



# Extensible Resource Descriptor (XRD) Version 1.0

OASIS Standard

1 November 2010

## Specification URIs:

### This Version:

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

### Previous Version:

<http://docs.oasis-open.org/xri/xrd/v1.0/cs01/xrd-1.0-cs01.xml> (Authoritative)  
<http://docs.oasis-open.org/xri/xrd/v1.0/cs01/xrd-1.0-cs01.html>  
<http://docs.oasis-open.org/xri/xrd/v1.0/cs01/xrd-1.0-cs01.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/) [<http://www.oasis-open.org/committees/xri/>]

## Chairs:

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

## Editors:

Eran Hammer-Lahav  
Will Norris, Google

## 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) [<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 OASIS Membership 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:

Copyright © OASIS Open 2010. All Rights Reserved.

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

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

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

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

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS 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" and "XRD" are trademarks of [OASIS](http://www.oasis-open.org/) [http://www.oasis-open.org/], 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

1. Introduction .....	5
1.1. Terminology .....	5
1.2. Normative References .....	5
1.3. Non-Normative References .....	5
1.4. Schema Organization and Namespaces .....	6
1.5. Common Data Types .....	6
1.5.1. String Values .....	6
1.5.2. URI Values .....	7
1.5.3. Time Values .....	7
2. XRD Document Structure .....	8
2.1. Element <XRD> .....	8
2.2. Element <Expires> .....	9
2.3. Element <Subject> .....	9
2.4. Element <Alias> .....	9
2.5. Element <Property> .....	10
2.6. Element <Link> .....	10
2.7. Element <Title> .....	11
3. XRD Extensibility .....	13
3.1. Identifier Extension .....	13
3.2. Schema Extension .....	13
4. Selecting Linked Resources .....	14
5. XRD Signature .....	15
5.1. Signing Formats and Algorithms .....	15
5.2. References .....	15
5.3. Canonicalization .....	15
5.4. Transforms .....	15
5.5. KeyInfo .....	15
6. XRD Sequence .....	16
7. Conformance .....	17
7.1. XRD Consumer .....	17
7.2. Signature-Capable XRD Consumer .....	17
7.3. XRD Provider .....	17
7.4. Signature-Capable XRD Provider .....	17
 <b>Appendixes</b>	
A. Acknowledgments (Non-Normative) .....	18
B. XRD Examples (Non-Normative) .....	19

---

# 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/xmlsig-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) [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/) [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) [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/os/xrd-1.0-os.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="XRDTType">
  <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:XRDS" />
<complexType name="XRDS" />
  <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/xmlsig#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/xmlsig#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, Planetnetwork
- 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, Planetnetwork
- Jeff Hodges
- Fen Labalme, Planetnetwork
- 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. The XML signature is valid, though the certificate is self-signed.

```
<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>yi2N42KYR6b8dl6TCBKjs4duPuo=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
```

```
<ds:SignatureValue>
  NGJ/tVRnK807FwTic3nQjrEwldo+SgWE/LKE/Q2bge+k4b3Go6d9fLZq0/DX8nyr
  x0nYfpTgxzMUDVUVaDyvnP0MfNmTSJ/yL5bXAV2jW6+NWJH73DXjQoPKn0j1WY2G
  UoTdgnMiiNzKYY+QhWYogy4QXJOMjOF+6OE+uONKvQU=
</ds:SignatureValue>
<ds:KeyInfo>
  <ds:X509Data>
    <ds:X509Certificate>
      MIICsDCCAhmgAwIBAgIJAK6eiEXk2FoiMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
      BAYTAKFVMRMwEQYDVQIQIEWpTb211LVN0YXR1MSEwHwYDVQQKEzhJbnRlcm5ldCBX
      aWRnaXRzIFB0eSBMdGQwHhcNMTAwNTA3MDQ1MDAzWhcNMzgwMTE5MDQ1MDAzWjBF
      MQswCQYDVQQGEwJBVTETMBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UEChMYSW50
      ZXJuZXQgV2lkZ210cyBQdHkgTHRkMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
      gQDVEftG6aMNRBRMu9hHaZUe4ZU5jrbtnsaexNlh4OWnIOj9Tyyk2NfI9w1b2hp5f
      KQf5B9HYeZjowuYKVuc+NQMYgkN7V+YvcJ9ohAjCBZuo9Xcm5CiKeFnz5E6Ad0Fs
      BPnAHch9kZu2joz+iQOp6Av+A78Gvam9giG9ZT3rIj2LZQIDAQABo4GnMIGkMB0G
      A1UdDgQWBBR3yN91g21EACpJ9WaKm3fM+PAPqTB1BgNVHSMEbjBsgBR3yN91g21E
      ACpJ9WaKm3fM+PAPqaFJpEcwRTELMAkGA1UEBhMCQVUxEzARBgNVBAgTC1NvbWUt
      U3RhdGUxITAFBgNVBAoTGEIudGVybmV0IFdpZGdpdHMgUHR5IEEx0ZIIJAK6eiEXk
      2FoiMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEFBQADgYEA3cepBp8h2rwwc+f
      lFahLmJNVOePhw+uCyO8tLWu7Jcq9todVmeCNyqB9hGm2Rvt5yQ69tRpMxQ7WmqS
      O6HbDYzW5APuCPHEtlXoafEq4oWZS8ICPNel68MX5mnXg+XkUOb8cjuY8CwRntBf
      Ehs3jFzXUCMITILlPmE7bb38Hug=
    </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<Link rel="http://spec.example.net/auth/1.0"
  href="http://services.example.com/auth" />
</XRD>
```