**DOI syntax for datasets held in DataBank**

DOIs will be assigned to datasets (of all types) held in the Bodleian Libraries’ DataBank. DOIs will only be assigned to datasets that can provide a freely accessible splash page/record. [Not all datasets may be assigned a DOI?

Currently assigning a DOI is an API call and can be decided by the user

The proposed syntax is based on Dryad model (see <https://www.nescent.org/wg_dryad/DOI_Usage>)

Takes the form <http://dx.doi.org/PREFIX/SUFFIX>

**XX.XXXX/bodleianNNNN.V?file=FFFF**

Where

X = DOI prefix (10.5287 for University of Oxford)

N = dataset accession code

V = version number (can be omitted to indicate most recent version)

F = file name [optional]

Example

**00.0000/bodleianfg1k.1/file=Readme**

Example with UUID

**00.0000/bodleian3A6cdeb184-d639-42d7-b6dd-074d68ce17aa**

**.1/filename?urlappend=%3fformat=rdf**

Identifiers for Bodleian Libraries’ datasets need to contain the following information:

1. The (DOI or Handle) prefix. DOI prefix XX.XXXX assigned by BL/DataCite
2. The string "bodleian" for branding purposes*.*
3. An accession code (N) for the data package. UUIDs will be assigned to each dataset and searchable/resolvable, but not used in the visibly assigned DOI – does it matter?. Long-term provision in case of demise of DOI resolver services and for continued discovery via search engine.
4. A version number (V) for the data package. Not all datasets will have DOIs and not all versions of a dataset need to have a DOI
5. Data file name (F). A file name for the data file (is always unique within the dataset)

**Notes [DRYAD]**

General notes:

* The majority of human use will be at the level of data packages, which have relatively short identifiers.
* a version number can be omitted to signify "the most recent version of this object", which is desirable in many situations (e.g., the Dryad search interface, OAI-PMH provider)
* DOIs are defined to be case-insensitive.
* CrossRef's guidelines discourage keeping version information in the DOI, but the guidelines are being revised.

Levels of Persistence:

* The most commonly used formats are registered as DOIs. We may call them "object identifiers". They are permanent and citable.
* The "bitstream identifiers" are persistent, but not at the level of a fist-class DOI. We will make an effort to keep them functional, but they may need to evolve as the implementation of the core DOI architecture changes.

Number of DOIs required:

* assume an average of 2 data files per package
* assume 2 versions per data file (base file, possible migration)
* For each package.... 1 default package DOI, 2 versioned package DOIs, 2 default file DOIs, 4 versioned file DOIs
* We have a minimum of 6 DOIs per "typical" data package, and we can expect 9-12 total identifiers per "typical" data package that undergoes some version changes.

Resolved Questions

1. Should the DOI for a data file be syntactically related to the DOI for a data package? Yes.
   * Reasons to relate them:
     + This is consistent with the Web architecture.
     + People who use Dryad identifiers can perform some simple manipulation on the DOI to get useful results. ("hackable identifiers")
     + It is more obvious when people are citing a file versus a package.
     + The journal representatives think it is important.
   * Reasons NOT to relate them:
     + The more semantics added to an identifier, the more brittle it becomes, and the more likely that problems will ensue. (But it this isn't too bad as long as we are clear to users that the identification is permanent, and the hackability may change.)
     + This structure in the identifiers complicates the use of version numbers in the identifiers (see below).
2. Should we reserve a DOI for "all available versions of this object"? No. This can be treated as a REST command. It will be available in human-readable form from the item display pages. There is no established standard for this, so it is not worth registering a DOI to reserve this functionality.
3. Should migration to a new version (the most common operation that "changes" an item in DSpace) create a new DOI? No. If all we are doing is adding a new bitstream without changing the existing bitstreams, there is no need to force a version number change.
4. Should we use http parameters as part of the "identifier", especially for bitstream identifiers that are unlikely to appear in citations? Yes. Although [DOI parameter passing](http://www.crossref.org/help/Content/07_advanced concepts/Passing_parameters.htm) is extremely clunky, and may change over time, it is moderately persistent. This loss of persistence is reasonable given that it will greatly cut down on the number of DOIs that need to be registered and maintained.
5. Does modification of a file force version changes of the other files, or can the other files continue to use a DOI based on the original package?

The other files can retain their original ID. This lessens hackability, but isn't terrible. We don't expect items to be versioned regularly. For systems that want to ensure they always have the latest version of an item, they can parse the metadata for the package object and download the latest versions. (Or they can just use the abstract DOI, which will always point to the latest version.)