Digital Curation Tools: Metadata Enhancement with Selenium IDE

Daniel Gelaw Alemneh
University of North Texas Libraries
Daniel.Alemneh@unt.edu

Andrew James Weidner
University of North Texas Libraries
Andrew.Weidner@unt.edu

Abstract

Maintaining usable and sustainable digital collections requires a complex set of actions that address the myriad challenges at various stages of the data lifecycle. Digital curation activities enhance access and retrieval, maintain quality, add value, and facilitate use and re-use over time. Digital resource lifecycle management is becoming an increasingly important topic as digital curators actively explore tools and applications that directly perform curation and management tasks. Accordingly, the University of North Texas (UNT) Libraries develop and/or adopt various tools, workflows, and quality control mechanisms that enable quick and effective analysis and quality assurance. This brief paper demonstrates automated metadata enhancement with Selenium IDE, an open source, Web-based tool which UNT has adopted for use during the post-ingestion stage of the data lifecycle.

Keywords: digital curation, lifecycle management, metadata, open source, curation tools

Introduction

Digital lifecycle management starts when an item is created (born-digital) or selected for digitization (analog) and continues through image post-processing, metadata capture, derivative creation, and preservation for long-term access. Quality metadata is crucial to implementing reliable, usable, and sustainable digital libraries (Sumner & Custard, 2005). Recognizing the role of standardized metadata in digital resource lifecycle management, the University of North Texas (UNT) Libraries actively promote metadata-based digital resource management.

The UNT Digital Libraries Division utilizes various tools to ensure metadata consistency and precision across all digital resources and facilitate digital curation activities. This paper describes a workflow that uses Selenium IDE to edit large sets of published metadata records quickly and accurately with minimal human intervention.

What is Selenium IDE?

Selenium IDE is a free and open source add-on for the Firefox Web browser. It is primarily used by the Web development community to perform automated testing of Web applications. Selenium IDE provides an integrated development environment in which to create, debug and run custom scripts that automate actions in a Web browser. Users write or record scripts in the Selenium IDE window (Figure 1) and use standard play controls to run single scripts, called test cases, or groups of scripts, called test suites. The Selenese syntax, encoded as an HTML table, sends commands to the browser that act on specified page elements in sequence. See Table 1 for a list of common Selenium IDE commands.

Alemneh, D. G., & Weidner, A. J. (2013). Digital curation tools: Metadata enhancement with Selenium IDE. *iConference 2013 Proceedings* (pp. 831-835). doi:10.9776/13411

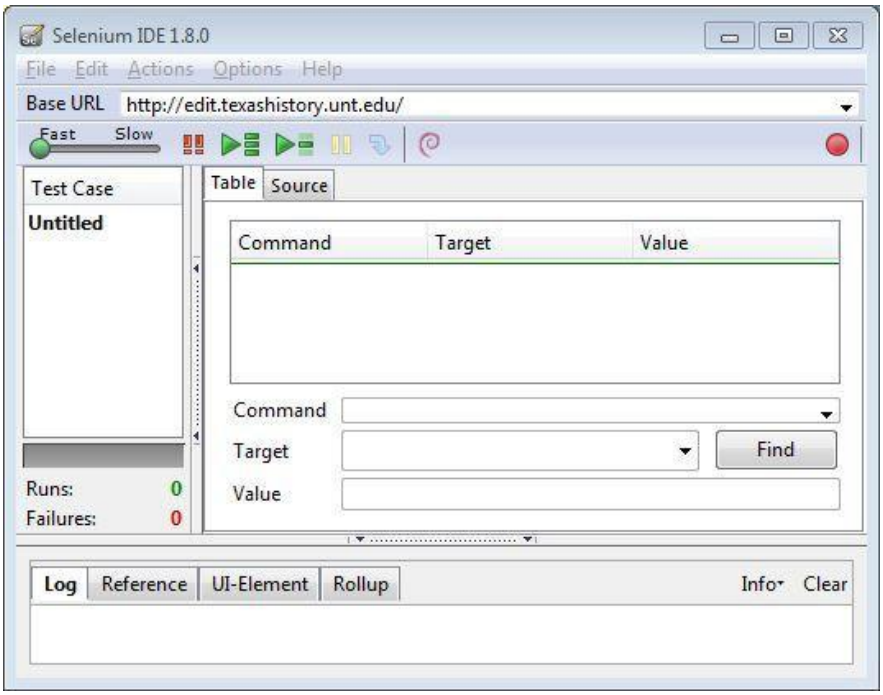

Figure 1: The Selenium IDE Window.

Selenium IDE Workflow for Post-Ingest Metadata Normalization

Selenium IDE is an important part of the digital curation toolkit for the UNT Digital Libraries Division. It has proven to be essential in the ongoing processes of improving and maintaining metadata quality for large collections of digital objects. Naturally, as careful repository stewards, we attempt to publish accurate and complete metadata when we upload items to our repository. We have a number of tools at our disposal that facilitate normalized metadata creation and eliminate mistakes before items are uploaded.

Sometimes, for a variety of reasons that are usually out of our control, we end up with incorrect or sub-standard metadata for published digital objects. After objects have been published, our content management system provides a single method for editing metadata: the object record. A human operator must open an object record in a Web browser and manually change the information via the editing interface. When large sets of records contain metadata that must be normalized in order to improve retrieval or meet our repository's data standards, the single object method is undesirable.

The single object paradigm means that editing large sets of records requires shifting staff time away from more important production activities. However, if the metadata that must be changed is standard across the entire set, we use Selenium IDE to automate the editing process. In the best case scenario, a Selenium IDE operator creates a test suite that automates the editing process for multiple object records. If a test suite is not feasible, an operator implements a test case that streamlines the editing process for individual object records.

A typical metadata editing workflow begins with identifying a set of objects that require normalization. If the set is large and the required changes are standard, an operator creates a Selenium IDE script that performs the changes, publishes the new metadata, and closes the browser tab. After testing and debugging to ensure that the script performs correctly, the operator creates a test suite. Using the content management system's search interface, the operator opens multiple object records as Web browser tabs. Finally, the operator runs the test suite. Each script in the test suite works on a tab in the browser window until there are either no more scripts in the suite or no more tabs in the browser. The operator repeats the process until the entire record set is normalized.

Table 1
Common Selenium Commands

Command	Function	Usage
assertValue	Tests an element's value.	Verify that a page element has a specific value. Accepts regular expressions.
click	Performs a mouse click.	Click links, buttons, and check boxes.
clickAndWait	Performs a mouse click and waits for the page to load.	Execute the next command in the script after the new page loads.
close	Closes the current tab or browser window.	Place at the end of a script to enable automated batch editing with a test suite.
keyPress	Sends a key press.	Delete or add characters in a text input element. Use with setCursorPosition.
select	Selects a value.	Select a choice from a drop-down menu.
setCursorPosition	Places cursor at a specific location in a text input field.	0 = beginning of text string. -1 = end of text string.
store	Assigns a value to a variable.	Use the variable with the type command to paste the assigned value in a text input field.
storeValue	Stores existing text in a variable.	Use the variable with the type command to paste the stored value in a text input field.
type	Enters specified text.	Populate a text input field. Overwrite existing text.

Use Case: The Portal to Texas History

The UNT Libraries (UNTL) metadata guidelines specify that the main title of an object is the title printed on the title or front page. We use a template tool to create standard metadata for large groups of similar objects, such as newspaper issues, during ingest. In some cases it is difficult to notice minor changes in a newspaper title during the quality control process. After the objects have been uploaded to The Portal to Texas History, however, we can easily view changes in a newspaper title by browsing through the thumbnail images created during the ingest process.

If we identify any title changes that are not reflected in the template-produced metadata, we use Selenium IDE to quickly bring our metadata in line with the UNTL standard. What follows is a step-by-step breakdown, with HTML code and a screenshot (Figure 2), of how a newspaper title script works:

1. Test that the original title is what we expect it to be with the "assertValue" command and a regular expression. Adding tests to our scripts ensures that we avoid making inadvertent changes to the metadata; the script will stop if the value does not match. If the title were the only information in the Main Title field, we would simply enter the title as the value. Because there is unique information present, such as volume and issue numbers, we use a regular expression to check the title from the beginning of the string.

```
<tr>
    <td>assertValue</td>
    <td>//div[@id='main']/div/div[2]/div/input</td>
    <td>regexp:^UNT Daily</td>
</tr>
```

2. Create a variable called "NewText" which contains the word "The" with the "store" command.

```
<tr>
    <td>store</td>
    <td>The</td>
    <td>NewText</td>
</tr>
```

3. Create a variable called “OriginalTitle” which contains the current text in the Main Title field with the “storeValue” command.

```
<tr>
    <td>storeValue</td>
    <td>//div[@id='main']/div/div[2]/div/input</td>
    <td>OriginalTitle</td>
</tr>
```

4. Paste the stored text from both variables, with a single space between them, into the Main Title field with the “type” command.

```
<tr>
    <td>type</td>
    <td>//div[@id='main']/div/div[2]/div/input</td>
    <td>${NewText} ${OriginalTitle}</td>
</tr>
```

5. Save the new metadata by clicking the Publish button with the “clickAndWait” command.

```
<tr>
    <td>clickAndWait</td>
    <td>name=publish</td>
    <td></td>
</tr>
```

6. Close the tab with the “close” command to allow the next script in the automation suite to work on the next tab in the browser window.

```
<tr>
    <td>close</td>
    <td></td>
    <td></td>
</tr>
```

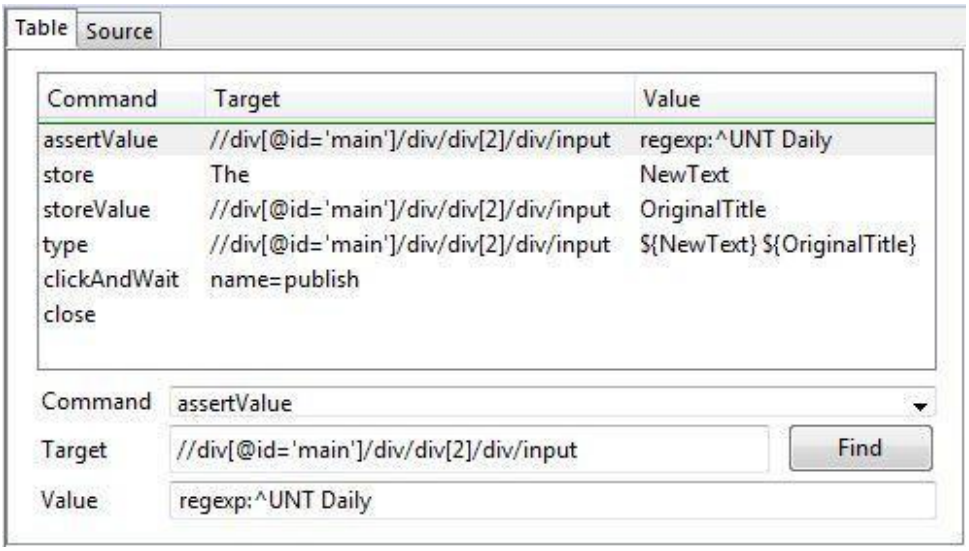

Figure 2: Newspaper Title Script in Selenium IDE’s Table View.

Combining multiple instances of the above script in a test suite automatically adds the word “The” to the beginning of the Main Title field for any number of digital object records loaded in browser tabs. In this manner we can quickly edit large sets of records and avoid the inevitable typographical errors introduced during manual data entry.

Conclusion

Large digital collections present challenges when producing descriptive metadata. Naming schemes and element definitions can vary widely, requiring substantial rework to meet local repository standards. Successful metadata enhancement strategies involve mechanisms for both pre- and post-ingest metadata normalization. Automated metadata normalization with Selenium IDE improves operator efficiency and accuracy during the time- and labor-intensive post-ingest data entry process (Figure 3).

From institutional repository platforms (e.g., DSpace) to commercial content hosting sites (e.g., Flickr), Selenium IDE can be used to edit metadata in any content management system that has a Web-based editing interface. Selenium IDE is a highly recommended addition to the metadata enhancement toolkit for any institution that serves content in a content management system with a Web-based administrative interface.

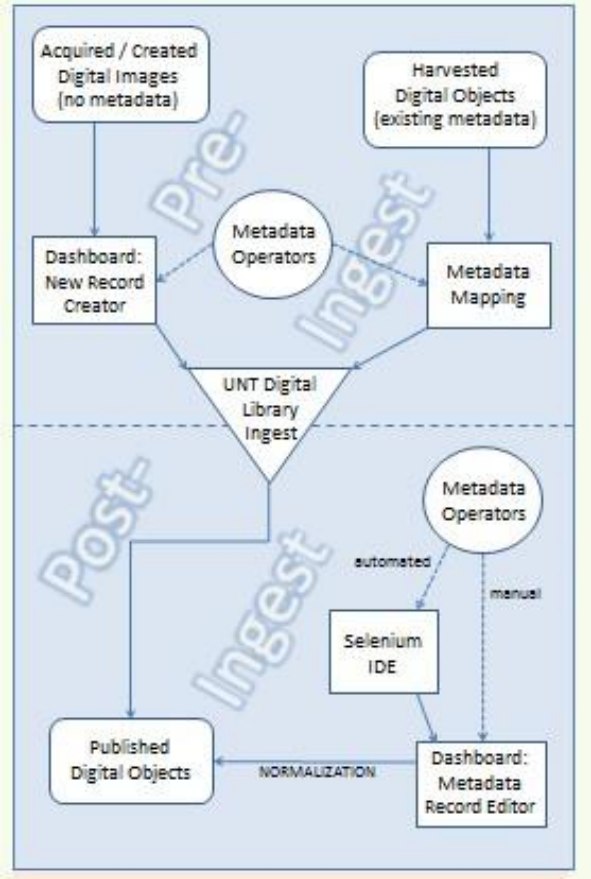

Figure 3: UNT Digital Library Workflow Modified for Post-Ingest Metadata Normalization. Adapted from "Metadata Quality Enhancement for Large Digital Collections: Web Browser Automation with Selenium IDE" by A. J. Weidner and D. G. Alemneh, 2012, *UNT Digital Library*.

References

Sumner, T & Custard, M. (2005). Using Machine Learning to Support Quality Judgments. *D-Lib Magazine 11*(10). Retrieved from http://www.dlib.org/dlib/october05/custard/10custard.html

The Portal to Texas History (2012). Retrieved from http://texashistory.unt.edu/

UNT Libraries' Input Guidelines for Descriptive Metadata (2012). Retrieved from http://www.library.unt.edu/digital-projects-unit/input-guidelines-descriptive-metadata

Weidner, A. J. & Alemneh, D. G. (2012). Metadata Quality Enhancement for Large Digital Collections: Web Browser Automation with Selenium IDE. *UNT Digital Library.* Retrieved from http://digital.library.unt.edu/ark:/67531/metadc86138/