



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 Standard, Approved October 2003

**Document identifier:**

os-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 and accepted as  
52 OASIS Open Standard October 2003. Two bindings, the SOAP/HTTP binding and the file binding, are  
53 defined.

### 54 1.1 SOAP/HTTP Binding

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

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

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

#### 65 1.1.1 Basic Operations

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

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

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

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

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

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

77 client specifies a request as asynchronous, the service MAY process the request synchronously and  
78 return the results in the response.  
79

## 80 **1.1.2 Authentication**

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

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

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

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

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

## 91 **1.1.3 Message Integrity**

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

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

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

97 1.1.4 Peter Cordery (solutions center).

## 98 **1.1.4 Message Confidentiality**

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

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

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

## 103 **1.1.5 Error Reporting**

104 An SPML Service that refuses to perform a message exchange with the SPML Client SHOULD return a  
105 "403 Forbidden" response. In this case, the content of the HTTP body is not significant.

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

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

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

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

## 119 **1.1.6 Declaring Supported SPML Operations**

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

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

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

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

## 160 **1.2 File Binding**

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

### 163 **1.2.1 Bulk Processing Files**

164 When using the SPML File binding for bulk processing of provisioning operations, the input file MUST  
165 consist of a single SPML request element. An SPML Application MAY output a file containing an SPML

166 response element that corresponds to the request element. If it exists, the resulting output file MUST  
167 consist of a single SPML response element.

168 The following table designates what are valid requests that may be used in an input file and the  
169 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

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

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

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

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

## 184 1.2.2 Error Reporting

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

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

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

## 192 1.2.3 Example Bulk Processing Files

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

```
194 <?xml version="1.0" encoding="utf-8"?>  
195 <batchRequest xmlns="urn:oasis:names:tc:SPML:1:0:req"  
196 xmlns:spml="urn:oasis:names:tc:SPML:1:0:core">
```

```

197 <spml:addRequest>
198   <spml:identifier type="urn:oasis:names:tc:SPML:1:0:core#GUID">
199     <spml:id>584D268K</spml:id>
200   </spml:identifier>
201   <spml:attributes>
202     <attr name="objectclass" xmlns="urn:oasis:names:tc:DSML:2:0:core">
203       <value>BasicAccount</value>
204     </attr>
205     <attr name="id" xmlns="urn:oasis:names:tc:DSML:2:0:core">
206       <value>584D268K</value>
207     </attr>
208     <attr name="firstname" xmlns="urn:oasis:names:tc:DSML:2:0:core">
209       <value>Jane</value>
210     </attr>
211     <attr name="lastname" xmlns="urn:oasis:names:tc:DSML:2:0:core">
212       <value>Doe</value>
213     </attr>
214   </spml:attributes>
215 </spml:addRequest>
216 ...

```

## 217 1.2.4 Provisioning Schema Files

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

## 221 1.2.5 Example Provisioning Schema File

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

```

223 <?xml version="1.0" encoding="utf-8"?>
224 <schema xmlns="urn:oasis:names:tc:SPML:1:0:schema">
225   <providerIdentifier providerIDType="urn:oasis:names:tc:SPML:1:0:core#URN">
226     <providerID xmlns="urn:oasis:names:tc:SPML:1:0:core">urn:com:acme</providerID>
227   </providerIdentifier>
228   <schemaIdentifier schemaIDType="urn:oasis:names:tc:SPML:1:0:core#GenericString">
229     <schemaID>BasicAccountSchema</schemaID>
230   </schemaIdentifier>
231   <objectClassDefinition name="BasicAccount" description="Basic account." >
232     <memberAttributes>
233       <AttributeDefinitionReference name = "id" required="true" />
234       <AttributeDefinitionReference name = "firstname"/>
235       <AttributeDefinitionReference name = "lastname"/>
236       <AttributeDefinitionReference name = "initial"/>
237       <AttributeDefinitionReference name = "email"/>
238     </memberAttributes>
239   </objectClassDefinition>
240   <attributeDefinition name="id" type="xsd:string" />
241   <attributeDefinition name="firstname" type="xsd:string" />
242   <attributeDefinition name="lastname" type="xsd:string" />
243   <attributeDefinition name="initial" type="xsd:string" />
244   <attributeDefinition name="email" type="xsd:string" />
245   ...

```

246

## Appendix A. Sample WSDL Definition

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  
309

```
<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>
```



```

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

```

```

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

```

```
446         <wsdl:input name="SPMLSchemaRequestInput" >
447             <soap:body use="literal" />
448         </wsdl:input>
449         <wsdl:output name="SPMLSchemaRequestOutput" >
450             <soap:body use="literal" />
451         </wsdl:output>
452     </wsdl:operation>
453
454 </wsdl:binding>
455
456 <wsdl:service name="SPMLService">
457 </wsdl:service>
458
459 </wsdl:definitions>
460
```

---

## 461 Appendix B. References

- 462 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
463 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 464
- 465 **[RFC2617]** *HTTP Authentication: Basic and Digest Access Authentication*,  
466 <http://www.ietf.org/rfc/rfc2617.txt>, IETF RFC 2617.
- 467
- 468 **[RFC2616]** Hypertext Transfer Protocol -- HTTP/1.1, <http://www.ietf.org/rfc/rfc2616.txt>
- 469
- 470 **[RFC1738]** Uniform Resource Locators (URL), <http://www.ietf.org/rfc/rfc1738.txt>
- 471
- 472 **[RFC1945]** Hypertext Transfer Protocol -- HTTP/1.0, <http://www.ietf.org/rfc/rfc1945.txt>
- 473
- 474 **[RFC2246]** The TLS Protocol Version 1.0, <http://www.ietf.org/rfc/rfc2246.html>
- 475
- 476 **[SOAP1.1]** D. Box et al., *Simple Object Access Protocol (SOAP) 1.1*, <http://www.w3.org/TR/SOAP>, World  
477 Wide Web Consortium Note, May 2000.
- 478
- 479 **[XMLSig]** D. Eastlake et al., *XML-Signature Syntax and Processing*, <http://www.w3.org/TR/xmlsig-core/>,  
480 World Wide Web Consortium.
- 481
- 482 **[SSLv3]** The SSL Protocol Version 3.0, <http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt>
- 483
- 484 **[WSS]** OASIS Web Services Security TC (WSS TC), [http://www.oasis-](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)  
485 [open.org/committees/tc\\_home.php?wg\\_abbrev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)
- 486
- 487 **[SPML]** SPML Specification, <http://www.oasis-open.org/committees/documents.php>
- 488

489

---

## Appendix C. Revision history

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

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  
520

---

## 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.