

Archivists' Toolkit
Preliminary Beta Testing Report
November 13, 2006

Beta Testing Community

In July and August, the Project Team identified testing repositories, some from within the project's original partnering repositories. We also found additional testers from those who'd expressed strong interest in the project following presentations at SAA, DLF, etc. The testing community includes a number of academic institutions, though museum and governmental repositories are also represented. In addition, a number of the academic institutions included specialized repositories, as well as more generalized archives and special collections repositories.

Nineteen institutions signed on to participate in the beta testing, though three of these were unable to continue due to issues with their firewall, staffing changes, or other reasons. The following institutions participated in beta testing to varying degrees:

- American Museum of Natural History
- Arizona State University
- Bates College
- Brooklyn Museum
- Center for Jewish History
- Dartmouth College
- Das Bundesarchiv
- Georgia Tech
- Getty Research Institute
- Harvard University
- Mount Holyoke College
- Nationaal Archief
- New York University
- Northern Arizona University
- Princeton University
- University of California San Diego
- University of Massachusetts Amherst
- University of Michigan (Bentley)
- University of Texas

In some cases, multiple repositories participated from a single institution. The number of testers ranged from one to seven at each institution, so that there were nearly 60 individual testers involved. The project team used a listserv to communicate with the testers, and testers were encouraged to post general questions and comments to the listserv as well.

Testing Schedule

The Project Team developed a testing schedule that identified specific functional areas to be tested each week. We approached testing in this manner so that we could better manage the time commitment of our testers, ensure that all functionality of the application would be tested, and begin testing of some functions while others were still in development.

The testing schedule was designed to allow the testers to develop familiarity with the Toolkit's interface and operating environment by having the testers do simpler tasks first before working with the Toolkit's more complex and advanced features. The first week included the tasks necessary for a repository to use the Toolkit. Testing activities then

progressed from some of the more straightforward tasks, like name and subject authority work and accessioning, to more complex tasks such as description and import of legacy data. The original testing schedule is detailed in the following table:

Assignment	Time	Dates
Install the Toolkit client, establish repository record, create necessary user accounts, and create location records for the repository	Week 1	August 8-11
Create 25 subject records/edit 2/delete 1	Week 2	August 14-18
Create 25 name records/edit 2/delete 1	Week 2	August 14-18
Create 15 accession records with names, subjects, and locations/edit 2/delete 1	Week 3	August 21-25
Create 5 deaccession records/edit 2/delete 1	Week 3	August 21-25
Create 10-15 descriptions with corresponding names, subjects, and locations/edit 1/delete 1 resource or series-level component	Weeks 4-5	August 28-September 8
Create 5 digital object records (for repositories creating digital objects)	Week 6	September 11-15
Ingest at least 25 accession records (for repositories with accession spread-sheets or databases)	Weeks 7-8	September 18-29
Ingest 5-10 EADs (for repositories with EADs)	Weeks 7-8	September 18-29
Ingest 5-10 MARC XML records (for repositories with MARC XML w/no EAD or who want to harvest controlled access terms from MARC records)	Weeks 7-8	September 18-29
Search the database for administrative use and generate reports	Weeks 9-10	October 2-13
Customize system	Weeks 9-10	October 2-13
Search the database for reference use	Week 11	October 16-20

During the testing process, the schedule was modified both to provide assignments more in line with our four hour per week estimate for testing and to allow additional development time where needed. Testing was completed November 3, and did not include testing of the Toolkit’s search capabilities beyond the searching needed for generating administrative reports. Rather, additional time was provided for the participants to test importing legacy data and testing output of administrative reports and export of standardized data.

Testing Process

We chose a structured testing process in order to ensure that certain features and functions of the application were well tested by everybody. At the beginning of each week of testing, the participants received a worksheet with assigned tasks related to a specific function, as well as notification of the user documentation available for the assignment. The worksheets also had space for reporting feedback or problems. (An

example of one of these worksheets is available on the Toolkit's web site: <http://archiviststoolkit.org/betaTest/BETAtest-assignments-08212006.xls>). Testers returned the worksheet with their feedback at the end of each week.

Testers were also encouraged to report bugs directly as they occurred, and the Lead Programmer created a bug reporting form in the application for this purpose. The Team uses JIRA bug reporting software to track bugs and feature requests, and the bug reporting form in the application was programmed to be directly posted to JIRA. Where testers reported bugs in the worksheet or on the testers' listserv rather than using the bug reporting form, project team members logged the bugs directly into JIRA.

Results

Ultimately, we averaged fourteen responses each week of testing with a range of five to twenty-three responses. The Project Team followed up on the responses where necessary to clarify issues and distilled all feedback into weekly summaries. These compilations are organized into four sections: bugs, notable issues, feature requests, and comments on the documentation. The Lead Programmer then reviewed the report for issues to be logged into JIRA and generated a separate list of these issues. One of the Archives Analysts reviewed the summary and generated a separate list of documentation issues to be incorporated into the final user manual. Each week, these reports were distributed to the beta testers for additional feedback.

Compiled responses from the beta testers are available at: <http://www.archiviststoolkit.org/betaTest/index.html>

Overall Feedback

The Project Team distributed an exit survey designed to assess how well the application meets the goals of the original proposal, to elicit feedback on successful features and areas that need work, and to help us evaluate the interface, user documentation, and the beta testing process. *As of November 13, 2006, these results are preliminary.*

Overall impressions of the Toolkit were positive, with many testers stating that it is a much needed application in the archival community. Testers were very positive about description, authority functions, and the ability to customize the application. For accession records, a number of the testers would like to see more data or the ability to customize fields. Asked if they would recommend the Toolkit to a colleague, none of the testers responded negatively. Some of the testers qualified their answer, feeling that the application would be particularly useful for small to medium sized repositories and/or repositories without legacy systems in place.

Testers agreed that the Toolkit performs the following functions very well:

- Name and subject authorities
- Resource description
- Standardization of data
- Creation, manipulation, and display of hierarchical description
- Sorting and filtering of records

They identified the following areas that need work:

- Location management (ease of use, flexibility, and/or granularity)
- Performance/speed
- Import of legacy data (particularly non-standard legacy data)
- Accession records (additional data or the ability to create custom fields requested)
- Ability to expedite creation of container lists with a simplified interface
- Support for non-English languages

Testers had mixed responses to EAD/MARC import and export, customization, and the application interface. All felt that EAD and MARC import and export was one of the key functions of the Toolkit, and some were impressed with its capabilities. At the time of testing there were, however, a number of bugs related to data mapping, and many testers agreed that these must be resolved in order for the application to be ready for release.

Regarding customization, many of the testers were impressed with the customization abilities the Toolkit offers. Some appreciated the functionality but suggested that the customization process should be simplified. A number of testers also expressed interest in the ability to customize outputs.

The Toolkit's interface also produced a mixed response. A number of testers felt that the interface was intuitive and easy to use, stating that they were often able to use the Toolkit without assistance from the user manual. Others had vague criticisms about the look and feel of the application, but some concrete improvements were also suggested. For example, testers commented that the number of screens involved in certain processes can reduce efficiency (the container list creation previously mentioned is an example). Navigating among the various screens can also require numerous mouse clicks, and some felt that these windows were not always intuitive. Additional recommendations were that the application could provide better feedback in response to user actions, and that the layout of fields in the records could be improved in order to support scanning of the content.

The testers also mentioned a number of functions they'd like to see added to the application, including preservation, improved administrative tracking, collection development, patron registration and use tracking, and better support for item-level description of formats such as photos, audio, and video.

Data from the exit survey will assist the Project Team in prioritizing remaining development work for version 1.0, and also indicates features to be explored for inclusion in future releases.