OASIS ebXML RegRep Registry Information Model Version 4.0

Draft

October 8, 2008

Specification URIs:

This Version:

http://docs.oasis-open.org/regrep/4.0-draft-1/specs/core/regrep-rim-4.0-draft-1.html

http://docs.oasis-open.org/regrep/4.0-draft-1/specs/core/regrep-rim-4.0-draft-1.odt

http://docs.oasis-open.org/regrep/4.0-draft-1/specs/core/regrep-rim-4.0-draft-1.pdf

Previous Version:

http://docs.oasis-open.org/regrep/v3.0/specs/regrep-rim-3.0-os.html

http://docs.oasis-open.org/regrep/v3.0/specs/regrep-rim-3.0-os.odt

http://docs.oasis-open.org/regrep/v3.0/specs/regrep-rim-3.0-os.pdf

Latest Version:

http://docs.oasis-open.org/regrep/4.0-draft-1/specs/core/regrep-rim-4.0-draft-1.html

http://docs.oasis-open.org/regrep/4.0-draft-1/specs/core/regrep-rim-4.0-draft-1.odt

http://docs.oasis-open.org/regrep/4.0-draft-1/specs/core/regrep-rim-4.0-draft-1.pdf

Latest Approved Version:

http://docs.oasis-open.org/regrep/v3.0/specs/regrep-rim-3.0-os.html

http://docs.oasis-open.org/regrep/v3.0/specs/regrep-rim-3.0-os.odt

http://docs.oasis-open.org/regrep/v3.0/specs/regrep-rim-3.0-os.pdf

Technical Committee:

OASIS ebXML Registry TC

Chair(s):

Kathryn Breininger, Boeing

Editor(s):

Farrukh Najmi, Wellfleet Software
Nikola Stojanovic, RosettaNet

Contributors:

Kathryn Breininger, Boeing
Carl Mattocks, MetLife
Farrukh Najmi, Wellfleet Software

Oliver Newell, MIT Lincoln Labs
Nikola Stojanovic, RosettaNet
David Webber, Individual

Related Work:

This specification replaces or supercedes:

- [specifications replaced by this standard - OASIS as well as other standards organizations]
- [specifications replaced by this standard - OASIS as well as other standards organizations]

This specification is related to:

- [specifications related to this standard - OASIS as well as other standards organizations]
- [specifications related to this standard - OASIS as well as other standards organizations]

Declared XML Namespace(s):

This following table lists the namespace prefixes defined and / or referenced by this specification.

Namespace Prefix	Namespace URI	Defining Specification
enc	http://www.w3.org/2003/05/soap-encoding	A normative XML Schema [XML Schema Part 1], [XML Schema Part 2] document for the "http://www.w3.org/2003/05/soap-encoding" namespace can be found at http://www.w3.org/2003/05/soap-encoding.
env	http://www.w3.org/2003/05/soap-envelope	SOAP Version 1.2 Part 1. A normative XML Schema [XML Schema Part 1], [XML Schema Part 2] document for the "http://www.w3.org/2003/05/soap-envelope" namespace can be found at http://www.w3.org/2003/05/soap-envelope.
lcm	urn:oasis:names:tc:ebxml-regrep:xsd:lcm:4.0	ebXML RegRep Services and Protocols 4.0 (ebRS)
mime	http://schemas.xmlsoap.org/wsdl/mime/	WSDL namespace for WSDL MIME binding.
query	urn:oasis:names:tc:ebxml-regrep:xsd:query:4.0	ebXML RegRep Services and Protocols 4.0 (ebRS)
rim	urn:oasis:names:tc:ebxml-regrep:xsd:rim:4.0	ebXML RegRep Registry Information Model 4.0 (ebRIM)
rs	urn:oasis:names:tc:ebxml-regrep:xsd:rs:4.0	ebXML RegRep Services and Protocols 4.0 (ebRS)
wsdl	http://schemas.xmlsoap.org/wsdl/	WSDL 1.1 namespace defined by WSDL 1.1 specification.
xs	http://www.w3.org/2001/XMLSchema	XML Schema [XML Schema Part 1], [XML Schema Part 2] specification
xsi	"http://www.w3.org/2001/XMLSchema-instance	W3C XML Schema specification [XML Schema Part 1], [XML Schema Part 2].

Table 1: Namespaces Used

Abstract:

This document defines the types of metadata and content that can be stored in an ebXML Registry.

A separate document, OASIS ebXML RegRep: Service and Protocols [ebRS], defines the services and protocols for an ebXML Registry.

Status:

This document is a draft specification for review, revision and approval by the OASIS ebXML Registry TC.

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/committeees/regrep/.

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/regrep/ipr.php.

The non-normative errata page for this specification is located at http://docs.oasis-open.org/regrep/4.0-draft-1/specs/core/errata.pdf

Notices

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 names "OASIS", [insert specific trademarked names, abbreviations, etc. here] are trademarks 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/who/trademark.php for above guidance.

Table of Contents

Illustration Index

Index of Tables

1 Introduction

All text is normative unless otherwise indicated.

1.1 Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 .

1.2 Normative References

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

[Reference] [reference citation]

1.3 Non-normative References

[Reference] [reference citation]

[Reference] [reference citation]

2 Overview

2.1

3 Core Information Model

3.1 Local Language Support

Some information model types have textual values that need to be represented in multiple native languages. Examples include the name and description elements of the RegistryObject class.

The rim.xsd schema defines InternationalStringType and the LocalizedStringType as complexTypes that work together to support textual values in multiple native languages.

3.1.1 InternationalStringType

The InternationalStringType type is used throughout the schema whenever a textual value needs to be represented in multiple local languages. Two global elements Name and Description are defined to use InternationalStringType as their type.

The InternationalStringType has a sequence of LocalizedString instances, where each String is specific to a particular locale.

Syntax

```
<complexType name="InternationalStringType">
  <sequence>
    <element ref="tns:LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Example

```
<rim:Name>
  <rim:LocalizedString
    xml:lang="en-US" charset="UTF-8" value="freebXMLRegistry"/>
</rim:Name>
```

Description

Node	Type	Cardinality	Default Value	Specified By	Mutable
localizedString	LocalizedStringType	0..*		Client	Yes

- Element LocalizedString - An InternationalStringType instance MAY have zero or more LocalizedString elements where each defines a string value within a specific local language

3.1.2 LocalizedStringType

This type allows the definition of a string value using the specified local language and character set. It is used within the InternationalStringType as the type of the LocalizedString sub-element.

Syntax

```
<complexType name="LocalizedStringType">
```

```
  <attribute ref="xml:lang" default="en-US" use="optional"/>
  <attribute default="UTF-8" name="charset" use="optional"/>
  <attribute name="value" type="tns:FreeFormText" use="required"/>
</complexType>
```

Example

```
  <rim:LocalizedString
    xml:lang="en-US" charset="UTF-8" value="freebXMLRegistry"/>
```

Description

Node	Type	Cardinality	Default Value	Specified By	Mutable
lang	language	0..1	en-US	Client	Yes
charset	String	0..1	UTF-8	Client	Yes
value	String	1		Client	Yes

- Attribute lang - Each LocalizedStringType instance MAY have a *lang* attribute that specifies the language used by that LocalizedStringType instance
- Attribute charset - Each LocalizedStringType instance MAY have a *charset* attribute that specifies the name of the character set used by that LocalizedStringType instance. The value of this attribute SHOULD be registered with IANA at: http://www.iana.org/assignments/character-sets
- Attribute value - Each LocalizedStringType instance MUST have a *value* attribute that specifies the string value used by that LocalizedStringType instance

3.2 SlotType

This type is a container or wrapper that is capable of containing any type of information that may be represented in an XML document. It is an important extensibility mechanism with ebRIM.

A SlotType instance contains a ValueList element which contains one or more ValueListItems. It is the valueListItems that represent the values associated with the SlotType instance.

Syntax

```
<complexType name="SlotType">
  <sequence>
    <element ref="tns:ValueList" minOccurs="1" maxOccurs="1"/>
  </sequence>
  <attribute name="name" type="tns:LongName" use="required"/>
  <attribute name="dataType" type="tns:referenceURI" use="optional"/>
  <attribute name="collectionType" type="tns:referenceURI" use="optional"/>
</complexType>
```

Example

The following example shows how a GML geometry value may be specified as a Slot.

```
<rim:Slot
  name="spatialSlot1"
  dataType="urn:ogc:def:dataType:ISO-19107:GM_Envelope">
  <rim:ValueList>
    <rim:ValueListItem xsi:type="rim:AnyValueType">
      <gml:Envelope srsName="urn:ogc:def:crs:OGC:2:WGS84">
        <gml:lowerCorner>-122.35 19.31</gml:lowerCorner>
        <gml:upperCorner>-61.80 48.93</gml:upperCorner>
      </gml:Envelope>
    </rim:ValueListItem>
  </rim:ValueList>
</rim:Slot>
```

Description

Node	Type	Cardinality	Default Value	Specified By	Mutable
collectionType	ObjectRef	0..1		Client	No
dataType	LongName??	0..1		Client	No
name	LongName	1		Client	No
ValueList	ValueListType	1		Client	Yes

- Attribute collectionType – Defines the type of collection for the ValueList collection. Must be an ObjectRef that references a ClassificationNode in the canonical ClassificationScheme CollectionTypeScheme. A server MUST enforce the following semantics associated with the following canonical collection types:
  - List – Server MUST maintain the order of the values in the collection
  - Set – Server MUST NOT allow duplicate values in the collection
  - Sorted Set – Server MUST NOT allow duplicate values in the collection and MUST maintain a sort order according to the alphanumeric ordering of its elements according to the default local associated with the server
  - Bag – Server MUST allow duplicate values and MAY not maintain order of values
- Attribute dataType – A string that specifies the datatype for the values in the ValueList
- Attribute name – The name of this SlotType instance
- Element ValueList – This element is the container for the actual values within a SlotType instance.

3.3 ValueListType

This type is a container for ValueListItem instances that represent the values associated with a SlotType instance.

Syntax

```
<complexType name="ValueListType">
  <sequence>
    <element name="ValueListItem"
      type="tns:ValueType" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Description

Node	Type	Cardinality	Default Value	Specified By	Mutable
ValueListItem	ValueType	0..*		Client	Yes

- Element ValueListItem – This element represents a value within the collection of values in a SlotType instance. The type of this element is ValueType. Since ValueType is abstract, the actual type of ValueListItem MUST be a sub-type of ValueType. The rim.xsd schema defines the following concrete sub-types of ValueType:
    - AnyValueType – This concrete sub-type of ValueType is used as a container for any well-formed XML element value in any namespace
    - ParameterValueType – This concrete sub-type of ValueType is used as a container for Parameter definitions for a ParameterizedQuery instance
    - StringValueType – This concrete sub-type of ValueType is used as a container for a string value

3.4 ExtensibleObjectType

This type is the root type for most other types in rim.xsd. It allows any type of information to be added to instances of this type using Slot sub-elements. It is an important extensibility mechanism with ebRIM.

Syntax

```
<complexType name="ExtensibleObjectType" abstract="true">
  <sequence>
    <element ref="tns:Slot" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Example

The following example shows how a <rim:Organization> instance which is of type ExtensibleObjectType MAY use Slot sub-elements to define a tax payer id for the organization.

```
<rim:Organization
  id="urn:freebxml:registry:Organization:freebXMLRegistry" ...>

  <rim:Slot name="urn:foo:slot:taxPayerId">
    <rim:ValueList>
      <rim:ValueListItem xsi:type="rim:StringValueType">
        <rim:Value>1234567890</rim:Value>
      </rim:ValueListItem>
    </rim:ValueList>
  </rim:Slot>

  ...
</rim:Organization>
```

Description

Node	Type	Cardinality	Default Value	Specified By	Mutable
Slot	Slot	0..*		Client	Yes

- Element Slot – Allows any type of information to be defined within it and may be added to any ExtensibleObjectType instance

3.5 IdentifiableObjectType

Extends: ExtensibleObjectType

This type extends ExtensibleObjectType and allows its instances to be uniquely identifiable by a unique id.

Syntax

```
<complexType name="IdentifiableType" abstract="true">
  <complexContent>
    <extension base="tns:ExtensibleObjectType">
      <attribute name="id" type="string" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

Example

```
<rim:Organization
  id="urn:freebxml:registry:Organization:freebXMLRegistry" ...>
  ...
</rim:Organization>
```

Description

Node	Type	Cardinality	Default Value	Specified By	Mutable
Attribute id	string	1		Client	Yes

- Attribute id – Specifies the unique identifier for an IdentifiableType instance. An IdentifiableType instance MUST have an id and that id MUST conform to the rules defined in section title “Unique ID Generation” in [ebRS]

3.6 RegistryObjectType

Extends: IdentifiableType

This type extends IdentifiableObjectType and is the common base type for all query-able metadata elements in ebRIM.

Syntax

```
<complexType name="RegistryObjectType">
```

```
  <complexContent>
    <extension base="tns:IdentifiableType">
      <sequence>
        <element ref="tns:Name" minOccurs="0" maxOccurs="1"/>
        <element ref="tns:Description" minOccurs="0" maxOccurs="1"/>
        <element name="VersionInfo" type="tns:VersionInfoType" minOccurs="0"
maxOccurs="1"/>
        <element ref="tns:Classification" minOccurs="0" maxOccurs="unbounded"/
>
        <element ref="tns:ExternalIdentifier" minOccurs="0"
maxOccurs="unbounded" />
        <element ref="tns:ExternalLink" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="lid" type="anyURI" use="optional"/>
      <attribute name="objectType" type="tns:referenceURI" use="optional"/>
      <attribute name="owner" type="string" use="optional"/>
      <attribute name="status" type="tns:referenceURI" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

Description

Node	Type	Cardinality	Default Value	Specified By	Mutable
Classification	ClassificationType	0..*		Client	Yes
Description	InternationalStringType	0..1		Client	Yes
ExternalIdentifer	ExternalIdentifierType	0..*		Client	Yes
ExternalLink	ExternalLinkType	0..*		Client	Yes
lid	string	0..1.		Client or Server	No
Name	InternationalStringType	0..1		Client	Yes
objectType	objectReference	0..1		Client or Server	No
owner	string	0..1		Server	Yes
status	objectReference	0..1		Server	Yes
VersionInfo	VersionInfoType	0..1		Server	No

- Element Classification - A RegistryObjectType instance MAY have zero or more Classification instances that are composed within the RegistryObject. A Classification instance classify the RegistryObject using a value within a ClassificationScheme
- Element Description - A RegistryObjectType instance MAY have textual description in a human readable and user-friendly form. This element is of type InternationalStringType and therefor capable of containing textual values in multiple locales and character sets.

- Element ExternalIdentifier - A RegistryObjectType instance MAY have zero or more ExternalIdentifier instances that are composed within the RegistryObject. A ExternalIdentifier instance represents an alternate identifier for the RegistryObject in addition to the identifier specified by its id attribute value.

- Attribute lid - A RegistryObjectType instance MUST have a lid (Logical Id) attribute . The lid is used to refer to a logical RegistryObject in a version independent manner.

  - All versions of a RegistryObject MUST have the same value for the lid attribute. Note that this is in contrast with the id attribute that MUST be unique for each version of the same logical RegistryObject.

  - The lid attribute MAY be specified by the client when creating the original version of a RegistryObject.

  - If the client assigns the lid attribute when submitting the original version of a RegistryObject, it must guarantee that it is a globally unique.

  - A server MUST honor a client specified LID. If the client does not specify a LID then the server MUST assign a LID and the value of the LID attribute MUST be identical to the value of the id attribute of the first (originally created) version of the logical RegistryObject. Make sure this is consistent with latest spec??

- Element Name - A RegistryObjectType instance MAY have a human readable name. The name does not need to be unique with respect to other RegistryObject instances. This element is of type InternationalStringType and therefor capable of containing textual values in multiple locales and character sets.

- Attribute objectType - A RegistryObjectType instance has an *objectType* attribute.

  - The value of the objectType attribute MUST be a reference to a ClassificationNode in the canonical ObjectType ClassificationScheme.

  - A server MUST support the object types as defined by the canonical ObjectType ClassificationScheme. The canonical ObjectType ClassificationScheme may easily be extended by adding additional ClassificationNodes to the canonical ObjectType ClassificationScheme.

  - The *objectType* attribute MUST be assigned by the server for all RegistryObjectType instances that are not instances of ExtrinsicObjectType.

  - The *objectType* attribute MAY be assigned by the client for all RegistryObjectType instances that are instances of ExtrinsicObjectType. In such cases it represents the objectType associated with the repository item for the ExtrinsicObjectType instance.

  - A client SHOULD specify the objectType for an ExtrinsicObject during submission whenever possible.

  - If the client does not specify an objectType for an ExtrinsicObject then the server MUST set its value to the id of the ClassificationNode representing ExtrinsicObject within the canonical ObjectType ClassificationScheme.

  - A server MUST set the correct objectType on a RegistryObject when returning it as a response to a client request.

- Attribute owner – Specified the identifier associated with the registered user that own the RegistryObjectType instance. It is used for authorization of access and may be referenced within custom access control policies.

- Attribute status - A RegistryObjectType instance MUST have a life cycle status indicator. The status is assigned by the server.

  - A server MUST set the correct status on a RegistryObject when returning it as a response to a client request.
  - A client SHOULD NOT set the status on a RegistryObject when submitting the object as this is the responsibility of the server.
  - A server MUST ignore the status on a RegistryObject when it is set by the client during submission or update of the object.
  - The value of the status attribute MUST be a reference to a ClassificationNode in the canonical StatusType ClassificationScheme.
  - A Registry MUST support the status types as defined by the StatusType ClassificationScheme. The canonical StatusType ClassificationScheme MAY easily be extended by adding additional ClassificationNodes to the canonical StatusType ClassificationScheme.

  The following table lists pre-defined choices for the RegistryObject status attribute:

Name	Description
Approved	Indicated that the objects has been approved after being submitted
Deprecated	Indicated that the objects has been deprecated or marked as obsolete
Submitted	Indicated that the objects has been submitted to the server.
Withdrawn	Indicated that the objects has been withdrawn from the server. This SHOULD be used with ExtrinsicObjects when their repository item has been removed or withdrawn.

- Element VersionInfo - Provides information about the specific version of a RegistryObject. The VersionInfo attribute is set by the server.

3.7 VersionInfoType

This type represents information about a specific version of a RegistryObject.

Syntax

```
<complexType name="VersionInfoType">
  <attribute name="versionName"
    type="tns:String16" use="optional" default="1.1"/>
  <attribute name="comment" type="string" use="optional"/>
</complexType>
```

Example

```
<rim:Organization ...>
  ...
  <rim:VersionInfo versionName="1.1" comment="Initial version"/>
  ...
</rim:Organization>
```

Description

Node	Type	Cardinality	Default Value	Specified By	Mutable
comment	LongName	0..1		Client	Yes
versionName	String16	0..1	1.1	Server	No

- Attribute comment - Represents a client-specified comment associated with the VersionInfo for a specific RegistryObject version. It is analogous to a commit comment in version control systems.
  - The value of the comment attribute MAY be indirectly provided by the client when the client specifies a value for the comment attribute of the <rim:Request> object when making a request to the server.
  - The value for this attribute MUST be set by the Registry implementation based upon the <rim:Request> comment attribute value provided by the client if any.
- Attribute versionName - Represents the version name identifying the VersionInfo for a specific RegistryObject version.
  - The value for this attribute SHOULD NOT be specified by the client
  - A server MUST ignore the value for this attribute if specified by the client
  - The value for this attribute MUST be automatically generated by the server and MUST be defined for RegistryObjectType instances returned by server responses.

4 Association Information Model

5 Classification Information Model

6 Provenance Information Model

7 Service Information Model

8 Event Information Model

9 Federation Information Model

10 Access Control Information Model

Appendix A. Acknowledgments

The following individuals have contributed significantly towards the creation of this specification and are gratefully acknowledged

Contributors:

- Rob Atkinson, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia
- Simon Cox, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia
- Lydia Gietler, Danish Ministry of the Environment
- Aleksei Valikov, Disy Informationssysteme GmbH

Appendix B. Revision History

[optional; should not be included in OASIS standards]

Appendix C. Non-Normative Text