



Caching in with Resolvers

Norman Walsh

XML Conference & Exposition 2003
07-12 December 2003

<http://www.sun.com/>



Version 1.0

Introduction

- Using URIs
- The Problem
- Solutions
 - Catalog-based Resolution
 - Proxy Caches
- Parting Thoughts and Q&A

Using URIs

Relative URIs

Absolute URIs on the Local File System

Absolute URIs on the Network

How do we use URIs to address resources?

Relative URIs

- dbpoolx.rng
- ../xml/docbookx.dtd
- ../../xsl/html/docbook.xsl

These are context dependent.

Absolute URIs on the Local File System

- `file:///c:/xml/docbook42/docbookx.dtd`
- `file:///share/schemas/relax-ng/docbook/4.2/docbook.rng`
- `file:///export/home/john/doctypes/xml/docbook/4.2/docbookx.dtd`

These identifiers are only useful (in general) on the system where they were created.

Absolute URIs on the Network

- <http://docbook.org/rng/4.2/docbook.rng>
- <http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd>
- <urn:publicid:-:OASIS:DTD+DocBook+XML+V4.2:EN>
- <http://docbook.sourceforge.net/release/xsl/current/html/docbook.xsl>

These offer the most unambiguous identification.

The Problem

In Theory...

In Practice...

Demo

Therefore...

An old chestnut: in theory, theory and practice are the same but in practice, they aren't.

In Theory...

Standards encourage us design for a perfect world. Namespace documents, schemas, stylesheets, and other files are identified by URIs on the global web where:

- The network is universally available and
- There is no latency

In Practice...

- Networks go down
- Firewalls and security measures interfere with access
- Our machines are sometimes physically disconnected
- Latency is sometimes significant



Demo

Demo 1

Therefore...

- While it's useful, interoperable, and important to identify documents with URIs on the global web,
- It is convenient, sometimes necessary, to store representations locally and
- To access them transparently instead of “hitting the web” for them each time

Brute Force

Brute Force
Example (1)
Example (2)

Brute Force

The brute force “solution” is to edit every document so that it explicitly references local resources:

- Replace absolute URIs to resources on the global web with absolute or relative references to URIs on the local file system
- Do this every time you exchange documents with colleagues
- (Hopefully you don't need any digital signatures)

Example (1)

```
<!DOCTYPE book SYSTEM
  "http://www.oasis-open.org/docbook/xml/4.2/docbook42.dtd" >
<?xml-stylesheet href="http://www.example.org/style.xsl" type="text/xsl" />
<book>...</book>
```

Becomes

```
<!DOCTYPE book SYSTEM
  "/path/to/docbookx.dtd" >
<?xml-stylesheet href="/local/copy/of/style.xsl" type="text/xsl" />
<book>...</book>
```

Example (2)

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                xmlns:exsl="http://exslt.org/comm
                version="1.0"
                exclude-result-prefixes="exsl">
```

```
<xsl:import href="http://docbook.sourceforge.net/
<xsl:import href="http://docbook.sourceforge.net/
<xsl:include href="http://docbook.sourceforge.net
...
```

Becomes

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                xmlns:exsl="http://exslt.org/comm
```

Example (2) (Continued)

```
version="1.0"
```

```
exclude-result-prefixes="exsl">
```

```
<xsl:import href="/local/copy/of/docbook.xsl"/>
```

```
<xsl:import href="/local/copy/of/chunk-common.xsl"/>
```

```
<xsl:include href="/local/copy/of/manifest.xsl"/>
```

```
...
```

Catalog-based Resolution

What Is It?

Catalog History

An XML Catalog

Catalog Features

Mapping External Identifiers

Mapping URIs

Chaining Catalogs

Rewriting

Delegation

Extension

Miscellany

Catalog: Pro

...

What Is It?

- Explicit mapping from global identifiers to local identifiers
- Maintained by hand (or by local system configuration, e.g., Debian)
- Relies on a *resolver* in the application
- The focus of this presentation is *XML Catalogs*, developed by the Entity Resolution Technical Committee at OASIS

Catalog History

XML Catalogs...

- Developed by the Entity Resolution Technical Committee at OASIS
- Have the same semantics as SGML Open Catalogs, where appropriate
- Support normatively only the SGML Open Catalog entries relevant to XML

An XML Catalog

```
<?xml version='1.0'?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml"
  <public publicId="-//OASIS//DTD DocBook XML V4.
    uri="/share/doctypes/docbook42/xml/docb
  <system systemId="http://www.oasis-open.org/doc
    uri="/share/doctypes/docbook42/xml/docb
  <uri name="http://docbook.org/rng/4.2/docbook.r
    uri="schema/relaxng/docbook.rng"/>
</catalog>
```

Catalog Features

What do catalogs provide? They can...

- Map external identifiers
- Map URIs
- Be chained together for modularity
- Rewrite system identifiers and URIs
- Delegate mapping to another catalog
- Be extended

Mapping External Identifiers

The “public” and “system” entries map external identifiers to local resources based on public and/or system identifiers.

```
<public publicId="-//OASIS//DTD DocBook XML V4.2/  
        uri="/share/doctypes/docbook42/xml/docbook
```

```
<system systemId="http://www.oasis-open.org/docbook  
        uri="/share/doctypes/docbook42/xml/docbook
```

Mapping URIs

The “uri” entry maps one URI to another.

```
<uri name="http://docbook.org/rng/4.2/docbook.rng"
      uri="schema/relaxng/docbook.rng" />
```

```
<uri name="http://docbook.sf.net/xsl/fo/docbook.xsl"
      uri="xsl/fo/docbook.xsl" />
```

```
<uri name="http://docbook.sf.net/bibl/bibl.xml"
      uri="file:/home/ndw/.bibliography.xml" />
```

```
<uri name="http://docbook.sf.net/images/draft.png"
      uri="xsl/images/draft.png" />
```

Chaining Catalogs

Catalogs can be chained together. For example, consider `/etc/catalog.xml`:

```
<?xml version='1.0'?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml"
  <nextCatalog catalog="/usr/share/docbook/4.2/catalog.xml"
  <nextCatalog catalog="/usr/share/html/4.01/catalog.xml"
  <nextCatalog catalog="/usr/share/entities/8879/catalog.xml"
</catalog>
```

Rewriting

Rewriting is a convenient method of moving an entire tree

```
<rewriteSystem systemIdStartString="http://www.w3.org/2002/xmlspec/dtd/2.3/xmlspec.dtd"
               rewritePrefix="/projects/w3c/WWW/2002/xmlspec/dtd/2.3/xmlspec.dtd">
```

This entry would map:

```
http://www.w3.org/2002/xmlspec/dtd/2.3/xmlspec.dtd
```

to

```
/projects/w3c/WWW/2002/xmlspec/dtd/2.3/xmlspec.dtd
```

Delegation

Delegation turns control over to another catalog.

```
<delegatePublic publicIdStartString="-//Example//  
catalog="http://example.com/catal
```

(We'll come back to this example later because the "http" URI in the `catalog` attribute is interesting.)

Extension

The XML Catalog format is extensible. For example, an “suffix rewriting” extension:

```
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
          xmlns:sfx="http://nwalsh.com/xcatalog/1.0"
          <sfx:systemSuffix suffix="docbookx.dtd"
            uri="/share/doctypes/docbook42/xml/docbook42.dtd"
          </catalog>
```

This extension replaces any system identifier that *ends in* “**docbookx.dtd**” with the specified URI.

Miscellany

- Relative URIs in the catalog are resolved against the current base URI
- XML Catalogs support **xml:base**
- Entries can be grouped for convenience

Catalog: Pro

Catalogs can be:

- Easily configured without privileged access to the machine
- Changed on a per-application basis, if necessary
- Managed automatically by install processes
- Configured manually, without every having network access to the resources managed
- Extended by resolvers

Catalog: Con

On the other hand, XML Catalogs:

- must be explicitly maintained, either directly by the user or by processes the user runs
- only function for applications that explicitly support XML Catalogs (or are built on top of libraries that explicitly support them)



Demo

Demo 2

Caching Proxies

What Are They?

Proxy Features

Configuration Example

Configuration Example

Cache: Pro

Cache: Con (1)

Cache: Con (2)

What Are They?

- A system-level cache of recently accessed resources
- A proxy cache sits between the system on which it is running and the rest of the network
- The concrete examples in this presentation are from the *World Wide Web Offline Explorer* or *WWWOffle*.

Proxy Features

- Configured on a system-wide basis.
- Stores anything that it doesn't consider "local"
- Sometimes offers features to control advertising and spam
- Can sometimes be tuned for better XML support.

Configuration Example

What's local?

LocalHost

```
{  
  localhost  
  127.0.0.1  
  mercury  
}
```

Configuration Example

What's important?

Purge

```
{  
  <http://www.w3.org/> age = 2y  
  <http://lists.w3.org/> age = 3y  
  <http://www.oasis-open.org/> age = 2y  
  <http://norman.walsh.name/> age = 1  
  <ftp://*> age = 7  
  age = 4w  
}
```

Additional rules could be added to address specific patterns of URI (e.g., URIs that end in **.xsl**).

Cache: Pro

Proxy caches:

- Operate transparently, requiring no explicit setup by the user
- Are applicable to almost every application that access the network

Cache: Con (1)

On the other hand, caches:

- are substantially more complex to configure and may require privileged access to the machine
- apply globally and cannot be configured on a per-application basis

Cache: Con (2)

- only cache resources that can be accessed at least once. You can't, for example, install a package that you received in email and expect it to work without actually receiving it at least once.
- may discard resources that have not been accessed for some period of time
- are not easily extensible

Other Mechanisms

RDDL

DDDS

RDDL

- Provides a level of indirection for locating representations associated with a URI.
- Allows applications to say not just what they want, but why.
- Solves a slightly different problem, operating conceptually above the resolver used in catalog systems.
- With some improvements to existing APIs, would be very useful indeed.

DDDS

- Designed as an extension of DNS for resolving URIs
- Is really a system for binding strings to data
- Not yet widely deployed
- Is likely to depend on network services that may not be readily available on disconnected machines

Conclusions

Catalogs and Caches

Parting Thoughts

References

Q&A

Catalogs *and* Caches

Remember our delegation example? Suppose you're offline?

```
<delegatePublic publicIdStartString="-//Example//  
catalog="http://example.com/catal
```

- The application asks the catalog to do resolution...
- The resolver asks for the delegated catalog...and the cache provides it
- The resolver continues processing the new catalog...

Parting Thoughts

- XML Catalogs are easy to use, easy to install, and easy to extend
- Caches are transparent and work with almost any application
- Best of all: *they can work together*
- In point of fact, I use both everyday and I wouldn't want to give either up.

References

- The OASIS Entity Resolution Technical Committee, <http://www.oasis-open.org/committees/entity/>
- The World Wide Web Offline Explorer, <http://www.gedanken.demon.co.uk/wwwoffle/>
- Apache XML Commons, <http://xml.apache.org/commons/>
- libxml2, <http://xmlsoft.org/>

Q&A

I can be reached at **<Norman.Walsh@Sun.COM>**