



S-RAMP Version 1.0. Part 2: Atom Binding

Committee Specification 01

23 December 2013

Specification URIs

This version:

<http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/cs01/part2-atom-binding/s-ramp-v1.0-cs01-part2-atom-binding.doc> (Authoritative)
<http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/cs01/part2-atom-binding/s-ramp-v1.0-cs01-part2-atom-binding.html>
<http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/cs01/part2-atom-binding/s-ramp-v1.0-cs01-part2-atom-binding.pdf>

Previous version:

N/A

Latest version:

<http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/s-ramp-v1.0-part2-atom-binding.doc> (Authoritative)
<http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/s-ramp-v1.0-part2-atom-binding.html>
<http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/s-ramp-v1.0-part2-atom-binding.pdf>

Technical Committee:

OASIS SOA Repository Artifact Model and Protocol (S-RAMP) TC

Chair:

Vincent Brunssen (brunssen@us.ibm.com), IBM

Editors:

Martin Smithson (msmiths@uk.ibm.com), IBM
Vincent Brunssen (brunssen@us.ibm.com), IBM

Additional artifacts:

This prose specification is one component of a Work Product that includes:

- XML schemas: <http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/cs01/schemas/>
- *S-RAMP Version 1.0. Part 1: Foundation*.
<http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/cs01/part1-foundation/s-ramp-v1.0-cs01-part1-foundation.html>.
- *S-RAMP Version 1.0. Part 2: Atom Binding*. (this document)
<http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/cs01/part2-atom-binding/s-ramp-v1.0-cs01-part2-atom-binding.html>.

Related work:

This specification is related to:

- Service Oriented Architecture Ontology (<http://www.opengroup.org/projects/soa-ontology/>)
- XML Schema Part 1: Structures Second Edition (<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>)
- Web Services Description Language (<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>)

Abstract:

Vendors offer tools to facilitate various activities across the life cycle of a SOA artifact, such as design, assembly, quality assurance, deployment and runtime operation of SOA based applications and business processes. The lack of a standardized information model and interaction protocol for artifacts and their metadata residing in a SOA repository means that tools must be customized for use with each different vendor's SOA repository product. This reduces choice, flexibility and adds costs for customers when choosing tools. This specification defines a SOA artifact data model together with bindings that describe the syntax for interacting with a SOA repository.

Status:

This document was last revised or approved by the OASIS SOA Repository Artifact Model and Protocol (S-RAMP) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/s-ramp/>.

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 Technical Committee web page (<http://www.oasis-open.org/committees/s-ramp/ipr.php>).

Citation format

When referencing this specification the following citation format should be used:

[S-RAMP-v1.0-atom-binding]

S-RAMP Version 1.0. Part 2: Atom Binding. Edited by Martin Smithson and Vincent Brunssen. 23 December 2013. OASIS Committee Specification 01. <http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/cs01/part2-atom-binding/s-ramp-v1.0-cs01-part2-atom-binding.html>. Latest version: <http://docs.oasis-open.org/s-ramp/s-ramp/v1.0/s-ramp-v1.0-part2-atom-binding.html>.

Notices

Copyright © OASIS Open 2013. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

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 section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, 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 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

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' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on 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 OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/policies-guidelines/trademark> for above guidance.

Table of Contents

1	Introduction.....	7
1.1	Terminology.....	7
1.2	Normative References.....	7
1.3	Non-Normative References.....	7
1.4	Abbreviations and Acronyms.....	8
1.5	XML Namespaces.....	8
2	S-RAMP Artifact to Atom Entry Mapping.....	9
2.1	Overview.....	9
2.2	Design Principles.....	9
2.3	Coarse Grained View.....	11
2.3.1	S-RAMP Atom Category Schemes & Terms.....	12
2.3.2	Atom Link Relation Values.....	14
2.3.3	Service Document.....	15
2.3.4	Atom Entries in S-RAMP.....	15
2.3.5	Publishing Artifacts in the Coarse Grained View.....	18
2.3.5.1	Publishing an Artifact Entry.....	18
2.3.5.2	Publishing Multiple Artifact Entries.....	24
2.3.5.3	Retrieving Repository Artifacts.....	31
2.3.5.4	Editing an Artifact Entry.....	33
2.3.5.5	Deleting an Artifact Entry.....	33
2.4	Fine Grained Views.....	34
2.4.1	S-RAMP Relationships.....	34
2.4.1.1	Relationship Feeds.....	34
2.4.1.2	Relationship Entry Documents.....	39
2.4.1.3	Relationship Type Entry Documents.....	44
2.4.1.4	Creating a Relationship Instance.....	46
2.4.1.5	Retrieving a Relationship Instance.....	51
2.4.1.6	Editing a Relationship Instance.....	52
2.4.1.7	Deleting a Relationship.....	52
2.4.2	S-RAMP Properties.....	54
2.4.2.1	Property Entry Documents.....	54
2.4.2.2	Creating Properties.....	57
2.4.2.3	Retrieving Properties.....	58
2.4.2.4	Editing Properties.....	59
2.4.2.5	Deleting Properties.....	60
2.4.3	S-RAMP Classifications.....	60
2.4.3.1	The Classification Entry Document.....	61
2.4.3.2	Creating Classifications.....	62
2.4.3.3	Retrieving Classifications.....	64
2.4.3.4	Editing Classifications.....	65
2.4.3.5	Deleting Classifications.....	65
3	S-RAMP Query Using Atom Binding.....	66
3.1	Searching Repository Artifacts.....	66
3.2	Inline Queries.....	66
3.3	Stored Queries.....	69

3.3.1 Stored Query Entry Documents	69
4 Security.....	72
5 Conformance.....	73
Appendix A. Acknowledgements.....	74
Appendix B. Non-Normative Text.....	75
Appendix C. Glossary	76
Appendix D. S-RAMP Atom Service Document	77
Appendix E. Notional S-RAMP URI Space	90
Appendix F. S-RAMP Atom Binding Schema	91
Appendix G. S-RAMP HTTP Response Codes.....	94
Appendix H. Revision History	96

Table of Tables

Table 1 - XML Namespace Prefixes Used.....	8
Table 2 - Mapping of built-in S-RAMP Artifact Properties to Atom Elements in an Entry Document	11
Table 3 - Category term Attributes for Entry Types	12
Table 4 - Link rel Attribute Values.....	14
Table 5 - S-RAMP URI Space.....	90
Table 6 - S-RAMP HTTP Response Codes.....	94
Table 7 - Error Attribute Values.....	95
Table 8 - SubElement Values	95

Table of Examples

Example 1 - Summary Artifact Entry.....	15
Example 2 - Full Artifact Entry with s-ramp:artifact Section.....	16
Example 3 - Publishing a Document Without Atom Multi-part POST	20
Example 4 - Initial Media Link Entry Returned Following a POST	20
Example 5 - Updating an Initial Media Link Entry with Metadata.....	21
Example 6 - Combined Publishing using Atom Multi-Part POST	23
Example 7 - Batch Post Construct Example	24
Example 8 - Successful Batch POST Response	26
Example 9 - Failed Batch POST Response – Complete Rollback	27
Example 10 - Failed Batch POST Response – Partial Create.....	27
Example 11 - Response from Publish using S-RAMP Package File Method	30
Example 12 – Error Response from Publish using S-RAMP Package File Method	31
Example 13 - Complex Relationship Scenario Summary Entry.....	36
Example 14 - Complex Relationship Scenario Relationships Feed	40
Example 15 – Backward Relationships Feed	42
Example 16 - Relationship Types Feed	45
Example 17 - Creating Generic Relationships - Before	47
Example 18 - Creating Generic Relationships - Adding the Relationship	48

Example 19 - Creating Generic Relationships - After	48
Example 20 - Adding a Relationship with No Targets	50
Example 21 - Retrieving a Relationship Entry Instance.....	51
Example 22 - Property Entry Feed.....	55
Example 23 - Creating a Property - Adding the Property	57
Example 24 - Creating a Property - After.....	57
Example 25 - Retrieving a Property Entry Document	59
Example 26 - Editing a Property Entry Document	60
Example 27 - Classification Entry Feed	61
Example 28 - Creating a Classification - Before	62
Example 29 - Creating a Classification - Adding the Classification Entry.....	63
Example 30 - Creating a Classification - After	64
Example 31 - Retrieving a Classification Entry Document	65
Example 32 - Ad-hoc Queries	67
Example 33 - Ad-hoc Query Response	68
Example 34 - Stored Query Entry Document.....	70

1 Introduction

The SOA - Repository Artifact Model and Protocol (S-RAMP) specification defines a common data model for SOA repositories to facilitate the use of common tooling and sharing of data. It provides a rich representation data model that supports fine-grained query. It includes binding(s) which document the syntax for interaction with a compliant Repository for create, read, update, delete, query and subscription operations within the context of each binding.

The specification is organized into several documents. This document, the SOA Repository Artifact Model and Protocol - Atom Binding document, builds upon the SOA Repository Artifact Model and Protocol – Foundation document. It describes the interaction protocol and syntax associated with using Atom to interact with an S-RAMP compliant Repository. Any other bindings will be expressed in their own separate binding documents.

When there is a discrepancy between this Atom Binding document and the Foundation document, the Atom Binding document takes precedence but only within the context of Atom based interaction within an S-RAMP compliant repository.

1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

This specification uses the following syntax to define normative outlines for messages:

- The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- The character "|" is used to indicate a choice between alternatives.
- The characters "{" and "}" are used to indicate that the contained item is a description of a value instead of the actual value itself.
- The characters "..." indicate where the syntax may be extended.
- XML namespace prefixes (see Table 1) are used to indicate the namespace of the element being defined.

1.2 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

1.3 Non-Normative References

- [URN] L. Daigle, D. W. van Gulik, R. Iannella and P. Faltstrom, *Uniform Resource Names (URN) Namespace Definition Mechanisms*, <http://www.ietf.org/rfc/rfc3406.txt>, IETF RFC 3406, October 2002.
- [APP] J. Gregorio and B. de hOra, *The Atom Publishing Protocol*, <http://www.ietf.org/rfc/rfc5023.txt>, IETF RFC 5023, October 2007.
- [ATOM] M. Nottingham and R. Sayre, *The Atom Syndication Format*, <http://www.ietf.org/rfc/rfc4287.txt>, IETF RFC 4287, December 2005.
- [ISO6392] *Codes for the Representation of Names and Languages – Part 2*, <http://www.loc.gov/standards/iso639-2/normtext.html>, ISO 639-2, 1998.

- 42 **[XML]** *Extensible Markup Language (XML) 1.0 Specification (Fifth Edition)*,
43 <http://www.w3.org/TR/2008/REC-xml-20081126/>, W3C Recommendation,
44 November 2008.
- 45 **[XMLNS]** *Namespaces in XML 1.0 (Second Edition)*, <http://www.w3.org/TR/2006/REC-xml-names-20060816/>, W3C Recommendation, August 2006.
- 47 **[XSD]** *XML Schema Part 1: Structures Second Edition, version 1.0*,
48 <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>, W3C
49 Recommendation, October 2004.
- 50 **[XPATH]** *XML Path Language (XPath) 2.0 (Second Edition)*,
51 <http://www.w3.org/TR/2010/REC-xpath20-20101214/>, W3C Recommendation,
52 December 2010.
- 53 **[UUID]** P. Leach, M. Mealling, and R. Salz, *A Universally Unique Identifier (UUID) URN
54 Namespace*, <http://www.ietf.org/rfc/rfc4122.txt>, IETF RFC 4122, July 2005.

55 1.4 Abbreviations and Acronyms

- 56 APP Atom Publishing Protocol
- 57 S-RAMP SOA Repository Artifact Model and Protocol
- 58 XPath2 XML Path Language (XPath) 2.0

59 1.5 XML Namespaces

60 The XML namespace URIs that MUST be used by implementations of this specification is:

61

62 <http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0> The namespaces used in this
63 document are provided in Table 1 below. The choice of any namespace prefix is arbitrary and not
64 semantically significant.

65

66 Table 1 - XML Namespace Prefixes Used

Prefix	XML Namespace	Specification(s)
atom	http://www.w3.org/2005/Atom	Atom Syndication Format
app	http://www.w3.org/2007/app	Atom Publishing Protocol
fn	http://www.w3.org/2005/xpath-functions	XPath 2
s-ramp	http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0	S-RAMP Foundation
wSDL	http://schemas.xmlsoap.org/wSDL/	WSDL [WSDL 1.1]
wsp	http://schemas.xmlsoap.org/ws/2002/12/policy	WS-Policy [WS-Policy]
xsd	http://www.w3.org/2001/XMLSchema	XML Schema [Part 1, 2]

2 S-RAMP Artifact to Atom Entry Mapping

68 This chapter describes how S-RAMP artifacts are represented in Atom as well as how to perform create,
69 retrieve, update, delete operations against the data in an S-RAMP compliant repository.

70 2.1 Overview

71 This specification suggests a single URI space for S-RAMP. This URI space is organized according to the
72 Artifact Type Model structure. This notional URI space is used throughout this document to reference
73 repository objects in all Atom requests and responses. See Appendix F for details. Note, however, that
74 the URI space for any given implementation is not prescribed, but **MUST** be discoverable via the Service
75 Document.

76 It is useful to discuss several other basic characteristics of the Atom mapping for S-RAMP.

- 77 • Atom entry documents are used to represent all S-RAMP artifacts in the repository.
- 78 • S-RAMP makes use of the APP media resource and media link entry concepts.
- 79 • In its full representation, all Atom entry documents (including Media Link Entries) used in S-
80 RAMP contain the metadata associated with the S-RAMP repository artifact which it represents.
- 81 • S-RAMP artifact metadata will be represented in the same way in an Atom entry document,
82 regardless of what kind of S-RAMP artifact it represents.
- 83 • For the convenience of Atom clients that do not understand S-RAMP data structures, applicable
84 components of S-RAMP specific artifact metadata have been mapped to specific Atom elements
85 for use in messages. Complete S-RAMP instance metadata, however, **SHALL** be provided in the
86 full representation of an Artifact Entry document using foreign markup and **SHALL** be compliant
87 with the S-RAMP schemas described in the Appendices of the SOA Repository Artifact Model
88 and Protocol Specification – Foundation Book. While this representation is duplicative, it provides
89 the greatest value for both S-RAMP aware as well as simple Atom clients. In all cases, however,
90 the foreign markup content of a message **SHALL** take precedence and be considered
91 authoritative by all S-RAMP compliant repositories.
- 92 • The HTTP POST message **SHALL** be used for creation of all new artifacts, and HTTP PUT
93 **SHALL** be used for updating existing S-RAMP artifacts (see exception noted in Section 2.3.5.2.2
94 concerning the use of S-RAMP package files to publish S-RAMP content).
- 95 • All S-RAMP artifacts are represented by the Atom Binding schema in Appendix Appendix F.
- 96 • Text in this document takes precedence over the schema.

97 2.2 Design Principles

98 Metadata in the S-RAMP data model comes in several forms:

- 99 • Built-in and user-defined properties which describe a repository artifact.
- 100 • User-defined classifications which describe a repository artifact.
- 101 • Built-in and user-defined relationships between two repository artifacts.

102 S-RAMP has adhered to several design criteria requirements which have guided mapping of the binding
103 independent S-RAMP Artifact Type Model onto an Atom binding:

- 104 • The Atom entry document is the Atom representation of an S-RAMP artifact object. There are two
105 possible Atom representations of an S-RAMP artifact object.
 - 106 1. A summary Atom entry that appears in an Atom feed document. Summary entries do not
107 include the S-RAMP structured extension element (S-RAMP foreign markup).
 - 108 2. A complete Atom entry which does include the S-RAMP structured extension element
109 (see Section 2.3.2).

- 110 • It MUST be possible to update all user-editable S-RAMP metadata with a single update to an
111 Atom entry document.
- 112 ○ Some of the S-RAMP Artifact Type Model built-in properties have been mapped to
113 existing Atom elements present in any legal Atom entry, wherever there is an obvious
114 and direct analog. This provides some basic value to any Atom client.
 - 115 ○ The full Atom representation of an S-RAMP object will also include foreign markup in the
116 form of a structured extension element (s-ramp:artifact). The XML markup within this
117 element is compliant with the binding independent S-RAMP schemas, and provides a
118 complete instance document describing the artifact represented by the Atom entry
119 document. While this does create some degree of duplication with those few items which
120 are directly mapped to Atom elements, it does permit single step operations on repository
121 artifacts and it facilitates various processing optimizations.
- 122 2. Given the richness and complexity of S-RAMP metadata, it is useful for Atom clients to have
123 greater contextual value and more convenient mechanisms for reading and editing that metadata
124 beyond what is otherwise only available as foreign markup (described above). S-RAMP defines
125 normative features whose implementation is OPTIONAL that provide clients with the ability to
126 manipulate individual pieces of S-RAMP metadata.
- 127 3. Whenever a feed is returned as the response to a client request to an S-RAMP server, the server
128 MAY return a partial list that includes the appropriate links to support client pagination of the feed
129 using the “first”, “last”, “previous” and “next” link rel types defined in the Atom Publishing Protocol
130 specification, Section 10.1 “Collection partial lists”.

131 These design principles have given rise to two major approaches to representing S-RAMP metadata in
132 Atom:

- 133 1. Coarse Grained View (normative). This is an Atom entry document (Artifact Entry) which
134 represents an S-RAMP object. It contains:
- 135 ○ A subset of S-RAMP built-in property values mapped to existing Atom elements, as
136 described in Table 2 of Section 2.3, in both summary and complete representations.
 - 137 ○ Only the complete Atom entry representation also contains a foreign markup section
138 describing a structured extension element called s-ramp:artifact. This extension contains
139 a complete S-RAMP XSD schema compliant instance document representing the S-
140 RAMP artifact object.
 - 141 ○ The value of each target of a relationship in s-ramp:artifact for the Atom Binding SHALL
142 be the URI of the Target Entry.
- 143 2. Fine Grained Views (optional features which have a normative interface when implemented):
- 144 ○ These provide a hierarchical representation for a given class of metadata (relationships,
145 properties or classifications)
 - 146 ○ Fine Grained Views allow an S-RAMP client to navigate to and manipulate the applicable
147 metadata without the need to retrieve the full Atom representation of the S-RAMP object,
148 which includes a potentially large structured extension element (see Section 2.3.2).
149 Detailed information concerning the various Fine Grained Views can be found in Section
150 2.4.
 - 151 ○ The Artifact Entry document MAY contain Atom link(s) to feed(s) which contain Fine
152 Grained View(s) for any of the three major classes of metadata described above.
 - 153 ○ The presence of these feed link(s) in each case indicates whether fine-grained
154 support is exposed by a given implementation, for the indicated class(es) of
155 metadata.
 - 156 ○ Using these fine-grained feeds, one can publish, retrieve, edit and delete
157 individual pieces of such metadata without having to edit the Artifact Entry
158 document which includes those feeds. For relationship metadata, it is also
159 possible to perform a bulk deletion of all relationships of a given Relationship
160 Type in one operation.

161 o When Fine Grained View(s) are supported, an S-RAMP server SHALL present
 162 the appropriate feed link(s) in an Atom entry document. Attempts by a client to
 163 alter any of these feed links are ignored by the server.

164 Other miscellaneous conventions which pertain to Atom entry documents used in S-RAMP:

- 165 • Artifact Entry atom:id elements SHALL be a UUID, which is mapped from the S-RAMP artifact
 166 UUID value, expressed as a legal URN (e.g., urn:uuid:{uuid value}). atom:id element values for all
 167 other entry documents in S-RAMP have no normative format and are server generated. They
 168 SHALL be unique values, but do not need to be parsed or understood. Examples presented
 169 throughout this document, however, will either contain a real URN compliant UUID value, or will
 170 use a symbolic short name representing a UUID value for a the artifact, such as
 171 {uuid:source.xsd}, {uuid:target.xsd} and so on, to simplify or add clarity as needed.
- 172 • Also note that servers SHALL retain atom:id values chosen by clients for Artifact Entry documents
 173 during initial publish.
- 174 • href values used in links and src values used in atom:content elements carry no special meaning.
 175 As above, examples presented throughout this specification MAY depart from this rule simply to
 176 add clarity.
- 177 • URN values are used in S-RAMP atom:category scheme and atom:link rel attribute values. By
 178 convention, S-RAMP assumes a format for these URN values which contains ":" separated terms.
 179 Multiple ":" separators in a URN are permitted. At the time of publication, S-RAMP URN values
 180 use the "x-" convention established by RFC 2611.
- 181 • For clarity, the sample URLs used throughout this document have generally not been URL
 182 encoded.
- 183 • URLs used throughout this document often omit the base URL component which precedes "/s-
 184 ramp/...". For example:

185 "http://host:port/mysoa/s-ramp/..."

186 is presented as simply:

187 "/s-ramp/..."

191 2.3 Coarse Grained View

192 Table 2 below defines a mapping for some S-RAMP artifact data items from the S-RAMP schema to built-
 193 in Atom elements in an Atom entry document. Only a small subset of such data is directly mapped. More
 194 is discussed on the remainder in subsequent sections.

195 *Table 2 - Mapping of built-in S-RAMP Artifact Properties to Atom Elements in an Entry Document*

S-RAMP Type Element	S-RAMP Property	Atom Element Mapping
BaseArtifactType	createdBy	Corresponds to the atom:name element of the atomPersonConstruct of the atom:author element. No mapping is defined for other elements in this construct.
	uuid	atom:id
	createdTimestamp	atom:published
	lastModifiedTimestamp	atom:updated
	lastModifiedBy	atom:contributor element

		<p>Corresponds to the atom:name element of the atom:personConstruct of the atom:contributor element. No mapping is defined for other elements in this construct.</p> <ul style="list-style-type: none"> <i>Note:</i> S-RAMP restricts the number of occurrences of atom:contributor in an entry document to one.
	name	atom:title
	description	atom:summary
DocumentArtifactType	contentType	<p><u>In the Media Link Entry:</u> atom:content, with this attribute:</p> <ul style="list-style-type: none"> type (for contentType)
XmlDocument	contentEncoding	<p>atom:content, <i>type</i> attribute</p> <ul style="list-style-type: none"> Including the type attribute using a charset attribute on the MIME media type. Example: type="application/xml; charset=utf-8"

196

197 2.3.1 S-RAMP Atom Category Schemes & Terms

198 S-RAMP pre-defines several atom:category scheme attribute and corresponding term values for
199 describing the data present in an Atom entry document. These term values are fixed.

- 200 urn:x-s-ramp:2013:type

- 201 o Indicates the type of S-RAMP artifact represented by the Entry.

- 202 o Appears in all entry documents and media link entry documents.

203 Defined values for the term attribute are described in table 3 below.

204 *Table 3 - Category term Attributes for Entry Types*

Type of Entry	Defined <i>term</i> value(s)
Artifact Entry (non-document)	"HumanActor" "AttributeDeclaration" "Binding" "BindingOperation" "BindingOperationInput" "BindingOperationOutput" "BindingOperationFault" "Choreography" "ChoreographyProcess" "Collaboration" "CollaborationProcess" "ComplexTypeDeclaration"

	"Composition" "Effect" "Element" "ElementDeclaration" "Event" "Fault" "InformationType" "Message" "Operation" "OperationInput" "OperationOutput" "Orchestration" "OrchestrationProcess" "Organization" "Part" "Policy" "PolicyAttachment" "PolicyExpression" "PolicySubject" "Port" "PortType" "Process" "SoapAddress" "SoapBinding" "SimpleTypeDeclaration" "Service" "ServiceComposition" "ServiceContract" "ServiceEndpoint" "ServiceInstance" "ServiceInterface" "ServiceOperation" "WsdExtension" "WsdService" "XsdType" "{Extended Artifact Type}"
Artifact (Media Link) Entry (corresponds to a document)	"Document" "PolicyDocument" "WsdDocument" "XmlDocument" "XsdDocument"
Relationship Entry	"relationship"

Relationship Type Entry	"relationshipType"
Property Entry	"property"
Classification Entry	"classification"
Stored Query Entry	"query"

- 205
- 206
- urn:x-s-ramp:2013:kind
 - Indicates the kind of the entry
 - Occurs in Artifact Entry, Relationship Target Entry, Relationship Type Entry, and Property Entry documents, except as noted below.
 - Legal values for the *term* attribute are
 - "derived"
 - Indicates entry is part of a Derived Model
 - "modeled"
 - Indicates entry is pre-defined and is part of the SOA or Service Implementation Models or is part of an extended artifact model
 - "generic"
 - Indicates entry is ad-hoc
 - Does not occur in Artifact Entry documents.

219 2.3.2 Atom Link Relation Values

220 Table 4 below summarizes all of the relation attribute (*rel*) values for links used in S-RAMP:

221 *Table 4 - Link rel Attribute Values*

Attribute Value	Where Used
self	All entry documents
edit-media	All media link entry documents
edit	All entry documents which can be edited by the client
urn:x-s-ramp:2013:relationships	Source Entry documents: Link to a feed of all Relationship entry documents
urn:x-s-ramp:2013:relationshipTypes	Source Entry and Relationship Type Entry documents: Link to a feed of all Relationship Type entry documents
urn:x-s-ramp:2013:relationships: {Relationship Type}	Source Entry documents: Link to a feed of Relationship entry documents which share the specified Relationship Type
urn:x-s-ramp:2013:backwardRelationships	Target Entry documents: Link to a feed of backward relationships of a Target Entry document (applies ONLY to modeled and derived relationships)
urn:x-s-ramp:2013:backwardRelationships: {Relationship Type}	Target Entry documents: Link to a feed of backward relationships for which the subject Target Entry document is the target of a relationship (of a given Relationship Type) in a Source Entry document. This feature ONLY applies to modeled and derived relationships. Each Backward Relationship Entry will only have a single target that points to the appropriate

	Source Entry Document, because a forward relationship from the Source Entry document never has duplicate targets for a given Relationship Type.
urn:x-s-ramp:2013:relationship:source	Relationship Entry Documents. Used in the Atom link to the Source Entry of the relationship.
urn:x-s-ramp:2013:relationship:target	Relationship Entry Documents. Used in the Atom link to the Target Entry of the relationship.
urn:x-s-ramp:2013:relationshipType	Relationship Entry Documents. Used in the Atom link to the Relationship Type Entry which corresponds to the Relationship Type value for this Relationship Entry.
urn:x-s-ramp:2013:properties	Artifact Entry documents: Link to a feed of all Property Entry documents
urn:x-s-ramp:2013:classifications	Artifact Entry documents: Link to a feed of all Classification Entry documents
urn:x-s-ramp:2013:query:results	Stored Query Entry Documents. Used in the results feed associated with execution of a given Stored Query.

222

223 2.3.3 Service Document

224 This section describes how S-RAMP implementations publicize the top level collections defined by this
225 specification in an Atom Publishing Protocol Service Document.

226 S-RAMP implementations SHALL return an Atom Publishing Protocol Service Document to clients who
227 perform an HTTP GET on the following URL:

228

229 {base URL}/s-ramp/servicedocument
230

231 The content of the Service Document that is returned is defined as follows:

- 232 • MUST contain a workspace for each of the artifact models identified in Section 3 of the SOA
233 *Repository Artifact Model & Protocol Specification – Foundation Document*.
- 234 • Each workspace MUST contain an app:collection element for each of the artifact types that are
235 defined within the corresponding artifact model for that workspace.
- 236 • Each collection in a workspace MUST specify an atom:categories element that will define the
237 categories that MUST be applied to the member resources of the collection as defined in Section
238 2.3.1.
- 239 • The workspace for the query artifact model MUST contain an app:collection element for each
240 Stored Query that exists in the S-RAMP implementation.

241 The workspace for the SOA or Service Implementation Artifact Model MUST contain an app:collection
242 element for each pre-defined type described in the S-RAMP specification.

243 2.3.4 Atom Entries in S-RAMP

244 An example of an S-RAMP summary (media link) entry which corresponds to the accountingTypes.xsd
245 resource is shown below. The mapping defined in Table 2 is illustrated in Example 1:

246 *Example 1 - Summary Artifact Entry*

```
247     <entry xmlns="http://www.w3.org/2005/Atom"
248         xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
249         <id>urn:uuid:aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a</id>
250         <updated>2009-05-26T13:13:55.013+02:00</updated>
```

```

251     <title type="text">accountingTypes.xsd</title>
252     <published>2009-05-26T13:13:55.013+02:00</published>
253     <author>
254         <name>Bellwood</name>
255     </author>
256     <contributor>
257         <name>Pospisil</name>
258     </contributor>
259     <summary type="text">Accounting types schema document</summary>
260     <content type="application/xml"
261         src="http://example.org/s-ramp/xsd/XsdDocument/
262             aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a/media"/>
263     <link type="application/atom+xml;type=entry" rel="self"
264         href="http://example.org/s-ramp/xsd/XsdDocument/
265             aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a" />
266     <link type="application/atom+xml;type=entry" rel="edit-media"
267         href="http://example.org/s-ramp/xsd/XsdDocument/
268             aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a/media" />
269     <link type="application/atom+xml;type=entry" rel="edit"
270         href="http://example.org/s-ramp/xsd/XsdDocument/
271             aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a" />
272
273     <!--
274         Links to of the optional feeds provided by server implementations which
275         support the various the Fine Grained Views defined in S-RAMP omitted for
276         brevity.
277         See Section 2.4 for complete information on these feeds and the Fine
278         Grained View.
279     -->
280     <!--
281         S-RAMP defined categorizations identifying class of data represented by
282         this entry
283     -->
284     <category term="XsdDocument" label="XML Schema"
285         scheme="urn:x-s-ramp:2013urn:x-s-ramp:2013:type" />
286 </entry>
287
288 As noted earlier, a full Atom entry representation of an S-RAMP object includes foreign markup in the
289 form of a structured extension element called s-ramp:artifact. This contains an S-RAMP schema
290 compliant XML instance fragment describing the complete S-RAMP artifact. A sample s-ramp:artifact
291 section is shown below in the full version of the Artifact Entry described in the example in Section 2.3.
292 The optional Fine Grained View feed links are again omitted here for brevity:
293
294 Example 2 - Full Artifact Entry with s-ramp:artifact Section
295
296 <entry xmlns="http://www.w3.org/2005/Atom"
297     xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-
298     v1.0http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0"
299     xmlns:xlink="http://www.w3.org/1999/xlink" >
300     <id>urn:uuid:aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a</id>
301     <updated>2009-05-26T13:13:55.013+02:00</updated>
302     <title type="text">accountingTypes.xsd</title>
303     <published>2009-05-26T13:13:55.013+02:00</published>
304     <author>

```



```

302     <name>Bellwood</name>
303 </author>
304 <contributor>
305     <name>Pospisil</name>
306 </contributor>
307 <summary type="text">accountingTypes.xsd schema document</summary>
308 <content type="application/xml;charset=utf-8"
309     src="http://example.org/s-ramp/xsd/XsdDocument/
310     aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a/media"/>
311 <link type="application/atom+xml;type=entry" rel="self"
312     href="http://example.org/s-ramp/xsd/XsdDocument/
313     aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a"/>
314 <link type="application/atom+xml;type=entry" rel="edit-media"
315     href="http://example.org/s-ramp/xsd/XsdDocument/
316     aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a/media" />
317 <link type="application/atom+xml;type=entry" rel="edit"
318     href="http://example.org/s-ramp/xsd/XsdDocument/
319     aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a" />
320
321 <!--
322     S-RAMP defined categorizations identifying class of data represented by
323     this entry
324 -->
325 <category term="XsdDocument" label="XML Schema"
326     scheme="urn:x-s-ramp:2013urn:x-s-ramp:2013:type" />
327 <s-ramp:artifact xsi:schemaLocation="http://docs.oasis-open.org/s-ramp/ns/s-
328 ramp-v1.0http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0
329     http://s-ramp.org/2010/specification/schemas/xsdmodel.xsd"
330     xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-
331 v1.0http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0"
332     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
333     <s-ramp:XsdDocument name="accountingTypes.xsd"
334         description="accountingTypes.xsd"
335         createdBy="Bellwood" version="1.0"
336         uuid="aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a"
337         createdTimestamp="2009-05-26T13:13:55.013+02:00"
338         lastModifiedTimestamp="2009-06-26T13:13:55.013+02:00"
339         lastModifiedBy="Pospisil" contentEncoding="UTF-8"
340         contentType="application/xml" contentsize="4096"
341         targetNamespace="http://example.org/accountingTypes">
342     <s-ramp:classifiedBy>
343         http://example.org/ontologies/accounting.owl/accounts
344     </s-ramp:classifiedBy>
345     <!--
346         Example of a user created generic relationship called "similarXsds"
347         between this xsd artifact and two others with UUID values of "...a6b"
348         and "...a6c"), respectively:
349     -->
350     <s-ramp:relationship>
351         <s-ramp:relationshipType>similarXsds</s-ramp:relationshipType>
352         <s-ramp:relationshipTarget>

```

```

353         <s-ramp:target
354           xlink:href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
355           aaaa-aaaaaaaaa6b">
356             aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6b
357         </s-ramp:target>
358     </s-ramp:relationshipTarget>
359     <s-ramp:relationshipTarget>
360         <s-ramp:target
361           xlink:href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
362           aaaa-aaaaaaaaa6c">
363             aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6c
364         </s-ramp:target>
365     </s-ramp:relationshipTarget>
366 </s-ramp:relationship>
367 <s-ramp:property>
368     <propertyName>myPropertyName</propertyName>
369     <propertyValue>myPropertyValue</propertyValue>
370 </s-ramp:property>
371 </s-ramp:XsdDocument>
372 <!-- Example of the "importedXsds" Derived relationship -->
373 <s-ramp:importedXsds>
374     <s-ramp:target
375       xlink:href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
376       aaaa-aaaaaaaaa6b">
377         aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6b
378     </s-ramp:target>
379 </s-ramp:importedXsds>
380 </s-ramp:artifact>
381 </entry>

```

382 2.3.5 Publishing Artifacts in the Coarse Grained View

383 The Coarse Grained View in S-RAMP supports a complete representation of all of the metadata which
384 describes an S-RAMP artifact, mapped to an Atom entry document. Some of S-RAMP built-in properties
385 are mapped directly to existing Atom entry elements for the convenience of clients, but as illustrated in
386 Section 2.3.2, the S-RAMP structured extension element (s-ramp:artifact) contains a complete S-RAMP
387 schema compliant representation of the artifact, and MUST be used to publish in the Coarse Grained
388 View.

389 Publishing in this view requires understanding the s-ramp:artifact structured extension, as it is the vehicle
390 by which all metadata associated with the entry is specified.

391 Publishing an artifact to the wrong collection will result in HTTP error "403" Forbidden.

392 2.3.5.1 Publishing an Artifact Entry

393 Publishing of new artifacts to an S-RAMP compliant repository is accomplished using HTTP POST. All S-
394 RAMP artifacts have an atom:entry representation, but only those which are not derived artifacts (such as
395 document or SOA Model artifacts) can be published directly by the client. By convention, S-RAMP
396 documents are treated as *media resources* by the Atom Binding. Publication of non-document artifacts is
397 accomplished using HTTP POST of an Atom entry document which represents it. Publication of a
398 document via HTTP POST will result in the creation of the document artifact in the repository AND a
399 Media Link Entry document which corresponds to it and is returned in response to the POST. This entry
400 will contain initial metadata associated with the document. The UUID in it is set by the server, and the
401 document includes an edit-media link to the media resource as does the atom:content element via the src
402 attribute, although the value of the IRI of the src attribute on the atom:content element does not need to

403 be the same as the media resource IRI contained in the edit-media link. This allows implementations to
404 point clients at a cached version of a Media Resource. The Media Link Entry can then be updated to
405 modify or add additional metadata and be PUT to the repository as desired. Unfortunately APP currently
406 makes this a multi-step process if one wishes to publish a document together with domain specific
407 metadata associated with the document. Typical steps might entail:

- 408 1. POST the document. This saves the document in the repository, and returns the initial Media Link
409 Entry.
- 410 2. Edit the Media Link Entry to insert and change metadata as desired
- 411 3. PUT the Media Link Entry to update its contents in the repository

412 To simplify this process for clients, S-RAMP implementations SHALL also support publication of media
413 resources and Media Link Entries in a single step using the procedure described in the *ATOM Multi-part*
414 *Media Resource Creation* draft document (hereafter referred to as "*Atom Multi-part POST*") under review
415 by the IETF. It specifies extensions to APP which allow simultaneous publishing of a media resource and
416 its corresponding Media Link Entry in a single HTTP request. The body of the request thus contains both
417 the media resource and a boundary delimited Media Link Entry containing all the desired metadata for the
418 resource. The result of this operation will save the document in the repository, and create a complete
419 Media Link Entry containing an edit-media link to that document with no additional steps.

420 Notes:

- 421 • A client MAY specify the name of an artifact using the Slug header in the HTTP POST. Clients
422 SHOULD not assume that the Slug header is used to influence the atom:id or the URIs.
- 423 • If an Atom Multi-part POST is used, then the name and UUID values SHOULD only be provided
424 as part of the s-ramp:metadata in the Media Link Entry portion of the POST.
- 425 • UUID values can be provided by the user for a non-document Artifact being published. In order to
426 conform to the APP specification (sections 4.3 and 9.2), the following HTTP operations will have
427 only these uses:
 - 428 ○ POST
 - 429 ▪ Only used to create new Atom entry documents.
 - 430 ▪ Any uuid property value supplied MUST NOT already be present in the
431 repository.
 - 432 ▪ If the supplied uuid property value duplicates one already in the repository, the
433 server SHALL return an HTTP error code of "409" indicating a Conflict.
 - 434 ▪ If a UUID is not supplied, the server SHALL create one.
 - 435 ○ PUT
 - 436 ▪ Only used to update an existing Atom entry document.
 - 437 ▪ The uuid property value CANNOT be changed.
 - 438 ▪ If the artifact being edited is not in the repository at the time of the PUT, then the
439 server SHALL return an HTTP error code of "404" indicating a Not Found.
- 440 • Within a full Coarse Grained Artifact Entry document, the s-ramp:artifact element might contain
441 Modeled or Derived relationships, whose type is s-ramp:target. It might also contain user
442 provided Generic relationships, whose s-ramp:relationshipTarget element is also of type s-
443 ramp:target). The Core Model Schema in Appendix A of the Foundation Document defines the s-
444 ramp:target type. The s-ramp:target element in the Core Model Schema terminates in the ##any
445 attribute. The Atom Binding requires that server responses include an xlink:href attribute in the
446 target element. The value of this attribute MUST contain a URL pointing to the target. During a
447 publish operation, the link need not be included and the server MUST ignore it. See Example 2
448 above for more.
- 449 • Document artifacts MAY only be published to a valid collection described in the Atom Service
450 Document (see Appendix A). For example, the ".../s-ramp/xsd/XsdDocument" collection.
- 451 • Documents which are part of a S-RAMP package file MAY be published to the top level URI for
452 the appropriate model in which they reside (e.g., /s-ramp/xsd). The repository will determine the

453 appropriate collection(s) in which to place its constituent entries based on the category
454 information in the entry document(s) included in the S-RAMP package file. For more on
455 publishing using S-RAMP package files, refer to Section 2.3.5.2.2. If the S-RAMP package file
456 contains files which belong in different S-RAMP models, the S-RAMP package file should be
457 published to the top level collection (i.e., /s-ramp).

458 Example 3 below illustrates publishing of the accountingTypes.xsd document without using the Atom
459 Multipart POST method. Publication is done in two steps. First we POST the document itself:

460

461 *Example 3 - Publishing a Document Without Atom Multi-part POST*

462 POST /s-ramp/xsd/XsdDocument HTTP/1.1

463 Host: example.org

464 Content-Type: application/xml

465 Content-Length: nnn

466 Slug: accountingTypes.xsd

467

468 {accountingTypes.xsd document content goes here}

469

470 In response to this POST, the server will return an initial Media Link Entry based upon the information
471 provided on the POST and within the XSD file. For a moment, we will assume that this XSD file does NOT
472 include or import any other XSD files and thus includes no external resource dependencies. The
473 atom:summary element value at this point is implementation defined, and there are no user defined
474 properties or classifications yet. Clients can adjust all of these later:

475

476 *Example 4 - Initial Media Link Entry Returned Following a POST*

477 HTTP/1.1 201 Created

478 Date: Tues, 26 May 2009 13:13:55 GMT+2:00

479 Content-Length: nnn

480 Content-Type: application/atom+xml;type=entry;charset="utf-8"

481 Location: http://example.org/s-ramp/xsd/XsdDocument/

482 aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a

483 ETag: "c181bb840673b5"

484

485 <?xml version="1.0"?>

486 <entry xmlns="http://www.w3.org/2005/Atom"

487 xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">

488 <id>urn:uuid:aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a</id>

489 <updated>2009-05-26T13:13:55.013+02:00</updated>

490 <title type="text">accountingTypes.xsd</title>

491 <published>2009-05-26T13:13:55.013+02:00</published>

492 <author>

493 <name>Bellwood</name>

494 </author>

495 <summary type="text">accountingTypes.xsd schema document</summary>

496 <content type="application/xml;charset=utf-8"

497 src="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
498 aaaa-aaaaaaaaa6a/media"/>

499 <link type="application/atom+xml;type=entry" rel="self"

500 href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
501 aaaa-aaaaaaaaa6a" />

502 <link type="application/atom+xml;type=entry" rel="edit-media"

```

503         href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
504         aaaa-aaaaaaaaaaaa6a/media" />
505         <link type="application/atom+xml;type=entry" rel="edit"
506         href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
507         aaaa-aaaaaaaaaaaa6a" />
508
509         <!--
510         S-RAMP defined categorizations identifying class of data represented by
511         this entry
512         -->
513         <category term="XsdDocument" label="XML Schema"
514         scheme="urn:x-s-ramp:2013urn:x-s-ramp:2013:type" />
515         <s-ramp:artifact xsi:schemaLocation="http://docs.oasis-open.org/s-ramp/ns/s-
516         ramp-v1.0/xsdmodel.xsd"
517         xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-
518         v1.0"
519         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
520         <s-ramp:XsdDocument name="accountingTypes.xsd"
521         description="accountingTypes.xsd schema document"
522         createdBy="Bellwood" version="1.0"
523         uuid="aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa6a"
524         createdTimestamp="2009-05-26T13:13:55.013+02:00"
525         lastModifiedTimestamp="2009-05-26T13:13:55.013+02:00"
526         lastModifiedBy="Bellwood" contentEncoding="UTF-8"
527         contentType="application/xml" contentSize="4096" >
528         </s-ramp:XsdDocument>
529         </s-ramp:artifact>
530     </entry>

```

531

532 Clients can update this Media Link Entry to add additional metadata. This is done using an HTTP PUT as

533 in Example 5 below. All such metadata added using the Coarse Grained View is done through the s-

534 ramp:artifact extension. This example illustrates adding a user-defined property and classification in the s-

535 ramp:artifact section.

536 Clients SHOULD perform a GET on the entry to insure it is complete and current. S-RAMP server

537 implementations SHALL support and return an ETag to allow conditional GET and PUT.

538

539 *Example 5 - Updating an Initial Media Link Entry with Metadata*

```

540     PUT /s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa6a HTTP/1.1
541     Host: example.org
542     Content-Type: application/atom+xml;type=entry
543     Content-Length: nnn
544     If-Match: "c181bb840673b5"
545
546     <?xml version="1.0"?>
547     <entry xmlns="http://www.w3.org/2005/Atom"
548     xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0" >
549     <id>urn:uuid:aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa6a</id>
550     <updated>2009-05-26T13:13:55.014+02:00</updated>
551     <title type="text">accountingTypes.xsd</title>
552     <published>2009-05-26T13:13:55.013+02:00</published>
553     <author>

```

```

554     <name>Bellwood</name>
555 </author>
556 <contributor>
557     <name>Pospisil</name>
558 </contributor>
559 <summary type="text">accountingTypes.xsd schema document</summary>
560 <content type="application/xml;charset=utf-8"
561     src="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
562     aaaa-aaaaaaaaaaaa6a/media"/>
563     <link type="application/atom+xml;type=entry" rel="self"
564         href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
565     aaaa-aaaaaaaaaaaa6a" />
566     <link type="application/atom+xml;type=entry" rel="edit-media"
567         href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
568     aaaa-aaaaaaaaaaaa6a/media" />
569     <link type="application/atom+xml;type=entry" rel="edit"
570         href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
571     aaaa-aaaaaaaaaaaa6a" />
572
573 <!--
574     S-RAMP defined categorizations identifying class of data represented by
575     this entry
576 -->
577 <category term="XsdDocument" label="XML Schema"
578     scheme="urn:x-s-ramp:2013urn:x-s-ramp:2013:type" />
579 <s-ramp:artifact xsi:schemaLocation="http://docs.oasis-open.org/s-ramp/ns/s-
580 ramp-v1.0 /xsdmodel.xsd"
581     xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-
582 v1.0"
583     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
584 <s-ramp:XsdDocument name="accountingTypes.xsd"
585     description="accountingTypes.xsd schema document"
586     createdBy="Bellwood" version="1.0"
587     uuid="aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa6a "
588     createdTimestamp="2009-05-26T13:13:55.013+02:00"
589     lastModifiedTimestamp="2009-05-26T13:13:56.013+02:00"
590     lastModifiedBy="Bellwood" contentEncoding="UTF-8"
591     contentType="application/xml" contentSize="4096" >
592 <s-ramp:classifiedBy>
593     http://example.org/ontologies/accounting.owl/accounts
594 </s-ramp:classifiedBy>
595 <s-ramp:property>
596     <propertyName>accountingCalendar</propertyName>
597     <propertyvalue>2009</propertyvalue>
598 </s-ramp:property>
599 </s-ramp:XsdDocument>
600 </s-ramp:artifact>
601 </entry>
602

```

603 The steps illustrated in Example 3 through Example 5 above could all be performed in a single HTTP
604 request using the Atom Multi-part POST. This method allows combining both the Atom Media Link Entry
605 and its corresponding media resource document in a single POST. Example 6 below presents the same
606 example in this combined approach:

```

607
608 Example 6 - Combined Publishing using Atom Multi-Part POST
609     POST /s-ramp/xsd/XsdDocument HTTP/1.1
610     Host: example.org
611     Content-Type: multipart/related;boundary="====1605871705===";
612     type="application/atom+xml"
613     MIME-Version: 1.0
614
615     -----1605871705==
616     Content-Type: application/atom+xml; charset="utf-8"
617     MIME-Version: 1.0
618
619     <?xml version="1.0"?>
620     <entry xmlns="http://www.w3.org/2005/Atom"
621         xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
622         <id>urn:uuid:aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a</id>
623         <updated>2009-05-26T13:13:55.014+02:00</updated>
624         <title type="text">accountingTypes.xsd</title>
625         <author>
626             <name>Bellwood</name>
627         </author>
628         <summary type="text">accountingTypes.xsd schema document</summary>
629
630         <!--
631             S-RAMP defined categorizations identifying class of data represented by
632             this entry
633         -->
634         <category term="XsdDocument" label="XML Schema"
635             scheme="urn:x-s-ramp:2013urn:x-s-ramp:2013:type" />
636         <s-ramp:artifact xsi:schemaLocation="http://docs.oasis-open.org/s-ramp/ns/s-
637 ramp-v1.0
638             http://s-ramp.org/2010/specification/schemas/xsdmodel.xsd"
639             xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-
640 v1.0"
641             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
642         <s-ramp:XsdDocument name="accountingTypes.xsd"
643             description="accountingTypes.xsd schema document"
644             createdBy="Bellwood" version="1.0"
645             uuid="aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a "
646             createdTimestamp="2009-05-26T13:13:55.013+02:00"
647             lastModifiedTimestamp="2009-06-26T13:13:55.013+02:00"
648             lastModifiedBy="Bellwood" contentEncoding="UTF-8"
649             contentType="application/xml" contentsize="4096" >
650         <s-ramp:classifiedBy>
651             http://example.org/ontologies/accounting.owl/accounts
652         </s-ramp:classifiedBy>
653         <s-ramp:property>
654             <propertyName>accountingCalendar</propertyName>
655             <propertyValue>2009</propertyValue>
656         </s-ramp:property>
657     </s-ramp:XsdDocument>

```

```
658     </s-ramp:artifact>
659 </entry>
660 -----1605871705==
661 Content-Type: application/xml
662 MIME-Version: 1.0
663
664 {XML content of accountTypes.xsd document goes here}
665 -----1605871705===--
666
```

667 Non-document artifacts are published directly using HTTP POST of the desired Artifact Entry document to
668 the appropriate S-RAMP collection and do not use the Atom Multi-part POST method.

669 *Note:* Derived Artifacts cannot be published (created or deleted) directly, as these are automatically
670 generated and managed by the server as part of the publication of the document to which the
671 corresponding Derived Artifact Model and all its constituent artifacts apply.

672 2.3.5.2 Publishing Multiple Artifact Entries

673 Many document types include or import other documents upon which they are dependent. For example,
674 XSD document A.xsd imports XSD document B. Since S-RAMP repositories require that all dependencies
675 be resolvable at the time of publication, allowing only the publication of one artifact at a time would mean
676 that the dependent document B.xsd would need to be published first, followed by publication of A.xsd, so
677 that the server could resolve this dependency. But this is clumsy, inefficient and potentially difficult for
678 clients to manage. To simplify publication of such documents for clients, this section discusses two
679 methods for publishing multiple resources (documents) and their associated metadata which S-RAMP
680 compliant repositories SHALL support. Each method can support documents with nested dependencies
681 since all dependent documents are published in a single step. All dependencies MUST be resolvable at
682 the time of publication.

683 2.3.5.2.1 Using Batch POST

684 RFC 2387, "The MIME Multipart/Related Content-type" describes how to perform a multipart POST of
685 binary documents. S-RAMP builds on this RFC to extend the IETF draft document "ATOM Multi-part
686 Media Resource Creation" to support the simultaneous publishing of a collection of non-dependent and
687 dependent resources and their associated metadata. With this approach it is possible, for example, to
688 publish an XSD document which imports two other XSD documents by including it as well as its
689 dependencies in a single POST to the repository.

690 The Batch POST method requires an atom entry document as the root body part. This root atom entry
691 document points to any dependent atom entry documents contained in other body parts using atom:link
692 elements with an *href* attribute whose value is the Content-ID of the relevant body part. It also points at
693 the actual document content by specifying the Content-ID of the relevant body part as the value of the *src*
694 attribute in the atom:content element of the atom entry document. This approach provides complete
695 linkage between a document and its dependencies and all corresponding binary resources.

696 The syntactic structure of the Batch POST method is illustrated in Example 7 which publishes an A.xsd
697 document which has a dependency on the B.xsd document. Notable items specific to the structure are in
698 bold:

699

700 *Example 7 - Batch Post Construct Example*

```
701     POST /s-ramp/xsd HTTP/1.1
702     Host: example.org
703     Content-Length: nnnn
704     Content-Type: multipart/related;version 1.1;msgtype=request;boundary=example-
705 bound;type="application/atom+xml;type=entry";start="<12@example.org>"
706     slug: The Beach
707     MIME-Version: 1.0
```



```

708
709  --example-bound
710  Content-Type: application/atom+xml;type=entry
711  Content-ID: <12@example.org>
712
713  <entry xmlns="http://www.w3.org/2005/Atom"
714    xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
715    <id>urn:uuid:aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a</id>
716    <updated>2009-05-26T13:13:55.014+02:00</updated>
717    <title type="text">A.xsd</title>
718    <published>2009-05-26T13:13:55.013+02:00</published>
719    <author>
720      <name>Bellwood</name>
721    </author>
722    <contributor>
723      <name>Pospisil</name>
724    </contributor>
725    <summary type="text">A.xsd schema document</summary>
726    <link href=" cid:56@example.org" type="application/atom+xml;type=entry"
727      rel="related" title="Imported XSD"/>
728    <content src=" cid:34@example.org"/>
729    <!--
730      S-RAMP defined categorizations identifying class of data represented by
731      this entry
732    -->
733    <category term="XsdDocument" label="XML Schema"
734      scheme="urn:x-s-ramp:2013:type" />
735    <s-ramp:artifact
736      ...rest of artifact definition goes here...
737    </ s-ramp:artifact>
738  </entry>
739
740  --example-bound
741  Content-Type: application/xml
742  Content-Description: The root XSD document
743  Content-Transfer-Encoding: base64
744  Content-ID: <34@example.org>
745
746  ...XML content for A.xsd...
747
748  --example-bound
749  Content-Type: application/atom+xml;type=entry
750  Content-ID: <56@example.org>
751
752  <entry xmlns="http://www.w3.org/2005/Atom"
753    xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
754    <id>urn:uuid:bbbbbbbbb-bbbb-bbbb-bbbb-bbbbbbbbb6b</id>
755    <title type="text">B.xsd</title>
756    <content src=" cid:78@example.org"/>
757    ...rest of entry document goes here...
758  </entry>

```

```

759
760     --example-bound
761     Content-Type: application/xml
762     Content-Description: The imported XSD document
763     Content-Transfer-Encoding: base64
764     Content-ID: <78@example.org>
765
766     ...XML content for B.xsd...
767
768     --example-bound--
769

```

770 The Batch POST method is intended to support the publication of new documents which have
771 dependencies, although it can also include non-document artifacts. Such artifacts will not have a
772 corresponding binary section in the body. S-RAMP servers SHALL process the entire encapsulated
773 payload in a Batch POST as a group and SHALL perform any dependent processing necessary. All parts
774 of the body SHOULD be published successfully, or the entire request SHOULD fail and be rolled back.
775 However, rolling back the entire batch POST on a failed request is implementation specific as some
776 implementations MAY choose to create only those artifacts which are valid, thus doing a partial create
777 from the Batch POST. Regardless of whether a failure results in a complete rollback or a partial create,
778 the implementation MUST return a failure response and in the body of the response provide an
779 explanation of the failure.

780 This specification does define any limits to number or size of artifacts that can be included in a batch
781 POST request. An S-RAMP implementation can decide to reject the request if it determines that the
782 request cannot be processed for some reason, for example, due to constrained resource. In such a case
783 the server must report failure as stated above.

784 The response from a Batch POST in S-RAMP SHALL provide a return code which indicates success or
785 failure. A successful response MUST be an HTTP 200 OK and a failure response MUST be an HTTP 409
786 Conflict.

787 In the case of a successful response from the "encapsulating" HTTP POST the response would contain a
788 set of boundary delineated HTTP responses, which in this example would be a set of boundary
789 delineated Media Link Entries corresponding to the two XSD files which were published.

790
791 *Example 8 - Successful Batch POST Response*

```

792     HTTP/1.1 200 OK
793     Date: Tues, 26 May 2009 13:13:55 GMT+02:00
794     Content-Length: 1234
795     Content-Type: multipart/mixed; boundary=batch
796     Mime-Version: 1.0
797
798     --batch
799     Content-ID: <aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaa12@example.org>
800     Content-Type: message/http; version=1.1;msgtype=response
801
802     HTTP/1.1 201 Created
803     Date: Tues, 26 May 2009 13:13:55 GMT+02:00
804     Content-Type: application/atom+xml;type=entry
805
806     {Updated Atom:entry for Resource A omitted for clarity}
807
808     --batch
809     Content-ID: <aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaa67@example.org>

```

810 Content-Type: message/http; version=1.1;msgtype=response

811

812 HTTP/1.1 201 Created

813 Date: Tues, 26 May 2009 13:13:55 GMT+02:00

814 Content-Type: application/atom+xml;type=entry

815

816 {Updated Atom:entry for Resource B omitted for clarity}

817

818 --batch--

819

820 In the case where the Batch POST was unsuccessful and the server rolled back the entire Batch POST
821 request, the unsuccessful response from the "encapsulating" HTTP POST would be an HTTP 409. The
822 response would contain an explanation of the error with enough information to allow the user to recognize
823 the conflict. Ideally, the information provided would also allow the user to fix the conflict, however this
824 MAY not always be possible.

825

826 *Example 9 - Failed Batch POST Response – Complete Rollback*

827 HTTP/1.1 409 Conflict

828 Date: Tues, 26 May 2009 13:13:55 GMT+02:00

829 Content-Length: 520

830 Content-Type: multipart/mixed; boundary=batch

831 Mime-Version: 1.0

832

833 --batch

834 Content-ID: <aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa67@example.org>

835 Content-Type: message/http; version=1.1;msgtype=response

836

837 HTTP/1.1 409 Conflict

838 Date: Tues, 26 May 2009 13:13:55 GMT+02:00

839

840 {Explanation of error condition. For details see appendix H}

841

842 --batch--

843

844 In the case where the Batch POST request was unsuccessful and the server chose to do a partial create
845 of the Batch POST request, the unsuccessful response from the "encapsulating" HTTP POST would be
846 an HTTP 409. The response would contain an explanation of the error with enough information to allow
847 the user to recognize the content of the content that were successfully published and the content that was
848 not successful. Ideally, the information provided would also allow the user to fix the conflict, however this
849 MAY not always be possible.

850

851 *Example 10 - Failed Batch POST Response – Partial Create*

852 HTTP/1.1 409 Conflict

853 Date: Tues, 26 May 2009 13:13:55 GMT+02:00

854 Content-Length: 520

855 Content-Type: multipart/mixed; boundary=batch

856 Mime-Version: 1.0

857

858 --batch

859 Content-ID: <aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa12@xample.org>

860 Content-Type: message/http; version=1.1;msgtype=response
861
862 HTTP/1.1 201 Created
863 Date: Tues, 26 May 2009 13:13:55 GMT+02:00
864 Content-Type: application/atom+xml;type=entry
865
866 {Updated Atom:entry for Resource A omitted for clarity}
867
868 --batch
869 Content-ID: <aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa67@example.org>
870 Content-Type: message/http; version=1.1;msgtype=response
871
872 HTTP/1.1 409 Conflict
873 Date: Tues, 26 May 2009 13:13:55 GMT+02:00
874
875 {Explanation of error condition. For details see appendix H}
876
877 --batch--
878

879 2.3.5.2.2 Package Publishing Using POST of Package File

880 This method of publishing to an S-RAMP compliant repository can provide significant convenience for
881 clients. The behavior described herein represents some departure from the APP specification in order to
882 support the REQUIRED capabilities, but wherever possible, it adheres to its spirit.

883 To publish using an S-RAMP package file, one simply performs an HTTP POST of a package file
884 containing the desired media resources and/or atom:entry documents to the appropriate S-RAMP
885 collection URI.

886 2.3.5.2.2.1 Package File Format

887 The format of An S-RAMP package file is based on ZIP file format and is used to aggregate a number of
888 files into one. The internal structure and publishing of the package file has following characteristics:

- 889 • The package file can have arbitrary internal structure (e.g., flat, or organized using folders as
890 desired). S-RAMP uses the UUID of an artifact as its official identifier rather than any user
891 supplied file structure. S-RAMP repositories MAY, but are not REQUIRED to, also utilize or retain
892 client supplied file structure information.
- 893 • The package file can contain
 - 894 ○ Media resources (e.g., XSD documents)
 - 895 ○ Media Link Entries that correspond to media resources which contain associated
896 metadata for those resources
 - 897 ○ Plain atom:entry file(s) (.atom files in the zip) representing nondocument based repository
898 artifacts.
- 899 • The convention for associating metadata with a resource in a package file is to append ".atom" to
900 its file-id for the Artifact Entry document. For example, for the resource "/somedir/somename.ext",
901 the file with the name "/somedir/somename.ext.atom" corresponds to its Atom Media Link Entry
902 containing the metadata associated with the document.
- 903 • When published, the S-RAMP server processes the package file, extracting the constituent files.
904 Each resource is published, together any associated Media Link Entry and any other Artifact
905 Entry documents provided.
- 906 • An S-RAMP repository does not persist or publish the package file itself after it has been
907 processed.

- 908 • All S-RAMP artifacts require a unique uuid property (in their s-ramp:artifact section) and atom:id.
909 Resources which lack corresponding Media Link Entries, or those whose Media Link Entry files
910 have no user specified uuid property and atom:id, or a user specified uuid property and atom:id
911 which does not already exist in the repository, are considered new and treated as if they were
912 published using a POST.
- 913 • Artifact Entry documents having a user supplied uuid property and atom:id which already exists in
914 the repository, are treated as if they were PUT, which causes the existing artifact to be replaced
915 with the copy in the package file.
- 916 • As discussed in Section 2.3.2 of the Foundation Document, documents which have a Derived
917 Model associated with them cannot be updated in the repository. They MUST be removed and
918 republished. The implicit updates of documents which have a Derived Model is, therefore, not
919 supported when publishing a ZIP file. Attempts to perform an update on such a document will
920 result in the HTTP error "403" Forbidden.

921
922 Package files are published to the /s-ramp S-RAMP URI space. The basic syntax for publication using S-
923 RAMP package files is then:

```
924     POST /s-ramp HTTP/1.1
925     Host: example.org
926     Content-Type: application/zip
927     Content-Length: 1234
928
929     { ...binary S-RAMP package data...}
930
```

931
932 The S-RAMP server will process the component files in the package file, placing each in the correct S-
933 RAMP collection in the S-RAMP URI space, based upon introspection of the files.

934 The response returned by an S-RAMP server as a result processing the POST of an S-RAMP package
935 file is the same as that returned as a result of processing an HTTP Batch POST, in other words:

```
936     multipart/mixed;boundary=package
937
```

938
939 which consists of a set of boundary delineated HTTP responses, one for each Atom:entry document
940 which was in the package file, as well as an Atom Media Link Entry document for each media resource in
941 the package (clients MAY provide these documents together with their corresponding media resource
942 documents in the package, or let the server create them and then update them in a second step). The
943 content of each entry document MAY have been altered by the server during publication (e.g., date/time
944 of the update, setting of the ID, if not provided by in the request, etc.) The client is responsible for noting
945 any updates which have been made. The Content-ID header is set to full path (without leading '/') of
946 resource or Atom:entry document in the package concatenated with '@package'.

947
948 For example, assume we wish to publish a set of related resources: a.wsdl, which has import statements
949 for b.xsd and c.xsd. The content of the MyFiles.zip is as follows:

- 950 • /a.wsdl
- 951 • /a.wsdl.atom
- 952 • /b.xsd
- 953 • /b.xsd.atom
- 954 • /c.xsd
- 955 • /c.xsd.atom

956 Upon receipt of a POST request, the S-RAMP server processes the S-RAMP package file and
957 instantiates an S-RAMP artifact in the repository for each of the three documents, which in the Atom
958 Binding are represented with Media Link Entries with document content elements for the a.wsdl, b.xsd
959 and c.xsd media resources respectively. The user supplied Media Link Entry files extracted from the
960 package are used to create these S-RAMP artifacts.

961

962 Example 11 below provides a response for the above package file scenario. It omits substantial content
963 for brevity:

964

965 *Example 11 - Response from Publish using S-RAMP Package File Method*

966 HTTP/1.1 200 OK

967 Date: Tues, 26 May 2009 13:13:55 GMT+02:00

968 Content-Length: 1234

969 Content-Type: multipart/mixed; boundary=package

970 Mime-Version: 1.0

971

972 --package

973 Content-ID: <c.xsd@packaga>

974 Content-Type: message/http; version=1.1;msgtype=response

975

976 HTTP/1.1 201 Created

977 Date: Tues, 26 May 2009 13:13:55 GMT+02:00

978 Content-Type: application/atom+xml;type=entry

979

980 {Created/updated Atom:entry for file /c.xsd omitted for clarity}

981

982 --package

983 Content-ID: <b.xsd@packaga>

984 Content-Type: message/http; version=1.1;msgtype=response

985

986 HTTP/1.1 201 Created

987 Date: Tues, 26 May 2009 13:13:55 GMT+02:00

988 Content-Type: application/atom+xml;type=entry

989

990 {Created/updated Atom:entry for file /b.xsd omitted for clarity}

991

992 --package

993 Content-ID: <a.wsdl@package>

994 Content-Type: message/http; version=1.1;msgtype=response

995

996 HTTP/1.1 201 Created

997 Date: Tues, 26 May 2009 13:13:55 GMT+02:00

998 Content-Type: application/atom+xml;type=entry

999

1000 {Created/updated Atom:entry for file /a.wsdl omitted for clarity}

1001

1002 --package--

1003

1004 As with the HTTP Batch approach described earlier, ALL operations implied by the package file contents
1005 MUST succeed in order for ANY of them to succeed. If any one fails, the entire package request is rolled

1006 back. However, rolling back the entire package on a failed request is implementation specific as some
1007 implementations MAY choose to create only those artifacts which are valid, thus doing a partial create
1008 from the package POST. Regardless of whether a failure results in a complete rollback or a partial create,
1009 the implementation MUST return a failure response and in the body of the response provide an
1010 explanation of the failure.

1011 This specification does define any limits to number or size of artifacts that can be included in a batch
1012 POST request. An S-RAMP implementation can decide to reject the request if it determines that the
1013 request cannot be processed for some reason, for example, due to constrained resource. In such a case
1014 the server must report failure as stated above.

1015 The unsuccessful response from the "encapsulating" HTTP POST would be an HTTP 409. The response
1016 would contain an explanation of the error with enough information to allow the user to recognize the
1017 conflict. Ideally, the information provided would also allow the user to fix the conflict, however this MAY
1018 not always be possible.

1019

1020 *Example 12 – Error Response from Publish using S-RAMP Package File Method*

```
1021 HTTP/1.1 409 Conflict
1022 Date: Tues, 26 May 2009 13:13:55 GMT+02:00
1023 Content-Length: 1234
1024 Content-Type: multipart/mixed; boundary=package
1025 Mime-Version: 1.0
1026
1027 --package
1028 Content-ID: a.wsd1@package
1029 Content-Type: message/http; version=1.1;msgtype=response
1030
1031 HTTP/1.1 409 Conflict
1032 Date: Tues, 26 May 2009 13:13:55 GMT+02:00
1033
1034 {Explanation of error condition. For details see appendix H}
1035
1036 --package--
```

1037 **2.3.5.3 Retrieving Repository Artifacts**

1038 HTTP GET is used to retrieve Artifact Entries. Entry documents can be retrieved individually by
1039 performing an HTTP GET against a member resource URI, or as a feed of Entry Documents by
1040 performing an HTTP GET against a collection. Requests to retrieve an Artifact Entry document from the
1041 incorrect Artifact Type Model will result an HTTP "404" Not Found.

1042 Several examples are provided below using this sample URL format:

1043

1044 Format to request a specific Artifact Entry document:

1045

```
1046 GET /s-ramp/{artifactModel}/{artifactType}/{uuid:artifact} HTTP/1.1
1047 Host: example.org
```

1048

1049 Format to request document content:

1050

```
1051 GET /s-ramp/{artifactModel}/{artifactType}/{uuid:artifact}/media HTTP/1.1
1052 Host: example.org
```

1053

1054 Format to request a feed of summary Artifact Entry documents from an artifact collection which is
1055 defined in the Service Document:

1056
1057 GET /s-ramp/{artifactCollection} HTTP/1.1
1058 Host: example.org

1059
1060 The examples below illustrate several requests:

1061
1062 To request a specific XsdDocument full Artifact Entry:

1063
1064 GET /s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a HTTP/1.1
1065 Host: example.org

1066
1067 To request a specific XsdDocument content itself:

1068
1069 GET /s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a/media
1070 HTTP/1.1
1071 Host: example.org

1072
1073 To request a feed of all Media Link Entry documents corresponding to each XSD document
1074 instance in the repository, in summary form:

1075
1076 GET /s-ramp/xsd/XsdDocument HTTP/1.1
1077 Host: example.org

1078 To request a feed of all Service Implementation Model entries representing Organization artifact
1079 instances, in summary form:

1080
1081 GET /s-ramp/serviceImplementation/Organization HTTP/1.1
1082 Host: example.org

1083
1084 Notes:

- 1085 • Requesting a feed of summary Artifact Entries for an entire collection can result in large result
1086 sets. The Atom defined mechanism for paging of feeds SHALL be supported by S-RAMP
1087 servers.
- 1088 • The HTTP allow header can be used to indicate the HTTP methods which can be executed
1089 against the request URI.

1090

1091 **2.3.5.3.1 Resolving Internal References**

1092 Some documents have dependencies on other documents. From our earlier scenario, a WSDL document
1093 might import an XSD document. When the WSDL document was published to the repository, the XSD
1094 document would have been published as well, but the original WSDL document published would not
1095 typically have the correct import statement to reference the XSD file which is actually persisted in the
1096 repository. More likely, it would be a local reference based on the development environment from which
1097 both files originated. It is important that the repository support the capability for tooling to resolve these
1098 dependencies to the correct version of the document which is actually published in the repository. S-
1099 RAMP provides two implementation choices for achieving this. Compliant implementations SHALL
1100 support at least one of them.

- 1101 1. When returning a requested document, the repository MAY dynamically re-write the document's
1102 references to other documents to correctly reference the appropriate file(s) stored in the
1103 repository. The actual document stored in the repository is unaffected. Only the serialized content
1104 returned in a GET is altered as the content is returned.
- 1105 2. When the URL of a document containing dependent links is requested, the repository MAY
1106 instead redirect to another URL of its own creation, against which all relative path references to
1107 dependent documents will resolve properly with the redirected URL as a base. S-RAMP does not
1108 describe how such an implementation is done, only the REQUIRED behavior when
1109 implementations choose to support this option.

1110 Regardless of which option is implemented, its activation is achieved by adding the type=relative query
1111 string to the URL in the request (a sample URL format is shown here):

1112
1113 /s-ramp/<artifactModel>/<artifactType>/{uuid:artifactDocument}/media?type=relative

1114 2.3.5.4 Editing an Artifact Entry

1115 Example 5 - Updating an Initial Media Link Entry with Metadata already described how to perform an
1116 update of an existing Atom entry, and as described in Section 2.3.5.2, it is also possible to use PUT as
1117 part of a multi-entry update using either the HTTP Batch or S-RAMP package publishing methods
1118 supported in S-RAMP. When the update uses the HTTP Batch technique described in Section 2.3.2, the
1119 boundary delineated section applying to the artifact to be updated simply uses a PUT instead of a POST.
1120 When the update is requested as part of a POST using a package file as described in Section 2.3.5.2.2,
1121 the PUT is implicit. All package file publishing is done using a POST of the S-RAMP package file, but
1122 when an individual entry within the package file references an Artifact Entry uuid property which already
1123 exists in the repository; it is treated as an update during processing.

1124 2.3.5.5 Deleting an Artifact Entry

1125 Deletion of an Artifact Entry is accomplished using HTTP DELETE, with this sample syntax:

1126
1127 To delete an Artifact Entry (for media link entries, this implicitly deletes the corresponding content
1128 as well):

1129
1130 DELETE /s-ramp/{artifactModel}/{artifactType}/{uuid:artifact} HTTP/1.1
1131 Host: example.org

1132
1133 To delete content (this implicitly deletes the corresponding media link entry as well):

1134
1135 DELETE /s-ramp/{artifactModel}/{artifactType}/{uuid:artifact}/media HTTP/1.1
1136 Host: example.org

1137
1138 The above URL for the entry with the indicated Artifact Entry uuid property MUST already resolve to an
1139 existing Artifact Entry in the repository in order for it to be deleted. All artifacts not belonging to a Derived
1140 Model can be deleted by a client. The deletion of an Artifact Entry also removes all of its relationships.
1141 Additional information on how this affects reporting of backward relationship feeds in the Fine Grained
1142 View can be found in Section 2.4.1.7.

1143 If an artifact is deleted which is the Target Entry of a relationship instance owned by some other Source
1144 Entry, then that relationship instance is also deleted in that Source Entry.

1145 The following HTTP errors MAY be returned:

- 1146
- 404 Not Found (no matching Artifact Entry uuid property in repository)
 - 403 Forbidden (returned in response to an attempted delete of a Derived Artifact)
- 1147

1148 2.4 Fine Grained Views

1149 The three Fine Grained Views in the Atom Binding for S-RAMP provide a mechanism for working with
1150 each of the three classes of S-RAMP metadata: relationships, properties and classifications. While S-
1151 RAMP compliant implementations MAY choose to implement none, any, or all of these features,
1152 implementation of any of these features SHALL conform with the applicable interfaces described in these
1153 sections. Updates to metadata items using the Fine Grained Views implicitly changes the Artifact Entry
1154 which owns the metadata.

1155 2.4.1 S-RAMP Relationships

1156 S-RAMP models relationship metadata in the Atom Binding as resources in order to facilitate their
1157 manipulation separately from the Atom Source Entry which manages them. This is particularly useful
1158 when many relationships are present, since the fine-grained method allows manipulation of one
1159 relationship at a time without having to explicitly update the Atom Source Entry with which it is associated.
1160 Several concepts and terms are useful:

- 1161 • A **relationship** is a concept that represents an association between a single Source Entry and a
1162 single Target Entry (each of which represents an S-RAMP artifact). Relationships are modeled in
1163 Atom with a **Relationship Entry**, which is an Atom entry document describing the relationship,
1164 including links to the Source Entry and Target Entry, the **Relationship Type**, as well as specific
1165 categorizations which are described later that provide relevant metadata pertaining to the
1166 Relationship Entry.
- 1167 • A **Relationship Type** is a name which describes the purpose or meaning of that relationship
1168 (e.g., "includedXsds" or "similarXsds"). There can be more than one relationship having the
1169 same **Relationship Type**. A **Relationship Type Entry** is an Atom entry document which
1170 describes a particular set of Relationship Types. It contains a link to the applicable Relationship
1171 feed as well as specific categorizations which are described later that provide relevant metadata
1172 pertaining to the Relationship Type Entry.
- 1173 • A **Backward Relationship Feed** is a special kind of feed whose members are Relationship Entry
1174 documents. Links to such *backwardRelationships* feed(s) are placed in the Target Entry
1175 document corresponding to the relationship target represented by a *modeled* or *derived*
1176 relationship's Target Entry link. These feeds are provided for the convenience of clients to
1177 simplify artifact navigation. Additional information is provided in Section 2.4.1.1.

1178 2.4.1.1 Relationship Feeds

1179 S-RAMP defines several Atom feeds which are used to access fine-grained support for relationships.
1180 These allow clients to retrieve details about each of the Relationship Types and instances associated with
1181 the Artifact Entry having these relationships. When a server implementation supports the Fine Grained
1182 View for relationships, the Atom entry document representing the relationship's source artifact (Source
1183 Entry) SHALL contain links to the following Atom feed(s):

- 1184 • Link to the **relationships** feed of all relationships. Resolving the link to this feed will return a feed
1185 of summary Relationship Entries for every relationship instance owned by the Source Entry,
1186 regardless of the Relationship Type associated with each. This **feed** link SHALL have a *rel*
1187 attribute as follows:

```
1188 rel="urn:x-s-ramp:2013:relationships"
```

1189
1190 For example:

```
1191  
1192  
1193 <link title="All Relationships"  
1194 href="http://example.org/s-ramp/xsd/XsdDocument/  
1195 {uuid:source.xsd}/relationships"  
1196 type="application/atom+xml;type=feed"  
1197 rel="urn:x-s-ramp:2013:relationships" />  
1198
```

- 1199
- 1200
- 1201
- 1202
- 1203
- 1204
- Link to the **relationshipTypes** feed of all Relationship Types (e.g., "includedXsds", "similarXsds"). Resolving the link to this feed will return a feed of all the summary Relationship Type Entries. There is only one such entry in this feed for each Relationship Type represented across the entire set of relationship instances owned by the Source Entry. This **feed** link SHALL have a *rel* attribute as follows:

1205

```
rel="urn:x-s-ramp:2013:relationshipTypes"
```

1206

1207 For example:

1208

1209

```
<link title="Relationships Types"
1210 href="http://example.org/s-ramp/xsd/XsdDocument/
1211 {uuid:source.xsd}/relationshipTypes"
1212 type="application/atom+xml;type=feed"
1213 rel="urn:x-s-ramp:2013:relationshipTypes" />
```

1214

- 1215
- Link(s) to individual **relationships** feed(s) of all relationships sharing a particular Relationship Type. These link(s) will appear in the Source Artifact corresponding to each Relationship Type Entry. These links provide a means to retrieve Relationship Entry documents owned by the Source Entry corresponding to a particular Relationship Type. These **feed** links SHALL have a *rel* attribute of the following form:

1220

```
rel="urn:x-s-ramp:2013:relationships:{Relationship Type}"
```

1222

1223 For example:

1224

1225

```
<link title="Relationships of type includedXsds"
1226 href="http://example.org/s-ramp/xsd/XsdDocument/
1227 {uuid:source.xsd}/relationships/includedXsds"
1228 type="application/atom+xml;type=feed"
1229 rel="urn:x-s-ramp:2013:relationships:includedXsds" />
```

1230

1231 While all S-RAMP relationships are unidirectional, the S-RAMP Atom Binding provides a convenience

1232 feature supported for at least all *modeled* and *derived* relationships which facilitates a client's ability to

1233 navigate backward from the Target Entry of a given relationship to the Source Entry, without the need for

1234 performing a query to discover the Source Entry side of a relationship instance. As noted in Section 2.4.1

1235 above, these are called Backward Relationship Feeds.

1236 A *backwardRelationships* feed is defined corresponding to each *relationships* feed (including those for

1237 specific Relationship Types). These *backwardRelationships* feed(s) are created by the server by placing

1238 the same Relationship Entry that occurs in the *relationships* feed of the relationship's Source Entry into

1239 the corresponding *backwardRelationships* feed of the Target Entry referenced by that relationship. In

1240 addition, applicable Relationship Type specific *backwardRelationships/{Relationship Type}* feeds are

1241 also present in the Target Entry. Clients can use these feeds to navigate backward using the Source

1242 Entry link associated with a relationship's Target Entry.

1243 Backward Relationship Feeds are read-only. Clients cannot add or remove a Relationship Type Entry

1244 documents from any *backwardRelationships* feed. The S-RAMP server provides these feeds in the

1245 Target Entry serialization only as a convenience to clients. All relationships are still managed from the

1246 applicable Source Entry's relationship feed links.

1247 The Target Entry document representing the relationship's target artifact SHALL contain links to the

1248 following Atom feed(s):

- 1249
- Link to the **backwardRelationships** feed. Resolving this link will return a feed of summary Relationship Entry documents for at least every *modeled* and *derived* kind of relationship instance, regardless of its Relationship Type, for which the Target Entry link resolves to this Artifact Entry. Note that this feed will be empty if there is no Source Entry with a relationship having this Artifact Entry as its target. This **feed** link SHALL have a *rel* attribute of the following
- 1250
- 1251
- 1252
- 1253

1254 form:
1255
1256 rel="urn:x-s-ramp:2013:backwardRelationships"

1257
1258 For example:

```
1259 <link title="Back Links from this Target Entry for all Relationship Types"  
1260 href="http://example.org/s-ramp/xsd/XsdDocument/  
1261 {uuid:foo.xsd}/backwardRelationships"  
1262 type="application/atom+xml;type=feed"  
1263 rel="urn:x-s-ramp:2013:backwardRelationships" />  
1264  
1265
```

- 1266 • Link(s) to individual Relationship Type specific *backwardRelationships*{Relationship Type}
1267 feed(s) of all relationships sharing a particular Relationship Type, and whose targets are this
1268 Artifact Entry. These provide a means to retrieve all Relationship Entry documents of a particular
1269 Relationship Type for use in backward navigation to the source side of each such relationship.
1270 These **feed** links SHALL have a *rel* attribute of the following form:

```
1271  
1272 rel="urn:x-s-ramp:2013:backwardRelationships:{Relationship Type}"
```

1273
1274 For example:

```
1275  
1276 <link title="Back Links for serviceImplementation model relationships of type  
1277 hasServiceEndpoint"  
1278 href="http://example.org/s-ramp/serviceImplementation/ServiceEndpoint/  
1279 {uuid:target.xsd}/backwardRelationships/hasServiceEndpoint"  
1280 type="application/atom+xml;type=feed"  
1281 rel="urn:x-s-ramp:2013:backwardRelationships:hasServiceEndpoint" />  
1282
```

1283 Example 13 below illustrates both *relationships* and *backwardRelationships* feeds using three types of
1284 relationships (*derived*, *modeled* and *generic*). It uses the same summary Atom (media link) entry which
1285 corresponds to the "accountingTypes" XSD document artifact found in Section 2.3.2, except that this
1286 version includes the necessary links to support the Fine Grained View for Relationships. To illustrate all
1287 these features, let's add three more XSD documents, and an S-RAMP Service Implementation Model
1288 ServiceInstance artifact called "myServiceInstance" which has a "describedBy" relationship to the
1289 "accountingTypes" document.

1290
1291 *Example 13 - Complex Relationship Scenario Summary Entry*

1292 Assume a set of four documents:

- 1293 1. "customer.xsd", which includes
- 1294 2. "accountingTypes.xsd", which includes
- 1295 3. "dataTypes.xsd"
- 1296 4. "related.xsd" (a similar schema which is related to "accountingTypes.xsd")

1297
1298
1299 The following relationships exist among the Artifact Entries representing each of these four document
1300 artifacts and the Service Implementation Model artifact:

- 1301
1302 ○ Relationship 1:
 - 1303 ▪ Type: "includedXsds"
 - 1304 ▪ Kind: "derived"
 - 1305 ▪ Source: "customer.xsd" entry

- 1306 ▪ Target: "accountingTypes.xsd" entry
- 1307 ○ Relationship 2:
 - 1308 ▪ Type: "includedXsds"
 - 1309 ▪ Kind: "derived"
 - 1310 ▪ Source: "accountingTypes.xsd" entry
 - 1311 ▪ Target: "dataTypes.xsd" entry
- 1312 ○ Relationship 3:
 - 1313 ▪ Type: "similarXsds"
 - 1314 ▪ Kind: "generic"
 - 1315 ▪ Source: "accountingTypes.xsd" entry
 - 1316 ▪ Target: "related.xsd" entry
- 1317 ○ Relationship 4:
 - 1318 ▪ Type: "describedBy"
 - 1319 ▪ Kind: "modeled"
 - 1320 ▪ Source: "myServiceInstance" entry
 - 1321 ▪ Target: "accountingTypes.xsd" entry

1322

The summary Source Entry below corresponding to the "accountingTypes.xsd" artifact has relationship instances for the "includedXsds" and "similarXsds" Relationship Types associated with it, and thus exposes the various feeds shown:

1326

```

1327 <entry xmlns="http://www.w3.org/2005/Atom"
1328       xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
1329   <id>urn:uuid:aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a</id>
1330   <updated>2009-05-26T13:13:55.013+02:00</updated>
1331   <title type="text">accountingTypes.xsd</title>
1332   <published>2009-05-26T13:13:55.013+02:00</published>
1333   <author>
1334     <name>Bellwood</name>
1335   </author>
1336   <contributor>
1337     <name>Pospisil</name>
1338   </contributor>
1339   <summary type="text">accountingTypes.xsd schema document</summary>
1340   <content type="application/xml"
1341     src="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1342     aaaaaaaaaa6a/media"/>
1343     <link type="application/atom+xml;type=entry" rel="self"
1344       href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1345       aaaaaaaaaa6a" />
1346     <link type="application/atom+xml;type=entry" rel="edit-media"
1347       href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1348       aaaaaaaaaa6a/media" />
1349     <link type="application/atom+xml;type=entry" rel="edit"
1350       href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1351       aaaaaaaaaa6a" />
1352   <!--
1353     Link to relationships feed of all Relationship entries over all
1354     Relationship Types. In this example, that includes Relationships #2 and #3
1355     above.
1356   -->

```

```

1357     <link title="All Relationships"
1358           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1359 aaaaaaaaaa6a/relationships"
1360           type="application/atom+xml;type=feed"
1361           rel="urn:x-s-ramp:2013:relationships" />
1362
1363     <!--
1364         Link to backwardRelationships feed containing all Relationship Entries of
1365         any Relationship Type whose target is this Artifact Entry. In this example,
1366         that includes Relationships #1 and #4 above.
1367     -->
1368     <link title="All Backward Relationship Targets"
1369           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1370 aaaaaaaaaa6a/backwardRelationships"
1371           type="application/atom+xml;type=feed"
1372           rel="urn:x-s-ramp:2013:backwardRelationships" />
1373
1374     <!--
1375         Link to relationshipTypes feed of Relationship Type Entries. In this
1376         example, that means entries for the "includedXsds" and "similarXsds"
1377         Relationship Types.
1378     -->
1379     <link title="All Relationship Types"
1380           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1381 aaaaaaaaaa6a/relationshipTypes"
1382           type="application/atom+xml;type=feed"
1383           rel="urn:x-s-ramp:2013:relationshipTypes" />
1384
1385     <!--
1386         Link to feed of all Relationship Entries whose Relationship Type =
1387         "includedXsds"
1388     -->
1389     <link title="All includedXsds Type Relationships"
1390           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1391 aaaaaaaaaa6a/relationships/includedXsds"
1392           type="application/atom+xml;type=feed"
1393           rel="urn:x-s-ramp:2013:relationships:includedXsds" />
1394     <!--
1395         Link to backwardRelationships feed containing all Relationship Entries of
1396         Relationship Type = "includedXsds", whose corresponding Target Entry is
1397         this Artifact Entry. In this example, that includes only Relationship #1
1398         above.
1399     -->
1400     <link title="All includedXsds Backward Relationship Targets"
1401           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1402 aaaaaaaaaa6a/backwardRelationships/includedXsds"
1403           type="application/atom+xml;type=feed"
1404           rel="urn:x-s-ramp:2013:backwardRelationships:includedXsds" />
1405
1406     <!--
1407         Link to backwardRelationships feed containing all Relationship Entries of
1408         Relationship Type = "describedBy", whose corresponding Target Entry is this
1409         Artifact Entry. In this example, that includes only Relationship #4 above.
1410         There are no forward describedBy relationships in this example.

```

```

1411     -->
1412     <link title="All describedBy Backward Relationship Targets"
1413           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1414 aaaaaaaaaa6a/backwardRelationships/describedBy"
1415           type="application/atom+xml;type=feed"
1416           rel="urn:x-s-ramp:2013:backwardRelationships:describedBy" />
1417     <!--
1418         Link to feed of all Relationship Entries whose Relationship Type =
1419         "similarXsds". In this example, that includes only Relationship #3 above.
1420         Note that there are no backward feeds for generic relationships.
1421     -->
1422     <link title="All similarXsds Type Relationships"
1423           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1424 aaaaaaaaaa6a/relationships/similarXsds"
1425           type="application/atom+xml;type=feed"
1426           rel="urn:x-s-ramp:2013:relationships:similarXsds" />
1427
1428     <!--
1429         S-RAMP defined categorizations identifying class of data represented by
1430         this entry
1431     -->
1432     <category term="xsdDocument" label="XML Schema Document"
1433             scheme="urn:x-s-ramp:2013:type" />
1434 </entry>
1435

```

1436 2.4.1.2 Relationship Entry Documents

1437 Resolving the link to a *relationships* feed will return a feed of Relationship Entry documents. A
1438 Relationship Entry document is a valid Atom entry document which contains information about a single
1439 relationship instance associated with a Source Entry. S-RAMP requires that the representation of
1440 summary and full Relationship Entry documents SHALL be the same. This makes it possible to retrieve
1441 all relationship information for a Source Entry in a single step by retrieving the *relationships* feed. The
1442 following items SHALL appear in a Relationship Entry Document:

- 1443 • The basic Atom elements REQUIRED for a legal entry document. The REQUIRED atom:author
1444 element value SHALL be set by the server to match the value found in the Source Entry.
- 1445 • An Atom link to the Source Entry. This link SHALL use the following *rel* attribute value:
1446 o rel="urn:x-s-ramp:2013:relationship:source"
- 1447 • An Atom link to the Target Entry. This link SHALL use the following *rel* attribute value:
1448 o rel="urn:x-s-ramp:2013:relationship:target"
- 1449 • An Atom link to the Relationship Type Entry from the *relationshipTypes* feed which corresponds
1450 to the Relationship Type of this Relationship Entry. This entry SHALL use the following *rel*
1451 attribute value:
1452 o rel="urn:x-s-ramp:2013:relationship:type"
- 1453 • Atom:content text element describing the entry document.
- 1454 • A structured extension element s-ramp:relationshipData containing the Relationship Type, source
1455 UUID and target UUID of this Relationship Entry.
- 1456 • Atom:category elements describing the particular Relationship Entry:
1457 o The entry type:
1458 ▪ scheme="urn:x-s-ramp:2013:type"
1459 ▪ The only valid value for the *term* attribute here is "relationship"
1460 o The kind of relationship:

- 1461 ▪ scheme="urn:x-s-ramp:2013:kind"
- 1462 ▪ Valid values for *term* attribute are:
- 1463 • "derived"
- 1464 • "modeled"
- 1465 • "generic"

1466

1467 The example below builds on the one in Section 2.4.1.1, with the *relationships* feed of all relationships
 1468 owned by the Source Artifact. With reference to that prior example, this feed contains Relationship
 1469 Entries for relationships #2 and #3:

1470

1471 *Example 14 - Complex Relationship Scenario Relationships Feed*

```

1472 <feed xmlns="http://www.w3.org/2005/Atom"
1473       xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
1474   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344ecc0</id>
1475   <link href="http://example.org/s-ramp/xsd/XsdDocument/1225c695-cfb8-4ebb-
1476   aaaa-80da344ecc0/relationships"
1477         rel="self" type="application/atom+xml;type=feed" />
1478   <updated>2009-05-26T13:13:55.013+02:00</updated>
1479   <title type="text">accountingTypes.xsd : All relationships feed</title>
1480   <author>
1481     <name>Bellwood</name>
1482   </author>
1483
1484   <!--First Relationship Entry in the feed -->
1485   <entry>
1486     <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344ecc1</id>
1487     <updated>2009-05-26T13:13:55.013+02:00</updated>
1488     <title type="text">
1489       includedXsds Relationship for accountingTypes.xsd Source Entry
1490     </title>
1491     <published>2009-05-26T13:13:55.013+02:00</published>
1492
1493     <!-- Note that derived relationship entries do not have an "edit" link -->
1494     <link type="application/atom+xml;type=entry"
1495           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-
1496   aaaa-aaaaaaaaaaaa6a/relationships/includedXsds/1225c695-cfb8-4ebb-aaaa-
1497   80da344ecc1" rel="self" />
1498
1499     <!-- Content element identifies this as a Relationship Entry -->
1500     <content type="text">Relationship Entry</content>
1501
1502     <!-- S-RAMP structured extension for Relationship Entry data -->
1503     <s-ramp:relationshipData>
1504       <s-ramp:relationshipType>includedXsds</s-ramp:relationshipType>
1505       <s-ramp:sourceId>aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa6a</sourceId>
1506       <s-ramp:targetId>aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa6b</targetId>
1507     </ s-ramp:relationshipData>
1508
1509     <!-- Link to relationship's Source Entry -->
1510     <link title="Relationship Source Entry"
  
```



```

1511         href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
1512 aaaa-aaaaaaaaa6a"
1513         type="application/atom+xml;type=entry"
1514         rel="urn:x-s-ramp:2013:relationship:source"/>
1515
1516         <!-- Link to relationship's Target Entry -->
1517         <link title="Relationship Entry"
1518             href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
1519 aaaa-aaaaaaaaa6b"
1520             type="application/atom+xml;type=entry"
1521             rel="urn:x-s-ramp:2013:relationship:target"/>
1522
1523         <-- Link to corresponding includedXsds Relationship Type Entry -->
1524         <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
1525 aaaa-aaaaaaaaa6a/relationshipTypes/
1526             1225c695-cfb8-4ebb-aaaa-80da344eddd1"
1527             type="application/atom+xml;type=entry"
1528             rel="urn:x-s-ramp:2013:relationship:type" />
1529
1530         <-- Categorizations describing the Relationship Entry -->
1531         <category term="derived" label="Derived S-RAMP relationship."
1532             scheme="urn:x-s-ramp:2013:kind" />
1533         <category term="relationship" label="Relationship Entry type"
1534             scheme="urn:x-s-ramp:2013:type" />
1535     </entry>
1536
1537
1538     <!--Second Relationship Entry in the feed -->
1539     <entry>
1540         <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eccc2</id>
1541         <updated>2009-05-26T13:13:55.013+02:00</updated>
1542         <title type="text">
1543             similarXsds relationship for accountingTypes.xsd Source Entry.
1544         </title>
1545         <published>2009-05-26T13:13:55.013+02:00</published>
1546         <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
1547 aaaa-aaaaaaaaa6a/relationships/similarXsds/1225c695-cfb8-4ebb-aaaa-
1548 80da344eccc2"
1549             type="application/atom+xml;type=entry" rel="self" />
1550
1551         <-- Generic Relationship Entry documents include an "edit" link: -->
1552         <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
1553 aaaa-aaaaaaaaa6a/relationships/similarXsds/1225c695-cfb8-4ebb-aaaa-
1554 80da344eccc2"
1555             type="application/atom+xml;type=entry" rel="edit" />
1556
1557         <!-- Content element identifies this as a Relationship Entry -->
1558         <content type="text">Relationship Entry</content>
1559
1560         <-- S-RAMP structured extension for Relationship Entry data -->
1561         <s-ramp:relationshipData>
1562             <s-ramp:relationshipType>similarXsds</s-ramp:relationshipType>
1563             <s-ramp:sourceId>aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a</sourceId>

```

```

1564         <s-ramp:targetId>aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6c</targetId>
1565     </s-ramp:relationshipData>
1566
1567     <!--Link to Relationship's Source Entry -->
1568     <link title="Relationship's Source Entry"
1569           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
1570 aaaa-aaaaaaaaa6a"
1571           type="application/atom+xml;type=entry"
1572           rel="urn:x-s-ramp:2013:relationship:source"/>
1573
1574     <!--Link to Relationship's Target Entry -->
1575     <link title="Relationship's Target Entry"
1576           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
1577 aaaa-aaaaaaaaa6c"
1578           type="application/atom+xml;type=entry"
1579           rel="urn:x-s-ramp:2013:relationship:target"/>
1580
1581     <-- Link to corresponding similarXsds Relationship Type Entry -->
1582     <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaa-aaaa-aaaa-
1583 aaaa-aaaaaaaaa6a/relationshipTypes/1225c695-cfb8-4ebb-aaaa-80da344eddd2"
1584           type="application/atom+xml;type=entry"
1585           rel="urn:x-s-ramp:2013:relationship:type" />
1586
1587     <-- Categorizations describing the Relationship Entry -->
1588     <category term="generic" label="Generic S-RAMP relationship."
1589             scheme="urn:x-s-ramp:2013:kind" />
1590     <category term="relationship" label="S-RAMP Relationship Entry"
1591             scheme="urn:x-s-ramp:2013:type" />
1592 </entry>
1593 </feed>
1594

```

1595 It is also useful to examine the *backwardRelationships* feed from the example in Section 2.4.1.1, since it
1596 contains a "describedBy" relationship entry because this Source Entry is the target of that relationship
1597 which is owned by the "myServiceInstance" Artifact Entry.

1598
1599 *Example 15 – Backward Relationships Feed*

```

1600 <feed xmlns="http://www.w3.org/2005/Atom"
1601       xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
1602   <id>{urn:uuid:backwardRelationships:feed}</id>
1603   <link href="http://example.org/s-
1604 ramp/xsd/XsdDocument/{uuid:backwardRelationships:feed}/backwardRelationships"
1605         rel="self" type="application/atom+xml;type=feed" />
1606   <updated>2009-05-26T13:13:55.013+02:00</updated>
1607   <title type="text">
1608     accountingTypes.xsd : All backward relationships feed
1609   </title>
1610   <author>
1611     <name>Bellwood</name>
1612   </author>
1613
1614   <!--

```

1615 For this example, this feed only contains Relationship #4 from Section
1616 2.4.1.1 although to be complete, it would also have contained Relationship
1617 #1.

```

1618 -->
1619 <entry>
1620   <id>urn:uuid:{myServiceInstance:1:describedBy}</id>
1621   <updated>2009-05-26T13:13:55.013+02:00</updated>
1622   <title type="text">
1623     The describedBy relationship for myServiceInstance:1 Source Entry
1624   </title>
1625   <published>2009-05-26T13:13:55.013+02:00</published>
1626
1627   <link href="http://example.org/s-
1628 ramp/serviceImplementation/ServiceInstance/{uuid:myServiceInstance:1}/relations
1629 hips/{uuid:describedBy:1}"
1630     type="application/atom+xml;type=entry" rel="self" />
1631   <link href="http://example.org/s-
1632 ramp/serviceImplementation/ServiceInstance/{uuid:myServiceInstance:1}/relations
1633 hips/{uuid:describedBy:1}"
1634     type="application/atom+xml;type=entry" rel="edit" />
1635
1636   <!-- Content element identifies this as a Relationship Entry -->
1637   <content type="text">Relationship Entry</content>
1638
1639   <-- S-RAMP structured extension for Relationship Entry data -->
1640   <s-ramp:relationshipData>
1641     <s-ramp:relationshipType>describedBy</s-ramp:relationshipType>
1642     <s-ramp:sourceId>{uuid:ServiceInstance:1}</sourceId>
1643     <s-ramp:targetId>{uuid:accountingTypes:1}</targetId>
1644   </s-ramp:relationshipData>
1645
1646   <!-- Link to relationship's Source Entry -->
1647   <link title="Relationship Source Entry"
1648     href="http://example.org/s-
1649 ramp/serviceImplementation/ServiceInstance/{uuid:ServiceInstance:1}"
1650     type="application/atom+xml;type=entry"
1651     rel="urn:x-s-ramp:2013:relationship:source"/>
1652
1653   <!-- Link to relationship's Target Entry -->
1654   <link title="Relationship Entry"
1655     href="http://example.org/s-
1656 ramp/xsd/XsdDocument/{uuid:accountingTypes:1}"
1657     type="application/atom+xml;type=entry"
1658     rel="urn:x-s-ramp:2013:relationship:target"/>
1659
1660   <-- Link to corresponding describedBy Relationship Type Entry -->
1661   <link href="http://example.org/s-
1662 ramp/serviceImplementation/ServiceInstance/{uuid:myServiceInstance:1}/relations
1663 hipTypes/{uuid:describedBy:1}"
1664     type="application/atom+xml;type=entry"
1665     rel="urn:x-s-ramp:2013:relationship:type" />
1666
1667   <-- Categorizations describing the Relationship Entry -->

```

1668 <category term="modeled" label="Modeled S-RAMP relationship."
1669 scheme="urn:x-s-ramp:2013:kind" />
1670 <category term="relationship" label="Relationship Entry type"
1671 scheme="urn:x-s-ramp:2013:type" />
1672 </entry>
1673 </feed>
1674

1675 2.4.1.3 Relationship Type Entry Documents

1676 A Relationship Type Entry document provides information about a particular Relationship Type (such as
1677 *includedXsds*). Exactly one such document exists for each Relationship Type associated with the Source
1678 Entry.

1679 Resolving the link to a *relationshipTypes* feed will return a feed of Relationship Type Entry documents.
1680 Each such document includes a link to the *relationships* feed of all relationship instances whose
1681 Relationship Type is represented by this Relationship Type Entry.

1682 The members of this feed are maintained by the S-RAMP server. The following behaviors are normative
1683 for S-RAMP server implementations supporting the Fine Grained View for relationships:

- 1684 • A Relationship Type Entry document is automatically generated and added to the
1685 *relationshipTypes* feed of the Source Entry whenever the client adds a relationship to the
1686 *relationships* feed of the Source Entry if its Relationship Type is not already represented in the
1687 *relationshipTypes* feed.
- 1688 • Relationship Type Entry documents are NOT deleted when all relationship instances having a
1689 Relationship Type value matching that of this entry are deleted from the *relationships* feed of the
1690 Source Entry. In this situation, the Relationship Type Entry represents a relationship of that
1691 Relationship Type which has no target.
- 1692 • Clients can create a Relationship Type Entry and add it to the Relationship Type feed if one of
1693 that type is not already present. This is useful when the client wants a relationship of a given type
1694 which has no targets.
- 1695 • Clients can delete a Relationship Type Entry from a *relationshipTypes* feed. Doing so will
1696 automatically delete all relationship instances in the *relationships* feed which share the same
1697 Relationship Type value. The link to the Relationship Type specific *relationships/{Relationship*
1698 *Type}* feed will also no longer appear in the Source Entry.

1699

1700 A Relationship Types Entry document contains the following items:

- 1701 • The basic Atom elements REQUIRED for a legal entry document. The REQUIRED atom:author
1702 element value SHALL be set by the server to match the value found in the Source Entry.
- 1703 • An Atom link to the applicable Relationship Targets feed corresponding to the Relationship Type
1704 associated with this Relationship Type Entry. This is the same link as provided in the Source
1705 Entry. As described in Section 2.4.1.1, the value of this link's *rel* attribute SHALL conform to this
1706 format:
1707

1708 **rel="urn:x-s-ramp:2013:relationships:{Relationship Type}"**
1709

- 1710 • Atom content element providing a text description of the entry document.
- 1711 • A structured extension element *s-ramp:relationshipTypeData* containing the Relationship Type
1712 value of this Relationship Type Entry (e.g., *includedXsds*, *similarXsds*, etc.).
- 1713 • Atom category elements describing the particular Relationship Type Entry:
 - 1714 ○ The entry type:
 - 1715 ▪ `scheme="urn:x-s-ramp:2013:type"`
 - 1716 ▪ The only valid value for the *term* attribute here is "relationshipType"

- 1717 ○ The kind of relationship:
- 1718 ▪ scheme="urn:x-s-ramp:2013:kind"
- 1719 ▪ Valid values for *term* attribute are:
- 1720 • "derived"
- 1721 • "modeled"
- 1722 • "generic"

1723 Below is an example of a Relationship Types Feed with two Relationship Type Entry summary
 1724 documents. Note that the summary and full versions of these entries are the same in S-RAMP:

1725

1726 *Example 16 - Relationship Types Feed*

```

1727 <feed xmlns="http://www.w3.org/2005/Atom"
1728       xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
1729   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eddd0</id>
1730   <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1731   aaaaaaaaa6a/relationshipTypes/1225c695-cfb8-4ebb-aaaa-80da344eddd0"
1732       rel="self" type="application/atom+xml;type=feed" />
1733   <updated>2009-05-26T13:13:55.013+02:00</updated>
1734   <title type="text">accountingTypes.xsd : Relationship Types feed</title>
1735   <author>
1736     <name>Bellwood</name>
1737   </author>
1738
1739   <!--First Relationship Type Entry in the feed -->
1740   <entry>
1741     <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eddd1</id>
1742     <updated>2009-05-26T13:13:55.013+02:00</updated>
1743     <title type="text">
1744       Relationship Type Entry for includedXsds relationship
1745     </title>
1746     <published>2009-05-26T13:13:55.013+02:00</published>
1747     <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1748     aaaaaaaaa6a/relationshipTypes/1225c695-cfb8-4ebb-aaaa-80da344eddd1"
1749         type="application/atom+xml;type=entry" rel="self" />
1750
1751     <!-- Content element identifies this as a Relationship Type Entry -->
1752     <content type="text">Relationship Type Entry</content>
1753
1754     <!-- S-RAMP structured extension for Relationship Type Entry data -->
1755     <s-ramp:relationshipTypeData>
1756       <s-ramp:relationshipType>includedXsds</s-ramp:relationshipType>
1757     </s-ramp:relationshipTypeData>
1758
1759     <!-- Link to relationships feed for includedXsds Relationship Type -->
1760     <link title="All includedXsds Type Relationship Instances"
1761           href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1762           aaaaaaaaa6a/relationships/includedXsds"
1763           type="application/atom+xml;type=feed"
1764           rel="urn:x-s-ramp:2013:relationships:includedXsds" />
1765
1766     <category term="derived" label="Derived S-RAMP Relationship"

```

```

1767         scheme="urn:x-s-ramp:2013:kind" />
1768     <category term="relationshipType" label="S-RAMP Relationship Type Entry"
1769         scheme="urn:x-s-ramp:2013:type" />
1770 </entry>
1771
1772 <!--Second Relationship Type Entry in the feed -->
1773 <entry>
1774     <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eddd2</id>
1775     <updated>2009-05-26T13:13:55.013+02:00</updated>
1776     <title type="text">
1777         Relationship for accountingTypes.xsd Source Entry: similarXsds
1778     </title>
1779     <published>2009-05-26T13:13:55.013+02:00</published>
1780     <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1781 aaaaaaaaaa6a/relationshipTypes/1225c695-cfb8-4ebb-aaaa-80da344eddd2"
1782         type="application/atom+xml;type=entry"
1783         type="application/atom+xml;type=entry" rel="self" />
1784
1785     <!-- Generic relationships include an "edit" link: -->
1786     <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1787 aaaaaaaaaa6a/relationshipTypes/1225c695-cfb8-4ebb-aaaa-80da344eddd2"
1788         type="application/atom+xml;type=entry"
1789         type="application/atom+xml;type=entry" rel="edit" />
1790
1791 <!-- Content element identifies this as a Relationship Type Entry -->
1792 <content type="text">Relationship Type Entry</content>
1793
1794 <!-- S-RAMP structured extension for Relationship Type Entry data -->
1795 <s-ramp:relationshipTypeData>
1796     <s-ramp:relationshipType>similarXsds</s-ramp:relationshipType>
1797 </s-ramp:relationshipTypeData>
1798
1799 <!-- Link to relationships feed for similarXsds Relationship Type -->
1800 <link title="All similarXsds Type Relationships"
1801     href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
1802 aaaaaaaaaa6a/relationships/similarXsds"
1803     type="application/atom+xml;type=feed"
1804     rel="urn:x-s-ramp:2013:relationships:similarXsds" />
1805
1806     <category term="generic" label="Generic Relationship"
1807         scheme="urn:x-s-ramp:2013:kind" />
1808     <category term="relationshipType" label="S-RAMP Relationship Type Entry"
1809         scheme="urn:x-s-ramp:2013:type" />
1810 </entry>
1811 </feed>
1812

```

1813 2.4.1.4 Creating a Relationship Instance

1814 The client's ability to create new relationships varies by the kind of relationship (*derived*, *modeled*, or
1815 *generic*), the artifact types permitted for the source and target of the relationship, and on the cardinality
1816 rules for the relationship.

1817

1818 **Creating Derived Relationships:**

1819 *Derived* relationships associated with an S-RAMP Derived Model cannot be directly created by the
1820 client. They are managed by the server based upon operations performed against document
1821 resources upon which the Derived Model containing that modeled relationship is based. To create
1822 such a relationship, it is necessary to alter the document resource itself (e.g., an XSD file).

1823

1824 **Creating Modeled Relationships:**

1825 Modeled relationships (i.e., Relationship Entries) can be created and deleted within the confines of
1826 the artifact types between which they are defined, and are subject to the cardinality rules defined for
1827 them:

- 1828 • The Source Artifact type and the Target Artifact type MUST match the types described in
1829 the model (i.e., the SOA Model or Service Implementation Model).
- 1830 • When the Minimum Cardinality ≥ 0 , relationships can be created.
- 1831 • When the Maximum Cardinality $<$ unbounded, relationships can only be created if doing
1832 so does not violate the upper limit on cardinality.

1833

1834 **Creating Generic Relationships:**

1835 Generic (ad-hoc) relationships can be created at any time in any type of Artifact Entry in any of the
1836 defined models supported by S-RAMP.

1837

1838 **Creating Extended Artifact Model Relationships:**

1839 The definition of Extended Artifact Models is outside the scope of the S-RAMP specification.
1840 However, such extended models may have relationships that (logically) are similar to Modeled
1841 Relationships (see above). These Extended Artifact Model Relationships are treated the same as
1842 Generic Relationships. It is left to the client to enforce Target Artifact Type and cardinality
1843 restrictions.

1844

1845

1846 As an example of creating a generic relationship, consider two Artifact Entries, conveniently called
1847 source.xsd and target.xsd. We wish to add a *similarXsds* relationship between them. Prior to doing this,
1848 performing a GET to resolve the link to the *relationships* feed in the Atom entry for source.xsd, might
1849 return an empty feed:

1850

1851 *Example 17 - Creating Generic Relationships - Before*

```
1852 GET /s-ramp/xsd/XsdDocument/{uuid:source.xsd}/relationships HTTP/1.1  
1853 Host: example.org
```

1854

1855 returns this empty feed:

```
1856 <feed xmlns="http://www.w3.org/2005/Atom"  
1857   xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">  
1858   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eccc0</id>  
1859   <link href="http://example.org/s-  
1860   ramp/xsd/XsdDocument/{uuid:source.xsd}/relationships"  
1861     rel="self" type="application/atom+xml;type=feed" />  
1862   <updated>2009-05-26T13:13:55.013+02:00</updated>  
1863   <title type="text">source.xsd : All relationships feed</title>  
1864   <author>
```

```
1865     <name>Bellwood</name>
1866     </author>
1867 </feed>
```

1868

1869 Now to add the desired *similarXsds* relationship, the client would POST the following Atom entry
1870 document to the source.xsd entry *relationships* feed:

1871

1872 *Example 18 - Creating Generic Relationships - Adding the Relationship*

```
1873 POST /s-ramp/xsd/XsdDocument/{uuid:source.xsd}/relationshipsTargets HTTP/1.1
```

```
1874 Host: example.org
```

```
1875 Content-Type: application/atom+xml;type=entry
```

```
1876 Content-Length: nnn
```

1877

```
1878 <?xml version="1.0" ?>
```

```
1879 <entry xmlns="http://www.w3.org/2005/Atom"
```

```
1880     xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
```

```
1881     <id>{urn:uuid:relationship:1}</id>
```

```
1882     <updated />
```

```
1883     <title />
```

1884

```
1885     <!-- Content element identifies this as a Relationship Entry -->
```

```
1886     <content type="text">Relationship Entry</content>
```

1887

```
1888     <!-- S-RAMP structured extension for Relationship Entry data -->
```

```
1889     <s-ramp:relationshipData>
```

```
1890         <s-ramp:relationshipType>similarXsds</s-ramp:relationshipType>
```

```
1891         <s-ramp:sourceId>{uuid:source.xsd}</sourceId>
```

```
1892         <s-ramp:targetId>{uuid:target.xsd}</targetId>
```

```
1893     </s-ramp:relationshipData>
```

1894

```
1895     <!--
```

```
1896         Note that Links to the source and target are not included on the POST,
```

```
1897         but the server will include them in the response to the POST and on
```

```
1898         subsequent GET requests.
```

```
1899     -->
```

```
1900     <category term="generic" label="This is a user-defined s-ramp relationship."
```

```
1901         scheme="urn:x-s-ramp:2013:kind" />
```

```
1902     <category term="relationship" label="Relationship Target"
```

```
1903         scheme="urn:x-s-ramp:2013:type" />
```

```
1904 </entry>
```

1905

1906 After the *similarXsds* relationship above has been added to the *relationships* feed, performing another
1907 GET on the Source Entry's *relationships* feed would return:

1908

1909 *Example 19 - Creating Generic Relationships - After*

```
1910 <feed xmlns="http://www.w3.org/2005/Atom"
```

```
1911     xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
```

```
1912     <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eccc0</id>
```



```

1913 <link href="http://example.org/s-
1914 ramp/xsd/XsdDocument/{uuid:source.xsd}/relationships"
1915 rel="self" type="application/atom+xml;type=feed" />
1916 <updated>2009-05-26T13:13:55.013+02:00</updated>
1917 <title type="text">source.xsd : All relationships feed</title>
1918 <author>
1919 <name>Bellwood</name>
1920 </author>
1921
1922 <!--First Relationship Entry in feed -->
1923 <entry>
1924 <id>{urn:uuid:relationship:1}</id>
1925 <updated>2009-05-26T13:13:55.013+02:00</updated>
1926 <title type="text">
1927 Relationship for source.xsd Source Entry: similarXsd
1928 </title>
1929 <published>2009-05-26T13:13:55.013+02:00</published>
1930
1931 <link href="http://example.org/s-
1932 ramp/xsd/XsdDocument/{uuid:source.xsd}/relationships/{uuid:relationship:1}"
1933 type="application/atom+xml;type=entry" rel="self" />
1934 <link href="http://example.org/s-
1935 ramp/xsd/XsdDocument/{uuid:source.xsd}/relationships/{uuid:relationship:1}"
1936 type="application/atom+xml;type=entry" rel="edit" />
1937
1938 <!-- Content element identifies this as a Relationship Entry -->
1939 <content type="text">Relationship Entry</content>
1940
1941 <!-- S-RAMP structured extension for Relationship Entry data -->
1942 <s-ramp:relationshipData>
1943 <s-ramp:relationshipType>similarXsds</s-ramp:relationshipType>
1944 <s-ramp:sourceId>{uuid:source.xsd}</sourceId>
1945 <s-ramp:targetId>{uuid:target.xsd}</targetId>
1946 </s-ramp:relationshipData>
1947
1948 <!--Link to relationship Source Entry -->
1949 <link title="Relationship Source Entry"
1950 href="http://example.org/s-ramp/xsd/XsdDocument/{uuid:source.xsd}"
1951 type="application/atom+xml;type=entry"
1952 rel="urn:x-s-ramp:2013:relationship:source"/>
1953
1954 <!--Link to relationship entry -->
1955 <link title="Relationship Entry"
1956 href="http://example.org/s-ramp/xsd/XsdDocument/{uuid:target.xsd}"
1957 type="application/atom+xml;type=entry"
1958 rel="urn:x-s-ramp:2013:relationship:target"/>
1959
1960 <!--
1961 Link to corresponding similarXsds Relationship Type Entry which the server
1962 creates if this is the first similarXsds type relationship
1963 -->

```

```

1964     <link href="http://example.org/s-
1965 ramp/xsd/XsdDocument/{uuid:source.xsd}/relationshipTypes/1225c695-cfb8-4ebb-aaaa-
1966 80da344eddd2"
1967         type="application/atom+xml;type=entry"
1968         rel="urn:x-s-ramp:2013:relationship:type" />
1969
1970     <category term="generic" label="This is a user-defined S-RAMP relationship."
1971             scheme="urn:x-s-ramp:2013:kind" />
1972     <category term="relationship" label="Relationship Entry type"
1973             scheme="urn:x-s-ramp:2013:type" />
1974 </entry>
1975 </feed>
1976

```

1977 As noted in Section 2.4.1.3, Relationship Type Entry documents are automatically created by the server
1978 and added to the *relationshipTypes* feed in response to the first creation of a relationship instance of a
1979 new Relationship Type. To create a relationship having no targets for a Relationship Type not already
1980 present in the Source Entry, it is necessary to POST a Relationship Type Entry for the new Relationship
1981 Type to the *relationshipTypes* feed. An example of adding a relationship with no targets whose
1982 Relationship Type is called "myNewRelationshipType" follows:

1983
1984 *Example 20 - Adding a Relationship with No Targets*

```

1985 POST /s-ramp/xsd/XsdDocument/{uuid:source.xsd}/relationshipsTypes HTTP/1.1
1986 Host: example.org
1987 Content-Type: application/atom+xml;type=entry
1988 Content-Length: nnn
1989
1990 <?xml version="1.0" ?>
1991 <entry>
1992   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eddd3</id>
1993   <updated>2009-05-26T13:13:55.013+02:00</updated>
1994   <title type="text">
1995     Relationship Type Entry for myNewRelationshipType relationship
1996   </title>
1997   <author>
1998     <name>Bellwood</name>
1999   </author>
2000
2001   <!-- Content element identifies this as a Relationship Type Entry -->
2002   <content type="text">Relationship Type Entry</content>
2003
2004   <!-- S-RAMP structured extension for Relationship Type Entry data -->
2005   <s-ramp:relationshipTypeData>
2006     <s-ramp:relationshipType>myNewRelationshipType</s-ramp:relationshipType>
2007   </s-ramp:relationshipTypeData>
2008
2009   <category term="generic" label="Generic S-RAMP Relationship"
2010           scheme="urn:x-s-ramp:2013:kind" />
2011   <category term="relationshipType" label="S-RAMP Relationship Type Entry"
2012           scheme="urn:x-s-ramp:2013:type" />
2013 </entry>

```

2014 **2.4.1.5 Retrieving a Relationship Instance**

2015 To retrieve the metadata for a particular relationship, the client simply performs a GET on the URL of the
2016 desired Relationship Entry. Following the example from the previous section, this might look like:
2017

2018 *Example 21 - Retrieving a Relationship Entry Instance*

```
2019 GET /s-ramp/xsd/XsdDocument/{uuid:source.xsd}/relationships/similarXsds  
2020 {uuid:relationshipEntry} HTTP/1.1  
2021 Host: example.org  
2022
```

2023 which would return the same Relationship Entry document as above:

```
2024  
2025 <entry>  
2026   <id>{urn:uuid:source.xsd:relationship:1}</id>  
2027   <updated>2009-05-26T13:13:55.013+02:00</updated>  
2028   <title type="text">  
2029     Relationship for source.xsd Source Entry: similarXsd  
2030   </title>  
2031   <author>  
2032     <name>Bellwood</name>  
2033   </author>  
2034   <published>2009-05-26T13:13:55.013+02:00</published>  
2035   <link href="http://example.org/s-  
2036   ramp/xsd/XsdDocument/{uuid:source.xsd}/relationships/similarXsds/{uuid:relationshi  
2037   pEntry}"  
2038     type="application/atom+xml;type=entry" rel="self" />  
2039   <link href="http://example.org/s-  
2040   ramp/xsd/XsdDocument/{uuid:source.xsd}/relationships/similarXsds/{uuid:relationshi  
2041   pEntry}"  
2042     type="application/atom+xml;type=entry" rel="edit" />  
2043  
2044   <!-- Content element identifies this as a Relationship Entry -->  
2045   <content type="text">Relationship Entry</content>  
2046  
2047   <-- S-RAMP structured extension for Relationship Entry data -->  
2048   <s-ramp:relationshipData>  
2049     <s-ramp:relationshipType>similarXsds</s-ramp:relationshipType>  
2050     <s-ramp:sourceId>{uuid:source.xsd}</sourceId>  
2051     <s-ramp:targetId>{uuid:target.xsd}</targetId>  
2052   </s-ramp:relationshipData>  
2053  
2054   <!--Link to relationship's Source Entry -->  
2055   <link title="Relationship Source Entry"  
2056     href="http://example.org/s-ramp/xsd/XsdDocument/{uuid:source.xsd}"  
2057     type="application/atom+xml;type=entry"  
2058     rel="urn:x-s-ramp:2013:relationship:source"/>  
2059  
2060   <!--Link to relationship's Target Entry -->  
2061   <link title="Relationship Entry"  
2062     href="http://example.org/s-ramp/xsd/XsdDocument/{uuid:target.xsd}"  
2063     type="application/atom+xml;type=entry"
```

```

2064         rel="urn:x-s-ramp:2013:relationship:target"/>
2065
2066         <!-- Link to corresponding similarXsds Relationship Type Entry -->
2067         <link href="http://example.org/s-ramp/xsd/XsdDocument/aaaaaaaa-aaaa-aaaa-aaaa-
2068         aaaaaaaaaa6a/relationshipTypes/1225c695-cfb8-4ebb-aaaa-80da344eccc2"
2069         type="application/atom+xml;type=entry"
2070         rel="urn:x-s-ramp:2013:relationship:type" />
2071
2072         <category term="generic" label="This is a user-defined S-RAMP relationship."
2073         scheme="urn:x-s-ramp:2013:kind" />
2074         <category term="relationship" label="Relationship Entry type"
2075         scheme="urn:x-s-ramp:2013:type" />
2076     </entry>

```

2077 **2.4.1.6 Editing a Relationship Instance**

2078 Editing of an existing Relationship Entry instance document is prohibited in S-RAMP. To accomplish an
2079 edit of a non-derived relationship, one first DELETES the existing relationship, then POSTs a new
2080 relationship with the desired changes.

2081 Similarly, editing of an existing Relationship Type Entry instance document is prohibited in S-RAMP. As
2082 previously noted, Relationship Type Entry documents are typically created and are managed by the
2083 server, but MAY also be by the client (see Section 2.4.1.3).

2084 **2.4.1.7 Deleting a Relationship**

2085 The client's ability to delete a relationship varies by what kind of relationship it is: *derived*, *modeled* or
2086 *generic*. Requests to delete a Relationship Entry as well as a Relationship Type Entry are discussed
2087 here for each kind of relationship. In neither case is the actual Source Entry nor Target Entry referenced
2088 in a relationship instance deleted, although the applicable feeds referenced in each are in general
2089 affected.

2090 Syntax for deleting a Relationship Entry document:

```

2091 DELETE /{relationshipEntryURL} HTTP/1.1
2092
2093 Host: example.org

```

2094 For example, to delete a particular generic Relationship Entry document for a relationship of type
2095 "similarXsds":

```

2096 DELETE /s-ramp/xsd/XsdDocument/{uuid:source.xsd}/
2097 relationships/similarXsds/{uuid:relationship:1} HTTP/1.1
2098
2099 Host: example.org

```

2100 Syntax for deleting a Relationship Type Entry document:

```

2101 DELETE /{relationshipTypeEntryURL} HTTP/1.1
2102
2103 Host: example.org

```

2104 For example, to delete a generic Relationship Type entry for "similarXsds":

```

2105 DELETE /s-ramp/xsd/XsdDocument/{uuid:source.xsd}/
2106 relationshipType/{uuid:Relationship Type Entry} HTTP/1.1

```

2112 Host: example.org

2113

2114 **Deleting Derived Relationships:**

2115 Derived relationships are part of an S-RAMP Derived Model and cannot be directly deleted by the client.
2116 This applies to both Relationship Entry and Relationship Type Entry documents. These relationships are
2117 managed by the server based upon operations performed against the document resource upon which the
2118 Derived Model containing that modeled relationship is based (e.g., the A.xsd document). To delete a
2119 derived relationship, one MUST make a material change to the referenced document itself in such a way
2120 as to cause its Derived Model to no longer contain that relationship. For example, removing the include
2121 of the B.xsd document from the A.xsd document will cause the Derived Model for A.xsd to be regenerated
2122 and no longer contain an includedXsds relationship with a target of B.xsd. The Relationship Type Entry
2123 for the includedXsds relationship will still exist on the A.xsd document artifact.

2124

2125 **Deleting Modeled Relationships:**

2126 Modeled Relationships are predefined (e.g., in the Service Implementation Model or SOA Model), but
2127 they can be deleted by the client within the constraints of their cardinality rules, although the behavior of
2128 the server for modeled relationships differs from other kinds of relationships:

- 2129
- 2130 • The server SHALL ensure that there is always a link to the applicable *relationships/{Relationship Type}* feed in the relationship's Source Entry, even when this feed is empty.
 - 2131 • The server SHALL ensure that there is always a Relationship Type Entry in the *relationshipTypes*
2132 feed for each modeled Relationship Type defined for the artifact type (e.g., see the SOA Model
2133 UML in Figure 2 of the the Foundation Document of this specification).

2134 Beyond these, the behavior for requests to delete modeled relationships is subject to these cardinality
2135 rules:

2136 1. Modeled Relationships with Minimum Cardinality = 0

- 2137
- 2138 • Requesting DELETE of a modeled Relationship Entry:
 - 2139 ○ Always permitted.
 - 2140 ○ Removes subject Relationship Entry instance document from the *relationships* feed
2141 and the applicable *relationships/{Relationship Type}* feed for the Source Entry.
 - 2142 ○ Does not affect the *relationshipTypes* feed.
 - 2143 ○ All occurrences of the deleted Relationship Entry instance document are removed
2144 from the *backwardRelationships* and *backwardRelationships/{Relationship Type}*
2145 feeds of the Target Artifact entry identified by the deleted relationship's target.
 - 2146 • Requesting DELETE of a Relationship Type Entry:
 - 2147 ○ Removes all Relationship Entry instances having the Relationship Type in the
2148 request from the *relationships* feed and the applicable *relationships/{Relationship Type}*
2149 feed.
 - 2150 ○ The Relationship Type Entry is NOT deleted from the *relationshipTypes* feed for the
2151 Source Entry. This indicates that the relationship still exists, but that it now has 0
2152 targets. The server SHALL still return an HTTP return status code of 200 OK upon
2153 successful completion, because the request completed correctly within the defined
2154 behavior of S-RAMP.
 - 2155 ○ The link to the (now empty) *relationships/{Relationship Type}* feed will remain in the
2156 Source Entry.
 - 2157 ○ The corresponding Relationship Type Entry remains in the *relationshipTypes* feed.
 - 2158 ○ All occurrences of the deleted Relationship Entry instance documents are removed
2159 from the *backwardRelationships* and *backwardRelationships/{Relationship Type}*
2160 feeds of the Target Artifact entries identified by the deleted relationship targets.

2160 2. Modeled Relationships with Minimum Cardinality > 0

- 2161
- DELETE of a modeled Relationship Entry:

- 2162 ○ Permitted, unless this operation would result in a violation of the minimum cardinality
- 2163 for this Relationship Type. Behavior when permitted is the same as for the
- 2164 Cardinality = 0 case.
- 2165 • DELETE of a modeled Relationship Type Entry:
- 2166 ○ Invalid operation. This would result in a violation of the minimum cardinality for
- 2167 relationships of this Relationship Type.
- 2168

2169 **Deleting Generic Relationships:**

2170 Since generic relationships are created and controlled by the client, they MAY always be deleted. Details
2171 on deletion behavior follow:

- 2172 • Requesting DELETE of a generic Relationship Entry:
- 2173 ○ Removes subject Relationship Entry instance document from the *relationships* feed
- 2174 and the applicable *relationships/{Relationship Type}* feed for the Source Entry.
- 2175 ○ Does not affect the *relationshipTypes* feed.
- 2176 • Requesting DELETE of a generic Relationship Type Entry:
- 2177 ○ Removes all Relationship Entry instances having the Relationship Type in the
- 2178 request from the *relationships* feed and the applicable *relationships/{Relationship*
- 2179 *Type}* feed.
- 2180 ○ Removes the Relationship Type Entry from the *relationshipTypes* feed for the Source
- 2181 Entry.
- 2182 ○ The link to the applicable *relationships/{Relationship Type}* feed is removed from the
- 2183 Source Entry.

2184 **2.4.2 S-RAMP Properties**

2185 If supported by the S-RAMP server implementation, a Fine Grained View is also available for S-RAMP
2186 properties in order to facilitate their manipulation separately from the Atom Source Entry with which they
2187 are associated. This is particularly useful when the s-ramp:artifact structured extension element in the
2188 Coarse Grained View contains a large amount of data, since this view allows manipulation of one
2189 property at a time without having to explicitly update the Atom Source Entry itself.

2190 In the Coarse Grained View, some of the built-in S-RAMP Artifact properties are mapped directly to
2191 existing Atom elements for the convenience of clients. These, together with the remaining built-in
2192 properties defined in the various models, as well as all user-defined properties, are available in the Fine
2193 Grained View. System defined properties are usually read-only. All user defined properties are editable.

2194 If the Fine Grained View for Properties is supported, then the Artifact Entry document which describes the
2195 Coarse Grained View will always contain a link to the *properties* feed (see Section 2.3.2). For example:

```
2196
2197 <link href="http://example.org/s-ramp/xsd/XsdDocument/
2198         aaaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a/properties"
2199         type="application/atom+xml;type=feed"
2200         rel="urn:x-s-ramp:2013:properties" />
```

2201 **2.4.2.1 Property Entry Documents**

2202 Resolving the link to a *properties* feed in an Artifact Entry will return a feed of Property Entry documents.
2203 A Property Entry document is a valid Atom entry document which contains information about a single S-
2204 RAMP property which is associated with the Artifact Entry in which the *properties* feed is found.

2205 The following items SHALL appear in a Property Entry document, both in its summary and full entry
2206 representations:

- 2207 • The basic Atom elements REQUIRED for a legal entry document. The REQUIRED atom:author
- 2208 element value SHALL be set by the server to match the value found in the Artifact Entry.
- 2209 • Atom content text element describing the property.

- 2210 • A structured extension element s-ramp:propertyData containing the name and value of the
2211 property.
- 2212 • Atom category elements describing the particular Property Entry:
 - 2213 ○ The entry type:
 - 2214 ▪ *scheme*="urn:x-s-ramp:2013:type"
 - 2215 ▪ The only valid value for the *term* attribute here is "property"
 - 2216 ○ The kind of property:
 - 2217 ▪ *scheme*="urn:x-s-ramp:2013:kind"
 - 2218 ▪ Valid values for *term* attribute are:
 - 2219 • "derived"
 - 2220 ○ Built-in property defined in a Derived Model. Never editable.
 - 2221 • "modeled"
 - 2222 ○ Pre-defined property in the Core Model, SOA Model, Service
2223 Implementation Model or a user defined model. Editable, but
2224 server can override values for properties in the Core Model.
 - 2225 • "generic"
 - 2226 ○ Client defined (ad-hoc) property. Always editable.

2228 Consistent with properties contained in the s-ramp:artifact structured extension element of the Coarse
2229 Grained View, the *properties* feed for the Fine Grained View can only contain one unique Property Entry
2230 document instance for a given property name.

2231 The representation of summary and full Property Entry documents SHALL be the same for all Property
2232 Entry documents. This makes it possible to retrieve all property data for the Artifact Entry in one step by
2233 resolving the link to the *properties* feed. Below is an example of a *properties* feed containing a user-
2234 defined (generic) Property Entry document instance, along with the modeled properties from the Core
2235 Model which are always present. For brevity, this feed only illustrates the *name* property among these:

2236
2237 *Example 22 - Property Entry Feed*

```

2238 <feed xmlns="http://www.w3.org/2005/Atom"
2239       xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
2240   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eeee0</id>
2241   <link href="http://example.org/s-
2242 ramp/xsd/XsdDocument/{uuid:accountingTypes.xsd}/properties"
2243         rel="self" type="application/atom+xml;type=feed" />
2244   <updated>2009-05-26T13:13:55.013+02:00</updated>
2245   <title type="text">accountingTypes.xsd : Properties feed</title>
2246   <author>
2247     <name>Bellwood</name>
2248   </author>
2249
2250   <!-- First Property Entry in feed -->
2251   <entry>
2252     <id>{urn:uuid:property:1}</id>
2253     <updated>2009-05-26T13:13:55.013+02:00</updated>
2254     <title type="text">
2255       Property for accountingTypes.xsd
2256     </title>
2257     <published>2009-05-26T13:13:55.013+02:00</published>
2258

```

```

2259     <link href="http://example.org/s-
2260 ramp/xsd/XsdDocument/{uuid:accountingTypes.xsd}/properties/{uuid:property:1}"
2261         type="application/atom+xml;type=entry" rel="self" />
2262     <link href="http://example.org/s-ramp/xsd/XsdDocument/{uuid:
2263 accountingTypes.xsd}/properties/{uuid:property:1}"
2264         type="application/atom+xml;type=entry" rel="edit" />
2265
2266     <!-- Content element identifies this as a Property Entry -->
2267     <content type="text">Property Entry</content>
2268
2269     <!-- S-RAMP structured extension for Property Entry data -->
2270     <s-ramp:propertyData>
2271         <s-ramp:property>
2272             <s-ramp:propertyName>foo</s-ramp:propertyName>
2273             <s-ramp:propertyValue>bar</s-ramp:propertyValue>
2274         </s-ramp:property>
2275     </s-ramp:propertyData>
2276
2277     <category term="generic" label="This is a user-defined S-RAMP property."
2278         scheme="urn:x-s-ramp:2013:kind" />
2279     <category term="property" label="Property Entry type"
2280         scheme="urn:x-s-ramp:2013:type" />
2281 </entry>
2282
2283 <!-- Second Property Entry in the feed -->
2284 <entry>
2285     <id>{urn:uuid:property:2}</id>
2286     <updated>2009-05-26T13:13:55.013+02:00</updated>
2287     <title type="text">
2288         Name property for accountingTypes.xsd
2289     </title>
2290     <published>2009-05-26T13:13:55.013+02:00</published>
2291
2292     <link href="http://example.org/s-
2293 ramp/xsd/XsdDocument/{uuid:accountingTypes.xsd}/properties/{uuid:property:2}"
2294         type="application/atom+xml;type=entry" rel="self" />
2295
2296     <!-- Content element identifies this as a Property Entry -->
2297     <content type="text">Property Entry</content>
2298
2299     <!-- S-RAMP structured extension for Property Entry data -->
2300     <s-ramp:propertyData>
2301         <s-ramp:property>
2302             <s-ramp:propertyName>name</s-ramp:propertyName>
2303             <s-ramp:propertyValue>accountingTypes</s-ramp:propertyValue>
2304         </s-ramp:property>
2305     </s-ramp:propertyData>
2306
2307     <category term="modeled" label="This is a modeled S-RAMP property."
2308         scheme="urn:x-s-ramp:2013:kind" />
2309     <category term="property" label="Property Entry type"
2310         scheme="urn:x-s-ramp:2013:type" />

```


2311 </entry>
2312 </feed>

2313 2.4.2.2 Creating Properties

2314 User-defined (generic) properties can be created by clients and associated with an Artifact Entry. To
2315 accomplish this, the client simply performs a POST of a Property Entry document to the Artifact Entry's
2316 *properties* feed.

2317 For example, consider again our Artifact Entry, which is still conveniently called artifact.xsd. We wish to
2318 add a property called "foo" with value "bar" to this entry. To add the desired *foo* property, the client would
2319 POST the following Atom entry document to the artifact.xsd entry's *properties* feed:

2320

2321 *Example 23 - Creating a Property - Adding the Property*

2322 POST /s-ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties HTTP/1.1

2323 Host: example.org

2324 Content-Type: application/atom+xml;type=entry

2325 Content-Length: nnn

2326

2327 <?xml version="1.0" ?>

2328 <entry xmlns="http://www.w3.org/2005/Atom"

2329 xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">

2330 <id>{urn:uuid:property:1}</id>

2331 <updated />

2332 <title />

2333 <author>

2334 <name>Bellwood</name>

2335 </author>

2336

2337 <!-- Content element identifies this as a Property Entry -->

2338 <content type="text">Property Entry</content>

2339

2340 <-- S-RAMP structured extension for Property Entry data -->

2341 <s-ramp:propertyData>

2342 <s-ramp:property>

2343 <s-ramp:propertyName>foo</s-ramp:propertyName>

2344 <s-ramp:propertyValue>bar</s-ramp:propertyValue>

2345 </s-ramp:property>

2346 </s-ramp:propertyData>

2347

2348 <category term="generic" label="This is a user-defined property."

2349 scheme="urn:x-s-ramp:2013:kind" />

2350 <category term="property" label="Property entry"

2351 scheme="urn:x-s-ramp:2013:type" />

2352 </entry>

2353

2354 After the *foo* property above has been added to the *properties* feed, performing a GET on the Artifact
2355 Entry's *properties* feed would return:

2356

2357 *Example 24 - Creating a Property - After*

2358 <feed xmlns="http://www.w3.org/2005/Atom"

```

2359     xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
2360     <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eeee0</id>
2361     <link href="http://example.org/s-
2362     ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties"
2363         rel="self" type="application/atom+xml;type=feed" />
2364     <updated>2009-05-26T13:13:55.013+02:00</updated>
2365     <title type="text">source.xsd : Feed of all properties</title>
2366     <author>
2367         <name>Bellwood</name>
2368     </author>
2369
2370     <!--First Property Entry in feed -->
2371     <entry>
2372         <id>{urn:uuid:property:1}</id>
2373         <updated>2009-05-26T13:13:55.013+02:00</updated>
2374         <title type="text">
2375             foo property for artifact.xsd Entry
2376         </title>
2377         <published>2009-05-26T13:13:55.013+02:00</published>
2378
2379         <link href="http://example.org/s-
2380         ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties/{uuid:property:1}"
2381             type="application/atom+xml;type=entry" rel="self" />
2382         <link href="http://example.org/s-
2383         ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties/{uuid:property:1}"
2384             type="application/atom+xml;type=entry" rel="edit" />
2385
2386         <!-- Content element identifies this as a Property Entry -->
2387         <content type="text">Property Entry</content>
2388
2389         <-- S-RAMP structured extension for Property Entry data -->
2390         <s-ramp:propertyData>
2391             <s-ramp:property>
2392                 <s-ramp:propertyName>foo</s-ramp:propertyName>
2393                 <s-ramp:propertyValue>bar</s-ramp:propertyValue>
2394             </s-ramp:property>
2395         </s-ramp:propertyData>
2396
2397         <category term="generic" label="This is a user-defined property."
2398             scheme="urn:x-s-ramp:2013:kind" />
2399         <category term="property" label="Property entry"
2400             scheme="urn:x-s-ramp:2013:type" />
2401     </entry>
2402     ...
2403 </feed>

```

2404 **2.4.2.3 Retrieving Properties**

2405 To retrieve the metadata for a particular property, the client simply performs a GET on the URL of the
2406 desired Property Entry. Following the example from the previous section, this might look like:
2407

2408 *Example 25 - Retrieving a Property Entry Document*

```
2409 GET /s-ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties/  
2410 {uuid:property:1} HTTP/1.1  
2411 Host: example.org
```

2412

2413 would return the same Property Entry document as above:

2414

```
2415 <entry>
```

```
2416   <id>{urn:uuid:property:1}</id>
```

```
2417   <updated>2009-05-26T13:13:55.013+02:00</updated>
```

```
2418   <title type="text">
```

```
2419     foo property for artifact.xsd Entry
```

```
2420   </title>
```

```
2421   <published>2009-05-26T13:13:55.013+02:00</published>
```

2422

```
2423   <link href="http://example.org/s-
```

```
2424   ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties/{uuid:property:1}"
```

```
2425     type="application/atom+xml;type=entry" rel="self" />
```

```
2426   <link href="http://example.org/s-
```

```
2427   ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties/{uuid:property:1}"
```

```
2428     type="application/atom+xml;type=entry" rel="edit" />
```

2429

```
2430   <!-- Content element identifies this as a Property Entry -->
```

```
2431   <content type="text">Property Entry</content>
```

2432

```
2433   <-- S-RAMP structured extension for Property Entry data -->
```

```
2434   <s-ramp:propertyData>
```

```
2435     <s-ramp:property>
```

```
2436       <s-ramp:propertyName>foo</s-ramp:propertyName>
```

```
2437       <s-ramp:propertyValue>bar</s-ramp:propertyValue>
```

```
2438     </s-ramp:property>
```

```
2439   </s-ramp:propertyData>
```

2440

```
2441   <category term="generic" label="This is a user-defined property."
```

```
2442     scheme="urn:x-s-ramp:2013:kind" />
```

```
2443   <category term="property" label="Property entry"
```

```
2444     scheme="urn:x-s-ramp:2013:type" />
```

```
2445 </entry>
```

2446 **2.4.2.4 Editing Properties**

2447 Editing of an existing Property Instance document is limited to altering the property value. The property
2448 name is always read only and cannot be changed by editing. Requests to alter the property name of an
2449 existing Property Entry document will return HTTP error "403" Forbidden.

2450 To edit the property value in a Property Entry document, the client performs an HTTP PUT of the
2451 complete Property Entry document with the changed value, to the member resource URI of the Property
2452 Entry document. The PUT operation will replace the property value with whatever value is specified
2453 here. An example which replaces the property value in the previous example with a new value is
2454 illustrated below:

2455

2456 *Example 26 - Editing a Property Entry Document*

2457 PUT /s-ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties HTTP/1.1

2458 Host: example.org

2459 Content-Type: application/atom+xml;type=entry

2460 Content-Length: nnn

2461

2462 <?xml version="1.0" ?>

2463 <entry>

2464 <id>{urn:uuid:property:1}</id>

2465 <updated>2009-05-26T13:13:55.013+02:00</updated>

2466 <title type="text">

2467 foo property for artifact.xsd Entry

2468 </title>

2469 <published>2009-05-26T13:13:55.013+02:00</published>

2470

2471 <link href="http://example.org/s-

2472 ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties/{uuid:property:1}"

2473 type="application/atom+xml;type=entry rel="self" />

2474 <link href="http://example.org/s-

2475 ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties/{uuid:property:1}"

2476 type="application/atom+xml;type=entry" rel="edit" />

2477

2478 <!-- Content element identifies this as a Property Entry -->

2479 <content type="text">Property Entry</content>

2480

2481 <-- S-RAMP structured extension for Property Entry data -->

2482 <s-ramp:propertyData>

2483 <s-ramp:property>

2484 <s-ramp:propertyName>foo</s-ramp:propertyName>

2485 <s-ramp:propertyValue>bar1</s-ramp:propertyValue>

2486 </s-ramp:property>

2487 </s-ramp:propertyData>

2488

2489 <category term="generic" label="This is a user-defined property."

2490 scheme="urn:x-s-ramp:2013:kind" />

2491 <category term="property" label="Property entry"

2492 scheme="urn:x-s-ramp:2013:type" />

2493 </entry>

2494 **2.4.2.5 Deleting Properties**

2495 To delete a *generic* (ad-hoc) property and remove it from the *properties* feed associated with an Artifact

2496 Entry, a client simply performs a DELETE against the URL of the desired Property Entry. Continuing with

2497 the *generic* property example from the previous sections this might look like:

2498

2499 DELETE /s-ramp/xsd/XsdDocument/{uuid:artifact.xsd}/properties/

2500 {uuid:property:1} HTTP/1.1

2501 Host: example.org

2502 **2.4.3 S-RAMP Classifications**

2503 S-RAMP classifications are another important class of metadata which describe an S-RAMP Artifact. The

2504 S-RAMP schema models a classification with a URL for the value of the s-ramp:classifiedBy element.

2505 Individual classification values within an S-RAMP Artifact Entry SHALL be unique. The same
2506 classification URL value can only be represented once in an s-ramp:artifact structured extension in the
2507 Artifact Entry.

2508 If supported by the S-RAMP server implementation, a Fine Grained View is also available for S-RAMP
2509 classifications in order to facilitate their manipulation separately from the Artifact Entry with which they are
2510 associated. This is particularly useful when the s-ramp:artifact element contains a large amount of data,
2511 since this view allows manipulation of one classification at a time without having to explicitly update the
2512 Artifact Entry itself.

2513 If the Fine Grained View for Classifications is supported, then the Artifact Entry document which describes
2514 the Coarse Grained View will always contain a link to the *classifications* feed (see Section 2.3.2). For
2515 example:

```
2516  
2517     <link href="http://{host}/s-ramp/{uuid:Artifact-Entry}/classifications"  
2518           type="application/atom+xml;type=feed"  
2519           rel="urn:x-s-ramp:2013:classification" />
```

2520 2.4.3.1 The Classification Entry Document

2521 Resolving the link to a *classifications* feed in an Artifact Entry will return a feed of Classification Entry
2522 documents. A Classifications Entry document is a valid Atom entry document which contains information
2523 about a single S-RAMP classification which is associated with the Artifact Entry in which this
2524 *classifications* feed is found.

2525 The following items SHALL appear in a Classifications Entry document, both in its summary and full entry
2526 representations:

- 2527 • The basic Atom elements REQUIRED for a legal entry document. The REQUIRED atom:author
2528 element value SHALL be set by the server to match the value found in the Artifact Entry.
- 2529 • Atom content text element describing the classification entry.
- 2530 • A structured extension element s-ramp:classificationData containing the URL of the OWL
2531 classification value.
- 2532 • Atom category element describing the particular Classification Entry:
 - 2533 ○ The entry type:
 - 2534 ▪ scheme="urn:x-s-ramp:2013:type"
 - 2535 ▪ The only valid value for the *term* attribute here is "classification"

2536

2537 As with the Coarse Grained View, the *classifications* feed for the Fine Grained View can only contain one
2538 unique Classification Entry document instance for a given classification URL value. These values
2539 correspond one-to-one with the s-ramp:classifiedBy element values in the s-ramp:artifact structured
2540 extension found in a full Artifact Entry document.

2541 The representation of both summary and full Classification Entry documents SHALL be the same for all
2542 Classification Entry documents. This makes it possible to retrieve all classification data for the Artifact
2543 Entry in one step by resolving the link to the *classifications* feed. Below is an example of a *classifications*
2544 feed containing a Classification Entry document instance:

2545

2546 *Example 27 - Classification Entry Feed*

```
2547     <feed xmlns="http://www.w3.org/2005/Atom"  
2548           xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">  
2549       <id>urn:uuid:aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaff0</id>  
2550       <link href="http://example.org/s-  
2551 ramp/xsd/XsdDocument/{uuid:accountingTypes.xsd}/classifications"  
2552           rel="self" type="application/atom+xml;type=feed" />  
2553       <updated>2009-05-26T13:13:55.013+02:00</updated>
```

```

2554     <title type="text">accountingTypes.xsd : Classifications feed</title>
2555     <author>
2556         <name>Bellwood</name>
2557     </author>
2558
2559     <!--First Classification Entry in feed -->
2560     <entry>
2561         <id>{urn:uuid:classification:1}</id>
2562         <updated>2009-05-26T13:13:55.013+02:00</updated>
2563         <title type="text">
2564             Classification for accountingTypes.xsd
2565         </title>
2566         <published>2009-05-26T13:13:55.013+02:00</published>
2567
2568         <link href="http://example.org/s-
2569 ramp/xsd/XsdDocument/{uuid:accountingTypes.xsd}/classifications/{uuid:classification:1}"
2570         type="application/atom+xml;type=entry" rel="self" />
2571         <link href="http://example.org/s-ramp/xsd/XsdDocument/{uuid:
2572 accountingTypes.xsd}/classifications/{uuid:classification:1}"
2573         type="application/atom+xml;type=entry" rel="edit" />
2574
2575         <!-- Content element identifies this as a Classification Entry -->
2576         <content type="text">Classification Entry</content>
2577
2578         <!-- S-RAMP structured extension for Classification Entry data -->
2579         <s-ramp:classificationData>
2580             <s-ramp:classifiedBy>
2581                 http://example.org/ontologies/accounting.owl/accounts#checking
2582             </s-ramp:classifiedBy>
2583         </s-ramp:classificationData>
2584
2585         <category term="classification" label="Classification Entry type"
2586                 scheme="urn:x-s-ramp:2013:type" />
2587     </entry>
2588 </feed>
2589

```

2590 2.4.3.2 Creating Classifications

2591 User-defined (generic) classifications can be created by clients and associated with an Artifact Entry. To
2592 accomplish this, the client simply performs a POST of a Classification Entry document to the Artifact
2593 Entry's *classifications* feed.

2594 For example, consider our Artifact Entry called *artifact.xsd* again. We wish to add a classification instance
2595 value denoting a savings account using the *accounting.owl* ontology. Prior to doing this, performing a
2596 GET to resolve the link to the *classifications* feed in the Atom entry for *artifact.xsd*, might return an empty
2597 feed:

2598

2599 *Example 28 - Creating a Classification - Before*

```

2600     GET /s-ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications HTTP/1.1
2601     Host: example.org
2602

```

2603 returns this empty feed:

```

2604
2605 <feed xmlns="http://www.w3.org/2005/Atom"
2606       xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
2607   <id>urn:uuid:aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaff0</id>
2608   <link href="http://example.org/s-
2609 ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications"
2610       rel="self" type="application/atom+xml;type=feed" />
2611   <updated>2009-05-26T13:13:55.013+02:00</updated>
2612   <title type="text">classifications.xsd : Feed of all classifications</title>
2613   <author>
2614     <name>Bellwood</name>
2615   </author>
2616 </feed>

```

2617
2618 Now to add the desired savings account classification, the client would POST the following Classification
2619 Entry document to the artifact.xsd entry's *classifications* feed:

2620

2621 *Example 29 - Creating a Classification - Adding the Classification Entry*

2622 POST /s-ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications HTTP/1.1

2623 Host: example.org

2624 Content-Type: application/atom+xml;type=entry

2625 Content-Length: nnn

2626

2627 <?xml version="1.0" ?>

2628 <entry xmlns="http://www.w3.org/2005/Atom"

2629 xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0"

2630 xmlns:s-rampatom="http://docs.oasis-open.org/s-ramp/ns/s-ramp-

2631 v1.0/atombinding">

2632 <id>{urn:uuid:classification:1}</id>

2633 <updated />

2634 <title />

2635 <author>

2636 <name>Bellwood</name>

2637 </author>

2638

2639 <!-- Content element identifies this as a Classification Entry -->

2640 <content type="text">Classification Entry</content>

2641

2642 <-- S-RAMP structured extension for Classification Entry data -->

2643 <s-ramp:classificationData>

2644 <s-ramp:classifiedBy>

2645 http://example.org/ontologies/accounting.owl/accounts#savings

2646 </s-ramp:classifiedBy>

2647 </s-ramp:classificationData>

2648

2649 <category term="classification" label="Classification entry"

2650 scheme="urn:x-s-ramp:2013:type" />

2651 </entry>

2652

2653 After the *savings account* classification above has been added to the *classifications* feed, performing
2654 another GET on the Artifact Entry's *classifications* feed would return:

2655

2656 *Example 30 - Creating a Classification - After*

```
2657 <feed xmlns="http://www.w3.org/2005/Atom"
2658       xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
2659   <id>urn:uuid:aaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaff0</id>
2660   <link href="http://example.org/s-
2661 ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications"
2662         rel="self" type="application/atom+xml;type=entry" />
2663   <updated>2009-05-26T13:13:55.013+02:00</updated>
2664   <title type="text">source.xsd : Feed of all classifications</title>
2665   <author>
2666     <name>Bellwood</name>
2667   </author>
2668
2669   <!-- First Classification Entry in feed -->
2670   <entry>
2671     <id>{urn:uuid:classification:1}</id>
2672     <updated>2009-05-26T13:13:55.013+02:00</updated>
2673     <title type="text">
2674       Account savings classification for artifact.xsd entry
2675     </title>
2676     <published>2009-05-26T13:13:55.013+02:00</published>
2677
2678     <link href="http://example.org/s-
2679 ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications/{uuid:classification:1}"
2680           type="application/xml;type=entry" rel="self" />
2681     <link href="http://example.org/s-
2682 ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications/{uuid:classification:1}"
2683           type="application/xml;type=entry" rel="edit" />
2684
2685     <!-- Content element identifies this as a Classification Entry -->
2686     <content type="text">Classification Entry</content>
2687
2688     <-- S-RAMP structured extension for Relationship Entry data -->
2689     <s-ramp:classificationData>
2690       <s-ramp:classifiedBy>
2691         http://example.org/ontologies/accounting.owl/accounts#savings
2692       </s-ramp:classifiedBy>
2693     </s-ramp:classificationData>
2694
2695     <category term="classification" label="Classification entry"
2696               scheme="urn:x-s-ramp:2013:type" />
2697   </entry>
2698 </feed>
```

2699 **2.4.3.3 Retrieving Classifications**

2700 To retrieve the metadata for a particular classification value, the client simply performs a GET on the URL
2701 of the desired Classification Entry. Following the example from the previous section, this might look like:

2702

2703 *Example 31 - Retrieving a Classification Entry Document*

```
2704 GET /s-ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications/  
2705 {uuid:classification:1} HTTP/1.1  
2706 Host: example.org
```

2707

2708 would return the same Classification Entry document as above:

2709

```
2710 <entry>  
2711   <id>{urn:uuid:classification:1}</id>  
2712   <updated>2009-05-26T13:13:55.013+02:00</updated>  
2713   <title type="text">  
2714     Account savings classification for artifact.xsd entry<  
2715   </title>  
2716   <published>2009-05-26T13:13:55.013+02:00</published>  
2717  
2718   <link href="http://example.org/s-  
2719 ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications/{uuid:classification:1}"  
2720     type="application/xml;type=entry" rel="self" />  
2721   <link href="http://example.org/s-  
2722 ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications/{uuid:classification:1}"  
2723     type="application/xml;type=entry" rel="edit" />  
2724  
2725   <!-- Content element identifies this as a Classification Entry -->  
2726   <content type="text">Classification Entry</content>  
2727  
2728   <!-- S-RAMP structured extension for Relationship Entry data -->  
2729   <s-ramp:classificationData>  
2730     <s-ramp:classifiedBy>  
2731       http://example.org/ontologies/accounting.owl/accounts#savings  
2732     </s-ramp:classifiedBy>  
2733   </s-ramp:classificationData>  
2734  
2735   <category term="classification" label="Classification entry"  
2736     scheme="urn:x-s-ramp:2013:type" />  
2737 </entry>
```

2738 **2.4.3.4 Editing Classifications**

2739 Editing of an existing Classification Entry document instance is prohibited in S-RAMP. To accomplish an
2740 edit of a classification using the Fine Grained View, the client first performs a DELETE of the existing
2741 classification, then a POST of a new classification with the desired changes.

2742 **2.4.3.5 Deleting Classifications**

2743 To delete a *generic* classification and remove it from the *classifications* feed associated with an Artifact
2744 Entry, a client simply performs a DELETE against the URL of the desired Classification Entry. Continuing
2745 with the classification example from the previous sections this might look like:

2746

```
2747 DELETE /s-ramp/xsd/XsdDocument/{uuid:artifact.xsd}/classifications/  
2748 {uuid:classification:1} HTTP/1.1  
2749 Host: example.org
```

2750

3 S-RAMP Query Using Atom Binding

2751

3.1 Searching Repository Artifacts

2752 S-RAMP supports a rich query capability, which is based upon the use of flexible XPath 2 based filter
2753 arguments. Refer to the *SOA Repository Artifact Model and Protocol Specification – Foundation*
2754 document, Section 4 for details on how to form S-RAMP query predicates. This document only describes
2755 the Atom specific syntax needed for query using the Atom Binding.

2756 A successful query using the Atom Binding will return a feed. Feeds contain summary Atom entry
2757 documents which cannot be assumed to be complete. To retrieve a full entry, it is necessary to perform a
2758 subsequent GET on the desired entry.

2759 S-RAMP supports execution of queries via an inline (ad-hoc) syntax, as well as through the use of Stored
2760 Queries which have been stored in the repository. Each is discussed in the following sections.

3.2 Inline Queries

2762 Ad-hoc queries can be performed in one of two ways:

- 2763 1. Using HTTP GET where the query arguments are included in the URL
- 2764 2. Using HTTP POST where the query arguments are the content being posted

2765

2766 To perform an ad-hoc query using HTTP GET use the following syntax:

2767

```
2768 GET /{query path}?query={predicate-filter-  
2769 string}&startIndex=0&count=25&orderBy=name&ascending=true&{label}={query parameter}...  
2770 HTTP/1.1
```

2771 Host: example.org

2772 To perform an ad-hoc query using HTTP POST use the following syntax:

2773

```
2774 POST /s-ramp  
2775 Content-Type: multipart/form-data; boundary=AaB03x
```

2776

```
2777 --AaB03x  
2778 Content-Disposition: form-data; name="query"
```

2779

```
2780 {query predicate}
```

2781

```
2782 --AaB03x  
2783 Content-Disposition: form-data; name="startIndex"
```

2784

```
2785 {start index}
```

2786

```
2787 --AaB03x  
2788 Content-Disposition: form-data; name="count"
```

2789

```
2790 {count}
```

2791

```
2792 --AaB03x  
2793 Content-Disposition: form-data; name="orderBy"
```

2794

2795 {order-by}
 2796
 2797 --AaB03x
 2798 Content-Disposition: form-data; name="ascending"
 2799
 2800 {ascending}
 2801
 2802 --AaB03x
 2803 Content-Disposition: form-data; name={label}
 2804
 2805 {query parameter}
 2806 --AaB03x
 2807
 2808 --AaB03x--
 2809

2810 As illustrated above, the following query arguments can (optionally) be specified when performing a query
 2811 (either via GET or POST):

- 2812 • startIndex (optional, default 0) – indicates the starting position within the result set to return
- 2813 • count (optional, default 'unbounded') – indicates the number of artifacts to return (if not included,
 2814 the number of artifacts returned is implementation specific)
- 2815 • startPage (optional, default 1) – mutually exclusive with the 'startIndex' parameter, indicates
 2816 which page within the result set to return
- 2817 • orderBy (optional, default 'unspecified') – indicates an S-RAMP property (either modeled or user-
 2818 defined) to use for ordering the results (if not specified, the ordering is implementation specific)
- 2819 • ascending (optional, default true) – indicates the direction of the ordering (true for ascending,
 2820 false for descending)

2821 Additionally, the legal values for {label} = {query parameter} are defined here. Sets of these can be
 2822 repeated an arbitrary number of times:

- 2823 • propertyName = {property name value}. This allows specifying property name(s) whose values
 2824 SHALL be included in applicable Entry documents in the results feed.

2825 The {query predicate} syntax is defined in the *Foundation* document. Example 32 below illustrates both
 2826 approaches for specifying an optional namespace and a propertyName to be returned in the results of the
 2827 query (note that uses of the HTML reserved character ":" in these examples would need to be URL
 2828 encoded as %3A):

2829
 2830 *Example 32 - Ad-hoc Queries*

2831 Query using HTTP GET with arguments contained in the URL:

```
2832 GET /s-ramp?query=serviceImplementation/ServiceInstance[@reliability!="high"]
2833 &propertyName="reliability"&xmlns:acme="http://acme.org/s-ramp/custom" HTTP/1.1
2834 Host: example.org
```

2836
 2837 Query using HTTP POST with arguments as content:

```
2838  

2839 POST /s-ramp
2840 Content-Type: multipart/form-data; boundary=AaB03x
2841  

2842 --AaB03x
```

2843 Content-Disposition: form-data; name="query"
 2844
 2845 query=serviceImplementation/ServiceInstance[reliability!="high"]&xmlns:acme=htt
 2846 p://acme.org/s-ramp/custom
 2847
 2848 --AaB03x
 2849 Content-Disposition: form-data; name="propertyName"
 2850
 2851 reliability
 2852
 2853 --AaB03x--
 2854

2855 The response from either form of ad-hoc query is an Atom *feed* of summary entry documents which
 2856 match the criteria of the query. If there are no matches, the feed will be empty. If one or more optional
 2857 propertyName values is included as a {query parameter}, this will cause each entry document returned to
 2858 include an *s-ramp:artifact* section containing the specific properties listed. The "s-ramp:artifact" section
 2859 included SHALL NOT be considered a complete representation of the entry. The propertyName
 2860 parameter option is a convenience to allow clients to recover all specifically requested properties in the
 2861 feed of entries. This MAY avoid the need to perform a subsequent GET on individual entries if the full
 2862 entry is not needed. Outside of explicit use of this parameter, S-RAMP does not prescribe which portions
 2863 of the Artifact Entry are included in the summary entries returned. This will vary by implementation.
 2864 Clients MUST perform a GET operation on the member resource URI in order to guarantee that they have
 2865 complete information for a particular artifact.
 2866

2867 *Example 33 - Ad-hoc Query Response*

```

2868 <feed xmlns="http://www.w3.org/2005/Atom"
2869       xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
2870   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344eazz0</id>
2871   <link href="http://example.org/s-ramp?query=s-
2872 ramp/serviceImplementation/ServiceInstance[@acme:reliability='high']
2873 &xmlns:acme='http://acme.org/s-ramp/custom'"
2874 rel="self" type="application/atom+xml;type=feed" />
2875   <updated>2009-05-26T13:13:55.013+02:00</updated>
2876   <title type="text">Query Response</title>
2877   <author>
2878     <name>Bellwood</name>
2879   </author>
2880
2881   <!--First Matching Entry in feed -->
2882   <entry xmlns="http://www.w3.org/2005/Atom"
2883         xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0">
2884     <id>urn:uuid:aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaabb</id>
2885     <updated>2009-05-26T13:13:55.013+02:00</updated>
2886     <title type="text">myServiceInstance</title>
2887     <published>2009-05-26T13:13:55.013+02:00</published>
2888     <author>
2889       <name>Bellwood</name>
2890     </author>
2891     <summary type="text">My Service Instance document</summary>
2892     <content type="application/xml"
2893           src="http://example.org/s-ramp/{path}/{artifact id}"/>
2894     <link type="application/atom+xml;type=entry" rel="self"
  
```

```

2895         href="http://example.org/s-ramp/{uuid:myServiceInstance}" />
2896 <link type="application/atom+xml;type=entry" rel="edit"
2897         href="http://example.org/s-ramp/{uuid:myServiceInstance}" />
2898
2899 <!-- Client defined classifications: -->
2900 <category term="http://example.org/ontologies/accounting.owl/accounts"
2901         label="User defined classification"
2902         scheme="urn:x-s-ramp:user-defined-classification" />
2903
2904 <!--
2905     S-RAMP defined categorizations identifying class of data represented by
2906     this entry
2907 -->
2908 <category term="ServiceInstance" label="Service Instance"
2909         scheme="urn:x-s-ramp:2013:type" />
2910 </entry>
2911 </feed>

```

2912 3.3 Stored Queries

2913 Query filters can be stored as artifacts in S-RAMP using the StoredQuery element described in the
2914 *Foundation* document, Section 5.6. Clients can request the server to execute a Stored Query, whose
2915 results are then made available at a particular URL. Stored Queries can be prepared for general use, or
2916 can be created by the client. Stored Queries are specified to an S-RAMP server using a Stored Query
2917 Entry document. Note that since the StoredQuery does not derive from baseArtifactType, it is much
2918 simpler. The name of the Stored Query SHALL be a unique client provided string.

2919 3.3.1 Stored Query Entry Documents

2920 The following items SHALL appear in a Stored Query Entry document, both in its summary and full entry
2921 representations:

- 2922 • The basic Atom elements REQUIRED for a legal entry document.
- 2923 • Atom content text element indicating that the document represents a Stored Query Entry.
- 2924 • S-RAMP structured extension (s-ramp:StoredQueryData element) as described in Appendix
2925 Appendix F, which contains the following items.
 - 2926 ○ A unique client provided name for the query (queryName)
 - 2927 ○ A query expression (queryString)
 - 2928 ○ An optional list of property names (propertyName values) which indicate to the server the
2929 properties whose values SHALL be included in the feed of Stored Query Entries made
2930 available at the results URL, to whatever extent those properties are present in each
2931 entry returned in the feed as a result of executing the query.
- 2932 • Link to the **results** collection of Artifact Entry documents returned as a result of executing the
2933 Stored Query. Resolving this link will return a feed of summary Artifact Entry documents which
2934 matched the criteria, including parameter substitution, of the queryString in the Stored Query.
2935 This link is not included on publication of the Stored Query Entry. The server sets it during
2936 processing of the POST. This feed link SHALL have a *rel* attribute of the following form:

```
2938     rel="urn:{host}:{version}:query/{queryName}/results"
2939
```

2940 For example:

2941

```
2942     <link title="Query Results for findImplsByVersion Stored Query"
2943           href="http://example.org/s-ramp/query/findImplsByVersion/results"
2944           type="application/atom+xml;type=feed"
2945           rel="urn:x-s-ramp:2013:query:results" />
2946
```

- 2947 • Atom category element identifying the document as a Stored Query Entry:
 - 2948 ○ The entry type:
 - 2949 ▪ `scheme="urn:x-s-ramp:2013:type"`
 - 2950 ○ The only valid value for the *term* attribute here is "query"

2951

2952 Stored Query Entry documents MAY also be used as templates, allowing simple substitution of client
2953 specified parameter values during execution. The syntax for parameter substitution follows the XPath2
2954 style to represent a variable within the query filter:

2955

```
2956     ${var-name}
```

2957

2958 A value for the var-name can then be specified as part of the query invocation. Default values are not
2959 supported.

2960

2961 Note that the client may augment the Stored Query result feed URL with the result set limiting query
2962 parameters (e.g. startIndex, count) defined in Section **Error! Reference source not found.** above.

2963

2964 All Stored Queries accessible to a given client are stored as members of the s-ramp *query* collection:

2965

```
2966     {host}/s-ramp/query
```

2967 Resolving this URL using HTTP GET will return a feed of all Stored Query Entry documents available to the client.

2968 An individual Stored Query Entry document follows from that root:

2969

```
2970     {host}/s-ramp/query/{queryName}
```

2971 Example 34 below illustrates a Stored Query Entry document which supports parameter substitution:

2972

2973 *Example 34 - Stored Query Entry Document*

```
2974     <entry>
```

```
2975         <id>{urn:uuid:findImplsByVersion}</id>
```

```
2976         <updated>2009-05-26T13:13:55.013+02:00</updated>
```

```
2977         <title type="text">
```

```
2978             Stored Query to retrieve ServiceInstance documents by version
```

```
2979         </title>
```

```
2980         <published>2009-05-26T13:13:55.013+02:00</published>
```

2981

```
2982         <link href="http://example.org/s-ramp/query/findImplsByVersion"
```

```
2983             type="application/atom+xml;type=entry" rel="self" />
```

```
2984         <link href="http://example.org/s-ramp/query/findImplsByVersion"
```

```
2985             type="application/atom+xml;type=entry" rel="edit" />
```

2986

```
2987     <!--
```

```
2988         when returned by the server after publication via POST, server includes
```

```
2989         a link to the results collection for this Store Query
```

```
2990     -->
```

```

2991     <link href="http://example.org/s-ramp/query/findImplsByVersion/results"
2992           type="application/atom+xml;type=feed"
2993           rel="urn:x-s-ramp:2013:query:results" />
2994
2995     <!-- Content element identifies this as a Stored Query Entry -->
2996     <content type="text">Stored Query Entry</content>
2997
2998     <!-- S-RAMP structured extension for Stored Query Data -->
2999     <s-ramp:StoredQueryData>
3000       <s-ramp:queryName>FindImplsByVersion</s-ramp:StoredQueryName>
3001       <s-ramp:queryString>
3002         s-ramp/serviceImplementation/ServiceInstance[@version >= ${MINVERSION}]>
3003       </s-ramp:queryString>
3004       <s-ramp:propertyName>version</s-ramp:propertyName>
3005       <s-ramp:propertyName>importantPropertyToKnow</s-ramp:propertyName>
3006     </s-ramp:StoredQueryData>
3007
3008     <category term="query" label="Stored Query entry"
3009             scheme="urn:x-s-ramp:2013:type" />
3010 </entry>

```

3011
3012 Stored Query Entry documents are managed in the same way as all other Artifact Entry documents are in
3013 the Atom Binding. HTTP POST, PUT, GET and DELETE.

3014 **4 Security**

3015 The S-RAMP Specification does not attempt to define a security model for products that implement it. For
3016 the Atom Binding, the only security requirement is that at a minimum, client and server implementations
3017 MUST be capable of being configured to use HTTP Basic Authentication in conjunction with a connection
3018 made with TLS.
3019

3020 **5 Conformance**

3021 An implementation is not compliant with this specification if it fails to satisfy one or more of the MUST,
3022 SHALL or REQUIRED level requirements defined herein.

3023 Normative text within this specification takes precedence over normative outlines, which in turn take
3024 precedence over the XML Schema [XML Schema Part 1, Part 2].

3025

3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054

Appendix A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

- Joel Fleck II, Hewlett-Packard
- Jishnu Mukerji, Hewlett-Packard
- Radek Pospisil, Hewlett-Packard
- Vincent Brunssen, IBM
- John Colgrave, IBM
- Diane Jordan, IBM
- Bernard Kuflik, IBM
- Kelvin Lawrence, IBM
- Martin Smithson, IBM
- Gershon Janssen, Individual
- Carl Mattocks, Individual
- Rex Brooks, Network Centric Operations Industry Consortium
- Jian Zhang, Primeton Technologies, Inc.
- Randall Hauch, Red Hat
- Kurt Stam, Red Hat
- Eric Wittmann, Red Hat
- Steve Fanshier, Software AG, Inc.
- Gary Woods, Software AG, Inc.
- Prasad Yendluri, Software AG, Inc.
- Eric Johnson, TIBCO Software Inc.
- Senaka Fernando, WSO2
- Paul Fremantle, WSO2
- Jonathan Marsh, WSO2

3055

Appendix B. Non-Normative Text

3056

This specification provides no additional non-normative information at this time. It is expected that best practices for S-RAMP Repositories will emerge over time as this specification is adopted by vendors and users.

3057

3058

3059

Appendix C. Glossary

Term	Definition
Artifact Type	The data type of an S-RAMP artifact
Artifact Feed	An Atom feed of S-RAMP Artifacts.
Artifact Type Model	The set of all Artifact Types used in the S-RAMP specification
Backward Relationship Feed	A feed whose members are Relationship Entry document(s), whose Target Entry link matches the Artifact Entry in which the feed is included
Service Implementation Model	Data types in the S-RAMP Artifact Model which describe business concepts and relationships.
Coarse Grained View	An Atom entry document which represents an S-RAMP artifact. In its full version, it can be used to modify any and all metadata for the artifact.
Core Model	Base Artifact Types used in the S-RAMP specification.
Derived Artifact	Any S-RAMP artifact which is part of a Derived Model (e.g., XSD Model).
Fine Grained View	An optional representation of one of three classes of S-RAMP metadata (relationships, properties or classifications), which permit updates to individual metadata items without changing the Artifact Entry document.
Policy Model	Policy document related Derived Artifact Types used in the S-RAMP specification.
Relationship Entry	An Atom entry document which represents a relationship, which consists of the triplet of the Relationship Type, a link to its Source Entry and a link to its Target Entry.
Relationship Type	A name which represents the type of the relationship (e.g., "includedXsds"). Multiple relationships can share the same Relationship Type.
Relationship Type Entry	An Atom entry document instance which represents the type of the relationship (e.g., "includedXsds"). There is one such document instance for each Relationship Type.
Source Entry	An Atom entry document representing the source side artifact of a directed S-RAMP relationship.
Stored Query	A query stored in the repository which may be executed on demand.
Stored Query Entry	An Atom entry document representing a Stored Query.
Target Entry	An Atom entry document representing the target side artifact of a directed S-RAMP relationship.
WSDL Model	WSDL document related Derived Artifact Types used in the S-RAMP specification.
XSD Model	XSD document related Derived Artifact Types used in the S-RAMP specification.

3061

Appendix D. S-RAMP Atom Service Document

3062 The Atom Service Document for S-RAMP defines a set of workspaces. Each of these workspaces
3063 contains a collection which can be published in a S-RAMP compliant repository:

- 3064 • Core Model Workspace
- 3065 • WSDL Model Workspace
- 3066 • Service Implementation Model Workspace
- 3067 • SOAP WSDL Model Workspace
- 3068 • SOA Model Workspace
- 3069 • XSD Model Workspace
- 3070 • Policy Model Workspace
- 3071 • Query Model Workspace

3072 All collections are classified according to the type of entry documents which they can contain.

```
3073
3074 <?xml version="1.0" encoding="UTF-8"?>
3075 <service xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://www.w3.org/2007/app">
3076   <workspace>
3077     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Query Model</atom:title>
3078     <collection href="http://example.org/s-ramp/query">
3079       <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Query Model Objects
3080       </atom:title>
3081       <accept>application/atom+xml; type=entry</accept>
3082       <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3083         <category scheme="urn:x-s-ramp:2013:type" term="query"
3084           label="Query" xmlns="http://www.w3.org/2005/Atom"></category>
3085       </categories>
3086     </collection>
3087   </workspace>
3088   <workspace>
3089     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Core Model</atom:title>
3090     <collection href="http://example.org/s-ramp/core/xmlDocument">
3091       <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">XML Documents</atom:title>
3092       <accept>application/xml</accept>
3093       <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3094         <category scheme="urn:x-s-ramp:2013:type" term="XMLDocument"
3095           label="XML Document" xmlns="http://www.w3.org/2005/Atom"></category>
3096       </categories>
3097     </collection>
3098     <collection href="http://example.org/s-ramp/core/document">
3099       <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Documents</atom:title>
3100       <accept>application/octet-stream</accept>
3101       <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3102         <category scheme="urn:x-s-ramp:2013:type" term="Document"
3103           label="Document" xmlns="http://www.w3.org/2005/Atom"></category>
3104       </categories>
3105     </collection>
3106     <collection href="http://example.org/s-ramp/core">
3107       <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Core Model Objects
3108       </atom:title>
```

```

3109     <accept>application/zip</accept>
3110     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3111         <category scheme="urn:x-s-ramp:2013:type" term="Document"
3112             label="Document" xmlns="http://www.w3.org/2005/Atom"></category>
3113         <category scheme="urn:x-s-ramp:2013:type" term="XMLDocument"
3114             label="XML Document" xmlns="http://www.w3.org/2005/Atom"></category>
3115     </categories>
3116 </collection>
3117 </workspace>
3118 <workspace>
3119     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">WSDL Model</atom:title>
3120 <collection href="http://example.org/s-ramp/wsd1/BindingOperationOutput">
3121     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Binding Operation
3122         Outputs</atom:title>
3123     <accept></accept>
3124     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3125         <category scheme="urn:x-s-ramp:2013:type" term="BindingOperationOutput" label="Binding
3126             Operation Output" xmlns="http://www.w3.org/2005/Atom"></category>
3127     </categories>
3128 </collection>
3129 <collection href="http://example.org/s-ramp/wsd1/BindingOperation">
3130     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Binding
3131         Operations</atom:title>
3132     <accept></accept>
3133     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3134         <category scheme="urn:x-s-ramp:2013:type" term="BindingOperation"
3135             label="Binding Operation" xmlns="http://www.w3.org/2005/Atom"></category>
3136     </categories>
3137 </collection>
3138 <collection href="http://example.org/s-ramp/wsd1/wsd1Document">
3139     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">WSDL Documents
3140 </atom:title>
3141     <accept>application/xml</accept>
3142     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3143         <category scheme="urn:x-s-ramp:2013:type" term="wsdlDocument"
3144             label="WSDL Document" xmlns="http://www.w3.org/2005/Atom"></category>
3145     </categories>
3146 </collection>
3147 <collection href="http://example.org/s-ramp/wsd1/Binding">
3148     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Bindings</atom:title>
3149     <accept></accept>
3150     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3151         <category scheme="urn:x-s-ramp:2013:type" term="Binding"
3152             label="Binding" xmlns="http://www.w3.org/2005/Atom"></category>
3153     </categories>
3154 </collection>
3155 <collection href="http://example.org/s-ramp/wsd1/OperationInput">
3156     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Operation Inputs
3157 </atom:title>
3158     <accept></accept>
3159     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3160         <category scheme="urn:x-s-ramp:2013:type" term="OperationInput"
3161             label="Operation Input" xmlns="http://www.w3.org/2005/Atom"></category>
3162     </categories>
3163 </collection>

```

```

3164 <collection href="http://example.org/s-ramp/wsd1/Message">
3165   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Messages</atom:title>
3166   <accept></accept>
3167   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3168     <category scheme="urn:x-s-ramp:2013:type" term="Message"
3169       label="Message" xmlns="http://www.w3.org/2005/Atom"></category>
3170   </categories>
3171 </collection>
3172 <collection href="http://example.org/s-ramp/wsd1/Fault">
3173   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Faults
3174   </atom:title>
3175   <accept></accept>
3176   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3177     <category scheme="urn:x-s-ramp:2013:type" term="Fault"
3178       label="Fault" xmlns="http://www.w3.org/2005/Atom"></category>
3179   </categories>
3180 </collection>
3181 <collection href="http://example.org/s-ramp/wsd1/Operation">
3182   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Operations</atom:title>
3183   <accept></accept>
3184   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3185     <category scheme="urn:x-s-ramp:2013:type" term="Operation"
3186       label="Operation" xmlns="http://www.w3.org/2005/Atom"></category>
3187   </categories>
3188 </collection>
3189 <collection href="http://example.org/s-ramp/wsd1">
3190   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">WSDL Model Objects
3191   </atom:title>
3192   <accept>application/zip</accept>
3193   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3194     <category scheme="urn:x-s-ramp:2013:type" term="Wsd1Document"
3195       label="WSDL Document" xmlns="http://www.w3.org/2005/Atom"></category>
3196     <category scheme="urn:x-s-ramp:2013:type" term="Wsd1DerivedArtifactType"
3197       label="WSDL Derived Artifact" xmlns="http://www.w3.org/2005/Atom"></category>
3198     <category scheme="urn:x-s-ramp:2013:type" term="NamedWsd1DerivedArtifactType"
3199       label="Named WSDL Derived Artifact"
3200       xmlns="http://www.w3.org/2005/Atom"></category>
3201     <category scheme="urn:x-s-ramp:2013:type" term="Service"
3202       label="Service" xmlns="http://www.w3.org/2005/Atom"></category>
3203     <category scheme="urn:x-s-ramp:2013:type" term="Port"
3204       label="Port" xmlns="http://www.w3.org/2005/Atom"></category>
3205     <category scheme="urn:x-s-ramp:2013:type" term="Wsd1Extension"
3206       label="WSDL Extension" xmlns="http://www.w3.org/2005/Atom"></category>
3207     <category scheme="urn:x-s-ramp:2013:type" term="Part"
3208       label="Part" xmlns="http://www.w3.org/2005/Atom"></category>
3209     <category scheme="urn:x-s-ramp:2013:type" term="Message"
3210       label="Message" xmlns="http://www.w3.org/2005/Atom"></category>
3211     <category scheme="urn:x-s-ramp:2013:type" term="Fault"
3212       label="Fault" xmlns="http://www.w3.org/2005/Atom"></category>
3213     <category scheme="urn:x-s-ramp:2013:type" term="PortType"
3214       label="Port Type" xmlns="http://www.w3.org/2005/Atom"></category>
3215     <category scheme="urn:x-s-ramp:2013:type" term="Operation"
3216       label="Operation" xmlns="http://www.w3.org/2005/Atom"></category>
3217     <category scheme="urn:x-s-ramp:2013:type" term="OperationInput"
3218       label="Operation Input" xmlns="http://www.w3.org/2005/Atom"></category>

```

```

3219     <category scheme="urn:x-s-ramp:2013:type" term="OperationOutput"
3220         label="Operation Output" xmlns="http://www.w3.org/2005/Atom"></category>
3221     <category scheme="urn:x-s-ramp:2013:type" term="Binding"
3222         label="Binding" xmlns="http://www.w3.org/2005/Atom"></category>
3223     <category scheme="urn:x-s-ramp:2013:type" term="BindingOperation"
3224         label="Binding Operation" xmlns="http://www.w3.org/2005/Atom"></category>
3225     <category scheme="urn:x-s-ramp:2013:type" term="BindingOperationInput"
3226         label="Binding Operation Input" xmlns="http://www.w3.org/2005/Atom"></category>
3227     <category scheme="urn:x-s-ramp:2013:type" term="BindingOperationOutput"
3228         label="Binding Operation Output"
3229         xmlns="http://www.w3.org/2005/Atom"></category>
3230     <category scheme="urn:x-s-ramp:2013:type" term="BindingOperationFault"
3231         label="Binding Operation Fault" xmlns="http://www.w3.org/2005/Atom"></category>
3232     </categories>
3233 </collection>
3234 <collection href="http://example.org/s-ramp/wsd1/wsd1Extension">
3235     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">WSDL Extensions
3236     </atom:title>
3237     <accept></accept>
3238     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3239         <category scheme="urn:x-s-ramp:2013:type" term="wsdlExtension"
3240             label="WSDL Extension" xmlns="http://www.w3.org/2005/Atom"></category>
3241     </categories>
3242 </collection>
3243 <collection href="http://example.org/s-ramp/wsd1/wsd1DerivedArtifactType">
3244     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">WSDL Derived Artifacts
3245     </atom:title>
3246     <accept></accept>
3247     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3248         <category scheme="urn:x-s-ramp:2013:type" term="wsdlDerivedArtifactType"
3249             label="WSDL Derived Artifact" xmlns="http://www.w3.org/2005/Atom"></category>
3250     </categories>
3251 </collection>
3252 <collection href="http://example.org/s-ramp/wsd1/OperationOutput">
3253     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Operation Outputs
3254     </atom:title>
3255     <accept></accept>
3256     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3257         <category scheme="urn:x-s-ramp:2013:type" term="OperationOutput"
3258             label="Operation Output" xmlns="http://www.w3.org/2005/Atom"></category>
3259     </categories>
3260 </collection>
3261 <collection href="http://example.org/s-ramp/wsd1/NamedWsd1DerivedArtifactType">
3262     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Named WSDL Derived
3263     Artifacts</atom:title>
3264     <accept></accept>
3265     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3266         <category scheme="urn:x-s-ramp:2013:type" term="NamedwsdlDerivedArtifactType"
3267             label="Named WSDL Derived Artifact"
3268             xmlns="http://www.w3.org/2005/Atom"></category>
3269     </categories>
3270 </collection>
3271 <collection href="http://example.org/s-ramp/wsd1/Port">
3272     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Ports</atom:title>
3273     <accept></accept>

```



```

3274     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3275         <category scheme="urn:x-s-ramp:2013:type" term="Port"
3276             label="Port" xmlns="http://www.w3.org/2005/Atom"></category>
3277     </categories>
3278 </collection>
3279 <collection href="http://example.org/s-ramp/wsd1/Part">
3280     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Parts</atom:title>
3281     <accept></accept>
3282     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3283         <category scheme="urn:x-s-ramp:2013:type" term="Part"
3284             label="Part" xmlns="http://www.w3.org/2005/Atom"></category>
3285     </categories>
3286 </collection>
3287 <collection href="http://example.org/s-ramp/wsd1/PortType">
3288     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Port Types</atom:title>
3289     <accept></accept>
3290     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3291         <category scheme="urn:x-s-ramp:2013:type" term="PortType"
3292             label="Port Type" xmlns="http://www.w3.org/2005/Atom"></category>
3293     </categories>
3294 </collection>
3295 <collection href="http://example.org/s-ramp/wsd1/BindingOperationFault">
3296     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Binding Operation
3297     Faults</atom:title>
3298     <accept></accept>
3299     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3300         <category scheme="urn:x-s-ramp:2013:type" term="BindingOperationFault"
3301             label="Binding Operation Fault" xmlns="http://www.w3.org/2005/Atom"></category>
3302     </categories>
3303 </collection>
3304 <collection href="http://example.org/s-ramp/wsd1/BindingOperationInput">
3305     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Binding Operation
3306     Inputs</atom:title>
3307     <accept></accept>
3308     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3309         <category scheme="urn:x-s-ramp:2013:type" term="BindingOperationInput"
3310             label="Binding Operation Input" xmlns="http://www.w3.org/2005/Atom"></category>
3311     </categories>
3312 </collection>
3313 <collection href="http://example.org/s-ramp/wsd1/Service">
3314     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service</atom:title>
3315     <accept>application/atom+xml; type=entry</accept>
3316     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3317         <category scheme="urn:x-s-ramp:2013:type" term="Service"
3318             label="Service" xmlns="http://www.w3.org/2005/Atom"></category>
3319     </categories>
3320 </collection>
3321 </workspace>
3322 <workspace>
3323     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service Implementation
3324     Model</atom:title>
3325     <collection href="http://example.org/s-ramp/serviceImplementation/ServiceOperation">
3326         <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service
3327         Operations</atom:title>
3328         <accept>application/atom+xml; type=entry</accept>

```

```

3329     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3330         <category scheme="urn:x-s-ramp:2013:type" term="ServiceOperation"
3331             label="Service Operation" xmlns="http://www.w3.org/2005/Atom"></category>
3332     </categories>
3333 </collection>
3334 <collection href="http://example.org/s-ramp/serviceImplementation/ServiceInstance">
3335     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service
3336     Instances</atom:title>
3337     <accept>application/atom+xml; type=entry</accept>
3338     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3339         <category scheme="urn:x-s-ramp:2013:type" term="ServiceInstance"
3340             label="Service Instance" xmlns="http://www.w3.org/2005/Atom"></category>
3341     </categories>
3342 </collection>
3343 <collection href="http://example.org/s-ramp/serviceImplementation/ServiceEndpoint">
3344     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service
3345     Endpoints</atom:title>
3346     <accept>application/atom+xml; type=entry</accept>
3347     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3348         <category scheme="urn:x-s-ramp:2013:type" term="ServiceEndpoint"
3349             label="Service Endpoint" xmlns="http://www.w3.org/2005/Atom"></category>
3350     </categories>
3351 </collection>
3352 <collection href="http://example.org/s-ramp/serviceImplementation">
3353     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service Implementation
3354     Objects</atom:title>
3355     <accept></accept>
3356     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3357         <category scheme="urn:x-s-ramp:2013:type" term="ServiceEndpoint"
3358             label="Service Endpoint" xmlns="http://www.w3.org/2005/Atom"></category>
3359         <category scheme="urn:x-s-ramp:2013:type" term="ServiceInstance"
3360             label="Service Instance" xmlns="http://www.w3.org/2005/Atom"></category>
3361         <category scheme="urn:x-s-ramp:2013:type" term="ServiceOperation"
3362             label="Service Operation" xmlns="http://www.w3.org/2005/Atom"></category>
3363         <category scheme="urn:x-s-ramp:2013:type" term="Organization"
3364             label="Organization" xmlns="http://www.w3.org/2005/Atom"></category>
3365     </categories>
3366 </collection>
3367 </workspace>
3368 <workspace>
3369     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">SOAP WSDL Model
3370 </atom:title>
3371 <collection href="http://example.org/s-ramp/soapwsdl/SoapBinding">
3372     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">SOAP Bindings</atom:title>
3373     <accept></accept>
3374     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3375         <category scheme="urn:x-s-ramp:2013:type" term="SoapBinding"
3376             label="SOAP Binding" xmlns="http://www.w3.org/2005/Atom"></category>
3377     </categories>
3378 </collection>
3379 <collection href="http://example.org/s-ramp/soapwsdl/SoapAddress">
3380     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">SOAP Addresses
3381 </atom:title>
3382     <accept></accept>
3383     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">

```

```

3384         <category scheme="urn:x-s-ramp:2013:type" term="SoapAddress"
3385             label="SOAP Address" xmlns="http://www.w3.org/2005/Atom"></category>
3386     </categories>
3387 </collection>
3388 <collection href="http://example.org/s-ramp/soapwsdl1">
3389     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">SOAP WSDL Model
3390     Objects</atom:title>
3391     <accept>application/zip</accept>
3392     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3393         <category scheme="urn:x-s-ramp:2013:type" term="SoapAddress"
3394             label="SOAP Address" xmlns="http://www.w3.org/2005/Atom"></category>
3395         <category scheme="urn:x-s-ramp:2013:type" term="SoapBinding"
3396             label="SOAP Binding" xmlns="http://www.w3.org/2005/Atom"></category>
3397     </categories>
3398 </collection>
3399 </workspace>
3400 <workspace>
3401     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">SOA Model</atom:title>
3402     <collection href="http://example.org/s-ramp/soa/ServiceInterface">
3403         <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service
3404         Interface</atom:title>
3405         <accept>application/atom+xml;type=entry</accept>
3406         <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3407             <category scheme="urn:x-s-ramp:2013:type" term="ServiceInterface"
3408                 label="Service Interface" xmlns="http://www.w3.org/2005/Atom"></category>
3409         </categories>
3410     </collection>
3411     <collection href="http://example.org/s-ramp/soa">
3412         <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">SOA Model Objects
3413         </atom:title>
3414         <accept>application/zip</accept>
3415         <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3416             <category scheme="urn:x-s-ramp:2013:type" term="HumanActor"
3417                 label="HumanActor" xmlns="http://www.w3.org/2005/Atom"></category>
3418             <category scheme="urn:x-s-ramp:2013:type" term="Choreography"
3419                 label="Choreography" xmlns="http://www.w3.org/2005/Atom"></category>
3420             <category scheme="urn:x-s-ramp:2013:type" term="ChoreographyProcess"
3421                 label="Choreography Process" xmlns="http://www.w3.org/2005/Atom"></category>
3422             <category scheme="urn:x-s-ramp:2013:type" term="Collaboration"
3423                 label="Collaboration" xmlns="http://www.w3.org/2005/Atom"></category>
3424             <category scheme="urn:x-s-ramp:2013:type" term="CollaborationProcess"
3425                 label="Collaboration Process" xmlns="http://www.w3.org/2005/Atom"></category>
3426             <category scheme="urn:x-s-ramp:2013:type" term="Composition"
3427                 label="Composition" xmlns="http://www.w3.org/2005/Atom"></category>
3428             <category scheme="urn:x-s-ramp:2013:type" term="Effect"
3429                 label="Effect" xmlns="http://www.w3.org/2005/Atom"></category>
3430             <category scheme="urn:x-s-ramp:2013:type" term="Element"
3431                 label="Element" xmlns="http://www.w3.org/2005/Atom"></category>
3432             <category scheme="urn:x-s-ramp:2013:type" term="Event"
3433                 label="Event" xmlns="http://www.w3.org/2005/Atom"></category>
3434             <category scheme="urn:x-s-ramp:2013:type" term="InformationType"
3435                 label="Information Type" xmlns="http://www.w3.org/2005/Atom"></category>
3436             <category scheme="urn:x-s-ramp:2013:type" term="Orchestration"
3437                 label="Orchestration" xmlns="http://www.w3.org/2005/Atom"></category>
3438             <category scheme="urn:x-s-ramp:2013:type" term="OrchestrationProcess"

```

```

3439         label="Orchestration Process" xmlns="http://www.w3.org/2005/Atom"></category>
3440     <category scheme="urn:x-s-ramp:2013:type" term="Policy"
3441         label="Policy" xmlns="http://www.w3.org/2005/Atom"></category>
3442     <category scheme="urn:x-s-ramp:2013:type" term="PolicySubject"
3443         label="Policy Subject" xmlns="http://www.w3.org/2005/Atom"></category>
3444     <category scheme="urn:x-s-ramp:2013:type" term="Process"
3445         label="Process" xmlns="http://www.w3.org/2005/Atom"></category>
3446     <category scheme="urn:x-s-ramp:2013:type" term="Service"
3447         label="Service" xmlns="http://www.w3.org/2005/Atom"></category>
3448     <category scheme="urn:x-s-ramp:2013:type" term="ServiceContract"
3449         label="Service Contract" xmlns="http://www.w3.org/2005/Atom"></category>
3450     <category scheme="urn:x-s-ramp:2013:type" term="ServiceComposition"
3451         label="Service Composition" xmlns="http://www.w3.org/2005/Atom"></category>
3452     <category scheme="urn:x-s-ramp:2013:type" term="ServiceInterface"
3453         label="Service Interface" xmlns="http://www.w3.org/2005/Atom"></category>
3454     <category scheme="urn:x-s-ramp:2013:type" term="System"
3455         label="System" xmlns="http://www.w3.org/2005/Atom"></category>
3456     <category scheme="urn:x-s-ramp:2013:type" term="Task"
3457         label="Task" xmlns="http://www.w3.org/2005/Atom"></category>
3458 </categories>
3459 </collection>
3460 <collection href="http://example.org/s-ramp/soa/CollaborationProcess">
3461     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Collaboration
3462     Process</atom:title>
3463     <accept>application/atom+xml;type=entry</accept>
3464     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3465         <category scheme="urn:x-s-ramp:2013:type" term="CollaborationProcess"
3466             label="Collaboration Process" xmlns="http://www.w3.org/2005/Atom"></category>
3467     </categories>
3468 </collection>
3469 <collection href="http://example.org/s-ramp/soa/Process">
3470     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Process</atom:title>
3471     <accept>application/atom+xml;type=entry</accept>
3472     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3473         <category scheme="urn:x-s-ramp:2013:type" term="Process"
3474             label="Process" xmlns="http://www.w3.org/2005/Atom"></category>
3475     </categories>
3476 </collection>
3477 <collection href="http://example.org/s-ramp/serviceImplementation/HumanActor">
3478     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">HumanActor</atom:title>
3479     <accept>application/atom+xml;type=entry</accept>
3480     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3481         <category scheme="urn:x-s-ramp:2013:type" term="HumanActor"
3482             label="HumanActor" xmlns="http://www.w3.org/2005/Atom"></category>
3483     </categories>
3484 </collection>
3485 <collection href="http://example.org/s-ramp/soa/Collaboration">
3486     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Collaboration</atom:title>
3487     <accept></accept>
3488     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3489         <category scheme="urn:x-s-ramp:2013:type" term="Collaboration"
3490             label="Collaboration" xmlns="http://www.w3.org/2005/Atom"></category>
3491     </categories>
3492 </collection>
3493 <collection href="http://example.org/s-ramp/soa/Composition">

```

```

3494     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Composition</atom:title>
3495     <accept>application/atom+xml;type=entry</accept>
3496     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3497         <category scheme="urn:x-s-ramp:2013:type" term="Composition"
3498             label="Composition" xmlns="http://www.w3.org/2005/Atom"></category>
3499     </categories>
3500 </collection>
3501 <collection href="http://example.org/s-ramp/soa/Element">
3502     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Element</atom:title>
3503     <accept>application/atom+xml;type=entry</accept>
3504     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3505         <category scheme="urn:x-s-ramp:2013:type" term="Element"
3506             label="Element" xmlns="http://www.w3.org/2005/Atom"></category>
3507     </categories>
3508 </collection>
3509 <collection href="http://example.org/s-ramp/soa/Event">
3510     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Event</atom:title>
3511     <accept>application/atom+xml;type=entry</accept>
3512     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3513         <category scheme="urn:x-s-ramp:2013:type" term="Event"
3514             label="Event" xmlns="http://www.w3.org/2005/Atom"></category>
3515     </categories>
3516 </collection>
3517 <collection href="http://example.org/s-ramp/soa/Orchestration">
3518     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Orchestration</atom:title>
3519     <accept>application/atom+xml;type=entry</accept>
3520     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3521         <category scheme="urn:x-s-ramp:2013:type" term="Orchestration"
3522             label="Orchestration" xmlns="http://www.w3.org/2005/Atom"></category>
3523     </categories>
3524 </collection>
3525 <collection href="http://example.org/s-ramp/soa/PolicySubject">
3526     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Policy Subject
3527 </atom:title>
3528     <accept>application/atom+xml;type=entry</accept>
3529     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3530         <category scheme="urn:x-s-ramp:2013:type" term="PolicySubject"
3531             label="Policy Subject" xmlns="http://www.w3.org/2005/Atom"></category>
3532     </categories>
3533 </collection>
3534 <collection href="http://example.org/s-ramp/soa/InformationType">
3535     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Information Type
3536 </atom:title>
3537     <accept>application/atom+xml;type=entry</accept>
3538     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3539         <category scheme="urn:x-s-ramp:2013:type" term="InformationType"
3540             label="Information Type" xmlns="http://www.w3.org/2005/Atom"></category>
3541     </categories>
3542 </collection>
3543 <collection href="http://example.org/s-ramp/soa/Task">
3544     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Task</atom:title>
3545     <accept>application/atom+xml;type=entry</accept>
3546     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3547         <category scheme="urn:x-s-ramp:2013:type" term="Task"
3548             label="Task" xmlns="http://www.w3.org/2005/Atom"></category>

```

```

3549     </categories>
3550 </collection>
3551 <collection href="http://example.org/s-ramp/soa/System">
3552   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">System</atom:title>
3553   <accept>application/atom+xml;type=entry</accept>
3554   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3555     <category scheme="urn:x-s-ramp:2013:type" term="System"
3556       label="System" xmlns="http://www.w3.org/2005/Atom"></category>
3557   </categories>
3558 </collection>
3559 <collection href="http://example.org/s-ramp/soa/Policy">
3560   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Policy</atom:title>
3561   <accept>application/atom+xml;type=entry</accept>
3562   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3563     <category scheme="urn:x-s-ramp:2013:type" term="Policy"
3564       label="Policy" xmlns="http://www.w3.org/2005/Atom"></category>
3565   </categories>
3566 </collection>
3567 <collection href="http://example.org/s-ramp/soa/Choreography">
3568   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Choreography</atom:title>
3569   <accept>application/atom+xml;type=entry</accept>
3570   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3571     <category scheme="urn:x-s-ramp:2013:type" term="Choreography"
3572       label="Choreography" xmlns="http://www.w3.org/2005/Atom"></category>
3573   </categories>
3574 </collection>
3575 <collection href="http://example.org/s-ramp/soa/Effect">
3576   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Effect</atom:title>
3577   <accept>application/atom+xml;type=entry</accept>
3578   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3579     <category scheme="urn:x-s-ramp:2013:type" term="Effect"
3580       label="Effect" xmlns="http://www.w3.org/2005/Atom"></category>
3581   </categories>
3582 </collection>
3583 <collection href="http://example.org/s-ramp/soa/ServiceContract">
3584   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service Contract
3585   </atom:title>
3586   <accept>application/atom+xml;type=entry</accept>
3587   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3588     <category scheme="urn:x-s-ramp:2013:type" term="ServiceContract"
3589       label="Service Contract" xmlns="http://www.w3.org/2005/Atom"></category>
3590   </categories>
3591 </collection>
3592 <collection href="http://example.org/s-ramp/soa/OrchestrationProcess">
3593   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Orchestration Process
3594   </atom:title>
3595   <accept>application/atom+xml;type=entry</accept>
3596   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3597     <category scheme="urn:x-s-ramp:2013:type" term="OrchestrationProcess"
3598       label="Orchestration Process" xmlns="http://www.w3.org/2005/Atom"></category>
3599   </categories>
3600 </collection>
3601 <collection href="http://example.org/s-ramp/serviceImplementation/Organization">
3602   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Organization</atom:title>
3603   <accept>application/atom+xml; type=entry</accept>

```

```

3604     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3605         <category scheme="urn:x-s-ramp:2013:type" term="Organization"
3606             label="Organization" xmlns="http://www.w3.org/2005/Atom"></category>
3607     </categories>
3608 </collection>
3609 <collection href="http://example.org/s-ramp/soa/Service">
3610     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Service</atom:title>
3611     <accept>application/atom+xml; type=entry</accept>
3612     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3613         <category scheme="urn:x-s-ramp:2013:type" term="Service"
3614             label="Service" xmlns="http://www.w3.org/2005/Atom"></category>
3615     </categories>
3616 </collection>
3617 <collection href="http://example.org/s-ramp/soa/ChoreographyProcess">
3618     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Choreography Process
3619 </atom:title>
3620     <accept>application/atom+xml;type=entry</accept>
3621     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3622         <category scheme="urn:x-s-ramp:2013:type" term="ChoreographyProcess"
3623             label="Choreography Process" xmlns="http://www.w3.org/2005/Atom"></category>
3624     </categories>
3625 </collection>
3626 </workspace>
3627 <workspace>
3628     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">XSD Model</atom:title>
3629 <collection href="http://example.org/s-ramp/xsd/XsdType">
3630     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">XSD Types</atom:title>
3631     <accept></accept>
3632     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3633         <category scheme="urn:x-s-ramp:2013:type" term="XsdType"
3634             label="XSD Type" xmlns="http://www.w3.org/2005/Atom"></category>
3635     </categories>
3636 </collection>
3637 <collection href="http://example.org/s-ramp/xsd/ElementDeclaration">
3638     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Element Declarations
3639 </atom:title>
3640     <accept></accept>
3641     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3642         <category scheme="urn:x-s-ramp:2013:type" term="ElementDeclaration"
3643             label="Element Declaration" xmlns="http://www.w3.org/2005/Atom"></category>
3644     </categories>
3645 </collection>
3646 <collection href="http://example.org/s-ramp/xsd/AttributeDeclaration">
3647     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Attribute Declarations
3648 </atom:title>
3649     <accept></accept>
3650     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3651         <category scheme="urn:x-s-ramp:2013:type" term="AttributeDeclaration"
3652             label="Attribute Declaration" xmlns="http://www.w3.org/2005/Atom"></category>
3653     </categories>
3654 </collection>
3655 <collection href="http://example.org/s-ramp/xsd/ComplexTypeDeclaration">
3656     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Complex Type Declarations
3657 </atom:title>
3658     <accept></accept>

```

```

3659     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3660         <category scheme="urn:x-s-ramp:2013:type" term="ComplexTypeDeclaration"
3661             label="Complex Type Declaration"
3662             xmlns="http://www.w3.org/2005/Atom"></category>
3663     </categories>
3664 </collection>
3665 <collection href="http://example.org/s-ramp/xsd/SimpleTypeDeclaration">
3666     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Simple Type Declarations
3667 </atom:title>
3668     <accept></accept>
3669     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3670         <category scheme="urn:x-s-ramp:2013:type" term="SimpleTypeDeclaration"
3671             label="Simple Type Declaration" xmlns="http://www.w3.org/2005/Atom"></category>
3672     </categories>
3673 </collection>
3674 <collection href="http://example.org/s-ramp/xsd/XsdDocument">
3675     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">XSD Documents</atom:title>
3676     <accept>application/xml</accept>
3677     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3678         <category scheme="urn:x-s-ramp:2013:type" term="XsdDocument"
3679             label="XSD Document" xmlns="http://www.w3.org/2005/Atom"></category>
3680     </categories>
3681 </collection>
3682 <collection href="http://example.org/s-ramp/xsd">
3683     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">XSD Model Objects
3684 </atom:title>
3685     <accept>application/zip</accept>
3686     <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3687         <category scheme="urn:x-s-ramp:2013:type" term="XsdDocument"
3688             label="XSD Document" xmlns="http://www.w3.org/2005/Atom"></category>
3689         <category scheme="urn:x-s-ramp:2013:type" term="AttributeDeclaration"
3690             label="Attribute Declaration" xmlns="http://www.w3.org/2005/Atom"></category>
3691         <category scheme="urn:x-s-ramp:2013:type" term="XsdType"
3692             label="XSD Type" xmlns="http://www.w3.org/2005/Atom"></category>
3693         <category scheme="urn:x-s-ramp:2013:type" term="ElementDeclaration"
3694             label="Element Declaration" xmlns="http://www.w3.org/2005/Atom"></category>
3695         <category scheme="urn:x-s-ramp:2013:type" term="SimpleTypeDeclaration"
3696             label="Simple Type Declaration" xmlns="http://www.w3.org/2005/Atom"></category>
3697         <category scheme="urn:x-s-ramp:2013:type" term="ComplexTypeDeclaration"
3698             label="Complex Type Declaration"
3699             xmlns="http://www.w3.org/2005/Atom"></category>
3700     </categories>
3701 </collection>
3702 </workspace>
3703 <workspace>
3704     <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Policy Model</atom:title>
3705     <collection href="http://example.org/s-ramp/policy/PolicyDocument">
3706         <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Policy Documents
3707         </atom:title>
3708         <accept>application/xml</accept>
3709         <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3710             <category scheme="urn:x-s-ramp:2013:type" term="PolicyDocument"
3711                 label="Policy Document" xmlns="http://www.w3.org/2005/Atom"></category>
3712         </categories>
3713     </collection>

```



```

3714 <collection href="http://example.org/s-ramp/policy">
3715   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Policy Model Objects
3716   </atom:title>
3717   <accept>application/zip</accept>
3718   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3719     <category scheme="urn:x-s-ramp:2013:type" term="PolicyDocument"
3720       label="Policy Document" xmlns="http://www.w3.org/2005/Atom"></category>
3721     <category scheme="urn:x-s-ramp:2013:type" term="PolicyExpression"
3722       label="Policy Expression" xmlns="http://www.w3.org/2005/Atom"></category>
3723     <category scheme="urn:x-s-ramp:2013:type" term="PolicyAttachment"
3724       label="Policy Attachment" xmlns="http://www.w3.org/2005/Atom"></category>
3725   </categories>
3726 </collection>
3727 <collection href="http://example.org/s-ramp/policy/PolicyAttachment">
3728   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Policy Attachments
3729   </atom:title>
3730   <accept></accept>
3731   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3732     <category scheme="urn:x-s-ramp:2013:type" term="PolicyAttachment"
3733       label="Policy Attachment" xmlns="http://www.w3.org/2005/Atom"></category>
3734   </categories>
3735 </collection>
3736 <collection href="http://example.org/s-ramp/policy/PolicyExpression">
3737   <atom:title type="text" xmlns:atom="http://www.w3.org/2005/Atom">Policy Expressions
3738   </atom:title>
3739   <accept></accept>
3740   <categories fixed="yes" xmlns:atom="http://www.w3.org/2005/Atom">
3741     <category scheme="urn:x-s-ramp:2013:type" term="PolicyExpression"
3742       label="Policy Expression" xmlns="http://www.w3.org/2005/Atom"></category>
3743   </categories>
3744 </collection>
3745 </workspace>
3746 </service>

```

3747

Appendix E. Notional S-RAMP URI Space

3748 The suggested URI space for S-RAMP is organized according to the logical structure of the S-RAMP
3749 Artifact Type Model. All S-RAMP artifacts can be mapped to this URI space. The following URI syntax
3750 applies:

3751

/s-ramp/Primary-Qualifier}/{Secondary-Qualifier}

3752

3753
3754 Typically, the Primary-Qualifier corresponds to the name of the Artifact Model, and the Secondary-
3755 Qualifier corresponds to an artifact type name. Exceptions include query and Service Document
3756 references. Table 5 defines the valid values for the components of S-RAMP URIs.

3757

3758 *Table 5 - S-RAMP URI Space*

Primary Qualifier	Secondary Qualifier
core	{Refer to the Artifact Type values in the core section of Table 6 of the Foundation Document}
XmlDocument	
xsd	{Refer to the Artifact Type values in the xsd section of Table 6 of the Foundation Document}
policy	{Refer to the Artifact Type values in the policy section of Table 6 of the Foundation Document}
soapWsdI	{Refer to the Artifact Type values in the soapwsdl section of Table 6 of the Foundation Document}
wSDL	{Refer to the Artifact Type values in the wsdl section of Table 6 of the Foundation Document}
soa	{Refer to the Artifact Type values in the soa section of Table 6 of the Foundation Document}
serviceImplementation	{Refer to the Artifact Type values in the xsd section of Table 6 of the Foundation Document}
ext	{Extended Artifact Type}
query	{StoredQuery name}
	Results
servicedocument	

3759

Appendix F. S-RAMP Atom Binding Schema

3760 This appendix describes the S-RAMP structured extensions used in the Atom Binding. For convenience,
3761 an S-RAMP Atom Binding Schema XSD file is also provided at:

3762

3763 <http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0/atombinding.xsd>

```
3764 <xsd:schema targetNamespace="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0" version="1.0"
3765 elementFormDefault="qualified" xmlns:tns="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0"
3766 xmlns:s-ramp="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0"
3767 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
3768 instance">
3769 <!--
3770 (c) 2010 Hewlett-Packard Company (HP), International Business Machines
3771 Corporation (IBM), Software AG (SAG) and TIBCO Software Inc. All
3772 rights reserved. Permission to copy and display the SOA Repository
3773 Artifact Model and Protocol (the "Specification"), in any medium
3774 without fee or royalty is hereby granted by Hewlett-Packard Company
3775 (HP), International Business Machines Corporation (IBM), Software AG
3776 (SAG) and TIBCO Software Inc. (collectively, the "Authors"), provided
3777 that you include the following on ALL copies of this document or
3778 portions thereof, that you make:
3779
3780 1. A link or URL to this document at this location:
3781 http://s-ramp.org/2010/s-ramp/specification/documents/{this document
3782 name}
3783 2. The copyright notice as shown in the Specification.
3784
3785 The Authors each agree to grant you a royalty-free license, under
3786 reasonable, non-discriminatory terms and conditions to their
3787 respective patents that they deem necessary to implement the "SOA
3788 Repository Artifact Model and Protocol" Specification, including all
3789 its constituent documents. THIS DOCUMENT IS PROVIDED "AS IS," AND THE
3790 AUTHORS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED,
3791 INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS
3792 FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE
3793 CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE
3794 IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY
3795 PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. THE AUTHORS WILL NOT
3796 BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR
3797 CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR
3798 DISTRIBUTION OF THIS DOCUMENT.
3799 -->
3800
3801 <xsd:include schemaLocation="serviceimplementationmodel.xsd" />
3802 <xsd:include schemaLocation="coremodel.xsd" />
3803 <xsd:include schemaLocation="wsdlmodel.xsd" />
3804 <xsd:include schemaLocation="xsdmodel.xsd" />
3805 <xsd:include schemaLocation="policymodel.xsd" />
3806 <xsd:include schemaLocation="soamodel.xsd" />
3807 <xsd:include schemaLocation="soapwsdlmodel.xsd" />
3808
3809 <!-- Base type for all Derived Artifacts in S-RAMP -->
3810 <xsd:element name="artifact">
3811 <xsd:complexType>
3812 <xsd:sequence>
3813 <xsd:choice>
3814 <!-- Concrete Artifact Types from Core Model -->
3815 <xsd:element name="Document" type="s-ramp:Document" minOccurs="1" maxOccurs="1" />
3816 <xsd:element name="XmlDocument" type="s-ramp:XmlDocument" minOccurs="1" maxOccurs="1"
3817 />
3818 <xsd:element name="ExtendedArtifactType" type="s-ramp:ExtendedArtifactType"
3819 minOccurs="1" maxOccurs="1" />
3820 <xsd:element name="ExtendedDocument" type="s-ramp:ExtendedDocument" minOccurs="1"
3821 maxOccurs="1" />
3822
3823 <!-- Concrete Artifact Types from Service Implementation Model -->
3824 <xsd:element name="Organization" type="s-ramp:Organization" minOccurs="1" maxOccurs="1"
3825 />
3826 <xsd:element name="ServiceEndpoint" type="s-ramp:ServiceEndpoint" minOccurs="1"
3827 maxOccurs="1" />
3828 <xsd:element name="ServiceInstance" type="s-ramp:ServiceInstance" minOccurs="1"
3829 maxOccurs="1" />
3830 <xsd:element name="ServiceOperation" type="s-ramp:ServiceOperation" minOccurs="1"
3831 maxOccurs="1" />
```

```

3832
3833     <!-- Concrete Artifact Types from SOA Model -->
3834     <xsd:element name="Actor" type="s-ramp:Actor" minOccurs="1" maxOccurs="1" />
3835     <xsd:element name="Choreography" type="s-ramp:Choreography" minOccurs="1" maxOccurs="1"
3836 />
3837     <xsd:element name="ChoreographyProcess" type="s-ramp:ChoreographyProcess" minOccurs="1"
3838 maxOccurs="1" />
3839     <xsd:element name="Collaboration" type="s-ramp:Collaboration" minOccurs="1"
3840 maxOccurs="1" />
3841     <xsd:element name="CollaborationProcess" type="s-ramp:CollaborationProcess"
3842 minOccurs="1" maxOccurs="1" />
3843     <xsd:element name="Composition" type="s-ramp:Composition" minOccurs="1" maxOccurs="1"
3844 />
3845     <xsd:element name="Effect" type="s-ramp:Effect" minOccurs="1" maxOccurs="1" />
3846     <xsd:element name="Element" type="s-ramp:Element" minOccurs="1" maxOccurs="1" />
3847     <xsd:element name="Event" type="s-ramp:Event" minOccurs="1" maxOccurs="1" />
3848     <xsd:element name="InformationType" type="s-ramp:InformationType" minOccurs="1"
3849 maxOccurs="1" />
3850     <xsd:element name="Orchestration" type="s-ramp:Orchestration" minOccurs="1"
3851 maxOccurs="1" />
3852     <xsd:element name="OrchestrationProcess" type="s-ramp:OrchestrationProcess"
3853 minOccurs="1" maxOccurs="1" />
3854     <xsd:element name="Policy" type="s-ramp:Policy" minOccurs="1" maxOccurs="1" />
3855     <xsd:element name="PolicySubject" type="s-ramp:PolicySubject" minOccurs="1"
3856 maxOccurs="1" />
3857     <xsd:element name="Process" type="s-ramp:Process" minOccurs="1" maxOccurs="1" />
3858     <xsd:element name="Service" type="s-ramp:Service" minOccurs="1" maxOccurs="1" />
3859     <xsd:element name="ServiceContract" type="s-ramp:ServiceContract" minOccurs="1"
3860 maxOccurs="1" />
3861     <xsd:element name="ServiceComposition" type="s-ramp:ServiceComposition" minOccurs="1"
3862 maxOccurs="1" />
3863     <xsd:element name="ServiceInterface" type="s-ramp:ServiceInterface" minOccurs="1"
3864 maxOccurs="1" />
3865     <xsd:element name="System" type="s-ramp:System" minOccurs="1" maxOccurs="1" />
3866     <xsd:element name="Task" type="s-ramp:Task" minOccurs="1" maxOccurs="1" />
3867
3868     <!-- Concrete Artifact Types from Policy Model -->
3869     <xsd:element name="PolicyAttachment" type="s-ramp:PolicyAttachment" minOccurs="1"
3870 maxOccurs="1" />
3871     <xsd:element name="PolicyExpression" type="s-ramp:PolicyExpression" minOccurs="1"
3872 maxOccurs="1" />
3873     <xsd:element name="PolicyDocument" type="s-ramp:PolicyDocument" minOccurs="1"
3874 maxOccurs="1" />
3875
3876     <!-- Concrete Artifact Types from XSD Model -->
3877     <xsd:element name="XsdDocument" type="s-ramp:XsdDocument" minOccurs="1" maxOccurs="1"
3878 />
3879     <xsd:element name="AttributeDeclaration" type="s-ramp:AttributeDeclaration"
3880 minOccurs="1" maxOccurs="1" />
3881     <xsd:element name="ElementDeclaration" type="s-ramp:ElementDeclaration" minOccurs="1"
3882 maxOccurs="1" />
3883     <xsd:element name="ComplexTypeDeclaration" type="s-ramp:ComplexTypeDeclaration"
3884 minOccurs="1" maxOccurs="1" />
3885     <xsd:element name="SimpleTypeDeclaration" type="s-ramp:SimpleTypeDeclaration"
3886 minOccurs="1" maxOccurs="1" />
3887
3888     <!-- Concrete Artifact Types from WSDL Model -->
3889     <xsd:element name="WsdDocument" type="s-ramp:WsdDocument" minOccurs="1" maxOccurs="1"
3890 />
3891     <xsd:element name="WsdService" type="s-ramp:WsdService" minOccurs="1" maxOccurs="1"
3892 />
3893     <xsd:element name="Port" type="s-ramp:Port" minOccurs="1" maxOccurs="1" />
3894     <xsd:element name="WsdExtension" type="s-ramp:WsdExtension" minOccurs="1"
3895 maxOccurs="1" />
3896     <xsd:element name="Part" type="s-ramp:Part" minOccurs="1" maxOccurs="1" />
3897     <xsd:element name="Message" type="s-ramp:Message" minOccurs="1" maxOccurs="1" />
3898     <xsd:element name="Fault" type="s-ramp:Fault" minOccurs="1" maxOccurs="1" />
3899     <xsd:element name="PortType" type="s-ramp:PortType" minOccurs="1" maxOccurs="1" />
3900     <xsd:element name="Operation" type="s-ramp:Operation" minOccurs="1" maxOccurs="1" />
3901     <xsd:element name="OperationInput" type="s-ramp:OperationInput" minOccurs="1"
3902 maxOccurs="1" />
3903     <xsd:element name="OperationOutput" type="s-ramp:OperationOutput" minOccurs="1"
3904 maxOccurs="1" />
3905     <xsd:element name="Binding" type="s-ramp:Binding" minOccurs="1" maxOccurs="1" />
3906     <xsd:element name="BindingOperation" type="s-ramp:BindingOperation" minOccurs="1"
3907 maxOccurs="1" />
3908     <xsd:element name="BindingOperationInput" type="s-ramp:BindingOperationInput"
3909 minOccurs="1" maxOccurs="1" />
3910     <xsd:element name="BindingOperationOutput" type="s-ramp:BindingOperationOutput"
3911 minOccurs="1" maxOccurs="1" />
3912     <xsd:element name="BindingOperationFault" type="s-ramp:BindingOperationFault"
3913 minOccurs="1" maxOccurs="1" />

```

```

3914
3915     <!-- Concrete Artifact Types from SOAP WSDL Model -->
3916     <xsd:element name="SoapAddress" type="s-ramp:SoapAddress" minOccurs="1" maxOccurs="1"
3917 />
3918     <xsd:element name="SoapBinding" type="s-ramp:SoapBinding" minOccurs="1" maxOccurs="1"
3919 />
3920     </xsd:choice>
3921   </xsd:sequence>
3922 </xsd:complexType>
3923 </xsd:element>
3924
3925 <!-- Relationship Data element used in S-RAMP Relationship Entry documents -->
3926 <xsd:element name="relationshipData">
3927   <xsd:complexType>
3928     <xsd:sequence>
3929       <xsd:element ref="s-ramp:relationshipType" minOccurs="0" maxOccurs="1" />
3930       <!-- sourceId is the UUID of the source artifact -->
3931       <xsd:element ref="tns:sourceId" minOccurs="0" maxOccurs="1" />
3932       <!-- targetId is the UUID of the target artifact -->
3933       <xsd:element ref="tns:targetId" minOccurs="1" maxOccurs="1" />
3934     </xsd:sequence>
3935   </xsd:complexType>
3936 </xsd:element>
3937
3938 <!-- Relationship Type Data element used in S-RAMP Relationship Type Entry
3939 documents. For now this only includes the s-ramp:relationshipType. -->
3940 <xsd:element name="relationshipTypeData">
3941   <xsd:complexType>
3942     <xsd:sequence>
3943       <xsd:element ref="s-ramp:relationshipType" minOccurs="1" maxOccurs="1" />
3944     </xsd:sequence>
3945   </xsd:complexType>
3946 </xsd:element>
3947
3948 <!-- Properties Data element used in S-RAMP Property Entry documents. -->
3949 <xsd:element name="propertyData">
3950   <xsd:complexType>
3951     <xsd:sequence>
3952       <xsd:element ref="s-ramp:property" minOccurs="1" maxOccurs="1" />
3953     </xsd:sequence>
3954   </xsd:complexType>
3955 </xsd:element>
3956
3957 <!-- Properties Data element used in S-RAMP Classification Entry documents. -->
3958 <xsd:element name="classificationData">
3959   <xsd:complexType>
3960     <xsd:sequence>
3961       <xsd:element ref="s-ramp:classifiedBy" minOccurs="1" maxOccurs="1" />
3962     </xsd:sequence>
3963   </xsd:complexType>
3964 </xsd:element>
3965
3966 <!-- Stored Query Data element used in S-RAMP Classification Entry documents. -->
3967 <xsd:element name="storedQueryData">
3968   <xsd:complexType>
3969     <xsd:sequence>
3970       <xsd:element name="queryName" type="xsd:string" minOccurs="1" maxOccurs="1" />
3971       <xsd:element name="queryString" type="xsd:string" minOccurs="1" maxOccurs="1" />
3972       <xsd:element ref="s-ramp:propertyName" minOccurs="0" maxOccurs="unbounded" />
3973     </xsd:sequence>
3974   </xsd:complexType>
3975 </xsd:element>
3976
3977 <xsd:element name="error">
3978   <xsd:complexType>
3979     <xsd:sequence>
3980       <xsd:element name="description" type="xsd:string" minOccurs="1" maxOccurs="1" />
3981       <xsd:element name="howtofix" type="xsd:string" minOccurs="0" maxOccurs="1" />
3982       <xsd:element name="detail" type="xsd:string" minOccurs="0" maxOccurs="1" />
3983     </xsd:sequence>
3984     <xsd:attribute name="responseCode" type="xsd:string" use="required" />
3985     <xsd:attribute name="name" type="xsd:string" use="required" />
3986     <xsd:attribute name="uuid" type="xsd:string" use="optional" />
3987   </xsd:complexType>
3988 </xsd:element>
3989
3990 </xsd:schema>

```

3991

Appendix G. S-RAMP HTTP Response Codes

3992

3993 While specific HTTP status codes are shown in the table 6 below, an S-RAMP client should be prepared
3994 to handle any status code. The S-RAMP Protocol uses the response status codes defined in HTTP to
3995 indicate the success or failure of an operation. Consult the HTTP specification [\[RFC2616\]](#) for detailed
3996 definitions of each status code.

3997

3998 *Table 6 - S-RAMP HTTP Response Codes*

3999

S-RAMP HTTP RESPONSE	Name	S-RAMP Description
Successful 2xx		
200	OK	Successful GET, PUT, DELETE
201	Created	Successful POST, artifact created successfully
Client Error 4xx		
400	Bad Request	Invalid request
401	Unauthorized	Not authorized request
403	Forbidden	Forbidden by the specification when <ul style="list-style-type: none"> - Attempt to update or delete derived content - Publishing artifacts to the wrong collection
404	Not Found	Artifact not found in repository using GET, PUT or DELETE
409	Conflict	Conflicting request <ul style="list-style-type: none"> - POST with UUID already existing - Batch POST encapsulating failure

4000

4001 Implementers are asked to note that according to the HTTP specification, HTTP 4xx and 5xx response
4002 entities *SHOULD* include a human-readable explanation of the error. The response structure *SHOULD*
4003 follow the error element definition in the atombinding.xsd. An example error message is given below.

4004

```

4005     HTTP/1.1 404 Not Found
4006     Date: Tues, 26 May 2009 13:13:55 GMT+02:00
4007     Content-Length: 520
4008     Content-Type: application/xml
4009     Mime-Version: 1.0
4010
4011     <?xml version="1.0" encoding="UTF-8"?>
4012     <error xmlns="http://docs.oasis-open.org/s-ramp/ns/s-ramp-v1.0"

```

4013 responseCode="404"
 4014 name="Not Found"
 4015 uuid="aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaa6a"
 4016 <description>You requested the deletion of artifact with UUID=aaaaaaaa-aaaa-aaaa-aaaa-
 4017 aaaaaaaaa6a but this artifact could not be found in the repository</description>
 4018 <howtofix>Check if you can find the artifact. If the artifact can't be found there is no need to re-issue
 4019 the delete request</howtofix>
 4020 <detail><stack-trace></detail>
 4021 </error>
 4022

4023 *Table 7 - Error Attribute Values*

Attribute Value	Description
responseCode	Required HTTP Response Code
name	Required, short message describing the error
uuid	Optional, the uuid of the artifact for which the error occurred, if this information is available at the time of the error

4024

4025 *Table 8 - SubElement Values*

SubElement Value	Description
description	Required, descriptive message
howtofix	Optional, pointing the user to a possible fix if this info is available at the time of the error
detail	Optional, additional detail about the error. This can be used to, for example, return a stack trace if that is deemed helpful

4026

4027

4028

Appendix H. Revision History

4029

Revision	Date	Editor	Changes Made
01	February 2, 2011	Martin Smithson	Initial conversion of the document
02	June 15, 2012	Martin Smithson	Applied modifications for issues SRAMP-8, 29, 36 and 37.
03	January 29, 2013	Vince Brunssen	Applied modifications for issues SRAMP-8, 15, 21, 22, 25, 44
04	February 7, 2013	Vince Brunssen	Applied modifications for new template and updated other miscellaneous issues found during the review process.
05	February 14, 2013	Vince Brunssen	Updated the document to reflect the namespace changes, acknowledgments and other miscellaneous changes that were found during an initial review.
06	February 20, 2013	Vince Brunssen	Accepted all changes that were reviewed in the February 20, 2013 TC Conference call. Made minor changes based on comments from the TC Meeting.
07	March 12, 2013	Vince Brunssen	S-RAMP JIRA SRAMP-33, fixing examples.

4030