Summary: This poster describes three simple shell scripts that facilitate the acquisition of born-digital archival collections from donors’ computers and streamline the creation of descriptive data for EAD finding aids. Mudd archivists created the first two shell scripts, which facilitate acquisition, using basic Bash and Batch commands that form part of the default shells in Unix (Mac) and DOS (Windows) computing environments, respectively. To streamline the creation of descriptive data, Mudd archivists wrote a third shell script using the default Linux (Ubuntu) shell that is packaged with the open-source BitCurator environment. Taken as a whole, the scripts have simplified the Library’s acquisition and access procedures for born-digital archives, empowering them to gain greater intellectual control of materials and provide enhanced access to researchers through richer description.

Acquisition Challenge: Efficiently acquiring born-digital records directly from donor’s computers. Initial attempts at this included using Bagger, which was ultimately abandoned due to Java dependencies and the amount of time required to complete large acquisitions.

Solution: To offset the issue of Java dependencies and to conduct efficient acquisitions, archivists wrote two shell scripts using Bash and Batch commands that form part of the default shells in Unix (Mac) and DOS (Windows) environments. Excluding temporary and other system files, the scripts receive variables from the user and then invoke the rsync and robocopy commands to move large volumes of data and generate log files for the acquisitions.

Digital Accessioning Goals
- Adhere to core archival principles of provenance and original order.
- Adhere to local principles of MPLP and “do no harm”
- Incorporate description practices that allow users to understand the context of born-digital records and facilitate access to born-digital materials.

Future developments
- Eliminate string values for <extent> elements and minimize post-processing of data.
- Index file and folder names to represent file structure within the finding aid or the repository.
- Create or incorporate scripts for modeling different classes of data (textual, visual, etc.).

Research Questions:
- How do we acquire born-digital records from different computer systems while minimizing impact on donors?
- For multi-level records, how does one create description elements programmatically?
- What are the key metadata points that we should extract from born-digital records and later represent in EAD?

Born-Digital Records Workflow

Acquisition
(Capture from Donor’s system)

Physical Control
(Virus Scan, Disk image)

Intellectual Control
(File ID, PII Scan, Description)

Patron Access
(Finding Aid)

Intellectual Control Challenge: Generating <unittitle>, <unitdate>, and <extent> values for born-digital collections with deep nesting of folders. Initial attempts at this included using the .csv output from FTK Imager and a simple Windows file browser.

Solution: Using the default Linux (Ubuntu) shell that is packaged with the open-source BitCurator environment, we wrote a Bash script that extracts these data points, which are already embedded in the filesystem and could be easily extracted without human error. The bash script consists of a simple for loop that uses a combination of the echo, find, sort, head, awk, and wc commands, the output of which we write to a text file and transform into an EAD component listing.

Description Bash Script

Conclusion: The shell is a simple yet powerful means to manipulate, interpret, and repurpose the data found in and around born-digital archival records. Archivists who acquire a basic control of this tool will find that it not only can streamline workflows but also provide users with greatly enhanced and contextualized representations of records.