



OASIS Service Provisioning Markup Language (SPML) v2 - XSD Profile

OASIS Standard 2006 April 1

Document identifier: pstc-spml2-xsd-profile-os.pdf

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

Send comments to: pstc-comment@lists.oasis-open.org

Editor:

Jeff Bohren, BMC (Jeff_Bohren@bmc.com)

Contributors:

Robert Boucher, CA
Doron Cohen, BMC
Gary Cole, Sun Microsystems
Cal Collingham, CA
Rami Elron, BMC
Marco Fanti, Thor Technologies
Ian Glazer, IBM
James Hu, HP
Ron Jacobsen, CA
Jeff Larson, Sun Microsystems
Hal Lockhart, BEA
Prateek Mishra, Oracle Corporation
Martin Raepple, SAP
Darran Rolls, Sun Microsystems
Kent Spaulding, Sun Microsystems
Gavenraj Sodhi, CA
Cory Williams, IBM
Gerry Woods, SOA Software

Abstract:

This specification defines usage of XML and XSD as a data model (profile) for SPML v2.

Status:

35 This is an OASIS Standard document produced by the Provisioning Services Technical
36 Committee. It was approved by the OASIS membership on 1 April 2006.

37 If you are on the provision list for committee members, send comments there. If you are not
38 on that list, subscribe to the provision-comment@lists.oasis-open.org list and send
39 comments there. To subscribe, send an email message to [provision-comment-](mailto:provision-comment-request@lists.oasis-open.org)
40 [request@lists.oasis-open.org](mailto:provision-comment-request@lists.oasis-open.org) with the word "subscribe" as the body of the message.

41	Table of contents	
42	1. Introduction (non-normative)	4
43	1.1. Concepts	4
44	1.2. Terminology	4
45	2. Notation	4
46	3. Overview (non-normative)	5
47	3.1. XML PSOs	5
48	3.1.1. PSO Identifier	5
49	3.1.2. PSO Data	5
50	3.2. Schema	6
51	3.3. Core Operations	6
52	3.3.1. Add Request	6
53	3.3.2. Add Response	6
54	3.3.3. Modify Request	7
55	3.3.4. Delete Request	7
56	3.3.5. Lookup Request	8
57	3.3.6. Lookup Response	8
58	3.4. Search Operations	8
59	3.4.1. Search Request	8
60	3.4.2. Search Response	9
61	4. Specification (Normative)	9
62	4.1. XPath Support	9
63	4.2. Core Capability	10
64	4.2.1. Element <spml:data>	10
65	4.2.2. Element <spml:modification>	10
66	4.2.3. Element <spml:schema>	11
67	4.2.4. Element <supportedSchemaEntity>	11
68	4.3. Search Capability	11
69	4.3.1. Element <spmlsearch:query>	11
70	4.3.2. Element <spmlsearch:select>	12
71	Appendix A. References	13
72	Appendix B. Acknowledgments	15
73	Appendix C. Notices	16
74		

75 1. Introduction (non-normative)

76 1.1. Concepts

77 SPML Version 2 (SPMLv2) defines a core protocol **[SPMLv2]** over which different data models can
78 be used to define the actual provisioning data. The combination of a data model with the SPML
79 core specification is referred to as a binding. The use of SPML requires that a specific binding is
80 used, although the choice of which binding is used to negotiated out-of-band by the participating
81 parties.

82 This document describes the use of the XML and XSD as a data model for SPML based
83 provisioning. This binding is optional.

84 1.2. Terminology

85 Within this document:

- 86 - The term "requestor" always refers to a [Requesting Authority \(RA\)](#).
- 87 - The term "provider" always refers to a [Provisioning Service Provider \(PSP\)](#).
- 88 - The term "target" always refers to a [Provisioning Service Target \(PST\)](#).
- 89 - The term "object" (unless otherwise qualified) refers to a [Provisioning Service Object \(PSO\)](#).
- 90 - The term "client" (unless otherwise qualified) refers to a [Requesting Authority \(RA\)](#).
- 91 - The term "server" (unless otherwise qualified) refers to a [Provisioning Service Provider \(PSP\)](#).

92 2. Notation

93 This specification contains schema conforming to W3C XML Schema and normative text to
94 describe the syntax and semantics of XML-encoded policy statements.

95 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
96 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be
97 interpreted as described in IETF RFC 2119 **[RFC2119]**

98 *"they MUST only be used where it is actually required for interoperation or to limit*
99 *behavior which has potential for causing harm (e.g., limiting retransmissions)"*

100 These keywords are thus capitalized when used to unambiguously specify requirements over
101 protocol and application features and behavior that affect the interoperability and security of
102 implementations. When these words are not capitalized, they are meant in their natural-language
103 sense.

104 This specification uses the following typographical conventions in text:

Format	Description	Indicates
attributeName	monospace font with <i>first letter lower-cased</i>	The name of an XML <i>attribute</i> .
SPMLElementName	monospace font with <i>first letter capitalized</i>	The name of an XML <i>element</i> that is defined as part of SPMLv2.
ns:ForeignElementName	monospace font with <i>namespace prefix</i>	The name of an XML element that is <i>defined by another specification</i> .

<SPMLElement>	monospace font <i>surrounded by <></i>	<i>An instance of an XML element that is defined as part of SPMLv2.</i>
<ns:ForeignElement>	monospace font <i>with namespace prefix surrounded by <></i>	<i>An instance of an XML element that is defined by another specification.</i>

105 Terms in ***italic bold-face*** are intended to have the meaning defined in the Glossary.

106 Listings of SPML schemas appear like this.

107

108 Example code listings appear like this.

109 Conventional XML namespace prefixes are used throughout the listings in this specification to
110 stand for their respective namespaces as follows, whether or not a namespace declaration is
111 present in the example:

- 112 - The prefix `saml:` stands for the SAML assertion namespace [**SAML**].
- 113 - The prefix `ds:` stands for the W3C XML Signature namespace [**DS**].
- 114 - The prefix `xsd:` stands for the W3C XML Schema namespace [**XS**].

115 3. Overview (non-normative)

116 3.1. XML PSOs

117 A PSO is represented in this binding by an XML structure. Thus structure should be defined by the
118 XSD that is returned as the schema for the containing target.

119 3.1.1. PSO Identifier

120 The PSO Identifier may be any opaque identifier for the PSO, such as a GUID, URN, or XPath
121 expression. If an XPath expression is used, it must resolve to a single PSO.

122 For instance if an opaque GUID is used for the PSO ID:

```
123 <spml:pso xmlns:spml="urn:oasis:names:tc:SPML:2:0">
124   <spml:psoID ID="2244" targetID="target2"/>
125 </spml:pso>
```

126 If for instance an XPath is used for the PSO ID:

```
127 <spml:pso xmlns:spml="urn:oasis:names:tc:SPML:2:0">
128   <spml:psoID ID="/Person/email='jdoe@acme.com'" targetID="target2"/>
129 </spml:pso>
```

130 3.1.2. PSO Data

131 The PSO Data element contains a root XML element that conforms to the XSD schema defined by
132 the target.

```
133 <spml:pso xmlns:spml="urn:oasis:names:tc:SPML:2:0">
134   <spml:data>
135     <user>
136       <cn>John Doe</cn>
137       <uid>jdoe</uid>
138       <email>jdoe@acme.com</email>
```

```
139     <phone>
140         <home>555-2323</home>
141         <work>555-6767x321</work>
142     </phone>
143 </user>
144 </spml:data>
145 </spml:ps0>
```

146 3.2. Schema

147 The schema defines the allowed attributes and elements. For the XSD Profile, the PSO schema is
148 defined using XSD. The XSD can be defined by inclusion in the spml:schema element, or by
149 reference to an external or well known XSD schema URI.

150 For instance if the XSD is defined by inclusion:

```
151 <spml:schema xmlns:spml="urn:oasis:names:tc:SPML:2:0">
152     <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
153 </spml:schema>
```

154 If the XSD is defined by reference:

```
155 <spml:schema ref="http://www.acme.com/schemas/user.xsd"
156     xmlns:spml="urn:oasis:names:tc:SPML:2:0" />
```

157 3.3. Core Operations

158 3.3.1. Add Request

159 The Add Request creates PSOs. The Add Request must contain a <data> element that contains an
160 XML element that defines the new PSO. The Add Request may also pass a PSO Identifier
161 (<psold> element). If a PSO identifier is not defined in the Add Request, the new PSO Identifier
162 must be returned in the Add Response.

```
163 <spml:addRequest targetID="target2" xmlns:spml="urn:oasis:names:tc:SPML:2:0">
164     <spml:data>
165         <user>
166             <cn>John Doe</cn>
167             <uid>jdoe</uid>
168             <email>jdoe@acme.com</email>
169             <phone>
170                 <home>555-2323</home>
171                 <work>555-6767x321</work>
172             </phone>
173         </user>
174     </spml:data>
175 </spml:addRequest>
```

176 3.3.2. Add Response

177 The Add Response would contain the status. If the request is successful, the response could
178 include the new PSO ID and data. For instance:

```
179 <spml:addResponse status="spml:success" xmlns:spml="urn:oasis:names:tc:SPML:2:0"
180 >
181     <spml:ps0ID ID="2244" targetID="target2"/>
182     <spml:data>
183         <user>
```

```

184     <cn>John Doe</cn>
185     <uid>jdoe</uid>
186     <email>jdoe@acme.com</email>
187     <phone>
188         <home>555-2323</home>
189         <work>555-6767x321</work>
190     </phone>
191 </user>
192 </spml:data>
193 </spml:addResponse>

```

194 3.3.3. Modify Request

195 The Modify Request modifies PSOs. The Modify Request always contains the PSO Identifier. The
 196 modification type can be either add, replace, or delete. If the modification is not being made to the
 197 root XML element of the PSO data, the request would specify a selector XPath that uniquely
 198 identifies the sub-element being modified.

199 For instance to add a sub-element to the root element of the PSO data:

```

200 <spml:modifyRequest xmlns:spml="urn:oasis:names:tc:SPML:2:0" >
201   <spml:psoid ID="2244" targetID="target2"/>
202   <spml:modification modificationMode = "spml:add">
203     <spml:component path="./phone" namespaceURI="http://www.w3.org/TR/xpath20"/>
204     <spml:data>
205       <mobile>555-1212</mobile>
206     </spml:data>
207   </spml:modification>
208 </spml:modifyRequest>

```

209 To replace a sub-element:

```

210 <spml:modifyRequest xmlns:spml="urn:oasis:names:tc:SPML:2:0" >
211   <spml:psoid ID="2244" targetID="target2"/>
212   <spml:modification modificationMode="spml:replace" >
213     <spml:component path="./phone" namespaceURI="http://www.w3.org/TR/xpath20"/>
214     <spml:data>
215       <phone>
216         <mobile>555-1212</mobile>
217         <home>555-2323</home>
218         <work>555-6767x321</work>
219       </phone>
220     </spml:data>
221   </spml:modification>
222 </spml:modifyRequest>

```

223 To delete a sub-element:

```

224 <spml:modifyRequest xmlns:spml="urn:oasis:names:tc:SPML:2:0" >
225   <spml:psoid ID="2244" targetID="target2"/>
226   <spml:modification modificationMode = "spml:delete" >
227     <spml:component path="./phone" namespaceURI="http://www.w3.org/TR/xpath20"/>
228   </spml:modification>
229 </spml:modifyRequest>

```

230 3.3.4. Delete Request

231 The Delete Request deletes PSOs. The Delete Request always contains the PSO Identifier.

```

232 <spml:deleteRequest xmlns:spml="urn:oasis:names:tc:SPML:2:0" >
233   <spml:psoid ID="2244" targetID="target2"/>

```

234 </spml:deleteRequest >

235 3.3.5. Lookup Request

236 The Lookup Request returns the data for an identified PSO. The Lookup Request always contains
237 the PSO Identifier.

```
238 <spml:lookupRequest returnData = "spml:everything"  
239 xmlns:spml="urn:oasis:names:tc:SPML:2:0" >  
240   <spml:psoid ID="2244" targetID="target2"/>  
241 </spml:lookupRequest>
```

242 3.3.6. Lookup Response

243 The Lookup Response (if successful) will return the data for the identified PSO.

```
244 <spml:lookupResponse status="spml:success"  
245 xmlns:spml="urn:oasis:names:tc:SPML:2:0" >  
246   <spml:psoid ID="2244" targetID="target2"/>  
247   <spml:data>  
248     <user>  
249       <cn>John Doe</cn>  
250       <uid>jdoe</uid>  
251       <email>jdoe@acme.com</email>  
252       <phone>  
253         <mobile>555-1212</mobile>  
254         <home>555-2323</home>  
255         <work>555-6767x321</work>  
256       </phone>  
257     </user>  
258   </spml:data>  
259 </spml:lookupResponse>
```

260 3.4. Search Operations

261 3.4.1. Search Request

262 The search request can specify a search base and an XPath selection statement.

```
263 <spmlsearch:searchRequest xmlns:spml="urn:oasis:names:tc:SPML:2:0"  
264 xmlns:spmlsearch="urn:oasis:names:tc:SPML:2:0:search">  
265   <spmlsearch:query scope = "spmlsearch:oneLevel" targetID="target2">  
266     <spml:select>/user</spml:select>  
267   </spmlsearch:query>  
268 </spmlsearch:searchRequest>
```

269 The select clause for the search request treats each target as a document root that (directly or
270 indirectly) contains all other objects as nodes. So, for example,

- 271 • "/Person" would select every Person object that the target directly contains.
- 272 • "//Person" would select every Person object on a target,
273 no matter which container was the Person object's parent.
- 274 • "/Group" would select every Group object that the target directly contains.
- 275 • "//Group" would select every Group object on a target,
276 no matter which container was the Group object's parent.

277 3.4.2. Search Response

278 The search response, if successful, would contain all of the PSOs that satisfied the search criteria.
279 For instance:

```
280 <spml:searchResponse status="spml:success"  
281 xmlns:spml="urn:oasis:names:tc:SPML:2:0">  
282   <spml:psso>  
283     <spml:pssoID ID="2244" targetID="target2"/>  
284     <spml:data>  
285       <user>  
286         <cn>John Doe</cn>  
287         <uid>jdoe</uid>  
288         <email>jdoe@acme.com</email>  
289       </user>  
290     </spml:data>  
291   </spml:psso>  
292   <spml:psso>  
293     <spml:pssoID ID="2245" targetID="target2"/>  
294     <spml:data>  
295       <user>  
296         <cn>Jane Smith</cn>  
297         <uid>jsmith</uid>  
298         <email>jsmith@acme.com</email>  
299       </user>  
300     </spml:data>  
301   </spml:psso>  
302 </spml:searchResponse>
```

303 4. Specification (Normative)

304 4.1. XPath Support

305 A provider **MUST** support the abbreviated syntax for XPath expressions. Put differently, a provider
306 **MUST** support any XPath location path that does not include an explicit axis specifier.

307 A provider **MAY** support explicitly specified axes. A provider **MAY** support arbitrary XPath
308 expressions. However, a requestor that deals with arbitrary providers should assume only that each
309 provider supports location paths in the abbreviated syntax format.

310 **Abbreviated Syntax.** An XPath expression that uses only the abbreviated syntax contains no
311 explicit axis specifier. Each step assumes the "child" axis by default. Any axis other than the "child"
312 axis is specified by one of the following abbreviations:

- 313 • "@" is short for "attribute:"
- 314 • "/" is short for "/descendant-or-self::node()/"
- 315 • "." is short for self::node()
- 316 • ".." is short for parent::node()

317 **Each target is a document root.** A provider **MUST** treat each target as a document root that
318 (directly or indirectly) contains all other objects as nodes.

319 4.2. Core Capability

320 4.2.1. Element <spml:data>

321 The <spml:data> element MAY contain any number of XML elements. The elements MUST
322 conform to the XSD specified in the *spml:schema* for that target.

323 4.2.2. Element <spml:modification>

324 The <spml:modification> element MAY contain any number of XML elements. The
325 <spml:modification> element MUST define the "modificationMode" attribute to be one of "add",
326 "replace", or "delete".

327 An <spml:modification> element MAY contain at most one <component> element. If the
328 modification is on a sub-element of the PSO data, the *component* element MUST be set to the
329 XPath state that uniquely identifies the sub-element within the PSO data root element. If the
330 modification is on the PSO data root element, the *component* element MAY be omitted.

331 An <spml:modification> element MAY contain at most one <data> element. If the
332 <spml:modification> contains a <component> element, then the <spml:modification> MUST contain
333 a <data> element.

334 **Modification component.** An <spml:component> element MUST have a "namespaceURI"
335 attribute and MUST have a "path" attribute.

336 The value of the "namespaceURI" attribute MUST specify the XML namespace of a query
337 language. The value of the "path" attribute MUST be an expression that is valid in the query
338 language that "namespaceURI" specifies. (For example, if a requestor uses XPath 2.0 as the
339 query language for the "path" attribute, the value of the "namespaceURI" attribute MUST be
340 "http://www.w3.org/TR/xpath20".)

341 The value of the "path" attribute MUST specify an attribute or a sub-element (or an attribute of a
342 sub-element) of the object that the provider is to modify. The specified attribute or element MUST
343 be valid (according to the schema of the target) for the schema entity of which the object to be
344 modified is an instance.

345 An <spml:component> element MAY include <spml:namespacePrefixMap> elements that defines
346 the namespace prefixes that are used in the XPath path. Each "prefix" attribute on the
347 <spml:namespacePrefixMap> element MUST exactly match one the namespace prefixes used in
348 the Xpath.

349 **Modification data.** A requestor must specify as the content of the <data> sub-element of a
350 <modification> any *value* that is to be added to, replaced within, or deleted from the element or
351 attribute that the <component> element specifies.

- 352 • In the XML Schema profile, a requestor that specifies a <component> element within a
353 <modification> element with "modificationMode='add'" or (within a <modification>
354 element with) "modificationMode='modify'" MUST specify a value that is to replace the
355 element or attribute that the <component> element specifies.
- 356 ▪ If the <component> element (XPath expression) specifies an XML element, then the *value*
357 (that is the content of the <data> element) MUST be one or more XML elements that are
358 valid (according to the schema of the target) for the element that the <component>
359 element specifies.

- 360 ▪ If the `<component>` element (XPath expression) specifies an XML attribute, then the *value*
361 MUST be valid (according to the schema of the target) for the attribute that the
362 `<component>` element specifies.
- 363 • In the XML Schema profile, a requestor that specifies a `<component>` element within a
364 `<modification>` element with “`modificationMode='delete'`” MUST NOT specify a
365 value. The (XPath expression that is the value of the) `<component>` element MUST specify the
366 set of elements or (MUST specify) the attribute that the provider should delete.
- 367 ▪ If the `<component>` element (XPath expression) specifies a set of XML elements, then
368 each XML element that the `<component>` element specifies must be optional (i.e.,
369 “`minOccurs='0'`”) according to the schema of the target for the object to be modified.
- 370 ▪ If the `<component>` element (XPath expression) specifies an XML attribute, then the
371 specified attribute MUST be *optional* (according to the schema of the target) for the object
372 to be modified.

373 **4.2.3. Element `<spml:schema>`**

374 If the schema is included as content of an `<spml:schema>` element, the `<spml:schema>` element
375 MUST contain at least one `<xsd:schema>` element. If the schema is not included as content of an
376 `<spml:schema>` element, the “`ref`” attribute on the `<spml:schema>` element MUST be set to the
377 URN of the referenced schema. If the schema is included as content of an `<spml:schema>`
378 element, a requestor should ignore any “`ref`” attribute on the `<spml:schema>` element.

379 If a provider supports only a subset of the top-level elements that are defined in the schema for a
380 target, then the `<spml:schema>` element MUST contain at least one
381 `<spml:supportedSchemaEntity>` element. Each `<spml:supportedSchemaEntity>` element specifies
382 a top-level schema element that the provider supports for that target.

383 If the `<spml:schema>` element contains no `<spml:supportedSchemaEntity>` element, then the
384 requestor may assume that the provider supports for that target all of the top-level elements that
385 the schema of the target defines.

386 **4.2.4. Element `<supportedSchemaEntity>`**

387 The “`entityName`” attribute on the `<spml:supportedSchemaEntity>` element MUST refer to a top-
388 level element that is defined in the schema for a target. The provider MUST support every sub-
389 element and attribute of the referenced schema element.

390 **4.3. Search Capability**

391 **4.3.1. Element `<spmlsearch:query>`**

392 The `<spmlsearch:query>` element MAY contain an `<spml:select>` element. If an `<spml:select>`
393 element is defined, it MUST be set to a valid XPath statement for the XSD schema defined by the
394 target. The “[XPath Support](#)” section specifies general requirements for XPath support.

395 **4.3.2. Element `<spmlsearch:select>`**

396 An `<spmlsearch:select>` element MUST have a “`namespaceURI`” attribute and MUST have a
397 “`path`” attribute.

398 The value of the "namespaceURI" attribute MUST specify the XML namespace of a query
399 language. The value of the "path" attribute MUST be an expression that is valid in the query
400 language that "namespaceURI" specifies. (For example, if a requestor uses XPath 2.0 as the
401 query language for the "path" attribute, the value of the "namespaceURI" attribute MUST be
402 "http://www.w3.org/TR/xpath20".)

403 The value of the "path" attribute MUST specify a *filter* that selects objects based on:

- 404 • The presence (or absence) of a specific element or attribute
- 405 • The presence (or absence) of a specific value in the content of an element
406 or (the presence of absence of a specific value) in the value of an attribute

407 An <spmlsearch:select> element MAY include <spml:namespacePrefixMap> elements that defines
408 the namespace prefixes that are used in the XPath path. Each "prefix" attribute on the
409 <spml:namespacePrefixMap> element MUST exactly match one the namespace prefixes used in
410 the Xpath.

411

412 Appendix A. References

413

- 414 **[AES]** National Institute of Standards and Technology (NIST), FIPS-197:
415 Advanced Encryption Standard,
416 <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>, National
417 Institute of Standards and Technology (NIST)
- 418 **[ARCHIVE-1]** OASIS Provisioning Services Technical Committee, email archive,
419 [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/provision/email/archives/index.html)
420 [open.org/apps/org/workgroup/provision/email/archives/index.html](http://www.oasis-open.org/apps/org/workgroup/provision/email/archives/index.html),
421 OASIS PS-TC
- 422 **[DS]** IETF/W3C, *W3C XML Signatures*, <http://www.w3.org/Signature/>,
423 W3C/IETF
- 424 **[DSML]** OASIS Directory Services Markup Standard, *DSML V2.0 Specification*,
425 <http://www.oasis-open.org/specs/index.php#dsmlv2>, OASIS DSML
426 Standard
- 427 **[GLOSSARY]** OASIS Provisioning Services TC, *Glossary of Terms*, [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/provision/download.php)
428 [open.org/apps/org/workgroup/provision/download.php](http://www.oasis-open.org/apps/org/workgroup/provision/download.php), OASIS PS-TC
- 429 **[RFC 2119]** S. Bradner., *Key words for use in RFCs to Indicate Requirement Levels*,
430 <http://www.ietf.org/rfc/rfc2119.txt>, IETF
- 431 **[RFC 2246]** T. Dierks and C. Allen, *The TLS Protocol*,
432 <http://www.ietf.org/rfc/rfc2246.txt>, IETF
- 433 **[SAML]** OASIS Security Services TC, [http://www.oasis-](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security)
434 [open.org/committees/tc_home.php?wg_abbrev=security](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security), OASIS SS-
435 TC
- 436 **[SOAP]** W3C XML Protocol Working Group, <http://www.w3.org/2000/xp/Group/>
- 437 **[SPML-Bind]** OASIS Provisioning Services TC, SPML V1.0 Protocol Bindings,
438 [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/1816/draft-pstc-bindings-03.doc)
439 [open.org/apps/org/workgroup/provision/download.php/1816/draft-](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/1816/draft-pstc-bindings-03.doc)
440 [pstc-bindings-03.doc](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/1816/draft-pstc-bindings-03.doc), OASIS PS-TC
- 441 **[SPML-REQ]** OASIS Provisioning Services Technical Committee, *Requirements*,
442 [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/2277/draft-pstc-requirements-01.doc)
443 [open.org/apps/org/workgroup/provision/download.php/2277/draft-](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/2277/draft-pstc-requirements-01.doc)
444 [pstc-requirements-01.doc](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/2277/draft-pstc-requirements-01.doc), OASIS PS-TC
- 445 **[SPML-UC]** OASIS Provisioning Services Technical Committee, *SPML V1.0 Use*
446 *Cases*, [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/988/draft-spml-use-cases-05.doc)
447 [open.org/apps/org/workgroup/provision/download.php/988/draft-](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/988/draft-spml-use-cases-05.doc)
448 [spml-use-cases-05.doc](http://www.oasis-open.org/apps/org/workgroup/provision/download.php/988/draft-spml-use-cases-05.doc), OASIS PS-TC
- 449 **[SPMLv2-Profile-DSML]** OASIS Provisioning Services Technical Committee, SPMLv2
450 DSMLv2 Profile, OASIS PS-TC
- 451 **[SPMLv2-Profile-XSD]** OASIS Provisioning Services Technical Committee, SPML V2
452 XSD Profile, OASIS PS-TC
- 453 **[SPMLv2-REQ]** OASIS Provisioning Services Technical Committee, *Requirements*, OASIS
454 PS-TC

484 **Appendix B. Acknowledgments**

485 The following individuals were voting members of the Provisioning Services committee at the time
486 that this version of the specification was issued:

487 Jeff Bohren, BMC
488 Robert Boucher, CA
489 Gary Cole, Sun Microsystems
490 Rami Elron, BMC
491 Marco Fanti, Thor Technologies
492 James Hu, HP
493 Martin Raeppele, SAP
494 Gavenraj Sodhi, CA
495 Kent Spaulding, Sun Microsystems
496
497
498
499
500

501 Appendix C. Notices

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

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

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

516 Copyright (C) OASIS Open 2006. All Rights Reserved.

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

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

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

532