OASIS Extensible Resource Identifier (XRI) TC FAQ

January 16, 2004

This FAQ is an overview of the work of the OASIS XRI Technical Committee. For additional questions or comments, please email the TC chairs, Gabe Wachob (gwachob@visa.com) or Drummond Reed (drummond.reed@onename.com).

Table of Contents

General Questions

What is an XRI?

An XRI is a new type of identifier that builds on the URI specification. The XRI specification adds a new layer of structure not present in generic URIs and defines a resolution scheme to make XRIs usable in a wide number of contexts.

What features do XRIs have that URIs don't?

The primary distinguishing features of XRIs are:

- XRIs are "location-independent" – the content of an XRI is decoupled from the network location of any data or services associated with the XRI. This means, among other things, that accessing a resource associated with an XRI isn't limited to a particular network location or protocol. Most URI schemes in wide use today specify identifiers that imply a close relationship to a network location or protocol.

- XRIs can assert "persistence" of *parts* of the identifier. XRIs allow two types of separators (permanent and reassignable) between segments of an XRI, so the XRI itself can suggest that certain parts of the identifier are intended to be long-lived "primary keys" (a concept borrowed from database technology) and others are subject to change. Most URI schemes, on the other hand, are all-or-nothing. With URNs, for example, the entire identifier is persistent. With HTTP URIs, the entire identifier is potentially transient. XRIs allow these characteristics to be combined in different parts of a single identifier.
- XRIs provide the ability to "contain" other URIs or XRIs in the form of *cross-references*. Conceptually this is similar to using quote marks in English to talk about someone else's text, Cross references allow a well-defined URI or XRI to identify not only a concept or resource, but also to identify that concept or resource relative to another concept or resource. For example, a URI could identify a book, and an XRI could then identify a particular library's copies of the book by using that URI in a cross-reference appended to the library's XRI. If urn:isbn:0-131-10362-8 represents *The C Programming Language* by Kernighan and Ritchie, and xri:@SeattlePublicLibrary represents the Seattle Public Library, then xri:@SeattlePublicLibrary/(urn:isbn:0-131-10362-8) represents *The C Programming Language* in the context of the Seattle Public Library.

  A *self-reference* is a special form of cross-reference that indicates that an XRI is not intended to be resolved on the network but is intended solely as a unique identifier. For example, (xri:@SeattlePublicLibrary/(urn:isbn:0-131-10362-8)) is a self-reference because it is entirely enclosed by parentheses, and as such represents the identifier itself.
- XRIs allow unlimited delegation of namespaces. While many URI schemes rely on DNS delegation, XRIs also have the ability to use abstract (non-DNS) names or identifiers that can contain a wider set of characters and strings. The XRI specification defines several global namespaces for abstract non-DNS identifiers (e.g., the "@" in xri:@example/foo represents a standard global namespace for organizations). In addition, a cross-reference can also serve as a namespace root, thus creating a private or community-based identifier space whose management is entirely community defined (as in xri:(http://www.example.com)/foo).
- XRIs are built from the ground up to use Unicode for internationalization. XRIs build on the IRI effort to internationalize URIs.

What makes XRIs "extensible"?

Applications using particular URI schemes have to invent new ways of structuring URIs to do things like:

- Versioning of resources
- Cross-context resource identification (e.g. identifying a resource such as an HTTP URI in the context of another HTTP URI)
- Parameterization of identifiers to allow programmatic creation of URIs (e.g. creating a URI which represents a specific user's home page in a portal web site by inserting the user's id in a particular part of the HTTP URI).

XRIs provide several mechanisms to standardize these functions (when possible) or to make them easier to specify completely. For example, XRIs provide a mechanism for declaring a version number using cross-references. Also, cross-references can contain arbitrary identifiers, and these can be used to attach special meaning to a segment (e.g. "the following segment should be interpreted as a Name"). This allows XRIs to be structured in an application-extensible manner (through URIs) while keeping them easy to define and parse because of the well-defined syntax of cross references.

What do XRIs identify?

XRIs can identify anything that can be identified. They can identify the same universe of things that can be identified by URIs, for example. They may identify abstract concepts, networked resources (such as documents or services), people, organizations, or things. An XRI may be bound to some network representation of a resource at resolution time, but this does not necessarily imply that this representation is actually the resource identified by the XRI. XRIs can be used in a variety of contexts, and the resource they identify cannot be determined by inspecting the XRI.

How are XRIs resolved?

The XRI specification defines a two-stage resolution protocol based on HTTP. The first stage is resolution of the first segment of the XRI (the *authority* portion) to the network endpoint of the authority that defines the local namespace. In the second stage, the resolver sends the remainder of the XRI (the *local access* portion) to the authority for resolution, using one of an extensible list of local access protocols. Currently, only HTTP is defined as a local access protocol, but defining new protocols is trivial.

While the XRI specification provides an HTTP-based resolution method that is suitable for use on the Internet, XRIs will likely also be used in closed systems such as databases, intranets, and even standalone computer systems where other resolution protocols may be more appropriate.

XRI Applications

For what types of applications will XRIs typically be used?

XRIs can be used to identify any type of resource for any type of application, in much the same way URIs (HTTP or otherwise) are used today. However, XRIs are optimized for domain-, location-, and application-independent identification of resources, so they are ideal for applications dealing with resources that have identity independent of a network, persist for long periods of time, or move or maintain relationships across multiple domains or locations. The types of resources for which XRIs are particularly well-suited include people, groups of people, devices, applications, physical assets, and digital assets.

One example of an application that might benefit from XRIs is a *dynamic address book.* XRIs enable the persistent identification of identity attributes, such as those required for PKI key distribution and revocation, electronic business cards, and other common attributes that need to be exchanged, linked, and synchronized across distributed directories.

What is the relevance of XRIs to online dictionaries and searching?

XRIs were designed for identifying and sharing fully abstract resources such as concepts, subjects, and topics (collections of which are commonly called *taxonomies* or *ontologies*) across a large population of domains and users. Three features of XRIs are particularly suited for this application:

1. The XRI *general namespace* (identified by the "+" prefix) is reserved for identifying generic concepts of all kinds.
2. Cross-references make it possible to address an instance of a concept (a phone number, a shoe size, a blood type, an invoice, a song) relative to a specific context (e.g., a person, a workgroup, a network, a device, a database, etc.)
3. Persistent XRI syntax makes it possible for multiple human-friendly XRIs (in one or multiple human languages) to be mapped to a persistent XRI *index value*, the Web equivalent of a global foreign key

in a database. This provides an efficient long-term way to identify an underlying concept even as human language undergoes "semantic drift."

The primary extensibility mechanism in XRI is similar to XML—the ability to add new "tags" through the use of cross-references (identifiers nested within other identifiers; see "What features do XRIs have that URIs don't?") As with XML, the number of cross-references that can be added by an XRI authority is unlimited, allowing authorities to extend the utility of XRIs while remaining interoperable across all XRI implementations.

Technical Questions

How is XRI syntax different from HTTP URI syntax?

For the most part, XRI syntax is a superset of HTTP URI syntax, and most HTTP URIs can be converted to XRIs simply by changing the scheme from "http" to "xri."

XRIs are structured much like HTTP URIs in that there is an initial *authority* segment followed by one or more "/"-separated *path* segments followed by optional *query* and/or *fragment* segments. XRIs differ from HTTP URIs, however, in that *all* XRI path segments are comprised of dot- or colon- separated sub-segments.

In one form of an XRI, the authority is a DNS name or an IP address. In this form, the XRI resembles an HTTP URI. This feature is provided primarily to ease the transition from HTTP URIs to XRIs.

In the more typical case, however, the XRI authority segment contains a dot- or colon- separated list of authorities. This *delegated authority identifier* is resolved left to right, with each authority supplying the location of the next. The last authority knows how to resolve the local access portion of the XRI.

The additional features of XRIs are reflected by a small set of enhancements to generic URI syntax, as shown in the table below. In some cases, extra escaping rules are defined to allow XRIs to be used in contexts where URIs are expected.

Syntax Feature	URIs	XRIs
Start of global authority	//	// DNS names and IP addresses + Generic concepts = People @ Organizations * User-relative (shortcuts) ! Annotations $ Special XRI metadata
Segment delimiters	Slash for all segments; dot for DNS names or IP address in authority segment	Slash for all segments; dot or colon for all sub-segments within segments (dots for reassignable identifiers, colons for persistent identifiers)
Cross-references (identifiers nested within other identifiers)	N/A	Parentheses can be used to nest XRIs or URIs to any depth
Self-references (identifiers not intended for resolution)	N/A	The entire identifier is placed in parentheses
Internationalization	DNS names can be partially internationalized using	Fully internationalized UTF-8 encoding for all segments (except DNS names and IP

ASCII encoding	addresses) per W3C IRI guidelines

What do XRIs look like?

The samples below are taken from the XRI 1.0 specification:

Global Context Symbols

```
xri:@ExampleCorp
xri:@ExampleCorp.www
xri:@ExampleCorp.website
xri:=JohnDoe
xri:=JohnDoe.home
xri:=JohnDoe.work
xri:+flowers
xri:+flowers.rose
xri:+flowers.daisy
```

Cross References

```
xri://www.example.com/(+management)/(+CEO)
xri:(urn:oasis:spec:2040)/(+index)
xri:(mailto:john.doe@example.com)/(+phone)
xri:=JohnDoe.home/(+email)
xri:=JohnDoe.home/(+email).($v/3)
```

Self References

```
xri:(+flowers.rose)
xri:(//www.example.com/dictionary/flowers/rose)
xri:(http://www.example.com/dictionary/flowers/rose)
```

HTTP URIs Converted to XRIs

```
xri://www.example.com/pages/index.html
xri://[2010:836B:4179::836B:4179]/pages/index.html
xri://www.example.com/inventory.parts/widget.subwidget.foobarator
xri://www.example.com/:inventory:parts/:12:7:234
```

Internationalized XRIs

French:	`xri:@ALaFrançaise/areté`
Hebrew:	`xri://בא.דג.ef/gh/יט/לכ.html`
Japanese (Kanji):	`xri:=崎村夏彦/(+本籍地)`

So are XRIs URNs?

XRIs are not URNs because they are not defined in the URN scheme. However, a fully persistent XRI – one in which all sub-segments in all segments are persistent—fulfills the same functional requirements as a URN.

With XRI syntax, you can also write partial or relative persistent identifiers, because individual segments of the XRI can be persistent even if the top-level authority or other parts are not. For example, an airline may want to assign permanent customer identifiers within its own domain/namespace, but that airline's identifier may need to be reassignable in case of a merger.

What security issues exist with XRIs?

In general, security issues should be analyzed with respect to the application using XRIs, not XRIs themselves. Resolution of XRIs, however, can be made secure in two ways. The first is using Secure HTTP for all resolution requests in the specification-defined resolution authority and local-access protocols. This provides sufficient security in many cases. In addition, an upcoming secure resolution specification will define the use of digital signatures to allow a digitally verifiable resolution trail for XRIs.

What other technologies and standards does XRI integrate with or leverage?

XRIs are designed to integrate fully with Web architecture, Web services, and any system that employs URIs. Because the XRI specification defines a process by which XRIs are converted to legal URIs, any application or system specifying the use of URIs can use XRIs.

Specifically, XRIs use the following standards or technologies:

- URI, IRI – XRIs are URI-compatible and rely on the IRI specification (currently in draft) for internationalized identifiers
- HTTP – XRI resolution depends on HTTP for discovering the network location for identifier authorities and for accessing resources on the network that are identified by XRIs or that represent the resources identified by the XRIs
- Unicode – XRIs leverage Unicode for internationalized characters
- REST Architecture – The default XRI local access protocol is inspired by REST and supports a REST-compliant mode of interaction
- XML – XRIs are useable anywhere an “anyURI” XML Schema Data type is required, and XRIs make convenient XML Namespace identifiers; furthermore, XML is used in the XRI-specified resolution scheme