

Identifier String Examples

This document is intended to contain examples possible of persistent identifier strings and their pros and cons.

Issues

Behaviors supported by the resolver

A straw man proposal

Start with a 3-part uri structure:

	domain name	prefix	identifier
simplest form	http://resolver.cornell.edu/	173/	1234.5678
variation 1	http://resolver.cornell.edu/	173/	1234.5678/v1
variation 2	http://resolver.cornell.edu/	173/	1234.5678/v1/pdf
variation 3	http://resolver.cornell.edu/	173/	1234.5678/ps

other variations

variation 4	http://resolver.cornell.edu/	173/	pdf/1234.5678/v1
variation 5	http://resolver.cornell.edu/	173/	1234.5678/v1.pdf

The resolver has one lookup table with an entry for each known prefix. We anticipate that a prefix will normally correspond to a collection, but it could be used for distinct services redirecting to the same collection – the resolver wouldn't care.

That lookup table specifies one of a finite set of behaviors for that prefix.

Strict behavior

The resolver would have a 2nd lookup table specific to that prefix (never mind for now how this would be efficiently implemented). If an entry matching the **entire** identifier part as an indivisible string is found, the resolver redirects to that entry. Otherwise the resolver returns an error. Any of the 6 distinct URI forms above would fail unless the resolver had been provided a redirection location uniquely matching the incoming URI (Note that the redirection locations need not be unique).

Pass through behavior

The resolver would never return an error, but instead pass the entire identifier string along to a **single** redirection address (e.g., <http://arxiv.org>). This allows the collection to treat the different forms of the URI however it needs to.

Other possibilities

It would be possible for a collection to use more than one prefix to allow different behaviors for different circumstances. There could also be additional behaviors such as checking for authentication and/or authorization before redirecting.

Components of the URI (and their ordering)

An example showing different paths for different representations of a resource as an alternative to content negotiation

Jon: why is it necessary to have multiple paths (commons/record, commons/html, and commons/xml)? There seems to be concern that content negotiation is the equivalent of polysemy (one name, many meanings), but this adds another component to every identifier, and we anticipate that only a relative few of our identifiers will offer more than one form of representation

From [URI-based Naming Systems for Science](#)

http://purl.org/commons/record/ncbi_gene/24866

denotes an Entrez Gene record "without commitment as to representation" - that is, the record's declarative content independent of whether the record is rendered as XML, ASN, or RDF. The information in the record may change over time as annotations are added and corrections are made, but it will always be about the same "gene" (a term that unfortunately is not defined).

http://purl.org/commons/xml/ncbi_gene/24866

denotes the XML version of the record.

http://purl.org/commons/html/ncbi_gene/24866

denotes a web page presenting information from the record in human-readable form.

Discussion at the 11/7/08 meeting did not favor this pattern of inserting a format or service designation in the middle of the URI (note that this would require a different behavior in the resolver than the 2 described above). Simeon also pointed out that many browsers do not behave correctly based on a MIME type specification alone, and that file extensions are usually needed (e.g., 1234.5678.pdf).

Layering to achieve branding without having branded identifiers

Registering a domain name, directing it to purl.org (CNAME), and basing identifiers on the registered name
From [URI-based Naming Systems for Science](#)

http://purl.obofoundry.org/obo/OBI_0000225 = http://purl.org/obo/OBI_0000225; both reference http://ashby.csail.mit.edu/cgi-bin/obitem?ref=OBI_0000225

Jon: This might be a way for a collection to continue its own branding when publishing otherwise meaningless persistent identifiers, as for example registering the domain name handle.ecommons.cornell.edu and directing it to <http://hdl.handle.net>, so that a URL in the form <http://handle.ecommons.cornell.edu/1813/734> could be published instead of <http://hdl.handle.net/1813/734>; both would resolve to <http://ecommons.cornell.edu/handle/1813/734>