



---

# XRI 2.0 FAQ

**1 December 2005**

This document is a comprehensive FAQ on the XRI 2.0 suite of specifications, with a particular emphasis on the XRI Syntax 2.0 Committee Specification which was submitted for consideration as an OASIS Standard on November 14, 2005.

<b>1</b>	<b>General</b>	<b>3</b>
1.1	What does the acronym XRI stand for?	3
1.2	What is the relationship of XRI to URI and IRI?	3
1.3	Why was XRI needed?	3
1.4	Who is involved in the XRI specification effort?	4
1.5	What is the XRI 2.0 specification suite?	4
1.6	Are there any intellectual property restrictions on XRI?	4
<b>2</b>	<b>Uses of XRI</b>	<b>5</b>
2.1	What things do XRI identify?	5
2.2	What are some example uses of XRI?	5
2.3	What are some applications that use XRI?	5
<b>3</b>	<b>Features of XRI Syntax</b>	<b>6</b>
3.1	What were some of the design requirements of XRI?	6
3.2	What are the major components of XRI syntax?	7
3.3	What are cross-references?	7
3.4	What is a subsegment?	8
3.5	What do the "*" and "!" delimiters mean?	8
3.6	How do you express a fully persistent XRI?	9
<b>4</b>	<b>Authorities</b>	<b>9</b>
4.1	What is an authority?	9
4.2	What types of authorities does XRI support?	9
4.3	What are the two new XRI authority options?	10
4.4	What is an XRI root authority?	10
4.5	What is a global context symbol (GCS)?	10
4.6	Is "xri://" optional if an XRI begins with a GCS character?	11
4.7	What is a cross-reference authority?	11
4.8	How does XRI authority delegation work?	12
4.9	How can I use DNS with XRI?	12

<b>5</b>	<b>Normalization and Comparison .....</b>	<b>13</b>
5.1	What are XRI-normal, IRI-normal, and URI-normal forms? .....	13
5.2	How do I tell if two XRIs are equivalent? .....	14
5.3	How do I tell if two XRIs identify the same resource? .....	14
<b>6</b>	<b>Features of XRI Resolution .....</b>	<b>14</b>
6.1	What is meant by XRI resolution? .....	14
6.2	What is authority resolution? .....	15
6.3	What is local resolution? .....	15
6.4	What is a service endpoint? .....	15
6.5	What can I put in an XRD? .....	16
6.6	What is trusted resolution? .....	16
6.7	What is lookahead resolution? .....	16
6.8	What is an XRI proxy resolver? .....	16
6.9	What is an HXRI? .....	17
6.10	How does a cross-reference get resolved? .....	17
6.11	Is it possible to map one XRI as an alias onto another? .....	17
<b>7</b>	<b>XRI Metadata.....</b>	<b>18</b>
7.1	What types of identifier metadata are defined in this specification? .....	18
7.2	How would this metadata be used? .....	18
7.3	What kind of identifier types are expected to be defined in the XRI Identifier Types specification? .....	19
<b>8</b>	<b>XRI and other Identifier Schemes .....</b>	<b>19</b>
8.1	Why not just use HTTP URIs? .....	19
8.2	Can you provide a specific example? .....	20
8.3	Should I use XRIs instead of URNs? .....	20
<b>9</b>	<b>Miscellaneous.....</b>	<b>20</b>
9.1	Is XRI related to something called XNS? .....	20
9.2	How does XRI relate to things like i-name, i-number, and XDI? .....	21
9.3	How are the global context symbol authority namespaces managed? .....	21
9.4	Are there implementations of XRI? .....	21
<b>10</b>	<b>Notices .....</b>	<b>22</b>

# 1 General

## 1.1 What does the acronym XRI stand for?

EXtensible Resource Identifier.

## 1.2 What is the relationship of XRI to URI and IRI?

URI ([RFC 3986](#)) is the IETF/W3C standard for addressing on the Web. IRI (Internationalized Resource Identifier, [RFC 3987](#)) builds on top of the URI specification by extending the syntax to include Unicode characters. It also defined a transformation from an IRI into a valid URI for applications that can only accept URIs.

XRI follows this same model. It builds on the IRI specification by extending the syntax to include features needed by abstract, cross-context identifiers that identify resources independent of any specific network location, domain, application, or protocol. It also defines a transformation from an XRI into an IRI for applications that only accept IRIs (which can then be transformed into a URI for applications that only accept URIs).

## 1.3 Why was XRI needed?

Early Web architecture divided Web addresses (URIs) into two types: URLs (Uniform Resource Locators) and URNs (Uniform Resource Names). The former are “concrete” addresses that identify resources at a specific location on the Internet (and thus can “break” if a resource moves). The latter are “abstract” addresses that persistently identify resources independent of any specific location, domain, or application.

However the URN scheme addressed only two requirements that apply to some (but not all) abstract identifiers: persistence (identifiers that never change) and delegation (decentralized identifier management). Over the past decade a much wider set of requirements for abstract, cross-context identification has emerged.

They have led to the need for a new identifier scheme that was:

- Compatible with URI and IRI.
- Independent of any specific transport or access protocol.
- Capable of expressing a wide variety of existing (and future) identifiers in a common, interoperable syntax (similar to what XML provides for data.)
- Capable of expressing structured or “tagged” identifiers that can incorporate metadata useful in identifier parsing, comparison, and resolution.
- Capable of fully qualifying local identifiers that otherwise may not be globally unique.
- Capable of reflecting a variety of authority relationships (particularly authority delegated between organizations, hierarchies, and federated systems.)

- Capable of expressing both human-friendly and machine-friendly identifiers.
- Resolvable on the Internet.
- Resolvable using a trustable resolution mechanism.

While other identifier schemes met some of these requirements, none met them all.

#### **1.4 Who is involved in the XRI specification effort?**

XRI 2.0 is the product of the [XRI Technical Committee](#) at [OASIS](#). Members of this technical committee include a variety of software and service vendors as well as XRI users of all types: major corporations, non-profit organizations, and community-based digital identity organizations.

#### **1.5 What is the XRI 2.0 specification suite?**

The XRI 2.0 specification suite includes:

- *XRI Syntax 2.0* defines the syntax, relative reference rules, normalization/comparison rules, and IRI/URI transformation rules.
- *XRI Resolution 2.0* defines a lightweight HTTP(S)- and XML-based resolution protocol that may be used with XRIs, together with a trusted version based on SAML 2.0 assertions.
- *XRI Metadata 2.0* defines a small set of globally interoperable XRIs that may be used as cross-references within other XRIs to “markup” an identifier, including its language, date, version, type, or other characteristics.
- *XRI Identifier Types 2.0* is an extension of the XRI Metadata specification that defines a set of XRIs for indicating the type of an identifier that is not natively an XRI, e.g., a UUID, OID, UID, X.500 distinguished name, etc.

Note that as of December, 2005, only XRI Syntax 2.0 has been advanced to Committee Specification status and been submitted for consideration as an OASIS Standard. The other three specifications are expected to be advanced to Committee Draft status in Q1 2006.

#### **1.6 Are there any intellectual property restrictions on XRI?**

No. The XRI specifications are licensed on Royalty-Free terms based on the [OASIS IPR policy](#) (<http://www.oasis-open.org/who/intellectualproperty.php>). See the [XRI TC IPR page](#) for details (<http://www.oasis-open.org/committees/xri/ipr.php>).

## 2 Uses of XRI

### 2.1 What things do XRIs identify?

XRIs can identify organizations, people, concepts, applications, machines, digital objects, or anything else. There is no defined limit on the scope of things (or “resources” as they are referred to in World Wide Web specifications) that can be identified by XRIs.

### 2.2 What are some example uses of XRI?

#### **Organizational or personal ID used for multiple purposes (e.g., contact page, IM, email, P2P, etc.)**

An XRI could be used to identify an individual or an organization across multiple methods of access. For example, the same XRI could be mapped to a contact web page, an instant messaging identifier, an email address, or even be used directly as an identifier in a completely decentralized P2P system such as JXTA.

#### **Single sign on**

Single sign-on systems, particularly those like OASIS SAML that operate across Internet domains, often focus on the interaction between authenticating and relying parties and do not require or specify a particular identifier scheme. XRIs, because they can be used anywhere URIs are used, and because they offer a human-friendly form of persistent identifier, can be a convenient and effective identifier for these SSO systems.

#### **Books in multiple libraries**

A single book title may be identified in several ways: by ISBN, by Dewey Decimal system, etc. If a user wishes to identify a physical book owned by one or more libraries, URI syntax does not provide a standard way to combine the identifier scheme for the book title (e.g. its ISBN number) with the identifier for the library. XRI cross-reference syntax solves this problem (see the sections below for specific examples.)

#### **Auditable and trusted resolution**

Regulatory requirements such as the Sarbanes-Oxley Act, HIPAA and EU data protection directives are forcing businesses to take much more care with the data they create and use and with applications they operate. XRI trusted resolution gives these organizations a way to ensure that they are communicating with and about known parties. Furthermore, XRI resolution provides an audit trail that can be saved for later review where it might be required by these new regulatory regimes.

#### **Purple numbers (transclusions)**

Purple numbers are a method for referring to chunks of content in a web site for the purposes of long term identification as well as the ability to include the content on other pages (transclusion). XRIs make a convenient way to identify these chunks of content across multiple sites.

### 2.3 What are some applications that use XRI?

[I-name](#) and [I-number](#) registry services for privacy-protected digital addressing use XRI.

The [iSSO](#) protocol under development at XDI.org uses XRI resolution to bootstrap usage of SAML 2.0 authentication assertions for cross-domain single sign-on.

The [i-tag](#) initiative for intelligent, identity-enabled tagging of structured content in the blogging and RSS syndication industries uses XRI.

XRIs also form the basis for the [XDI](#) trusted data sharing protocol under development by the [OASIS XDI Technical Committee](#) (<http://www.oasis-open.org/committees/xdi>).

See the [XRI TC home page](#) (<http://www.oasis-open.org/committees/xri>) for additional links.

## 3 Features of XRI Syntax

### 3.1 What were some of the design requirements of XRI?

#### **URI- and IRI-compatibility**

Being able to use an XRI wherever a generic URI or IRI is required.

#### **Transport- and protocol-independence**

No reliance on one transport protocol to be used with the identifier scheme.

#### **Structured or “tagged” identifiers**

The ability to combine identifiers from multiple identifier schemes in order to share/reuse identifiers across contexts and express shared identifier metadata (similar to the way XML standardizes the use of structured data and metadata)

#### **Delegation/Extensibility**

Allowing one identifier authority to control some subset of another identifier authority’s namespace.

#### **Federation**

Bringing together two or more identifier namespaces under a common root or otherwise enabling two or more communities to access each other’s identifiers.

#### **Decentralization**

No requirements for identifier assignment or resolution to be managed or performed by a single entity.

#### **Resolution**

The ability to resolve an XRI using a simple, extensible Internet-based resolution protocol.

### Trusted/secure resolution

The ability for XRI resolution to provide a mechanism for obtaining trusted responses.

### Internationalization

The ability to use all the Unicode features of IRI.

### Persistence

The ability to express the intent that all or parts of an XRI are persistent (i.e., will not be reassigned to another resource.)

### Human-friendliness and machine-friendliness

The ability to support identifiers that can be structured and easily parsed by both humans and machines.

## 3.2 What are the major components of XRI syntax?

The structure of XRIs derives directly from the structure of URIs. XRIs usually start with the string "xri://" followed by an **authority segment** and then the **path** portion of the XRI (if any). As with in generic URI syntax, the authority and path parts may be followed by optional **query** and **fragment** components.

For example, the XRI "xri://@oasis\*xri/docs/syntax?section=security#encoding" can be broken into the following parts:

<b>Scheme</b>	xri://
<b>Authority</b>	@oasis*xri
<b>Path</b>	/docs/syntax
<b>Query</b>	?section=security
<b>Fragment</b>	#encoding

**Table 1: Components of the XRI “xri://@oasis\*xri/docs/syntax?section=security#encoding”**

Note that in the authority (and possibly in the path), there is a further structure consisting of a list of **subsegments** separated by **subsegment delimiters**. In the example above, "\*" is the subsegment delimiter used to separate "oasis" and "xri" subsegments. (Technically, the "@" sign, called a global context symbol (GCS), is also a separate subsegment. An "\*" is implied when it is not present after a GCS character such as "@". See the Authority section below for more information.)

## 3.3 What are cross-references?

Cross-references are a special type of XRI subsegment, delimited using parentheses, that allow any URI, IRI or XRI value (i.e., an XRI without the 'xri://' part) to be inserted in an XRI.

Examples:

<b>XRI</b>	<b>Description</b>
@city*library/(urn:isbn:23232232)	A URN for a book used as a cross-reference within an XRI for a library
=JoeSmith*(+email)	An XRI cross-reference used to indicate a well-known concept (email address) relative to a containing XRI (i.e. =JoeSmith)

**Table 2: Example of cross-references**

Cross-references syntactically set off the encapsulated identifier so it can be used "whole" within the context of another identifier. This allows reuse of an identifier across contexts, a pattern useful for many types of identifiers, including both "proper nouns" (identifier for people or organizations) or "generic nouns" (identifiers for subjects, topics, concepts, or other dictionary terms). During XRI resolution, the cross-reference is treated as a single subsegment, regardless of the syntactic structure of the content in the parentheses.

### 3.4 What is a subsegment?

First, a **segment** is any portion of a URI, IRI, or XRI delimited by either two forward slashes ("/segment1/segment2/"), or a forward slash and the start or end of the identifier ("segment1/segment2".) The **authority segment** is the first segment of an absolute URI, IRI, or XRI (delimited by "/" and "/", e.g., "xri://authoritysegment/").

In XRI syntax a **subsegment** is a subcomponent inside a segment delimited by "\*" or "!". (See below for more about the difference.) For example, the XRI segment "/city\*seattle\*roadmap!1234/" has four subsegments, three delimited with "\*" and one with "!".

Note that URIs and IRIs have a similar concept of subsegments but only within DNS names and IP addresses used in the authority segment. In the case of DNS names and IPv4 addresses, the subsegment delimiter is "." (period). In the case of IPv6 addresses, it is ":" (colon).

### 3.5 What do the "\*" and "!" delimiters mean?

The "\*" (asterisk, splat, star) delimiter indicates that authority who assigned the following subsegment might reassign it to a different resource at some future date. This is called a "reassignable subsegment".

The "!" (exclamation point, bang) delimiter indicates the authority who assigned the following subsegment intends never to reassign it to a different resource. This is called a "persistent subsegment".

Note that merely declaring a subsegment persistent using the "!" delimiter does not by itself make it persistent. As explained in the IETF URN specification ([RFC 2141](#)), persistence is an operational characteristic that can only be maintained by identifier authorities. Such management is outside of the scope of the XRI specification.



### 3.6 How do you express a fully persistent XRI?

By using all persistent subsegments (including persistent cross-references), both in the authority segment and in any path segments. For example, all of the examples below are fully-persistent XRIs:

- `xri:///1002`
- `xri:///1002!82A7`
- `xri:///1002!82A7/!C345.D22A`

(Note that these XRIs begin with “!” as the special starting character reserved for fully-persistent identifiers. It must always be followed by a persistent subsegment, which explains the second “!”. See the “Global Context Symbol” question below.)

Although these XRIs do not use the URN scheme, they fulfill the requirements of fully persistent identifiers as defined by [RFC 1737](#). You may also create a persistent XRI using a cross-reference to one or more fully persistent identifiers such as a URN. For example:

- `xri:///!(urn:isbn:0-395-36341-1)`
- `xri:///1002!(urn:isbn:0-395-36341-1)`
- `xri:///1002!(urn:isbn:0-395-36341-1)/!4!7!92`
- `xri:///1002!82A7/!(hdl:18975/22745)`

Note that while it is valid to include both reassignable and persistent subsegments in an XRI, an XRI that includes at least one reassignable subsegment is not persistent as a whole.

## 4 Authorities

### 4.1 What is an authority?

In the context of identifiers, an authority is the entity that controls a namespace. With XRIs, an authority may host a server that provides XRI resolution services. These servers are analogous to nameservers for DNS. Multiple authority namespaces may be hosted on a single server.

### 4.2 What types of authorities does XRI support?

XRI supports two categories of authorities:

- *IRI authorities*, which include the same two types of authorities supported in URIs and IRIs, i.e., IP addresses and DNS names.
- *XRI authorities*, which include two new types of authority identifiers, GCS characters and cross-references (see below.)

By supporting both categories, XRI authority syntax is a superset of URI/IRI authority syntax, so any authority identifier that can be used in a URI or IRI may also be used in an XRI.

### 4.3 What are the two new XRI authority options?

The two new types of authorities supported by XRI syntax are:

- *Global context symbols (GCS)*—single-character symbols used to indicate the global context of the authority type (see below).
- *Cross-references*, which enable any identifier to be used to identify an authority (see below).

Both GCS characters and cross-references support delegation to other authorities using either reassignable or persistent subsegments. Note that a GCS character may only be used as the first authority subsegment, called the **root authority**, while a cross-reference may be used as the root or at any level of delegation.

### 4.4 What is an XRI root authority?

An XRI root authority is the first subsegment in an XRI authority segment. An XRI root authority may either be a GCS character or a cross-reference. For example,

`xri://@foo*bar` uses the "@" root authority and `xri://(http://example.com)` uses the "(http://example.com)" root authority.

Root authorities allow communities to share a common namespace and uniquely identify shared resources (or resources that are used by multiple parties). XRI resolver clients must be configured with the "bootstrap" XRI resolution metadata necessary to locate a root authority. This is similar to the configuration of root DNS servers in a DNS resolver.

Root authorities may be established by any community that needs interoperable identifier resolution. For instance, communities such as governments, the telecommunications industry, consortia, etc. might manage their own root authorities. There are also organizations that offer public root authorities for global context symbols. For example, see XDI.org (<http://www.xdi.org>).

### 4.5 What is a global context symbol (GCS)?

A global context symbol is one option for specifying the root authority of an XRI authority segment. GCS characters are intended to allow a community of users to agree on a handful of defined namespaces for interoperability and to avoid namespace collisions.

There are five GCS characters, each specifying the type of identifier authority they represent:

@	Organizations
=	People
+	Concepts, topics, or “tags” (typically used only in cross-references)
\$	Reserved for XRIs defined in specifications, such as specifications from the XRI TC (e.g., XRI Resolution, XRI Metadata, XRI Identifier Type); other OASIS Technical Committees (e.g., the XDI TC); or other standards bodies.
!	A root used to identify resources persistently without any associated semantics.

**Table 3: Global Context Symbols**

Examples:

- `xri://@ucsd`
- `xri://@joes.grill`
- `xri://=john.smith`
- `xri://=elvis`
- `xri://=(mailto:joe@yahoo.com)`
- `xri://+address` (*Usage might be: xri://=john.smith/(+address)*)
- `xri://!!1234`
- \$ used for versioning: `xri://=john.smith*(+address)*($v*3)`
- \$ used for language: `xri://@ucsd*($l*fr)*françois`

Resolution of an XRI rooted by a GCS character requires pre-configuration of a resolver (or its proxy) with the location of the GCS root. This is out of scope for the XRI specification suite.

#### **4.6 Is “xri://” optional if an XRI begins with a GCS character?**

Yes. This is a specific feature of XRI syntax to aid in human usability. It enables extremely simple strings to be recognized and interpreted as XRIs by electronic editing software. Examples:

- `=john.smith`
- `@example`
- `+flower`

#### **4.7 What is a cross-reference authority?**

A cross-reference authority is an authority subsegment that contains a cross-reference. A cross-reference may be used at any level within an XRI authority segment, including as the root authority.

Use of cross-references to specify a root authority enables the creation of **private root authorities** that do not require any centralized registration. Cross-referencing a globally unique identifier for the root helps ensure against name collisions (however there is no ironclad guarantee that two or more communities will not choose the same cross-reference for their root.)

Typically a private root authority is identified by a URI. For example:

- `xri://(mailto:bigman@example.com)*friends/name`

As with GCS authorities, an XRI resolver must be preconfigured to recognize a private root.

#### 4.8 How does XRI authority delegation work?

Each subsegment in the authority segment of an XRI corresponds to a separate authority. For example, in `@us*california*oakland`, the following prefixes identify four separate authorities:

- `@`
- `@us`
- `@us*california`
- `@us*california*oakland`

Note that an authority may be identified with several different "paths" of authority subsegments. The following two authority identifiers may resolve to the same authority:

- `@san.francisco*baseball`
- `@mlb*giants`

If they resolve to the same authority, then any subsegments underneath these two authority identifiers will always resolve to the same resource. So, if the above two authority identifiers resolve to the same resource, then the following must also resolve to the same resource:

- `@mlb*giants*manager` and
- `@san.francisco*baseball*manager`

As would the following:

- `@mlb*giants*pitcher` and
- `@san.francisco*baseball*pitcher`

#### 4.9 How can I use DNS with XRI?

As discussed above, the authority portion of an XRI may also contain an IRI authority, which may be either a DNS name or an IP address. These are unambiguous because they start with either a letter or a number, respectively, while a GCS character is always symbol and a cross-reference always starts with a left parentheses. For example, the following are all valid IRI authority segments in an XRI.

- `xri://www.example.com`
- `xri://10.0.0.1/index`
- `xri://[fec0::7]/user/docs/some.html`

XRI resolution, which is based on HTTP(S), is slightly different when using a DNS name or IP address. The DNS name or IP address is used as the authority portion of an HTTP URL with no path (and using HTTP content negotiation) to retrieve an XML file containing the XRI resolution metadata for this authority.

A second way a DNS name can be used within an XRI is embedded as a cross-reference. Since cross-references are opaque strings from the standpoint of XRI resolution process, this method does not use DNS resolution directly.

## 5 Normalization and Comparison

### 5.1 What are XRI-normal, IRI-normal, and URI-normal forms?

Just as the IRI specification defines a transformation of an IRI into a valid URI for applications that can only accept URIs, the XRI Syntax specification defines a transformation of an XRI into an IRI for applications that can only accept an IRI (which can then be transformed into a URI following the rules in the IRI specification.)

Thus an XRI can exist in three different normalization forms, depending on where it is being used. These forms differ only in what characters are escaped (converted to a hexadecimal representation form) and which ones are not. An XRI is considered equivalent in all three normal forms.

**XRI normal form** is the most minimally escaped form. In this form, any Unicode characters allowed by IRI syntax but not URI syntax are normalized according to Unicode Normalization Form KC (NFKC) and UTF-8 encoded. Other than this, no further escaping is performed. This is the preferred form for applications, languages, and protocols that accept XRIs.

**IRI normal form** requires escaping of the small set of delimiter characters that have special purposes in an XRI but may be misinterpreted in the transformation to IRI syntax as defined by [RFC 3987](#). No other escaping is necessary. This is the preferred form for applications, languages, and protocols that accept IRIs (such as XML).

**URI normal form** requires escaping of the UTF-8 encoded characters allowed in an IRI but not in a URI to be fully compliant with [RFC 3986](#). This transformation is defined in [RFC 3987](#).

Note that XRI resolution requires an XRI be converted to URI-normal form in order to be compatible with the HTTP(S) protocol, which requires URIs.

## 5.2 How do I tell if two XRIs are equivalent?

Exactly the same way URI or IRI comparison is performed, i.e., by doing a character-by-character comparison after applying a small set normalization and comparison rules defined in the XRI syntax specification.

## 5.3 How do I tell if two XRIs identify the same resource?

Other than the simple case where the two XRIs are character-for-character equivalent (as discussed above), the only answer is by resolving them to see if they return the same resource. However there is a mechanism in XRI resolution for an XRI authority to assert that two or more XRIs are **synonyms** (identify the same resource) even if they are not character-for-character equivalent. This is particularly useful for determining the equivalence of a persistent and a reassignable XRI for the same resource.

# 6 Features of XRI Resolution

*[Special Note: Based on feedback received during the public review, the XRI Resolution 2.0 Committee Draft 01 specification is currently undergoing revisions to add support for full HTTP proxy resolution of XRIs. This revision is introducing new functionality and associated terminology. The following answers will be further revised once this revision process is complete.]*

## 6.1 What is meant by XRI resolution?

XRI resolution is the conversion of an XRI into metadata describing the resource identified by the XRI, typically including one or more a **service endpoints** (URIs) that can be used to further interact with the resource via a specific network protocol (HTTP, FTP, SMTP, etc.)

This resolution metadata is contained in a very simple, extensible XML document called an **XRD** (Extensible Resource Descriptor). Multiple XRD documents can be returned inside a single **XRDS** (Extensible Resource Descriptor Set) container document. (Note that this set of two documents was formerly a single document called an XRID in Committee Draft 01.)

XRI resolution can be applied to both the authority segment of the XRI and to the local part. The former is called **authority resolution** and the latter **local resolution**. Once authority and (optionally) local resolution has been completed, additional metadata or resource representations may be obtain via one or more concrete network endpoints provided in the resolution metadata.

A final form of XRI resolution, **proxy resolution**, enables XRIs encoded in an HTTP URL format (called an HXRI) to be resolved via an HTTP proxy resolver.

## 6.2 What is authority resolution?

Authority resolution is the first phase of XRI resolution that takes the XRI authority segment and resolves it into an XRD describing the final target authority (the authority identified by the rightmost subsegment).

The authority resolution protocol uses HTTP(S) and a URI construction algorithm to resolve each subsegment from left to right, starting with the community root subsegment (identified either by a Global Context Symbol or a cross-reference).

Note that authority resolution is also defined for IRI authorities (DNS names or IP addresses) through making a single HTTP(S) request to an HTTP URL composed of the DNS name or IP address as the authority segment.

## 6.3 What is local resolution?

Local resolution adds the step of requesting an XRD corresponding to the local part (path and/or query components) of an XRI. This XRD will typically contain one or more service endpoints with which to interact further with the resource described by the local part of the XRI.

Local resolution also uses HTTP(S) and a URI construction algorithm based on the local part of the XRI.

## 6.4 What is a service endpoint?

A service endpoint is a set of elements contained in a Service element inside an XRD that describe a concrete network endpoint at which to obtain further metadata or otherwise interact with the resource described by the XRD. These elements include:

- *Type*—an IRI specifying the type of the service.
- *MediaType*—a string that specifies the media type supported by this endpoint (for use in HTTP(S) content negotiation).
- *URI*—an IRI that specifies the concrete network endpoint at which to obtain this service.

An unlimited number of services can be defined using the contents of these elements without the need to extend the XRD schema. The XRI resolution protocol itself is defined via a set of service endpoint types.

## 6.5 What can I put in an XRD?

An XRD is a very simple and highly extensible resource descriptor format. It has a very small set of defined elements used for XRI resolution and service endpoint description. However, there are numerous places where arbitrary XML may be placed in the document. Examples of such XML could include RDF statements, security tokens, or almost any other metadata about the described resource.

## 6.6 What is trusted resolution?

**Trusted resolution** is the use of signed SAML assertions in XRDs to form a chain of trust from a community root to a target XRI authority. In trusted resolution, each XRD contains unique IDs both for the authority it describes (the **described authority**), as well as the authority producing the XRD (the **describing authority**). In addition it contains an XML DSig element containing the public key the described authority will use to sign the next XRD. This chain of XRDs can be followed through to any depth.

Trusted resolution is an optional feature of XRI authority resolution and need not be implemented by every resolution client or server.

## 6.7 What is lookahead resolution?

**Lookahead resolution** is when a resolution client asks a resolution server to try to resolve more than one authority subsegment at a time on behalf of the client. If the resolution server complies with this request, it will return a chain of XRDs wrapped in an XRDS document to the client.

Note that the difference between this functionality and proxy resolution (below) is subtle and interested parties should consult the XRI Resolution specification for more details.

## 6.8 What is an XRI proxy resolver?

An XRI proxy resolver is an HTTP or HTTPS server that performs XRI resolution on behalf of a resolving client. The resolving client can ask for an entire XRI to be resolved at once and the proxy can perform the resolution steps for the resolving client.

This "on behalf of" resolution is very similar to the notion of recursive resolution in DNS.

XRI proxy resolvers use HTTP content negotiation to determine the resolution response. If asked for an XRDS document in the HTTP Accept header, the proxy resolver will return the complete chain of XRDs corresponding to all the authority subsegments (and, if requested, the local part of the XRI). If the client is non-XRI-aware and does not



specify XRDS as the Accept content type, the proxy resolver will an HTTP redirect based on the service endpoints in the final XRD.

## 6.9 What is an HXRI?

To facilitate backwards compability of XRI with installed HTTP(S) clients and servers, the XRI Resolution specification also defines a standard syntax for expressing an XRI as an HTTP URL. An XRI expressed in this format, called an **HXRI**, will work normally in a non-XRI-aware client such as a standard browser because it will be resolved by an XRI proxy resolver which will issue a redirect to the target resource.

Examples:

- `http://xri.example.com/=john.smith`
- `http://xri.example.com/=john.smith/(+contact)`
- `http://xri.example.org/@company*division/!1234.5678`
- `http://xri.example.net/!!1002!9876!A4C2/!98!8!14`

## 6.10 How does a cross-reference get resolved?

In XRI resolution, a cross-reference is treated as a single opaque subsegment, no matter how complicated its structure is.

For example, in the context of the following XRI...

- `xri://@whitehouse*(=bill.clinton*brother)`

...the following are subsegments for the purpose of resolution:

- `@`
- `*whitehouse`
- `*(=bill.clinton*brother)`

Note that even though “`=bill.clinton*brother`” is itself a standalone XRI with its own authority segment, when contained within a cross-reference its subsegments are NOT resolved independently. Instead, the entire XRI “`=bill.clinton*brother`” is treated as an opaque set of characters and treated identically to the “`whitehouse`” subsegment.

## 6.11 Is it possible to map one XRI as an alias onto another?

Yes, XRDs have an element called "Synonym" which defines two ways to map one XRI to another. An **authoritative synonym** is an XRI assigned to the same resource described by the current XRD by the same authority producing the XRD and thus, if resolved, would produce the same XRD. A **cross-reference synonym** is an XRI assigned to the same resource described by the current XRD by a *different* authority than the one producing the current XRD and thus, is resolved, MAY produce a different XRD.

## 7 XRI Metadata

*[Special Note: Based on feedback received during the public review, the XRI Metadata 2.0 Committee Draft 01 specification is currently undergoing revisions to add support for identifier type metadata. The following answers will be further revised once this revision process is complete.]*

### 7.1 What types of identifier metadata are defined in this specification?

The XRI Metadata specification is intended to define a small set of XRIs for general-purpose identifier metadata that aids cross-context interoperability. It includes five metadata types:

- *Language*—metadata that specifies the language in which an identifier is intended to be understood, interpreted, or pronounced (using [RFC 3066](#) language tags.)
- *Date*—metadata representing the date and time an identifier was assigned (in XML datetime format.)
- *Version*—metadata representing the version of an identifier (in simple dot- or dash-delimited integer format.)
- *Type*—metadata indicating the defined type of an identifier (an enumerated list to be defined in the Identifier Type specification.)
- *Annotation*—non-significant metadata used for human-readable annotations (the XRI equivalent of comments in XML.)

Additional metadata types may be added over time if they prove to have very broad applicability.

### 7.2 How would this metadata be used?

The metadata is specified in the form of XRIs assigned using the global context symbol “\$”, which is reserved for specifications such as this. All metadata in this specification is assigned using single character namespaces—\$l for language, \$d for date, \$v for version, \$t for type and \$- for annotations.

This metadata is then intended to be used in XRI cross-references to describe other segments or subsegments of an XRI. Following are several examples:

- `xri://@ucsd*($l*fr)*françois`
- `xri://=john.smith/(+address)*($d*2004-09-23T11:52:45-05:00Z)`
- `xri://@example*division/resource*(v*3.2)`
- `xri://xri.example.org/($t*oid)/2.57.18.7`
- `xri://!!1002!9876!A4C2*($-human.resources)/!98!8!14`

### 7.3 What kind of identifier types are expected to be defined in the XRI Identifier Types specification?

The purpose of the Identifier Types specification is to specify a set of XRIs in the \$t namespace that can be used to “tag” commonly used identifiers for purposes of cross-context interoperability. Many of the candidate identifier types will be drawn from the work currently underway at the [Core Identifier Workgroup](#) and at the [Network Applications Consortium XRI Profile Working Group](#). These include UUIDs, OIDs, URNs, X.500 distinguished names, operating system usernames, etc.

## 8 XRI and other Identifier Schemes

### 8.1 Why not just use HTTP URIs?

*(Note: this question is also answered in Section 1 where the motivations for creating XRI are discussed.)*

First, with the revisions underway to the XRI Resolution specification, there will now be a defined HTTP URI format in which all XRIs can be expressed. So in essence, XRIs *can* be treated entirely as HTTP URIs for the purpose of backwards compatibility with HTTP infrastructure.

From a broader perspective, however, the reason HTTP URI syntax itself was not used is that it does not fulfill several of the most important requirements for abstract, cross-context identifiers. Specifically:

- *HTTP URIs are bound to a specific transport protocol.* A fully abstract identifier scheme needs to be independent of any specific transport or access protocol.
- *HTTP URIs do not support sharing structured, “tagged” identifiers across contexts.* A fully abstract identifier scheme will enable “identifier markup” just like XML enables data markup.
- *HTTP URIs do not define a clear way to get at metadata (as opposed to data) about a resource.* A fully abstract identifier scheme can cleanly distinguish access to resource metadata vs. a resource itself.
- *HTTP URIs offer only a very limited authority delegation model.* Abstract identifiers need to be able to express logical authorities and authority delegation relationships.
- *HTTP URIs do not offer a trusted resolution mechanism.* A fully abstract identifier scheme can define a trusted resolution protocol independent of DNS.

- *HTTP URIs do not offer a standard syntax for persistence.* A fully abstract identifier scheme can clearly distinguish between persistent and reassignable identifiers.

## 8.2 Can you provide a specific example?

The library book scenario in section 2 is a good example. Say a library system uses URNs in the ISBN namespace to identify books and DNS subdomains to identify its library branches. HTTP URI syntax does not provide a standard way to express the URN for the book title in the context of the DNS name for the library branch. XRI cross-reference syntax solves this problem by allowing the library (and even automated programs running at the library) to programmatically construct the XRIs necessary to address any book at any branch. Examples:

```
xri://broadview.library.example.com/(urn:isbn:0-395-36341-1)
xri://shoreline.library.example.com/(urn:isbn:0-395-36341-1)
xri://northgate.library.example.com/(urn:isbn:0-395-36341-1)
```

This is just one example of the power of structured identifiers. Say the library wanted to indicate the type of each book available. By establishing a simple XRI dictionary of book types, it can now programmatically construct XRIs that include this metadata,

```
xri://broadview.library.example.com/(urn:isbn:0-395-36341-1)/(+hardcover)
xri://broadview.library.example.com/(urn:isbn:0-395-36341-1)/(+softcover)
xri://broadview.library.example.com/(urn:isbn:0-395-36341-1)/(+reference)
```

## 8.3 Should I use XRIs instead of URNs?

Fully persistent XRIs are functionally equivalent to URNs. They also have the advantage of the other features of XRI syntax, resolution, and metadata. In addition, XRI architecture is unique in that it allows resources to be simultaneously identified with both persistent and reassignable synonyms—the former for unambiguous, long-lived identification and the latter for human usability.

All these would be good reasons to consider XRI for persistent resource identification.

# 9 Miscellaneous

## 9.1 Is XRI related to something called XNS?

XRI is only indirectly related to XNS. Several of the members of the XRI technical committee were involved in developing XNS and saw a need for the creation of an

identifier standard that met the criteria discussed in previous questions. The XNS Addressing specification was submitted to the XRI TC and provided much of the inspiration for what became XRI.

## 9.2 How does XRI relate to things like i-name, i-number, and XDI?

These are external efforts which layer on top of the XRI specification for digital identity and data sharing purposes.

An **i-name** is a reassignable XRI used for personal or organizational digital identity. An **i-number** is a persistent XRI used for personal or organizational digital identity which consists entirely of persistent subsegments.

**XDI** (XRI Data Exchange) is a separate OASIS technical committee working on an architecture and specification for privacy-controlled data exchange where all data is identified using XRIs.

## 9.3 How are the global context symbol authority namespaces managed?

The XRI specifications do not specify any particular management solution for the GCS namespaces. The XRI Technical Committee's expectation is that communities that need interoperable XRI resolution will manage their own GCS namespaces.

There is at least one organization that is offering global GCS namespace management. XDI.org is a non-profit organization founded by a number of interested parties, some of which are members of the XRI TC. However, this effort is not related to or sponsored by the XRI TC.

## 9.4 Are there implementations of XRI?

All of the following are Open Source

- OpenXRI (<http://www.openxri.org>) - a Java XRI resolution server and client implementation.
- [IDCommons I-Broker](http://ibroker.idcommons.net/) (<http://ibroker.idcommons.net/>) - a system building on top of XRI to provide i-name and i-number registration and related identity services for individuals and organizations. Includes an XRI resolver. Built in PHP.
- Gabe Wachob's XRI Escaping Tool (<http://www.wachob.com/xriescape>) - an online tool (with Python source) of the URI and IRI normal form escaping mechanism.

## 10 Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights.

Information on OASIS's procedures with respect to rights in OASIS specifications can be found at 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 implementors or users of this specification, can be obtained from the OASIS President.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS President.

**Copyright © OASIS Open 2005. All Rights Reserved.**

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 paragraph are included on all such copies and derivative works. However, this document itself does not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document 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 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.