



Version History

Version	Date	Status & changes	Expression identifiers
V1.0	2007-12-07	Release	PILIN/7G6KM4PQH hdl:102.100.272/7G6KM4PQH

PILIN Project Guidelines

Form of Labels

To cite the *latest* version of this work use <http://resolver.net.au/hdl/102.100.272/0HJ9X8JQH>
 To cite *this* version of this work, use <http://resolver.net.au/hdl/102.100.272/7G6KM4PQH>

1 Purpose/Issue

These guidelines present considerations for projects deciding a format for their identifiers.

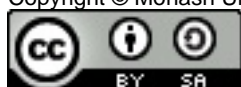
2 Background

This document uses the following definitions and concepts from the PILIN abstract identifier model [1]:

- An *identifier* is an association of a *name* and a *thing*.
- A *name* is a pair of a *label*, and a *context* within which the label is unique.
- An identifier is *arbitrary* if there is no direct relationship between the name and the thing identified. An identifier is *meaningful* if there is a direct relationship between the name and the thing identified.
- Names are *encoded* for presentation, to fulfil specific purposes.
- Names can be parameters of *services* acting on the identifier.
- Service *calls* (including their name parameters) have their own encodings, which may be distinct from the usual encoding of names in isolation.

Much of this discussion is inspired by the justification of label policy provided for the ARK identifier scheme [2].

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 is created as part of the PILIN – Persistent Identifier Linking Infrastructure project. The PILIN project is sponsored by the Australian Commonwealth Department of Education, Science and Training under the ARROW Project.

3 Scope

This document deals with criteria to be considered in a policy for labels. It does not address whether labels should be meaningful or not: that is discussed in a separate guidelines document [3]. Some of the criteria are specific to arbitrary labels.

4 Guidelines

4.1 Labels used in URLs should be URL-safe

Identifiers which will be used within URLs (e.g. as query parameters, or parts of URLs) will be URL encoded [4]. To prevent confusion, the labels within those identifiers should already be URL-safe: that is, they should not contain any characters which will be changed on URL-encoding. Otherwise, there will be confusion on what the preferred form of the label is when it is used within a URL.

Example: The label "a&b" is URL-encoded as "a%26b". A user seeing "a%26b" in a URL query will assume that the actual label is "a%26b", not "a&b".

URL-safe characters for labels are US-ASCII letters, digits and the characters "-", ".", "_", and "~".

4.2 Labels should exclude punctuation where possible

Punctuation is often used to separate components of a name, or components of a service call.

Example: The Handle 102.100.272/0N8J991QH uses a slash to separate the context 102.100.272 from the label 0N8J991QH.

Example: The service call <http://hdl.handle.net/102.100.272/0N8J991QH> uses a slash to separate the call to the resolver service at <http://hdl.handle.net/> from the parameter 102.100.272/0N8J991QH.

For this reason, punctuation within labels should be avoided, or else chosen to avoid confusion with punctuation used in names and service calls. For example, full stop is used to delimit hierarchy in OIDs, but probably would not be confused with the full stops in the authority component of a URL.

Using punctuation as a delimiter within a label (e.g. "721-2") counts as making the label meaningful, which has consequences for its long-term persistence [3].

When punctuation is used within a label, it should be URL-safe where possible (see previous section).

4.3 Labels should not be presented with variant forms

Where possible, labels should not be presented with multiple equivalent encodings or formats. This leads to confusion as to which of the formats is preferred. Moreover, systems having to mapping between multiple equivalent forms of labels may as a result be less robust, particularly if each thing in a domain is expected to have a single identifier.

Labels should not be presented in such a form that systems must preprocess them before operating on them further. Preprocessing means that there is a distinction between the raw and the processed label, which may again lead to confusion as to which is canonical, and makes any systems dealing with the label less robust.

Example: The ARK system allows the labels "712-4", "71-24", and "7124" to be treated as the same: hyphens are optional in ARK, used to break long strings up into chunks, and are stripped out in preprocessing. ARK users may be confused as to whether an ARK identifier with the label "712-4" is the same as an ARK identifier with the label "71-24".

4.4 Labels should be short

Labels intended for human users (e.g. label to be typed in, as opposed to clicked in a hyperlink) should be short enough to be stored in short-term memory.

One rule of thumb from the concept of "chunking" (dividing information into atomic pieces) is that there should be no more than 7 ± 2 "chunks" in a label. For an arbitrary label, this can mean seven characters, though it is possible to increase the size of the chunk through presentation (e.g. "99 05 05 90" can be considered four chunks, not eight.) A meaningful word used as a label counts as a single chunk.

It is useful to think of labels as subject to a "paper napkin" test: if someone can read a label out to me over the phone, and I can write it down on a paper napkin within a few seconds, with no real risk of information loss, then the label will likely be robust enough to be used by humans.

4.5 Labels should be large enough

Notwithstanding the preceding suggestion, labels should be large enough that all the things likely to be identified in the context can be identified through a unique

label. If the number of things to be identified is in the billions, three-letter label will not allow enough labels to be generated to cope.

The following table shows that it is possible to identify a great number of things with reasonably short labels.

	Decimal characters (base 10)	Hexadecimal characters (base 16)	Uncased alphanumeric characters less vowels (base 31)	Uncased alphanumeric characters (base 36)	Case-sensitive alphanumeric characters (base 62)
3 character labels	10^3	$> 10^3$	$> 10^4$	$> 10^4$	$> 10^5$
6 character labels	10^6	$> 10^7$	$> 10^8$	$> 10^9$	$> 10^{10}$
9 character labels	10^9	$> 10^{10}$	$> 10^{13}$	$> 10^{14}$	$> 10^{16}$
12 character labels	10^{12}	$> 10^{14}$	$> 10^{17}$	$> 10^{18}$	$> 10^{21}$

Table 1: Number of identifiers available for different label lengths and character repertoires.

4.6 Labels should be uncased

Labels should avoid case sensitivity. This is both because some systems strip case from labels, and because spoken citation of labels, in particular, tends to ignore case (e.g. reading out a label). (This also falls under the “paper napkin” test.)

4.7 Avoid confusable characters

If possible, characters that look or sound the same should be avoided, as they can be confused when the label is cited. For instance, a completely arbitrary label should not use both 1 and uppercase I (or lowercase l), since these look similar in print.

4.8 Arbitrary Labels should have the same length

Arbitrary labels should have uniform length. This is because discrepancies in length between labels may be interpreted as meaningful. It also allows easy error

checking for arbitrary labels, which are easy to mistype in the absence of meaning.

4.9 Label generation should avoid collisions

Label generation algorithms should avoid creating labels that have been previously used in identifiers. This makes it quicker to generate labels, as the generator does not have to check whether a label is available. It also reduces the risk of a label being accidentally assigned to the wrong thing.

5 References

- [1] *PILIN Glossary*,
<http://resolver.net.au/hdl/102.100.272/HHYMV8JQH>
hdl: 102.100.272/HHYMV8JQH
- [2] Kunze, J. 2007. *The ARK Persistent Identifier Scheme*.
<http://www.ietf.org/internet-drafts/draft-kunze-ark-14.txt>
- [3] *Meaningfulness of Labels in Identifiers*,
<http://resolver.net.au/hdl/102.100.272/D6N8F0DQH>
hdl:102.100.272/D6N8F0DQH
- [4] Berners-Lee, T., Fielding, R., Masinter, L. *RFC 3986: Uniform Resource Identifier (URI): Generic Syntax*, <http://tools.ietf.org/html/rfc3986>

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.