

# Premis Controlled Vocabularies

Following are the controlled vocabularies used in the Medusa repository metadata. These document only lists the values currently in use or imminently anticipated for use. It is expected that this list will grow as over time as new content is ingested into the repository.

## Identifiers

### Object Identifier Types

HANDLE
<p>This identifier type is used for CNRI Handles, for example 10111/MEDUSA:14511575-41a9-47e7-bd6a-946a948609f8-8. The primary identified for each package must be a handle in the UIUC Library's registered namespace of "10111". The first part of the local name must be "MEDUSA:" and after this is a special GUID with an extra check-digit at the end. For details see <a href="#">Package Identifiers</a>.</p> <p>The HANDLE identifier type may be used for other CNRI handles as well.</p>
FILENAME
<p>This identifier type is used for local filenames, including a relative path component if applicable. For a SIP package the filename value must be relative to the root package directory.</p>
CONTENTDM_NUMBER
<p>This identifier type is used when packaging ContentDM records. The corresponding value is the number assigned to the object by ContentDM in its containing collection.</p>
URL
<p>This identifier type is used for a URL associated with the object. At least at some point in time the object could be retrieved from this URL, but may no longer be available from this URL; therefore, the URL is mostly used as an alternate identifier for historical purposes.</p>

### Event and Rights Statement Identifier Types

LOCAL
<p>This identifier type represents an identifier which is only used within the repository. Local identifiers are a concatenation of the handle for the package and an index number that is sequentially incremented within this package, for example 10111/MEDUSA:14511575-41a9-47e7-bd6a-946a948609f8-8/2 where the 2 at the end after the slash is the index number. Because these numbers include the package handle, they should be globally unique. For details see <a href="#">Package Identifiers</a>.</p>

### Agent Identifier Types

UIUC_NETID
<p>This identifier represents a UIUC NetID value taken from the University's LDAP directory. It must be fully qualified with the domain prefix, such as "UIUC\thabing".</p>

EMAIL
This identifier type represents an email address. It is just the email address, it must not include the mailto: part, but just the address, such as "thabing@illinois.edu".

## Object Characteristics

### Message Digest Algorithms

SHA1
This indicates a SHA1 hash has been used for the message digest.

## Events

### Event Types

CREATION
This event type is used for the creation of a new information package.

CAPTURE
This event type is used for the capture, either of metadata or digital files, that make up the package.

MIGRATION
The event type is used for the migration of files between formats or for the translation of metadata formats using XSLT or other means.

## Rights

### Rights Basis

COPYRIGHT
The basis for the granted rights is copyright – used for public domain objects.

### Copyright Status

PUBLIC_DOMAIN
Use when the rights basis is copyright and the work has been determined to be in the public domain.

### Copyright Jurisdiction

Use the Country Name from the [MARC Code List for Countries](#).

## Rights Granted Actions

DISSEMINATE
This "act" value indicates that the repository has the right to disseminate the digital object to anyone.

## Rights Granted Action Restrictions

DISSEMINATE / CAMPUS_ONLY
This "restriction" value indicates that the repository has the right to disseminate the digital object only to people authenticated as members of the University of Illinois at Urbana-Champaign campus community.

DISSEMINATE / LIBRARY_ONLY
This "restriction" value indicates that the repository has the right to disseminate the digital object only to people authenticated as members of the University of Illinois at Urbana-Champaign Library community.

## Object Relationships

### Relationships Types

NOTE: Using the DLF Aquifer Asset Actions for ideas for relationships when possible

COLLECTION
This type of relationship is between objects and collections.

METADATA
This type of relationship is between objects and their metadata.

DERIVATION
This type of relationship is between objects which are derived from other objects.

BASIC_IMAGE_ASSET
This relationship type is used to tie a representation to a specific file which represents a specific rendering of a simple image.

BASIC_COMPOUND_ASSET
This relationship type is used to tie together the various parts of a compound object. All the objects participating in BASIC_COMPOUND_ASSET relationships must be representation type objects. The actual files or bitstreams making up the representation are then related to the BASIC_COMPOUND_ASSET objects by using another appropriate ASSET relationship like BASIC_IMAGE_ASSET.

### Relationship Types and Subtypes

**COLLECTION / IS\_MEMBER\_OF**

This value provides a link between a package and the collection to which it belongs. This relationship is used on the root representation object to link it to containing collection object.

**METADATA / HAS\_ROOT**

This value provides a link between the root representation object and the primary metadata file object.

**BASIC\_IMAGE\_ASSET / PRODUCTION\_MASTER**

This value provides a link between the root representation object and the primary image file object which represents the 'production master'. A production master is the highest resolution image available after the original raw image has been cropped, deskewed, color-corrected, etc.

**BASIC\_IMAGE\_ASSET / ARCHIVAL\_MASTER**

This value provides a link between the root representation object and the image file object which represents the 'archival master'. An archival master is the highest resolution image available before any sort of quality processing, such as cropping, deskewing, or color-correcting, has been done.

**BASIC\_IMAGE\_ASSET / SCREEN\_SIZE**

This value provides a link between the root representation object and the image file object which represents a typical screen-sized preview image.

**BASIC\_IMAGE\_ASSET / THUMBNAIL**

This value provides a link between the root representation object and the image file object which represents a typical thumbnail-sized preview image.

**BASIC\_COMPOUND\_ASSET / HAS\_ROOT**

This value provides a link between the root representation object and the root object node for the compound object. Each compound object must have a single root. The root of a compound object is also a representation type object.

**BASIC\_COMPOUND\_ASSET / FIRST\_CHILD**

This value provides a link between a branch representation object and the first child object of the branch which must be another representation object.

**BASIC\_COMPOUND\_ASSET / PARENT**

This value provides a link between a child object and its parent.

**BASIC\_COMPOUND\_ASSET / NEXT\_SIBLING**

This value provides a link between an object and its next sibling object.

**BASIC\_COMPOUND\_ASSET / PREVIOUS\_SIBLING**

This value provides a link between an object and its previous sibling object.

<b>DERIVATION/ HAS_SOURCE</b>
This value provides a link between an object and the source object from which it was derived. May be used for metadata and digital objects.

<b>DERIVATION/ IS_SOURCE_OF</b>
This value provides a link between an object and an object from which it is derived. May be used for metadata and digital objects.