Processing and Formatting System for Digital Collections

Frances Webb
Mann Library, Cornell University
fbw4@cornell.edu

Abstract

This system is being used to build structure data for the HEARTH digital collection and to manage the collection under the DLXS system. It allows student workers or unskilled employees to build structure metadata from scanned images for both monographs and serials, and manages the process of delivering the titles under DLXS once prepared. It allows supervisors to manage the work, simplifying tasks like re-assigning the in-progress work of graduated students.

1. Purpose of software

The software system was designed to automate many of the technical steps involved in processing digital titles before they may be included in a digital collection, and to offer an interface that would allow unskilled employees to build structure data.

Several steps are automated, including the importation of MARC records from our electronic catalogue, and the combining of the MARC records, OCR data, and structure data into the TEILite format metadata used by the DLXS system.

The entire system is web-based, allowing our collection management tools to be accessed from any networked computer without the installation of any specialized software. It is designed with the goal of allowing digital collections to be built and maintained, as much as possible, by the staff of Collection Development rather than Information Technology Services.

2. Document structuring

The document structuring and quality review steps are combined in the same process. This is the largest and most important piece of the system. A minimum of a 19” monitor is recommended for this work because it is necessary to work with the structure information and the documents themselves simultaneously.

The interface consists of a web-based form to be used while a staff member reviews full-quality scan images on the same screen. The interface is designed to allow student employees with minimal training to handle any of the structuring or quality control issues that are likely to be encountered.

The web form allows images to be associated with page numbers and important page features (e.g. TOC, title page, index). It allows images to be easily resequenced or removed, and blank images to be inserted into an image sequence. The quality control feature allows a user to review all of the versions of an image that exist in the system and select the one that should be used, or mark the image as a scanning issue if there is no acceptable image available.

The serials portion of the form allows the definition of issue and article units. Articles may be assigned titles, authors, and page sequences. Broken page sequences (e.g. “continued on page 44”) are handled seamlessly. By necessity, the issue units are more complex, but they are designed to not require the skills of a serials cataloguer.

Issues are assigned image ranges, dates, and enumerations (e.g. volume and issue numbers). A fairly sophisticated date processor forces the date of issue to be very clearly identified - for the purpose of issue sorting - while date display formats are determined more flexibly. The assumption is made that dates and not issue enumerations are the most fool-proof way for any system to automatically sort serials.

The review and structuring work for a title may be saved at any time, and restored from any saved version.

3. User account system

A simple home-grown user account system serves the dual purpose of restricting access to the staff-only interface to users with permission to be there, and of allowing a simple means of keeping track of work progress. Users either have administrator or user privileges. The primary difference is that administrators may maintain system accounts, assign users to review specific titles or un-assign them - which comes in handy when student employees leave.

4. Notes

This system was written under the IMLS grant for the HEARTH digital collection, with Joy Paulson as project lead. The HEARTH collection is available at: http://hearth.library.cornell.edu. The system is coded in C and Perl. It is currently running on Red Hat Linux 7.1, and relies on the Apache web server.