

The Arboreal Catalog System

Mark J. Schiefsky, Malcolm D. Hyman

July 8, 2003

1 Purpose of the Catalog System

Arboreal's catalog system allows for abstract URIs in a document to be mapped to real URIs. The abstract URI functions as a permanent identifier for an object. This strategy offers two benefits:

1. It allows for stable references to objects.
2. It allows a user to switch between several instantiations of the same object.

Since the locators used in the Archimedes project are valid URIs, they may function as abstract URIs.

Example 1: A user creates a terminology file that contains hundreds of links to the primary source document with locator `036.xml`. The catalog system allows for the links in the terminology file to be interpreted properly whether the text is instantiated on the user's machine as

```
file:///Users/mjschief/monte.mecha.01.1a.1577.xml
```

or on the network as

```
http://archimedes.fas.harvard.edu/texts/monte.xml
```

Example 2: The user wishes to switch between a series of page images on her local machine and an equivalent series stored on a Digilib Image Server. She can accomplish the switch by loading a new catalog file that remaps the abstract URIs for the page images.

2 Syntax of Abstract URIs

Any relative URI may be considered an abstract URI. In addition, URIs that begin with a set prefix may be considered abstract URIs. In the Archimedes project, we use the prefix `http://archimedes/`. For the Archimedes Project, the following are examples of abstract URIs:

- `http://archimedes/036/01/001.jpg`
- `036/01/001.jpg`

These are **not** abstract URIs:

- `http://mjschief.mpiwg-berlin.mpg.de/test.xml`
- `http://archimedes.fas.harvard.edu/036/01/001.jpg`

3 Semantics of Abstract URIs

When Arboreal encounters a potentially abstract URI, it uses the following lookup behavior:

1. Arboreal checks to see if the URI is mapped in the current catalog. If it is, Arboreal considers the URI to be equivalent to the mapped URI.
2. If the URI is not mapped in the current catalog, Arboreal treats it as a real URI.

Example 1: A user clicks on a link with the target `007/01/002.jpg`. A catalog file has been loaded that maps this abstract URI onto

```
http://archimedes.fas.harvard.edu:8080/docuserver/  
digitallibrary/digilib.jsp?007-01-pageimg+2
```

Arboreal opens a web browser window displaying the page image in Digilib.

Example 2: A user clicks on a link with the target `007/01/002.jpg`. This URI is not mapped in the current catalog. Arboreal treats the URI as relative to the location from which the current document was loaded.

4 Prefix mapping

The catalog also allows URI *prefixes* to be mapped — in this case, the portion of the abstract URI that is specified in the catalog will be replaced with the value specified in the catalog.

Example 1: The catalog maps `TestData/` to

- `http://archimedes.fas.harvard.edu/samples/`

The abstract URIs `TestData/1.jpg` and `TestData/2.jpg` will then be translated as:

- `http://archimedes.fas.harvard.edu/samples/1.jpg`
- `http://archimedes.fas.harvard.edu/samples/2.jpg`

Prefix mapping allows an entire range of mappings to be specified with a single catalog entry.

5 Using the Catalog Features in Arboreal

Task 1: Loading a catalog. Select **Load catalog** from the **File** menu.

When a new catalog is loaded, Arboreal adds the entries to a lookup table stored in memory. If a catalog entry maps an abstract URI that has already been defined, the entry in the new catalog overrides the previous definition. Once a document that has a locator has been loaded into memory, Arboreal considers the loaded version canonical and will access that version in lieu of referring to the catalog lookup table.

Task 2: Adding catalog entries. There are two ways of adding mappings to the current catalog within Arboreal:

1. If a document has already been loaded as a master text, it may be added to the current lookup table in memory by selecting **Add to catalog** from the **File** menu.
2. Most file choosers include a button labeled **Add to catalog...**; this button will be enabled if a file is selected in the chooser.

When adding XML files to the catalog, the file will be parsed, and metadata for the catalog will be extracted automatically. In the case of files where metadata is not available, the user will be prompted to enter the metadata manually.

Arboreal uses the docspec system to determine the appropriate metadata elements for a file. If a locator is not specified for the file in a way that Arboreal recognizes, Arboreal uses the filename as the locator. (The docspec system is described in a separate document.)

Task 3: Saving a catalog file. A catalog file that contains all the entries that have been manually added within Arboreal can be saved by selecting **Write new catalog** from the **File** menu. This file will *not* include entries that have been loaded from other catalog files.

6 Default Catalogs

Arboreal recognizes two sorts of default catalogs:

1. When a primary text is loaded from within a bundle, Arboreal will automatically load a file called `catalog.xml`, if it is present at the top level of the bundle. The Archimedes bundle scheme is described in a separate document.
2. Arboreal may be configured to load a particular catalog file automatically on start-up. The URI for this catalog file can be specified in the Arboreal preferences dialog (accessible by selecting **Preferences...** from the **File** menu).

7 Structure of a Catalog File

The root element of the catalog file is `<catalog>`. This element may contain any number of entries. A catalog entry for an object is contained within an `<entry>` element. The `xlink:href` attribute of the `<entry>` element is set to the abstract URI of the object. The `<entry>` element may have three children: `<author>`, `<title>`, and `<locator>`. The `<author>` and `<title>` elements are optional. Arboreal does not make use of these elements, but using them makes the catalog file more readable. If they are present, the catalog file may be opened as a master text and (assuming that it has also been loaded as a catalog) used as a table of contents from which documents can be opened directly. The `<locator>` element is required; its `xlink:href` attribute is a URI that specifies the actual location of the object. If this URI is a *relative* URI, Arboreal understands it as relative to the base from which the catalog file was loaded.

The following is an example of a complete catalog entry for an Archimedes primary text:

```
<entry type="text" xlink:href="001.xml">
  <author>Agricola, Georgius</author>
  <title>De re metallica</title>
  <location xlink:href="http://archimedes.fas.
    harvard.edu/texts/agric_remet_01_la_1556.xml"/>
</entry>
```

Appendix: The Catalog DTD

```
<!ELEMENT catalog (entry*)>

<!ATTLIST entry
  xmlns:xlink      CDATA
  #FIXED "http://www.w3.org/1999/xlink">

<!ELEMENT entry (author?, title?, location)>

<!ELEMENT author (#PCDATA)>
<!ELEMENT title (#PCDATA)>

<!ELEMENT location EMPTY>

<!ATTLIST entry
  xlink:type      (simple)          #FIXED "simple"
  xlink:href      CDATA             #REQUIRED
  type            CDATA             #IMPLIED>

<!ATTLIST location
  xlink:type      (simple)          #FIXED "simple"
  xlink:href      CDATA             #REQUIRED>
```