Extensible Resource Descriptor (XRD) Version 1.0

Committee Draft 01

22 October 2009

Specification URIs:

This Version:

`http://docs.oasis-open.org/xri/xrd/v1.0/cd01/xrd-1.0-cd01.xml` (Authoritative)
`http://docs.oasis-open.org/xri/xrd/v1.0/cd01/xrd-1.0-cd01.html`
`http://docs.oasis-open.org/xri/xrd/v1.0/cd01/xrd-1.0-cd01.pdf`

Previous Version:

N/A

Latest Version:

`http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.xml` (Authoritative)
`http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.html`
`http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.pdf`

Technical Committee:

OASIS Extensible Resource Identifier (XRI) TC [http://www.oasis-open.org/committees/xri/]

Chairs:

Peter Davis, NeuStar Inc.
Drummond Reed, XDI.org

Editors:

Eran Hammer-Lahav
Will Norris, Internet2

Related Work:

This specification replaces or supersedes:

- Extensible Resource Identifier (XRI) Resolution Version 2.0, Committee Specification 01, April 2008 [http://docs.oasis-open.org/xri/2.0/specs/xri-resolution-V2.0.html]

This specification is related to:

- Extensible Resource Identifier (XRI) Version 3.0, Working Draft 02, August 2009 [http://www.oasis-open.org/committees/download.php/33877/xri-syntax-3.0-wd02.pdf]

Declared XML Namespace:

- http://docs.oasis-open.org/ns/xri/xrd-1.0

Abstract:

This document defines XRD, a simple generic format for describing and discovering resources.

Status:

This document was last revised or approved by the XRI Technical Committee on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

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/xri.

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/xri/ipr.php).

The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/xri.

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 Final Deliverable, 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 deliverable.

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 OASIS Final Deliverable 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 OASIS Final Deliverable. 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 OASIS Final Deliverable 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 Final Deliverable, 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", “Extensible Resource Descriptor”, and “XRD” 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

Appendixes

1. Introduction

This document defines XRD, a simple generic format for describing resources. Resource descriptor documents provide machine-readable information about resources (resource metadata) for the purpose of promoting interoperability, and assist in interacting with unknown resources that support known interfaces.

For example, a web page about an upcoming meeting can provide in its descriptor document the location of the meeting organizer's free/busy information to potentially negotiate a different time. The descriptor for a social network profile page can identify the location of the user's address book as well as accounts on other sites. A web service implementing an API protocol can advertise which of the protocol's optional components are supported.

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 [RFC 2119].

1.2. Normative References

[Exclusive Canonicalization] J. Boyer, D. Eastlake, J. Reagle *Exclusive XML Canonicalization* *[http://www.w3.org/TR/xml-exc-c14n/]*. W3C Recommendation. 2002

[RFC 2046] N. Freed, N. Borenstein *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types* *[http://tools.ietf.org/html/rfc2046]*. IETF (Internet Engineering Task Force). 1996.

[RFC 2119] S. Bradner *Key words for use in RFCs to Indicate Requirement Levels* *[http://tools.ietf.org/html/rfc2119]*. IETF (Internet Engineering Task Force). 1997.

[RFC 3023] M. Murata, S. St. Laurent, D. Kohn *XML Media Types* *[http://tools.ietf.org/html/rfc3023]*. IETF (Internet Engineering Task Force). 2001.

[RFC 4288] N. Freed, J. Klensin *Media Type Specifications and Registration Procedures* *[http://tools.ietf.org/html/rfc4288]*. IETF (Internet Engineering Task Force). 2005.

[Web Linking] M. Nottingham *Web Linking* *[http://tools.ietf.org/html/draft-nottingham-http-link-header]*. IETF (Internet Engineering Task Force) Draft. 2009.

[XML 1.0] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, F. Yergeau *Extensible Markup Language (XML) 1.0* *[http://www.w3.org/TR/REC-xml/]* W3 Recommendation. 2008

[XML Schema] H. Thompson, D. Beech, M. Maloney, N. Mendelsohn *XML Schema Part 1: Structures Second Edition* *[http://www.w3.org/TR/xmlschema-1/]*. W3C Recommendation. 2004

[XML Schema Datatypes] P. Biron, A. Malhotra *XML Schema Part 2: Datatypes Second Edition* *[http://www.w3.org/TR/xmlschema-2/]*. W3 Recommendation. 2004

[XML Signature] D. Eastlake, J. Reagle, D. Solo, F. Hirsch, T. Roessler *XML Signature Syntax and Processing* *[http://www.w3.org/TR/xmldsig-core/]*. W3 Recommendation. 2008

[xml:id] J. Marsh, D. Veillard, N. Walsh *xml:id* *[http://www.w3.org/TR/xml-id/]*. W3 Recommendation. 2005

1.3. Non-Normative References

[Atom 1.0] M. Nottingham, R. Sayre *The Atom Syndication Format [http://tools.ietf.org/html/rfc4287]*. IETF (Internet Engineering Task Force). 2005.

[HTML 4.01] D. Raggett, A. Le Hors, I. Jacobs *HTML 4.01 Specification [http://www.w3.org/TR/html401/]*. W3C Recommendation. 1999

[XRI Resolution 2.0] OASIS Committee Draft 03, *Extensible Resource Identifier (XRI) Resolution V2.0 [http://docs.oasis-open.org/xri/2.0/specs/xri-resolution-V2.0.html]*. February 2008.

1.4. Schema Organization and Namespaces

The XRD document structure is defined in a schema associated with the following XML namespace:

```
http://docs.oasis-open.org/ns/xri/xrd-1.0
```

The schema for [XML 1.0] (the "xml:" namespace) is imported into the XRD schema, which is associated with the following XML namespace:

```
http://www.w3.org/XML/1998/namespace
```

The following [XML Schema] fragment defines the XML namespaces and other header information for the XRD schema:

```
<schema targetNamespace="http://docs.oasis-open.org/ns/xri/xrd-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xrd="http://docs.oasis-open.org/ns/xri/xrd-1.0"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  blockDefault="substitution"
  version="1.0">

    <import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>

    <annotation>
        <documentation>
            Document identifier: xrd-schema-1.0
            Location: http://docs.oasis-open.org/xri/xrd/v1.0/
        </documentation>
    </annotation>
...
</schema>
```

The location of the normative XML Schema file for an XRD document as defined by this specification is: `http://docs.oasis-open.org/xri/xrd/v1.0/cd01/xrd-1.0-cd01.xsd`

The following URI will always reference the latest version of this file: `http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.xsd`

1.5. Common Data Types

The following sections define how to use and interpret common data types that appear throughout the XRD schema.

1.5.1. String Values

All XRD string values have or extend the type `xs:string`, which is built in to the W3C [XML Schema Datatypes] specification. Unless otherwise noted in this specification or particular profiles, all strings in XRD documents MUST consist of at least one non-whitespace character (whitespace is defined in section 2.3 of [XML 1.0]).

The following schema fragment defines the `xrd:string` complex type, which extends `xs:string` to allow for artibtrary attributes (see Section 3.2, “Schema Extension”):

```
<complexType name="string">
    <simpleContent>
        <extension base="string">
            <anyAttribute namespace="##other" processContents="lax"/>
        </extension>
    </simpleContent>
</complexType>
```

1.5.2. URI Values

All XRD URI reference values have or extend the type `xs:anyURI`, which is built in to the W3C [XML Schema Datatypes] specification. Unless otherwise noted in this specification or particular profiles, all URIs in XRD documents MUST consist of at least one non-whitespace character.

The following schema fragment defines the `xrd:anyURI` complex type, which extends `xs:anyURI` to allow for artibtrary attributes (see Section 3.2, “Schema Extension”):

```
<complexType name="anyURI">
    <simpleContent>
        <extension base="anyURI">
            <anyAttribute namespace="##other" processContents="lax"/>
        </extension>
    </simpleContent>
</complexType>
```

1.5.3. Time Values

All XRD time values have the type `xs:dateTime`, which is built in to the W3C [XML Schema Datatypes] specification, and MUST be expressed in UTC form, with no time zone component (represented by the UTC 'Z' timezone). XRD providers MUST NOT generate time instants that specify leap seconds.

2. XRD Document Structure

XRD provides a simple and extensible XML format for describing resources. An XRD document may describe the properties of the resource itself, as well as the relations the resource has with other resources. XRD builds directly on the typed link relations framework defined by [Web Linking], and used by [HTML 4.01], [Atom 1.0], and other protocols.

The XRD schema defines only the elements necessary to support the most common use cases, with the explicit intention that applications will extend XRD as defined in Section 3, “XRD Extensibility” to include any other metadata about the resources and links they describe.

2.1. XRD Elements

The XRD elements provide information about the resource described by the XRD document and list its properties. They also provide administrative information as to how the document should be cached, as well as information necessary for the authentication and trust verification of the XRD document.

2.1.1. Element `<XRD>`

The `<XRD>` element encapsulates the entire resource descriptor, and is most commonly the root element of the document. It contains the following attributes and elements:

`xml:id` [Optional]

> This attribute, of type `xs:ID`, is defined by [xml:id]. It provides a unique identifier for this XRD, and is most commonly used as a signature reference.

`<Expires>` [Zero or One]

> Specifies when this document expires, as described in Section 2.1.2, “Element `<Expires>`”.

`<Subject>` [Zero or One]

> Provides the identifier of the resource described by this XRD. See Section 2.1.3, “Element `<Subject>`”.

`<Alias>` [Zero or More]

> Provides an additional identifier for the resoure described by this XRD. See Section 2.1.4, “Element `<Alias>`”.

`<Type>` [Zero or More]

> Declares a property of the resource described by this XRD, as described in Section 2.1.5, “Element `<Type>`”.

`<Link>` [Zero or More]

> Identifies another resource which is related to the resource described by this XRD, and describes the semantics of that relation. See Section 2.2.1, “Element `<Link>`”.

`<ds:Signature>` [Zero or More]

> This is an XML Signature, included from the [XML Signature] schema, that protects the integrity of the document, as described in Section 5, “XRD Signature”.

> Although [XML Signature] allows a single document to contain multiple signatures, the signing profile described in Section 5, “XRD Signature” requires only a single <Signature> element. Use of multiple <Signature> elements in an XRD document is therefore undefined. In order to aid certain types of XRD consumers, it is RECOMMENDED that XRD providers place the <Signature> element of a signed XRD as near the beginning of the document as possible.

The following schema fragment defines the <XRD> element and its XRDType complex type:

```
<element name="XRD" type="xrd:XRDType"/>
<complexType name="XRDType">
    <sequence>
        <element ref="xrd:Expires" minOccurs="0"/>
        <element ref="xrd:Subject" minOccurs="0"/>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xrd:Alias"/>
            <element ref="xrd:Type"/>
            <element ref="xrd:Link"/>
            <any namespace="##other" processContents="lax"/>
        </choice>
    </sequence>
    <attribute ref="xml:id" use="optional"/>
    <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

2.1.2. Element <Expires>

This dateTime value indicates the time instant after which the document is no longer valid. The value MUST use the UTC "Z" time zone, as specified in Section 1.5.3, “Time Values”, and MUST NOT use fractional seconds. The semantics of this element apply to the metadata available in the XRD document and are independent of the caching semantics of any transport protocol used to retrieve the document.

The following schema fragment defines the <Expires> element and its ExpiresType complex type:

```
<element name="Expires" type="xrd:ExpiresType"/>
<complexType name="ExpiresType">
    <simpleContent>
        <extension base="dateTime">
            <anyAttribute namespace="##other" processContents="lax"/>
        </extension>
    </simpleContent>
</complexType>
```

2.1.3. Element <Subject>

<Subject> is a URI value which identifies a resource. This value MUST be an absolute URI.

The following schema fragment defines the <Subject> element.

```
<element name="Subject" type="xrd:anyURI"/>
```

2.1.4. Element <Alias>

This URI value provides an additional identifier for the resource described by the XRD. This value MUST be an absolute URI.

The following schema fragment defines the `<Alias>` element:

```
<element name="Alias" type="xrd:anyURI"/>
```

2.1.5. Element `<Type>`

The `<Type>` element declares a property of the resource described by the XRD using a URI value. The meaning of the `<Type>` value is application-specific, and is used by the XRD provider to describe the resource to consumers familiar with the type identifier. `<Type>` contains the following attributes:

`required` [Optional]

> The `required` attribute is a boolean used to indicate to a consumer that some pre-defined knowledge is required in order to interact with the resource, without which undefined or potentially harmful side-effects can occur.
>
> If the `required` attribute is omitted or explicitly set to `false`, an XRD consumer SHOULD ignore any `<Type>` elements with values it does not recognize, and interact with the resource based on the values it does recognize. If the `required` attribute is set to `true`, an XRD consumer MUST NOT interact with the resource if it does not recognize the type identifier. The `required` attribute SHOULD NOT be used unless such harmful side-effects are likely.

The following schema fragment defines the `<Type>` element and its `TypeType` complex type:

```
<element name="Type" type="xrd:TypeType"/>
<complexType name="TypeType">
    <simpleContent>
        <extension base="xrd:anyURI">
            <attribute name="required" type="boolean" use="optional"/>
        </extension>
    </simpleContent>
</complexType>
```

2.2. Linked Resource Elements

One of the primary uses of XRD is to describe the relations the resource described by the XRD has with other resources. The following elements identify and describe these related resources.

2.2.1. Element `<Link>`

The `<Link>` element serves as a container for metadata about a relation to a related resource, and carries similar semantics as the [HTML 4.01] Link element, the [Atom 1.0] Link element, and the HTTP Link Header. The one distinction is that link relations described by the `<Link>` element are between the resource described by the XRD (referred to as the *context resource* by [Web Linking]) and the linked resources (referred to as the *target resource* by [Web Linking]), and not between the XRD document itself and the linked resource. `<Link>` has the following elements and attributes:

`<Rel>` [Zero or One]

> Defines the semantics of the link relation. See Section 2.2.2, "Element `<Rel>`".

`<MediaType>` [Zero or One]

> Provides a hint at the media type of the linked resource. See Section 2.2.3, "Element `<MediaType>`".

<URI> [Zero or One]

The URI of the linked resource. Only <URI> or <URITemplate> may be present, but not both. See Section 2.2.4, "Element <URI>".

<URITemplate> [Zero or One]

Provides a template which can be used to obtain the URI of the linked resource. Only <URI> or <URITemplate> may be present, but not both. See Section 2.2.5, "Element <URITemplate>".

<Title> [Zero or More]

Provides a human-readable description of the linked resource. See Section 2.2.6, "Element <Title>".

The following schema fragment defines the <Link> element and its LinkType complex type:

```
<element name="Link" type="xrd:LinkType"/>
<complexType name="LinkType">
    <sequence>
        <element ref="xrd:Rel" minOccurs="0"/>
        <element ref="xrd:MediaType" minOccurs="0"/>
        <choice minOccurs="0">
            <element ref="xrd:URI"/>
            <element ref="xrd:URITemplate"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xrd:Title"/>
            <any namespace="##other" processContents="lax"/>
        </choice>
    </sequence>
    <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

2.2.2. Element <Rel>

This URI value defines the semantics of the relation between the resource described by the XRD and the linked resource. <Rel> is semantically and syntactically equivalent to the Link Relation Types defined in [Web Linking], with the exception that it only allows for a single relation type and does not allow for multiple space delimited values. It is important to note that this value does not identify any property of the linked resource. Rather, it describes only how the linked resource is related to the resource described by the XRD.

The following schema fragment defines the <Rel> element:

```
<element name="Rel" type="xrd:anyURI"/>
```

2.2.3. Element <MediaType>

This string value provides a hint as to the media type of the linked resource. The value of this element MUST be of the form of a media type defined in [RFC 2046]. The IANA media types registry can be found at http://www.iana.org/assignments/media-types/.

The following schema fragment defines the <MediaType> element:

```
<element name="MediaType" type="xrd:string"/>
```

2.2.4. Element <URI>

The <URI> element provides the URI of the linked resource. If no <URI> element is defined, it is assumed the URI can be obtained from a <URITemplate> element or by other means outside this specification.

The following schema fragment defines the <URI> element:

```
<element name="URI" type="xrd:anyURI"/>
```

2.2.5. Element <URITemplate>

URI Templates provide a mechanism for URI construction, taking a list of variables as input and producing a URI string as an output. The template syntax and vocabulary are dependent upon the combination of the application through which the XRD document is obtained, and the link relation type indicated by the <Rel> child element of the corresponding <Link>. Applications that wish to utilize the template mechanism need to define the variable vocabulary for each relation type, as well as the template syntax and processing rules.

If a template uses an unknown syntax or contains unknown variable names (without rules on how to handle such variables), the parent <Link> element SHOULD be ignored.

The following schema fragment defines the <URITemplate> element:

```
<element name="URITemplate" type="xrd:string"/>
```

2.2.6. Element <Title>

This string value provides a human-readable description for the linked resource. The semantics of this value are intended only for human consumption and MUST NOT be used to imply any additional meaning. <Title> contains the following attributes:

xml:lang [Optional]

> This attribute is defined by the [XML 1.0] specification, and is used to identify the natural language in which this element's content is written.

The following schema fragment defines the <Title> element and its TitleType complex type:

```
<element name="Title" type="xrd:TitleType"/>
<complexType name="TitleType">
    <simpleContent>
        <extension base="xrd:string">
            <attribute ref="xml:lang" use="optional"/>
        </extension>
    </simpleContent>
</complexType>
```

3. XRD Extensibility

The XRD schema defines only the elements necessary to support the most common use cases, with the explicit intention that applications will extend XRD to include any other metadata about the resources they describe. XRD documents can be extended by providing custom, meaningful values for certain URI-based elements, as well as by extending the XML elements directly.

3.1. Identifier Extension

XRD uses URI-based identifiers for describing resources as well as for describing the relations between resources. It is expected that applications will use appropriate established URI identifiers for these purposes, or define new identifiers as necessary. It is RECOMMENDED that any new identifiers be defined in a formal specification of use. In no case should the meaning of a given URI used as such an identifier significantly change, or be used to mean two different things.

New relation types between resources MUST follow the extensibility and registration requirements defined in [Web Linking].

3.2. Schema Extension

The XRD schema allows for the inclusion of attributes from arbitrary namespaces (except for the XRD namespace) in almost all XRD elements. Additionally, the `<XRD>` and `<Link>` elements allow for the inclusion of child elements from arbitrary namespaces (except for the XRD namespace).

XML extensions MUST NOT require new interpretation of elements defined in this document. If an extension attribute or element is present, an XRD consumer MUST be able to ignore it and still correctly process the XRD document.

4. Processing XRD Documents

Once an XRD document has been obtained, the consumer typically inspects the list of resource properties looking for known values, and performs resource selection to find the links relevant to it. The selection process involves iterating through the list of linked resource descriptions and filtering them based on various metadata.

4.1. Linked Resource Selection

Link selection criteria is determined by the XRD consumer's needs, and is based on the presence, absence, or value of `<Link>` element attributes or child elements. The selection criteria may be any combination of metadata describing the linked resources such as `<Rel>`, `<MediaType>`, `<URI>`, or non-XRD extension elements or attributes. For example, a consumer might look for all linked resources with an image media-type, a linked resource with a specific authentication relation, or a linked resource with a URI matching a pattern.

Selection based on multiple criteria can be handled by performing multiple selections. Each selection is assigned preference order based on the consumer's needs. The selection results are compared to determine the most desired set. For example, if a consumer is looking for all image resources, giving higher preference to the JPEG formats over PNG, it will perform two selection processes, one for each media-type, and assign the resources in the JPEG set a higher preference value.

If multiple `<Link>` elements are matched by a given selection criteria, they MUST be processed in the order in which they appear in the XRD document. Therefore, XRD providers MAY indicate element priority by placing them in a specific order. If the first `<Link>` is subsequently disqualified from the set of selected elements, the consumer SHOULD attempt to select the next qualified element in document order. This process SHOULD be continued for all other qualified `<Link>` elements until success is achieved or all elements are exhausted.

5. XRD Signature

An XRD provider MAY digitally sign an XRD document in order to enable XRD consumers to verify the authenticity and integrity of the document. The [XML Signature] specification defines a general XML syntax for signing data that includes many options for flexibility. This section details constraints on these options so that XRD consumers do not have to implement the full generality of XML Signature processing.

5.1. Signing Formats and Algorithms

While XML Signature has three ways of relating a signature to a document (enveloping, enveloped, and detached), XRD documents MUST use enveloped signatures when signing. Any algorithm defined for use with the XML Signature specification MAY be used.

5.2. References

XRD documents MUST supply a value for the `xml:id` attribute on the root element of the XRD being signed. The XRD's root element may or may not be the root element of the actual XML document containing the signed XRD (e.g., it might be included within another document schema).

Signatures MUST contain a single `<ds:Reference>` containing a same-document reference to the `xml:id` attribute value of the root element of the XRD being signed. For example, if the `xml:id` attribute value is `foo`, then the `URI` attribute in the `<ds:Reference>` element MUST be `#foo`.

5.3. Canonicalization

XRD implementations MUST use [Exclusive Canonicalization], with or without comments, both in the `<ds:CanonicalizationMethod>` element of `<ds:SignedInfo>`, and as a `<ds:Transform>` algorithm. Use of Exclusive Canonicalization facilitates the verification of signatures created over XRD instances when placed into a different XML context than present during signing.

Note that use of this algorithm alone does not guarantee that a particular signed object can be moved from one context to another safely, nor is that a requirement of signed XRD instances in general, though it MAY be required by particular profiles.

5.4. Transforms

Signatures in XRD documents MUST NOT contain transforms other than the enveloped signature transform (with the identifier `http://www.w3.org/2000/09/xmldsig#enveloped-signature`) or the exclusive canonicalization transforms (with the identifier `http://www.w3.org/2001/10/xml-exc-c14n#` or `http://www.w3.org/2001/10/xml-exc-c14n#WithComments`).

5.5. KeyInfo

XML Signature defines usage of the `<ds:KeyInfo>` element. XRD does not require the use of `<ds:KeyInfo>`, nor does it impose any restrictions on its use. Therefore, `<ds:KeyInfo>` MAY be absent.

6. XRD Sequence

In cases where an application requires a sequence of `<XRD>` elements in a single XML document, this specification defines an alternate top-level element, `<XRDS>`. Its semantics are very simple: it contains a sequence of `<XRD>` elements, one optional attribute, and is not otherwise extensible.

`ref` [Optional]

> This attribute, of type `xs:anyURI`, identfies the resource described by the sequence of `<XRD>` elements.

`<XRD>` [One or More]

> See Section 2.1.1, “Element `<XRD>`”.

The following schema fragment defines the `<XRDS>` element and its `XRDSType` complex type:

```
<element name="XRDS" type="xrd:XRDSType"/>
<complexType name="XRDSType">
    <sequence>
        <element ref="xrd:XRD" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="ref" type="anyURI"/>
</complexType>
```

IMPORTANT: In past practice, `<XRDS>` was often used as the root element even if it contained only a single `<XRD>`. That practice is now deprecated. The `<XRDS>` element SHOULD NOT be used unless it contains a sequence of two or more `<XRD>` elements.

7. Conformance

An implementation is a *conforming* XRD consumer if the implementation meets the conditions in Section 7.1, “XRD Consumer”. An implementation is a *conforming* XRD provider if the implementation meets the conditions in Section 7.2, “XRD Provider”. An implementation may serve as both an XRD consumer and provider.

7.1. XRD Consumer

An implementation conforms to this specification as an XRD consumerif it meets the following conditions:

1. It MUST implement parsing of XRD documents which conform to the XRD schema as specified in Section 2, “XRD Document Structure”.
2. It MUST conform to the processing rules as specified in Section 4, “Processing XRD Documents”.
3. In calculating XRD Signatures, it SHOULD support the use of RSA signing and verification for public key operations in accordance with the signing algorithm identified by `http://www.w3.org/2000/09/xmldsig#rsa-sha256`.

7.2. XRD Provider

An implementation conforms to this specification as an XRD provider if it meets the following conditions:

1. Any published XRD documents MUST conform to the XRD schema as specified in Section 2, “XRD Document Structure”.
2. Any published XRD documents that are digitally signed using [XML Signature] MUST conform to the signature profile specified in Section 5, “XRD Signature”.

A. Acknowledgments (Non-Normative)

The editors would like to thank the following current and former members of the OASIS XRI TC for their particular contributions to this and previous versions of this specification:

- Dirk Balfanz, Google
- Bill Barnhill, Booz Allen Hamilton
- John Bradley
- Scott Cantor, Internet2
- Les Chasen, NeuStar
- Steven Churchill, XDI.org
- Brian Eaton, Google
- George Fletcher, AOL
- Victor Grey, Planetwork
- Joseph Holsten
- Nika Jones
- Breno de Medeiros, Google
- Bob Morgan, Internet2
- Markus Sabadello, XDI.org
- Nat Sakimura, NRI
- Tatsuki Sakushima, NRI
- William Tan, NeuStar
- Gabe Wachob

The editors would also like to acknowledge the contributions of the other members of the OASIS XRI Technical Committee, whose other voting members at the time of publication were:

- Giovanni Bartolomeo, University of Rome "Tor Vergata"
- Owen Davis, Planetwork
- Jeff Hodges
- Fen Labalme, Planetwork
- Ben Laurie, Google
- XiaoDong Lee, China Internet Network Information Center
- Nick Nicholas, Australian Department of Education
- Marty Schleiff, The Boeing Company
- Paul Trevithick

B. XRD Examples

Example B.1. Simple XRD Example

```
<XRD xmlns="http://docs.oasis-open.org/ns/xri/xrd-1.0">
  <Expires>1970-01-01T00:00:00Z</Expires>
  <Subject>http://example.com/gpburdell</Subject>
  <Link>
    <Rel>http://spec.example.net/auth/1.0</Rel>
    <URI>http://services.example.com/auth</URI>
  </Link>
  <Link>
    <Rel>http://spec.example.net/photo/1.0</Rel>
    <MediaType>image/jpeg</MediaType>
    <URI>http://photos.example.com/gpburdell.jpg</URI>
  </Link>
</XRD>
```

Example B.2. Signed XRD Example

Following is an example of a signed XRD document. Line breaks have been added for readability; the signatures are not valid and cannot be successfully verified.

```
<XRD xmlns="http://docs.oasis-open.org/ns/xri/xrd-1.0" xml:id="foo">
  <Expires>1970-01-01T00:00:00Z</Expires>
  <Subject>http://example.com/gpburdell</Subject>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
      <ds:SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#foo">
        <ds:Transforms>
          <ds:Transform
           Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <InclusiveNamespaces PrefixList="#default xrd ds xs xsi"
              xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>TCDVSuG6grhyHbzhQFWFzGrxIPE=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
      x/GyPbzmFEe85pGD3c1aXG4Vspb9V9jGCjwcRCKrtwPS6vdVNCcY5rHaFPYWkf+5
      EIYcPzx+pX1h43SmwviCqXRjRtMANWbHLhWAptaK1ywS7gFgsD01qjyen3CP+m3D
      w6vKhaqledl0BYyrIzb4KkHO4ahNyBVXbJwqv5pUaE4=
    </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
          MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT
          MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT
          F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ
          bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgQ0Eg
          LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVowgYsx
          CzAJBgNVBAYTAlVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW5uIEFy
          Ym9yMQ4wDAYDVQQKEwVVQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZXQyLmVk
          dTEnMCUGCSqGSIb3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWR1MIGfMA0G
          CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj
          IHRYQgIv6IqaGG04eTcyVMhoekE0b45QgvBIaOAPSZBl13R6+KYiE7x4XAWIrCP+
          c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE
          pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq
          hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n
          qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX
          8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w==
        </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
  <Link>
    <Rel>http://spec.example.net/auth/1.0</Rel>
    <URI>http://services.example.com/auth</URI>
  </Link>
</XRD>
```

C. Media Type Definition for `application/xrd+xml`

This section is prepared in anticipation of filing a media type registration meeting the requirements of [RFC 4288].

Type name:

`application`

Subtype name:

`xrd+xml`

Required parameters:

None

Optional paramters:

"charset": This parameter has identical semantics to the charset parameter of the "application/xml" media type as specified in [RFC 3023].

Encoding considerations:

Identical to those of `application/xml` as described by [RFC 3023] section 3.2.

Security considerations:

As defined in this specification. In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [RFC 3023], Section 10.

Interoperability considerations:

There are no known interoperability issues.

Published specification:

This specification

Applications that use this media type:

No known applications currently use this media type.

Magic Number:

As specified for [RFC 3023] section 3.2.

File Extension:

None.

Fragnent Identifiers:

As specified for [RFC 3023] section 5.

Base URI:

As specified for [RFC 3023] section 6.

Macintosh File Type code:

TEXT.

Person & email address to contact for further information:

Eran Hammer-Lahav, eran@hueniverse.com

Intended usage:

COMMON

Author / Change controller:

OASIS XRI Technical Comittee

D. Revision History

Table D.1.

Revision	Date	Editor	Changes Made
Committee Draft 01	22 October 2009	willnorris	• Approved by the XRI Technical Committee as Committee Draft 01.
Working Draft 09	15 October 2009	willnorris	• Cleanup references section (some where no longer referenced at all, some were only informative). • Fix acknowledgements to properly include XRI Resolution 2.0 editors
Working Draft 08	14 October 2009	willnorris	• Remove "Subject" "ds:keyInfo" as child elements of Link. These only had clear meaning in the context of a linked XRD. • Remove default URI template syntax and change text to make it application+relation specific • Clarified that rel values are not allowed to contain space-delimited relation types
Working Draft 07	12 October 2009	willnorris	• Remove "Extensions" element. Revert to previous extension model, resolving the "ambiguous schema" issue by simply not defining the signature elements in the XRD schema. • Add "Title" element under "Link" for human readable name of linked resource • Add signature algorithm support to conformance • Greatly reduce complexity of Link element. Reduce cardinality of Rel,

Revision	Date	Editor	Changes Made
			MediaType, URI, and URITemplate elements to zero or one. URI or URITemplate is allowed, but not both. Processing section updated to reflect these changes. • Remove definition of linked XRD documents. • Various minor editorial changes
Working Draft 06	04 September 2009	willnorris	• Combine "Document Property Elements" and "Resource Property Elements" into "XRD Elements" • Move schema and references to first section • Promote "XRD Extensions" section, and move schema fragment • Add example for URI / URITemplate processing order • Move XRD Example into an appendix • Various minor rewording
Working Draft 05	01 September 2009	willnorris	• Remove priority attribute on Link, URI, and URITemplate elements. Instead, element priority is implied by document order. Additionally, requirement for consumers to respect priority strengthened from "should" to "must". • New "Extensions" element added to XRD and Link elements as the sole location to extend XRD with arbitrary child elements.

Revision	Date	Editor	Changes Made
			• Define "XRDS" element to contain a sequence of XRD elements. • Removed "match" attribute from Subject. • Added requirement to follow normal rules for Rel values (either use a URI, or register value with IANA) • Switched from Relax NG to XSD as the authoritative schema language for the XRD Schema. (Primarily due to the lack of a Relax NG schema for XML DSig) • Clarify language regarding URIs and URI Templates • Define "Common Data Types" for XRD • Various minor editorial and grammatical changes
Working Draft 04	12 August 2009	willnorris	• Remove XRD Trust section, pushing that work to separate trust profiles. Move XRD Signature section up one level. • Remove requirement for explicit Link Subject on linked XRDs • Use non-information URI for rel value to designate linked XRD • Flesh out subject matching rules • Remove "must not be used" from Expires element description
Working Draft 03	04 August 2009	willnorris	• Revert to previous processing flow for related resources -- first filter, then sort by priority

Revision	Date	Editor	Changes Made
			• Add media type definition for "application/xrd+xml" • Clarify text for URI templates • Strengthen requirement to use excl-c14n from "should" to "must" • Move Signature element to bottom of the document for readability • Add conformance section • Add "match" attribute to Subject element. Also add stub section for subject matching. • Add XSD schema (in addition to RELAX NG) • Various editorial and grammatical changes.
Working Draft 02	03 July 2009	willnorris	• Remove XRD Trust namespace and elements (TargetSubject replaced by Subject, TargetAuthority replaced by ds:KeyInfo) • Section added for XML Digital Signature, primarily copied from SAML 2.0, which chnages as necessary • Language clarified on priority attribute values ('null' is not a valid value) • Add section for XRD Extensibility • Only require XML element order for elements with cardinality of "zero or one" • Add section for defining linked XRD documents • Processing rules changed for related resources to first sort

Revision	Date	Editor	Changes Made
			by priority, then filter. Also add processing rule for linked XRD documents. - Various editorial and grammatical changes.
Working Draft 01	09 May 2009	willnorris	Initial Publication