Extensible Resource Descriptor (XRD) Version 1.0

Committee Draft 02

9 March 2010

Specification URIs:

This Version:

`http://docs.oasis-open.org/xri/xrd/v1.0/cd02/xrd-1.0-cd02.xml` (Authoritative)
`http://docs.oasis-open.org/xri/xrd/v1.0/cd02/xrd-1.0-cd02.html`
`http://docs.oasis-open.org/xri/xrd/v1.0/cd02/xrd-1.0-cd02.pdf`

Previous 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`

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]

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:

Table of Contents

Appendixes

1. Introduction

This document defines XRD (Extensible Resource Descriptor), a simple generic format for describing resources. Resource descriptor documents provide machine-readable information about resources (resource metadata) for the purpose of promoting interoperability. They also 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 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels* *[http://tools.ietf.org/html/rfc2119]*. IETF, 1997.

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

[RFC 3986] T. Berners-Lee, R. Fielding, L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax* *[http://tools.ietf.org/html/rfc3986]*. IETF, 2005.

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

[Web Linking] M. Nottingham. *Web Linking* *[http://tools.ietf.org/html/draft-nottingham-http-link-header]*. IETF 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, 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] G. Wachob, D. Reed, L. Chasen, W. Tan, S. Churchill *Extensible Resource Identifier (XRI) Resolution V2.0 [http://docs.oasis-open.org/xri/2.0/specs/xri-resolution-V2.0.html]*. OASIS, 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), which is associated with the following XML namespace, is imported into the XRD schema:

```
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/cd02/xrd-1.0-cd02.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

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 arbitrary 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 (whitespace is defined in section 2.3 of [XML 1.0]).

The following schema fragment defines the `xrd:anyURI` complex type, which extends `xs:anyURI` to allow for arbitrary 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. Time values MUST be represented with the UTC designator 'Z'. 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 a resource. 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.

An XRD document MUST (a) be a well-formed XML document as defined by [XML 1.0] with a root element of `<XRD>`, (b) validate against the normative XRD schema identified in Section 1.4, "Schema Organization and Namespaces", and (c) adhere to the additional syntactic constraints defined by Section 1.5, "Common Data Types" and this section.

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 other metadata about the resources and links they describe.

2.1. Element <XRD>

The `<XRD>` element encapsulates the entire resource descriptor. 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 used as a signature reference.

`<Expires>` [Zero or One]
: Specifies when this document expires. See Section 2.2, "Element `<Expires>`".

`<Subject>` [Zero or One]
: Provides the identifier of the resource described by this XRD. See Section 2.3, "Element `<Subject>`".

`<Alias>` [Zero or More]
: Provides an additional identifier for the resource described by this XRD. See Section 2.4, "Element `<Alias>`".

`<Property>` [Zero or More]
: Declares a property of the resource described by this XRD. See Section 2.5, "Element `<Property>`".

`<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 2.6, "Element `<Link>`".

`<ds:Signature>` [Zero or More]
: This XML Signature, included from the [XML Signature] schema, 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:Property"/>
            <element ref="xrd:Link"/>
            <any namespace="##other" processContents="lax"/>
        </choice>
    </sequence>
    <attribute ref="xml:id" use="optional"/>
    <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

2.2. Element `<Expires>`

The `<Expires>` element contains a time value which specifies the instant at and after which the document has expired and SHOULD NOT be used. The value MUST be expressed in UTC form, 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. If present, any cache expiration date specified by the transport protocol SHOULD NOT be later than the time instant indicated by the `<Expires>` element.

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>
```

2.3. Element `<Subject>`

The `<Subject>` element contains a URI value which identifies the resource described by this XRD. This value MUST be an absolute URI. If `<Subject>` is not specified, it is expected that the resource described by the XRD will be identified by other means. Comparison of this value MUST be performed using the scheme-specific normalization rules for the URI, as specified in Section 6.2.3 of [RFC 3986].

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

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

2.4. Element `<Alias>`

The `<Alias>` element contains a URI value that is an additional identifier for the resource described by the XRD. This value MUST be an absolute URI. The `<Alias>` element does not identify additional resources the XRD is describing, but rather provides additional identifiers for the same resource. Comparison of this value MUST be performed using the scheme-specific normalization rules for the URI, as specified in Section 6.2.3 of [RFC 3986].

The following schema fragment defines the <Alias> element:

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

2.5. Element <Property>

The <Property> element declares a property of a resource (when used as a child of the <XRD> element) or link relation (when used as a child of the <Link> element), expressed as a key-value pair. The key is identified by the type attribute, and the value expressed as the string content of the <Property> element. A property MAY have no value if the type identifier alone is sufficient. <Property> elements that contain no value MUST include the xsi:nil attribute with a value of true as defined in [XML Schema]. <Property> has the following attributes:

type [Required]

The type attribute is a URI that identifies the property being declared. This value MUST be an absolute URI. This URI value is application-specific, and is used by the XRD provider to declare a property to consumers familiar with the type identifier. Comparison of this value MUST follow the same comparison rules used for comparing Link Relation Types as defined in [Web Linking].

The following schema fragment defines the <Property> element and its PropertyType complex type:

```
<element name="Property" type="xrd:PropertyType" nillable="true"/>
<complexType name="PropertyType">
    <simpleContent>
        <extension base="xrd:string">
            <attribute name="type" type="anyURI" use="required"/>
        </extension>
    </simpleContent>
</complexType>
```

2.6. Element <Link>

The <Link> element serves as a container for metadata about a relation between the resource described by the XRD and a related resource.

The semantics of this element are similar to the [HTML 4.01] Link element, the [Atom 1.0] Link element, and the HTTP Link Header. The one distinction is that the link relation described by the <Link> element is between the resource described by the XRD (referred to as the *context* resource by [Web Linking]) and the linked resource (referred to as the *target* resource by [Web Linking]), and not between the XRD document itself and the linked resource.

The <Link> element contains the following attributes and elements:

rel [Optional]

This URI value defines the semantics of the relation between the resource described by the XRD and the linked resource. This value MUST be an absolute URI or a registered relation type, as defined in [Web Linking]. Comparison of this value MUST follow the comparison rules for Link Relation Types defined in [Web Linking].

With one exception, the rel attribute is semantically and syntactically equivalent to the Link Relation Types defined in [Web Linking]. It differs in 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.

`type` [Optional]

This string value identifies the media type of the linked resource, and MUST be of the form of a media type as defined in [RFC 4288]. The IANA media types registry can be found at http://www.iana.org/assignments/media-types/. Comparison of this value MUST be done in accordance with [RFC 4288].

Note that this is only a hint and does not override the media type declared by the linked resource itself (e.g. the Content-Type header of a HTTP response obtained by following the link).

`href` [Optional]

The `href` attribute provides the URI of the linked resource. If no `href` attribute is defined, it is assumed the URI can be obtained from a `template` attribute or by application-specific means.

A `<Link>` element MAY contain an `href` attribute or a `template` attribute, but MUST NOT contain both.

`template` [Optional]

The `template` attribute provides a URI template which can be used to obtain the URI of the linked resource. 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 determined by the application through which the XRD document is obtained and processed, and MAY be specific to the link relation type indicated by the `rel` attribute of the corresponding `<Link>` element. Applications utilizing the template mechanism MUST define the template syntax and processing rules (including error handling) as well as the variable vocabulary.

A `<Link>` element MAY contain an `href` attribute or a `template` attribute, but MUST NOT contain both.

`<Title>` [Zero or More]

Provides a human-readable description of the linked resource. See Section 2.7, “Element `<Title>`”.

`<Property>` [Zero or More]

Declares a property of this link relation, as described in Section 2.5, “Element `<Property>`”. It is important to note that this value does not identify any property of the linked resource or the resource described by the XRD, but rather of the link relation between the linked resources.

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

```
<element name="Link" type="xrd:LinkType"/>
<complexType name="LinkType">
    <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="xrd:Title"/>
        <element ref="xrd:Property"/>
        <any namespace="##other" processContents="lax"/>
    </choice>
    <attribute name="rel" type="anyURI" use="optional"/>
    <attribute name="type" type="string" use="optional"/>
    <attribute name="href" type="anyURI" use="optional"/>
    <attribute name="template" type="string" use="optional"/>
    <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

2.7. Element `<Title>`

The `<Title>` element contains a string value that provides a human-readable description for the link. This value is intended only for human consumption and MUST NOT be used by an XRD consumer to affect the processing of the document. `<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 other metadata about the resources they describe. XRD documents can be extended by providing custom, meaningful values for certain URI-based attributes and elements, as well as by extending the XML schema directly.

3.1. Identifier Extension

XRD uses URI-based identifiers for describing resources as well as for describing the relations between resources. Whenever possible, applications SHOULD use well-established URI identifiers for these purposes to promote interoperability and shared semantics. Only when absolutely necessary should new URI identifiers be defined. It is RECOMMENDED that any new identifiers be defined in a formal specification of use. The meaning of a given URI used as such an identifier SHOULD NOT significantly change over time, and the identifier SHOULD NOT be used to mean two different things.

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.

This specification does not define generic rules for the comparison of string or URI values. Therefore, specifications that include XRD schema extensions MUST specify such comparison rules where necessary.

4. Selecting Linked Resources

Link selection criteria is determined by the XRD consumer's needs, and SHOULD be based on the presence, absence, or value of the `<Link>` element attributes or child elements. The selection criteria is usually based on the value of the `rel` attribute with the value of the `type` attribute used as a hint (helping to determine if the linked resource uses a familiar media type).

Selection based on multiple criteria SHOULD be handled by performing multiple selections. Each selection is assigned preference order based on the consumer's needs, and the selection results are compared to determine the most desired set. For example, an XRD consumer processing an XRD document describing an article may wish to select linked resources about the article's author. If that consumer prefers HTML documents over plain text, then the linked resource selection would occur in two steps. First, all links with the `author` relation type would be selected, and if more than one are found, then the most appropriate link would be selected based on its media type.

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 matching element in document order. This process SHOULD be continued for all other matching `<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

XRD documents MUST use enveloped signatures as defined by [XML Signature] when signing. Any signature algorithm defined by [XML Signature] 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).

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] 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 transform (with the identifier `http://www.w3.org/2001/10/xml-exc-c14n#`).

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>. This element SHOULD contain either zero or more than one <XRD> elements. It has the following attributes and elements, and is not otherwise extensible:

`ref` [Optional]
This URI value identifies the resource described by the sequence of <XRD> elements.

<XRD> [Zero or More]
See Section 2.1, “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" use="optional"/>
</complexType>
```

7. Conformance

An implementation is a *conforming* XRD Consumer if it meets the conditions in Section 7.1, “XRD Consumer”. An implementation is a *conforming* Signature-Capable XRD Consumer if it meets the conditions in Section 7.2, “Signature-Capable XRD Consumer”. An implementation is a *conforming* XRD Provider if it meets the conditions in Section 7.3, “XRD Provider”. An implementation is a *conforming* Signature-Capable XRD Provider if it meets the conditions in Section 7.4, “Signature-Capable 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 Consumer if it meets the following conditions:

1. It MUST implement parsing of XRD documents as defined in Section 2, “XRD Document Structure”. Support for XRDS documents, as defined in Section 6, “XRD Sequence”, is OPTIONAL.
2. It MUST conform to the processing rules as specified in Section 4, “Selecting Linked Resources”.

7.2. Signature-Capable XRD Consumer

An implementation conforms to this specification as a Signature-Capable XRD Consumer if it meets the following conditions:

1. It MUST meet all conformance requirements of an XRD Consumer as defined by Section 7.1, “XRD Consumer”.
2. It MUST support the verification of signed XRD documents as defined by Section 5, “XRD Signature”, and MUST support the digital signature algorithm identified by `http://www.w3.org/2000/09/xmldsig#rsa-sha256`, as defined by [XML Signature].

7.3. XRD Provider

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

1. It MUST support the creation of XRD documents as defined in Section 2, “XRD Document Structure”. Support for XRDS documents, as defined in Section 6, “XRD Sequence”, is OPTIONAL.

7.4. Signature-Capable XRD Provider

An implementation conforms to this specification as a Signature-Capable XRD Provider if it meets the following conditions:

1. It MUST meet all conformance requirements of an XRD Provider as defined by Section 7.3, “XRD Provider”.
2. It MUST support the creation of signed XRD documents as defined by Section 5, “XRD Signature”, and MUST support the digital signature algorithm identified by `http://www.w3.org/2000/09/xmldsig#rsa-sha256`, as defined by [XML 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 (Non-Normative)

Example B.1. Simple XRD Example

```
<XRD xmlns="http://docs.oasis-open.org/ns/xri/xrd-1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Expires>1970-01-01T00:00:00Z</Expires>
  <Subject>http://example.com/gpburdell</Subject>
  <Property type="http://spec.example.net/type/person" xsi:nil="true" />
  <Link rel="http://spec.example.net/auth/1.0"
    href="http://services.example.com/auth" />
  <Link rel="http://spec.example.net/photo/1.0" type="image/jpeg"
    href="http://photos.example.com/gpburdell.jpg">
    <Title xml:lang="en">User Photo</Title>
    <Title xml:lang="de">Benutzerfoto</Title>
    <Property type="http://spec.example.net/created/1.0">1970-01-01</Property>
  </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>
  <Alias>http://people.example.com/gpburdell</Alias>
  <Alias>acct:gpburdell@example.com</Alias>
  <Property type="http://spec.example.net/version">1.0</Property>
  <Property type="http://spec.example.net/version">2.0</Property>
  <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"
    href="http://services.example.com/auth" />
</XRD>
```

C. Media Type Definition for `application/xrd+xml` (Non-Normative)

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 parameters:
    "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.

Fragment 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 Committee

D. Revision History (Non-Normative)

Table D.1.

Revision	Date	Editor	Changes Made
Committee Draft 02	9 March 2010	willnorris	• Approved by the XRI Technical Committee as Committee Draft 02.
Working Draft 15	24 February 2010	willnorris	• Minor editorial changes with comparison language.
Working Draft 14	18 February 2010	willnorris	• Reference rfc4288 instead of rfc2046 for definition of media types • Add explicit comparison rules for Subject, Alias, Link type, and Property type. XRD schema extensions MUST declare their own comparison rules where appropriate. • Minor editorial changes.
Working Draft 13	21 January 2010	willnorris	• Clarify different uses of Property element • Regarding URI templates, Applications MUST define template syntax • In Identifier Extension section, emphasize re-use of existing identifers • In Identifier Extension section, make last sentence normative "SHOULD NOT" • Move "Linked Resource Selection" section up one level • Various minor editorial changes

Revision	**Date**	**Editor**	**Changes Made**
Working Draft 12	16 January 2010	willnorris, cantor.2	• Remove XRI 3.0 as a "related specification" • Remove intro paragraph for common data types • Combine element definitions into single section • More consistent language for descripting elements • Reword several descriptions to place additional emphasis on declaritive description before noting similarities, additional notes, etc. (Link, Link/@rel, Link/@type, XRDS) • Clarify semantics of Expires element as well as relationship with transport level expiration date • URI for a Link with an href or template attribute is "application-specific" • Clarify restrictions on use of Title element • Make XRD processing text normative • Cleanup language in XRD Signature sections • Reworked conformance text and created separate conformance targets for signature handling • Demonstrate multiple Titles and Link-level Property in example XRD
Working Draft 11	17 December 2009	willnorris	• Clarify that XRD Signatures must use Exclusive Canonicalization without comments.

Revision	Date	Editor	Changes Made
Working Draft 10	19 November 2009	willnorris	• Replace Type element with Property, which allows for a key-value pair. Add Property as a child element of Link • Change Rel, MediaType, URI, and URITemplate elements to be attributes of Link named rel, type, href, and template respectively. • Fix cardinality of XRD child elements of XRDS. • Additional text to clarify intended use of Subject, Alias, and Property elements, as well as type attribute. • Links MUST NOT contain both 'uri' and 'template' attributes. • Focus link selection on 'rel', using 'type' only as a useful hint. • Replace text about ignoring links with unknown template syntax with instructions to follow the protocol-specific rules on handling bad templates • Update examples to reflect new schema and demonstrate use of a few more elements.
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

Revision	Date	Editor	Changes Made
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, 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

Revision	**Date**	**Editor**	**Changes Made**
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

Revision	Date	Editor	Changes Made
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. • 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

Revision	Date	Editor	Changes Made
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 • 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.

Revision	Date	Editor	Changes Made
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 changes 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 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