tc:DSML:2:0:core">
```

```

202     <value>BasicAccount</value>
203   </attr>
204   <attr name="id" xmlns="urn:oasis:names:tc:DSML:2:0:core">
205     <value>584D268K</value>
206   </attr>
207   <attr name="firstname" xmlns="urn:oasis:names:tc:DSML:2:0:core">
208     <value>Jane</value>
209   </attr>
210   <attr name="lastname" xmlns="urn:oasis:names:tc:DSML:2:0:core">
211     <value>Doe</value>
212   </attr>
213 </spml:attributes>
214 </spml:addRequest>
215 ...

```

1.2.4 Provisioning Schema Files

216 The SPML File Binding can be used to document Provisioning Schema to be used in an SPML based
 217 system. When using the SPML File binding to document Provisioning Schema, the schema file MUST
 218 consist of a single SPML schema element.
 219

1.2.5 Example Provisioning Schema File

220 The following is an example of a Provisioning Schema File using the SPML File Binding:
 221

```

222 <?xml version="1.0" encoding="utf-8"?>
223 <schema xmlns="urn:oasis:names:tc:SPML:1:0:schema">
224   <providerIdentifier providerIDType="urn:oasis:names:tc:SPML:1:0:core#URN">
225     <providerID xmlns="urn:oasis:names:tc:SPML:1:0:core">urn:com:acme</providerID>
226   </providerIdentifier>
227   <schemaIdentifier schemaIDType="urn:oasis:names:tc:SPML:1:0:core#GenericString">
228     <schemaID>BasicAccountSchema</schemaID>
229   </schemaIdentifier>
230   <objectClassDefinition name="BasicAccount" description="Basic account." >
231     <memberAttributes>
232       <AttributeDefinitionReference name = "id" required="true" />
233       <AttributeDefinitionReference name = "firstname"/>
234       <AttributeDefinitionReference name = "lastname"/>
235       <AttributeDefinitionReference name = "initial"/>
236       <AttributeDefinitionReference name = "email"/>
237     </memberAttributes>
238   </objectClassDefinition>
239   <attributeDefinition name="id" type="xsd:string" />
240   <attributeDefinition name="firstname" type="xsd:string" />
241   <attributeDefinition name="lastname" type="xsd:string" />
242   <attributeDefinition name="initial" type="xsd:string" />
243   <attributeDefinition name="email" type="xsd:string" />
244   ...

```

245

Appendix A. Sample WSDL Definition

247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308

```
<wsdl:definitions
name="SPML Service"
targetNamespace="urn:oasis:names:tc:SPML:1:0:wsdl"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="urn:oasis:names:tc:SPML:1:0:wsdl"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:spml="urn:oasis:names:tc:SPML:1:0:req"
xmlns="http://schemas.xmlsoap.org/wsdl/"

  <wsdl:import namespace="urn:oasis:names:tc:SPML:1:0:req"
    schemaLocation="draft pstc schema request 13.xsd"/>

  <wsdl:message name="SPMLBatchRequestMessage">
    <wsdl:part name="body" element="spml:batchRequest" />
  </wsdl:message>

  <wsdl:message name="SPMLBatchResponseMessage">
    <wsdl:part name="body" element="spml:batchResponse" />
  </wsdl:message>

  <wsdl:message name="SPMLAddRequestMessage">
    <wsdl:part name="body" element="spml:addRequest" />
  </wsdl:message>

  <wsdl:message name="SPMLAddResponseMessage">
    <wsdl:part name="body" element="spml:addResponse" />
  </wsdl:message>

  <wsdl:message name="SPMLModifyRequestMessage">
    <wsdl:part name="body" element="spml:modifyRequest" />
  </wsdl:message>

  <wsdl:message name="SPMLModifyResponseMessage">
    <wsdl:part name="body" element="spml:modifyResponse" />
  </wsdl:message>

  <wsdl:message name="SPMLDeleteRequestMessage">
    <wsdl:part name="body" element="spml:deleteRequest" />
  </wsdl:message>

  <wsdl:message name="SPMLDeleteResponseMessage">
    <wsdl:part name="body" element="spml:deleteResponse" />
  </wsdl:message>

  <wsdl:message name="SPMLSearchRequestMessage">
    <wsdl:part name="body" element="spml:searchRequest" />
  </wsdl:message>

  <wsdl:message name="SPMLSearchResponseMessage">
    <wsdl:part name="body" element="spml:searchResponse" />
  </wsdl:message>

  <wsdl:message name="SPMLExtendedRequestMessage">
    <wsdl:part name="body" element="spml:extendedRequest" />
  </wsdl:message>

  <wsdl:message name="SPMLExtendedResponseMessage">
    <wsdl:part name="body" element="spml:extendedResponse" />
  </wsdl:message>
```



```

309
310 <wsdl:message name="SPMLSchemaRequestMessage">
311     <wsdl:part name="body" element="spml:schemaRequest" />
312 </wsdl:message>
313
314 <wsdl:message name="SPMLSchemaResponseMessage">
315     <wsdl:part name="body" element="spml:schemaResponse" />
316 </wsdl:message>
317
318 <wsdl:portType name="SPMLRequestPortType">
319
320     <wsdl:operation name="SPMLBatchRequest">
321         <wsdl:input name="SPMLBatchRequestInput"
322 message="tns:SPMLBatchRequestMessage"/>
323         <wsdl:output name="SPMLBatchRequestOutput"
324 message="tns:SPMLBatchResponseMessage"/>
325     </wsdl:operation>
326
327     <wsdl:operation name="SPMLAddRequest">
328         <wsdl:input name="SPMLAddRequestInput"
329 message="tns:SPMLAddRequestMessage"/>
330         <wsdl:output name="SPMLAddRequestOutput"
331 message="tns:SPMLAddResponseMessage"/>
332     </wsdl:operation>
333
334     <wsdl:operation name="SPMLModifyRequest">
335         <wsdl:input name="SPMLModifyRequestInput"
336 message="tns:SPMLModifyRequestMessage"/>
337         <wsdl:output name="SPMLModifyRequestOutput"
338 message="tns:SPMLModifyResponseMessage"/>
339     </wsdl:operation>
340
341     <wsdl:operation name="SPMLDeleteRequest">
342         <wsdl:input name="SPMLDeleteRequestInput"
343 message="tns:SPMLDeleteRequestMessage"/>
344         <wsdl:output name="SPMLDeleteRequestOutput"
345 message="tns:SPMLDeleteResponseMessage"/>
346     </wsdl:operation>
347
348     <wsdl:operation name="SPMLSearchRequest">
349         <wsdl:input name="SPMLSearchRequestInput"
350 message="tns:SPMLSearchRequestMessage"/>
351         <wsdl:output name="SPMLSearchRequestOutput"
352 message="tns:SPMLSearchResponseMessage"/>
353     </wsdl:operation>
354
355     <wsdl:operation name="SPMLExtendedRequest">
356         <wsdl:input name="SPMLExtendedRequestInput"
357 message="tns:SPMLExtendedRequestMessage"/>
358         <wsdl:output name="SPMLExtendedRequestOutput"
359 message="tns:SPMLExtendedResponseMessage"/>
360     </wsdl:operation>
361
362     <wsdl:operation name="SPMLSchemaRequest">
363         <wsdl:input name="SPMLSchemaRequestInput"
364 message="tns:SPMLSchemaRequestMessage"/>
365         <wsdl:output name="SPMLSchemaRequestOutput"
366 message="tns:SPMLSchemaResponseMessage"/>
367     </wsdl:operation>
368
369 </wsdl:portType>
370
371 <wsdl:binding name="SPMLSoapBinding" type="tns:SPMLRequestPortType">
372     <soap:binding style="document"
373 transport="http://schemas.xmlsoap.org/soap/http"/>
374
375     <wsdl:operation name="SPMLBatchRequest">
376         <soap:operation style="document"

```

```

377         soapAction="urn:oasis:names:tc:SPML:1:0:req/batchRequest" />
378     <wsdl:input name="SPMLBatchRequestInput" >
379         <soap:body use="literal" />
380     </wsdl:input>
381     <wsdl:output name="SPMLBatchRequestOutput" >
382         <soap:body use="literal" />
383     </wsdl:output>
384 </wsdl:operation>
385
386     <wsdl:operation name="SPMLAddRequest">
387     <soap:operation style="document"
388         soapAction="urn:oasis:names:tc:SPML:1:0:req/addRequest" />
389     <wsdl:input name="SPMLAddRequestInput" >
390         <soap:body use="literal" />
391     </wsdl:input>
392     <wsdl:output name="SPMLAddRequestOutput" >
393         <soap:body use="literal" />
394     </wsdl:output>
395 </wsdl:operation>
396
397     <wsdl:operation name="SPMLModifyRequest">
398     <soap:operation style="document"
399         soapAction="urn:oasis:names:tc:SPML:1:0:req/modifyRequest" />
400     <wsdl:input name="SPMLModifyRequestInput" >
401         <soap:body use="literal" />
402     </wsdl:input>
403     <wsdl:output name="SPMLModifyRequestOutput" >
404         <soap:body use="literal" />
405     </wsdl:output>
406 </wsdl:operation>
407
408     <wsdl:operation name="SPMLDeleteRequest">
409     <soap:operation style="document"
410         soapAction="urn:oasis:names:tc:SPML:1:0:req/deleteRequest" />
411     <wsdl:input name="SPMLDeleteRequestInput" >
412         <soap:body use="literal" />
413     </wsdl:input>
414     <wsdl:output name="SPMLDeleteRequestOutput" >
415         <soap:body use="literal" />
416     </wsdl:output>
417 </wsdl:operation>
418
419     <wsdl:operation name="SPMLSearchRequest">
420     <soap:operation style="document"
421         soapAction="urn:oasis:names:tc:SPML:1:0:req/searchRequest" />
422     <wsdl:input name="SPMLSearchRequestInput" >
423         <soap:body use="literal" />
424     </wsdl:input>
425     <wsdl:output name="SPMLSearchRequestOutput" >
426         <soap:body use="literal" />
427     </wsdl:output>
428 </wsdl:operation>
429
430     <wsdl:operation name="SPMLExtendedRequest">
431     <soap:operation style="document"
432         soapAction="urn:oasis:names:tc:SPML:1:0:req/extendedRequest"
433 />
434     <wsdl:input name="SPMLExtendedRequestInput" >
435         <soap:body use="literal" />
436     </wsdl:input>
437     <wsdl:output name="SPMLExtendedRequestOutput" >
438         <soap:body use="literal" />
439     </wsdl:output>
440 </wsdl:operation>
441
442     <wsdl:operation name="SPMLSchemaRequest">
443     <soap:operation style="document"
444         soapAction="urn:oasis:names:tc:SPML:1:0:req/schemaRequest" />

```

```
445         <wsdl:input name="SPMLSchemaRequestInput" >
446             <soap:body use="literal" />
447         </wsdl:input>
448         <wsdl:output name="SPMLSchemaRequestOutput" >
449             <soap:body use="literal" />
450         </wsdl:output>
451     </wsdl:operation>
452
453 </wsdl:binding>
454
455 <wsdl:service name="SPMLService">
456 </wsdl:service>
457
458 </wsdl:definitions>
459
```

460 Appendix B. References

- 461 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
462 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 463
- 464 **[RFC2617]** *HTTP Authentication: Basic and Digest Access Authentication*,
465 <http://www.ietf.org/rfc/rfc2617.txt>, IETF RFC 2617.
- 466
- 467 **[RFC2616]** Hypertext Transfer Protocol -- HTTP/1.1, <http://www.ietf.org/rfc/rfc2616.txt>
- 468
- 469 **[RFC1738]** Uniform Resource Locators (URL), <http://www.ietf.org/rfc/rfc1738.txt>
- 470
- 471 **[RFC1945]** Hypertext Transfer Protocol -- HTTP/1.0, <http://www.ietf.org/rfc/rfc1945.txt>
- 472
- 473 **[RFC2246]** The TLS Protocol Version 1.0, <http://www.ietf.org/rfcs/rfc2246.html>
- 474
- 475 **[SOAP1.1]** D. Box et al., *Simple Object Access Protocol (SOAP) 1.1*, <http://www.w3.org/TR/SOAP>, World
476 Wide Web Consortium Note, May 2000.
- 477
- 478 **[XMLSig]** D. Eastlake et al., *XML-Signature Syntax and Processing*, <http://www.w3.org/TR/xmlsig-core/>,
479 World Wide Web Consortium.
- 480
- 481 **[SSLv3]** The SSL Protocol Version 3.0, <http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt>
- 482
- 483 **[WSS]** OASIS Web Services Security TC (WSS TC), [http://www.oasis-](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)
484 [open.org/committees/tc_home.php?wg_abbrev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)
- 485
- 486 **[SPML]** SPML Specification, <http://www.oasis-open.org/committees/documents.php>
- 487

488

Appendix C. Revision history

Rev	Date	By whom	What
CS-1.0	3 June 2003	Editor	1.0 Committee Specification

489

490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519

Appendix D. 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 implementers or users of this specification, can be obtained from the OASIS Executive Director.

OASIS has been notified of intellectual property rights claimed in regard to some or all of the contents of this specification. For more information consult the online list of claimed rights.

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 (C) OASIS Open 2002. 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 may 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.