

 <p>persistent identifier linking infrastructure</p>	<p>web: <a href="http://resolver.net.au/hdl/102.100.272/0N8J991QH">http://resolver.net.au/hdl/102.100.272/0N8J991QH</a>          email: <a href="mailto:policy@pilin.net.au">policy@pilin.net.au</a></p>
---	--

#### Version History

Version	Date	Status & changes	Expression identifiers
V1.0	2007-12-19	RELEASE: Initial release to public	PILIN/4SG2RKNQH hdl:102.100.272/4SG2RKNQH
V1.1	2008-05-22	Tightened intro, changed section headings	PILIN/YVLKJ75RH hdl:102.100.272/YVLKJ75RH

## PILIN Project Guidelines

### Using URIs as Persistent Identifiers

To cite the *latest* version of this work use <http://resolver.net.au/hdl/102.100.272/DMGVQKNQH>

To cite *this* version of this work, use <http://resolver.net.au/hdl/102.100.272/YVLKJ75RH>

#### 1 Purpose/Issue

This guideline presents considerations for projects which choose to use URIs as persistent identifiers. In particular, it focuses on considerations when using http: URIs and URLs as persistent identifiers.

#### 2 Background

The PILIN project models and encourages the use of persistent digital identifiers, to deal with problems arising from the use of URLs to identify resources. The PILIN project considers many persistent identifier solutions which are not URIs to be useful. This contradicts current thinking from bodies like the W3C and the IETF, which promote the use of URIs as *persistent* identifiers as well as locators; they suggest the use of URIs as potentially persistent identifiers to the exclusion of other schemes. See for example the current definition of URIs in RFC 3896 [1], or the W3C finding "URNs, Namespaces, and Registries" [2].

Different identifier schemes interact with the HTTP protocol in different ways. Some are http: URIs (e.g. PURLs [9], ARKs [10]); others are explicitly dissociated from the HTTP protocol (e.g. the Handle System [7], Life Sciences ID [8]). Depending on how it interacts with HTTP, different strategies are needed to ensure that an identifier (including an http: URI identifier) remain persistent.

This document uses definitions and concepts in the PILIN Ontology for identifiers, summarised in [4]:

- An **identifier** is an association of a name with a thing.
- A component is **persistent** if it is managed and maintained for a defined timespan. The main type of persistence considered here is persistence of resolution: the identifier is maintained to resolve to the same thing over its lifespan.
- An **identifier management system** is a collection of definitions, information models, policies, data sources, and services used to manage identifiers.

This document uses the term **Universal Resource Identifier (URI)** in the most general sense defined in [1]:

Copyright © Monash University



This work is licensed under the Creative Commons Attribution-Share Alike 2.5 Australia License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/au/>

This work was created as part of the PILIN project. The PILIN project is funded by the Australian Commonwealth Department of Education, Science and Training, (DEST) under the Systemic Infrastructure Initiative (SII) as part of the Commonwealth Government's Backing Australia's Ability – An Innovation Action Plan for the Future (BAA) under the ARROW Project.

- "... a compact sequence of characters that identifies an abstract or physical resource."
- "The URI itself only provides identification; access to the resource is neither guaranteed nor implied by the presence of a URI. Instead, any operation associated with a URI reference is defined by the protocol element, data format attribute, or natural language text in which it appears."

We also use the definition of **Universal Resource Locator (URL)** from [1]:

- "... the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism (e.g., its network "location")."

We use the term **http: URI** for URIs that use the http: scheme.

### 3 Scope

These guidelines apply to projects considering using http: URIs as persistent identifiers. Since http: URIs remain the default mechanism for accessing online content, the guidelines are applicable to most projects.

The PILIN project position is that an http: URI can be used as a persistent identifier, but at the cost of additional maintenance, which needs to be planned for through explicit strategies.

Additionally, we maintain that if non-HTTP identifiers are used, then services on those identifiers should be exposed via HTTP, to allow interoperability with the rest of the web.

This document is intended to inform projects of some of the costs and responsibilities involved in using http: URIs as persistent identifiers.

### 4 Guidelines

#### 4.1 History

Protocols constrain the services that use them to communicate requests and responses. The URI specification [1] discusses two common identifier management services: resolution (determining how to access a thing) and retrieval (accessing a representation of the thing). As services, resolution and retrieval can be bound to HTTP. An identifier itself is not itself bound to a protocol: it merely associates a name with a thing identified. Protocols only matter when the identifier is used with services like resolution and retrieval, that rely on those protocols.

However, among the many possible schemes for identifiers, protocol-specific schemes are pervasive on the Internet: the protocol only constrains the identifier services, but is also used to identify a scheme typically used with those services. These protocol-specific schemes are grouped together as URI.

In the past users were expected to choose which identifier scheme to use (corresponding to a particular protocol), through the URI prefix. The registered URI prefixes reflect the state of affairs at the beginning of the World Wide Web: there was a large number of identifier schemes available, each scheme mapped to a specific resolution protocol, and no one scheme was privileged.

#### Example:

- ftp://ftp.example.com/1.txt is an identifier in the FTP space,
- http://www.example.com/1.txt is an identifier in the HTTP space,
- gopher://gopher.example.com/1.txt is an identifier in the Gopher

space, and so forth.

URI schemes named after protocols typically use those protocols for their retrieval service, and originally used only those protocols. So an identifier in the FTP space bound retrieval on that identifier to happen through the FTP protocol. However, a URI may now be retrieved through any protocol, in recognition of the fact that a URI is only an identifier, and not a service request [20]. The confusion of identifiers and service requests is long-standing, and an issue this document responds to.

Even when identifiers are not bound to the HTTP protocol, their identifier management systems often expose some functionality via HTTP.

**Example:**

- `http://purl.org/EXAMPLEDOMAIN/1.txt`, is an identifier in the PURL scheme. The PURL identifier management system exposes its retrieval via HTTP, not a distinct PURL protocol.
- `http://hdl.handle.net/102.100.example/1` is a retrieval service call exposed by the Handle System via HTTP. The HTTP retrieval request is translated to a Handle protocol retrieval request by the proxy server at `hdl.handle.net`

Several things have changed in the way identifiers are used online over the past 15 years. Specifically, unlike the early '90s, the HTTP protocol has become the dominant protocol for accessing resources online. This dominance has been exploited to establish the current profusion of web services and web-enabled architectures, and would not have been possible had the protocol space remained as fragmented as it was.

## **4.2 Identifier Schemes**

HTTP protocol identifiers were initially explicitly defined as locators (i.e. URLs) [3]: the context and label of the identifier correspond to a virtual file path and file name on some online retrieval system. This tied them to the retrieval service acting on the locator, so specifying the protocol was essential. But an identifier is defined only as an association between a name and a thing identified; and the string of virtual file path and file name still counts as a name. By that definition, a URL locator is just as much of an identifier as a UUID, Handle, or an ISBN.

Because the context and label of a URL tell you something about the digital object (where it is stored), URLs are meaningful identifiers (see [16] for a discussion of meaningful identifiers). It became clear in time that meaningful identifiers give a poor guarantee of persistence, particularly when based on an object attribute as changeable as network location. Several initiatives tried to address this problem.

- One approach was promoting URNs as location-independent identifiers, as an alternative to URLs [5]. The `info-uri` scheme [6] has a similar structure, although it has a much narrower domain of use. Both are used in small communities: `info-uri` is used in the bibliographic community, and URN in initiatives like the Life Sciences ID [8] for the biology community. Although it had considerable support, URN has not succeeded in displacing URLs, and has not passed into general use.

- A similar approach was to devise identifier schemes not bound to HTTP, again in reaction to the use of locators in URLs. The Handle System [7] is an example of this approach.
- A third approach was to use the HTTP protocol to present identifiers, but to dissociate the identifiers from locators by policy and branding. PURL [9] and ARK [10] are examples of this approach. Note that though these are true URIs, and use the `http://` prefix, they are not called URLs.
- The final approach is currently taken by the W3C: URLs are now explicitly defined as general identifiers (i.e. as URIs) [1], without any expectation that the name correspond to a network path. This acknowledges the problems in allowing locators to be used as identifiers. But rather than set up a URN or PURL scheme in opposition to locator URLs, the W3C redefines URLs as location-agnostic, and the best-practice recommendation ("Cool URLs" [11]) is to ensure URLs are persistent to begin with.

#### Examples:

- URN, Life Sciences ID: *urn:lsid:ipni.org:names:30000959-2*
- Info-uri, National Library of Australia digital collections: *info:nla/nla.mus-an7579855-s2*
- Handle: *hdl:102.100.272/0N8J991QH*
- PURL: *http://purl.oclc.org/OCLC/PURL/FAQ*
- ARK: *http://loc.gov/ark:/12025/654xz321*
- URI Identifier:  
*http://www.w3.org/2001/tag/doc/URNsAndRegistries-50*

Now that URLs are defined as URIs, the W3C and the IETF have discouraged alternative schemes to HTTP for identifiers, and promote `http:` URIs as a universal identifier scheme online [2]. `http:` URIs satisfy the persistence requirements that locators did not. Keeping identifiers bound to the HTTP protocol (though not to the exclusion of others) allows the ongoing leverage of a single protocol for all identification on the Web—whether to identify digital resources or more abstract targets. An example of the latter is the use of URIs as namespaces in XML, with no expectation of resolution to a web page. Another example is the use of URIs in the Semantic Web to identify offline things, e.g. in RDF.

This reality of a single protocol has affected the way identifiers are presented on the web in general. PURL and ARK are anchored to the HTTP protocol by design. Handles are not URIs, yet are usually presented online through resolution service calls in HTTP (that is, they are presented to look like `http:` URIs); the PILIN project policy on Handle citation recommends this [12]. The fact that URNs are not `http:` URIs has impeded their take-up, and while `info-uri` fulfils a clear requirement in the bibliographic space, it is constrained to a specific domain.

### 4.3 Binding identifiers to protocols

Binding identifiers to a protocol is a practical measure that makes sense in the contemporary digital world.

- Identifiers bound to a widely adopted protocol can be used with whichever services are based on that protocol, and can leverage other identifiers in the shared protocol. For instance, an ARK or a PURL can occur anywhere a

URL occurs, and be used the same way a URL is used: they can form hyperlinks, web service calls, targets of HTTP redirection, RDF resources, and so on. By default, a Handle in the hdl: protocol cannot be resolved by a browser, and so cannot leverage all that infrastructure.

- Moreover, binding an identifier to a protocol provides a specific operational context for the identifier name. This helps establish the uniqueness of the name in general: an HTTP identifier will not be confused with an FTP or a Handle identifier. It situates the identifier within the identifier management systems using that protocol: we know that URLs are managed one way, and Handles another.

So in light of the universal use of the HTTP protocol, why would anyone continue to use identifiers that aren't http: URIs?

- For a long time, locators were confused with identifiers: because a locator *can* be used as an identifier, it was initially the only thing used online as an identifier. Realising that locators don't make for persistent identifiers led to the uncoupling of location and identification.
- Similarly, for a long time identifiers for online resources were primarily used with specific protocols, because they were primarily used with specific services relying on those protocols (resolution and retrieval); so they were presented only in the context of those protocols. But an identifier need not be tightly coupled to a protocol. In particular, an identifier can be managed through an identifier management system that supports interfaces in a number of protocols.
- Binding online identifiers to protocols has happened because of a longstanding conflation of identifiers with actions on identifiers, as we have noted: in particular, resolution requests on identifiers. This leads to a confusion between identifiers and services [17]; e.g. a URL can be thought of as an identifier (i.e. a URI), or as a service call (request to resolve the URI). Protocols are necessary for services, but are not necessary for identifiers: an identifier can be used with more than one service, and (as [20] allows) with services bound to more than one protocol (e.g. HTTP URI over WAP, Handle over REST).
- An identifier does not have to be used to invoke a service like resolution; it should not be presented as if it does. This applies to identifiers for offline resources, which will not end up retrieved. But it also holds for identifiers used in services other than conventional resolution.

**Example:** In OpenURL [13], info-uri identifiers are used as parameters of service calls, not service calls themselves. Because an info-uri is a parameter, binding it as an identifier to a protocol would not be useful: the protocol is bound to the OpenURL service call, not its identifier parameter. So in:

***http://resolver.example.edu/cgi?url\_ver=Z39.88-2004&rft.id=info:nla/nla.mus-an7579855-s2***

The info:nla identifier is not used to call a service, but as a parameter; it does not need a protocol of its own. The service call does have a protocol, HTTP; but that involves the service call rather than the info:nla parameter. The info:nla identifier could be used unchanged if the OpenURL call were instead made through SOAP across FTP, precisely because the info:nla identifier is not bound to the HTTP protocol.

Even though OpenURL parameters have nothing to do with HTTP,

OpenURL service calls themselves are often realised through HTTP. So the service call capitalises on the wide availability of the protocol, even if the identifier parameters do not.

## 5 Considerations for URI persistence

While we are reluctant to enter the long-running religious war on HTTP vs non-HTTP identifiers, there are some considerations worth advancing. If your http: URIs are to be persistent and not just locators...

### 5.1.1 ... Do *manage* them for *persistence*. (Do not manage them like throwaway URLs.)

Whether the retrieval key for your resources is a URI, a Handle, or a folksonomy tag, it should still be treated as an identifier. The policy infrastructure needed to make a Handle or an ARK persistent is just as necessary to make http: URIs persistent. (Conversely, without such policy infrastructure, a Handle or an ARK is not inherently more persistent than any URL.) So if you want your URI to be persistent, you should manage it accordingly, and plan to make sure your URIs do not break. (This is no less true of Handles, of course: there exist broken Handles just as there exist broken URLs.)

To date, http URLs do not have a well-defined standard for managing them, as conceded in [21]:

- “there don't seem to be interoperable standards for encoding version metadata, guaranteeing that representations are 1:1 with names, supporting replicated deployment, etc.”

The lack of such URI identifier management standards has been a strength in making the Web grow as a decentralised, informal network. Its growth would have been impossible unless anyone with an Apache distribution could publish http: URIs—without being encumbered by standards for identifier management. But this informality works against persistence from an institutional point of view (not everyone with an Apache box can be trusted to make their http: URIs persistent: see discussion below). It also works against persistence from an administrative point of view: not everyone with an Apache box can keep track of their http: URIs, what they identify, who is allowed to edit them, what metadata they are associated with, how redundancy across hosts and protocols is provided for, etc. etc.

If a non-http: URI scheme provides at least some infrastructure which makes it easier to manage these aspects of identifiers, that infrastructure should be put to work. The decoupling of identifier from service in non-http: URIs makes such management easier—although of course, with the new understanding of http: URIs as just identifiers, good identifier management should now be possible for those URIs as well.

### 5.1.2 ... Do *present* them as *location-independent*. (Do not present them like throwaway URLs.)

Although W3C has changed its mind on the proper definition of a URL, the association of HTTP URIs with location is a cultural reality that will take time to undo. Outside specific domains (XML namespaces, RDF) and formats (URL query strings), “http: URI = locator” is the default assumption a user makes seeing an http: URI. PURL and ARK work around this assumption by labelling their identifiers as something other than URLs. A persistent http: URI needs a strong

policy apparatus behind it, and realistic attempts to educate its user community, if it is to undo the expectation that it is a locator.

One of the most important things to do is to ensure that identifiers are dissociated from file names (as in Cool URIs [11]), and from meaningful attributes in general [18].

**Examples:** The most straightforward way to persuade users that http: URIs are not locators is to present them as http: URI queries, with a distinct identifier parameter: e.g.  
<http://example.com?id=info:nla/nla.mus-an7579855-s2>. Users expect static URLs to be locators, but are prepared to consider URI queries as non-locators.

An alternate strategy, used in Cool URIs and by the W3C, is to treat the http: URI as a directory (with no file suffix), rather than a locator for an individual file: e.g.  
<http://www.w3.org/2001/tag/doc/URNsAndRegistries-50> . The Handle resolver behaves similarly:  
<http://www.handle.net/102.100.272/0N8J991QH>

### 5.1.3 ... Do allocate them as tokens of a contract. (Do not allocate them like throwaway URLs.)

Because of the policy apparatus required to keep identifiers persistent, a guarantee of persistence must be made, and must be seen as trustworthy: a persistent identifier represents a contract entered into by the identifier manager, to keep the identifier accessible and up to date. The more tightly controlled the identifier is, the likelier that the contract can be met. So there needs to be a barrier to entry for persistent identifiers: a party should only be allowed to manage a persistent identifier if they can be trusted to maintain it (see e.g. discussion on Life Sciences ID at [19]).

The point of http: URIs is their universality: anyone with an Apache distribution can publish them. But this low barrier to entry works against persistence. If http: URIs are to be maintained persistently, they need to be managed closely and within a well-defined curation boundary [14]. It should also be made clear to users that the URIs are being managed to be persistent; this is easier if they are located in a trusted domain than if they are not.

### 5.1.4 ... Do expose them like URLs. (Do not expose them as theoretical abstractions.)

Notwithstanding the arguments above, HTTP is unchallenged as the main protocol of the Web, and indeed is what makes the Web possible. Its widespread use effectively future-proofs it, as [2] points out: any future change which obsoletes the HTTP protocol will force a new mapping scheme into being, to take care of all legacy http: URIs. Note that the network effects of the universality of HTTP are given as the major argument for http: URIs and against LSIDs in [20].

Given this reality, a persistent identifier cannot realistically be actionable through the web but avoid the HTTP protocol. This applies even if the action does not involve retrieval, but an offline resource: the growing infrastructure of the Semantic Web entrenches http: URIs in that context as well. Non-HTTP identifier schemes like Handle, the info-uri identifier family, and LSID have demonstrated usefulness online, and allow a cleaner model of identification, uncoupling identifiers from protocols and services just as they have been uncoupled from

locators. But services on these identifiers need to be exposed through the HTTP protocol if they are to interoperate with the rest of the web. For info-uri, this happens through OpenURL; for Handle, through Handle proxy servers.

## 6 References

- [1]. Berners-Lee, T., Fielding, R. & Masinter, L. 2005. *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986. <http://www.ietf.org/rfc/rfc3986.txt>
- [2]. Thompson, Henry S. & Orchard, David. 2006. *URNs, Namespaces and Registries*. <http://www.w3.org/2001/tag/doc/URNsAndRegistries-50>
- [3]. Berners-Lee, T., Masinter, L. & McCahill, M. 1994. *Uniform Resource Locators (URL)*. RFC 1738. <http://www.ietf.org/rfc/rfc1738.txt>
- [4]. Blinco, K., Nicholas, N., Rehak, D., Ward, N. & Wilson, R. 2007. *PILIN Ontology for identifiers and identifier services*. <http://resolver.net.au/hdl/102.100.272/G9JR4TLQH>
- [5]. Moats, R. 1997. *URN Syntax*. RFC 2141. <http://www.ietf.org/rfc/rfc2141.txt>
- [6]. Van de Sompel, H., Hammond, T., Neylon, E. & Weibel, S. 2006. *The "info" URI Scheme for Information Assets with Identifiers in Public Namespaces*. RFC 4452. <http://www.ietf.org/rfc/rfc4452.txt>
- [7]. Sun, S., Reilly, S., & Lannom, L. 2003. *Handle System Namespace and Service Definition*. RFC 3651. <http://www.ietf.org/rfc/rfc3651.txt>
- [8]. Object Management Group 2004. *Life Sciences Identifiers: OMG Adopted Specification*. <http://www.omg.org/cgi-bin/doc?dtd/04-05-01>
- [9]. Shafer, Keith, Weibel, Stuart, Jul, Erik & Fausey, Jon. 1996. *Introduction to Persistent Uniform Resource Locators*. <http://www.omg.org/cgi-bin/doc?dtd/04-05-01>
- [10]. Kunze, J. 2007. *The ARK Persistent Identifier Scheme*. <http://www.ietf.org/internet-drafts/draft-kunze-ark-14.txt>
- [11]. Berners-Lee, B. 1998. *Cool URIs don't change*. <http://www.w3.org/Provider/Style/URI>
- [12]. PILIN Project 2007. *PILIN Project Policy: Citation of Handles within PILIN documentation*. <http://resolver.net.au/hdl/102.100.272/R67T0T0QH>
- [13]. National Information Standards Organization (U.S.) 2004. *The OpenURL Framework for Context-Sensitive Services*. ANSI/NISO Z39.88-2004. [http://www.niso.org/standards/standard\\_detail.cfm?std\\_id=783](http://www.niso.org/standards/standard_detail.cfm?std_id=783)
- [14]. Treloar, Andrew, Groenewegen, David & Harboe-Ree, Cathrine. 2007. *The Data Curation Continuum: Managing Data Objects in Institutional Repositories*. *D-Lib* 13.9/10. <http://www.dlib.org/dlib/september07/treloar/09treloar.html>
- [15]. Mockapetris, P. 1987. *Domain Names—Concepts and Facilities*. RFC 1034. <http://www.ietf.org/rfc/rfc1034.txt>
- [16]. PILIN Project 2007. *PILIN Project Guidelines: Persistence of Identifiers Guidelines*. <http://resolver.net.au/hdl/102.100.272/V89DC0DQH>
- [17] PILIN Project 2007. *PILIN Project Guidelines: Identifier Service Guidelines*. <http://resolver.net.au/hdl/102.100.272/1KKBLPDQH>
- [18] PILIN Project 2007. *PILIN Project Guidelines: Meaningfulness of Labels in Identifiers*. <http://resolver.net.au/hdl/102.100.272/D6N8F0DQH>
- [19] Donald Hobern, *Taxonomic Databases Working Group and LSIDs*. 2006-08-29. Dev Archives Standards Forum.

<http://archives.devshed.com/forums/standards-105/taxonomic-databases-working-group-and-lsids-1958821.html>

[20] Noah Mendelsohn, 2006. *URI Schemes and Web Protocols*.  
<http://www.w3.org/2001/tag/doc/SchemeProtocols.html>

[21] Noah Mendelsohn, *My conversation with Sean Martin about LSIDs*. 2006-07-25. www-tag@w3.org Mailing list. <http://lists.w3.org/Archives/Public/www-tag/2006Jul/0041>