Identifiers

Current Identifier Standards

In the DLP, we use semantically-rich identifiers (see differences between identifiers with and without semantics below). There are many types of identifiers in use:

- "Authoritative" Identifiers – These are the identifiers we use to unambiguously refer to objects. Whenever possible, their use should be preferred over other types of identifiers. They are always:
    - shortItemID (Hoh001.001.0001)
        - These are used in master/non-derived metadata sources, like EAD.
        - These form the first part of filenames.
        - In an "item-level" PURL, these appear after the final slash.
        - These are unique within a single collection within IU.
    - collectionID (lilly/hohenberger)
        - Portion of the PURL that uniquely identifies the collection
        - Does not begin or end with a slash, but may have slashes in the middle.
    - unitID (lilly)
        - Any prefix on the PURL up to the last slash in the "collectionID"
    - collectionName (hohenberger)
        - A shortened form of the collectionID (still unique, but not as clear for human readers) that identifies the collection within an application (often used in situations where slashes are not allowed).
    - fullItemID (/lilly/hohenberger/Hoh001.001.0001)
        - Created from the collectionID and the shortItemID
        - Must begin with a slash
        - These are used in all derived metadata.
        - These are used internally for all objects.
        - These are unique within IU.
    - PURL (http://purl.dlib.indiana.edu/iudl/lilly/hohenberger/Hoh001.001.0001)
        - Concatenation of the purlBase and the fullItemID
        - These are used as external references.
        - These are globally unique.
- Fedora Indentifiers
    - PID (iudl:3413)
    - Fedora URI (info:fedora/iudl:3413)
    - In an attept to minimize the number of identifiers we support, we try to only use Fedora identifiers in internal processes, and do not publish them for public consumption.

Misc Identifier Notes

We may eventually assign a DOI for each item, but we don't have a need to do that yet.

Some older collections use the shortItemID instead of the fullItemID in their OAI records. For example, in Hohenberger, we have Hoh007.013.0016 instead of lilly/hohenberger/Hoh007.013.0016. We will have to support the old forms of the ID.

The old NOTIS system generated IDs consisting of three letters and four numbers (like VAA1234). To easily connect with IUCAT, we have continued using IDs of this form for IU holdings that do not belong to a special collection. Items that use these IDs always include them as the "identifier" portion of the filename. These identifiers are unique across all DLP holdings. We would hope they are unique across the IU library system, but we have no way of knowing if other groups have adopted conflicting standards that emulate the old NOTIS system.

To generate new NOTIS-form identifiers, use the script at http://www.dlib.indiana.edu/cgi-bin/testtrackedit/filename.pl which keeps track of numbers in /usr/local/variations/lastFileName.txt. It is possible to assign a batch of numbers by simply increasing the value in lastFileName.txt, though you should be careful to look at /usr/local/variations/lastFileNameLog.txt to see whether any IDs were generated for other users around the time you were modifying the file.

Semantics in identifiers

Semantic IDs have "meaning" in at least some portion of the ID.

Pros:

- Memorable
- May encode some metadata, increasing the probability that media files can be matched to metadata records at a later date.
- May include a `brand'

Cons:

- "Ultimately, all semantic identifiers are incorrect" --Sean McGrath
- Changes in language use may make some semantic terms confusing or offensive over time.

Opaqueness in identifiers

Opaque IDs have no apparent meaning.

Pros:

- Can encode no metadata (avoids problems when the metadata changes)
- Include no branding, which can be useful when one project/collection absorbs another
- Allow automatic generation
- May be easier to manage (there is no need to maintain the semantics)

Cons:

- Not memorable
- It is nearly impossible to create a "purely opaque" identifier system:
    - Globally-unique identifiers typically have a portion to indicate the instituion that generated each identifier.
    - Many identifier generators include some sort of sequential information, from which users will often make inferences about the objects.
    - Even John Kunze's noid program allows users to enter a prefix for each identifier sequence, which will likely end up having some meaning.

NOID (Nice Opaque IDentifier)

NOID avoids the use of vowels, to prevent the unintentional creation of words (which may be misleading, as in a Variations ID like bad6666). It also avoids the use of lowercase 'l', to minimize confusion with the number 1.

The noid program allows creation of an unlimited number of "minters", which can generate IDs of differing types. IDs may consist of:

- a fixed prefix
- numeric digits
- lower-case ASCII characters, except 'l' and vowels (see above)
- a single checksum character to allow detection of mis-typed IDs

If a minter is set to produce IDs with a fixed number of digits, it may be set to generate the IDs in a random order, otherwise they are generated sequentially.

Identifier resolution systems

PURL: Persistent URL. Simply a URL redirect, which an institution plans to maintain in a working state indefinitely. PURLs can always be resolved by a web browser.

ARK: Archival Resource Key. An identifier with the form ark:/NAME_GENERATING_AUTHORITY/NAME. More commonly, a URL of the form http://NAME_MAPPING_SERVICE/ark:/NAME_GENERATING_AUTHORITY/NAME. The name mapping service is replacable, and there is a system for looking up a new service if an old one is non-functional. Eventually, ARK hopes to drop everything before the "ark:", and have the name mapping service by dynamic, but it is included for now so that web browsers can support name resolution. The URL can be appended with "?" to retrieve metadata about the object, and with "??" to retrieve a commitment statement. Note: Even though CDL primarily relys on ARK, they use PURL for items they do not control, because they don't want to adhere to an ARK persistence statement for these items.

DOI/handle: Digital Object Identifier. An identifier of the form NAME_GENERATING_AUTHORITY/NAME, but sometimes seen written as a URI (with "doi:"). DOIs are usually associated with a resolver via a URL (like "http://doi.acm.org/DOI").

URI: Uniform Resource Identifier. A string, followed by a colon, followed by a string. The initial string should be registered with IANA. Web browsers internally support resolution of a subset of URIs. All URLs are URIs. ARKs and DOIs have not been recognized as standard URIs yet. Of course, the URL form of an ARK is a URI. ARKs may also be referenced through the "info:" URI.

There is much similarity between PURLs and Handles. MIT has published a useful, but slightly biased, comparison (biased because the author had much experience with handles and no prior experience with PURLs).