Advanced Electronic Records Institute

Digital Forensics for Electronic Records Management

Cal Lee, University of North Carolina

This project made possible by a grant from the U.S. Institute of Museum and Library Services
How should we approach processing of born-digital records on media like this?

Goals in Appraisal, Ingest, and Processing of Born-Digital Materials

• Select the subset of materials that should be subject to ongoing preservation
• Ensure integrity of materials
• Allow users to make sense of materials and understand their context
• Prevent inadvertent disclosure of sensitive data
Relevant Principles

Provenance
• Reflect “life history” of records
• Records from a common origin or source should be managed together as an aggregate unit

Original Order
Organize and manage records in ways that reflect their arrangement within the creation/use environment

Chain of Custody
• “Succession of offices or persons who have held materials from the moment they were created”\(^1\)
• Ideal recordkeeping system would provide “an unblemished line of responsible custody”\(^2\)

More, Product, Less Process (MPLP)* – A Recent Theme in Archival Discussions

• Goal: Complete processing of archival backlogs and expedite processing of new acquisitions
• Questions the utility of giving equal, thorough treatment to all materials (some should be processed much more “lightly” than others)
• Primary focus is processing of analog materials, context in which almost all tasks require detailed, human attention

But digital is different...
“No computation without representation”
Representation Information

• “Information that maps a Data Object into more meaningful concepts” (OAIS) - makes humanly-perceptible properties happen

• Examples: file format, encoding scheme, data type

Reference Model for an Open Archival Information System (OAIS). Consultative Committee for Space Data Systems, 2002.: Figure 2-2
## Digital Resources - Levels of Representation

<table>
<thead>
<tr>
<th>Level</th>
<th>Label</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Aggregation of objects</td>
<td>Set of objects that form an aggregation that is meaningful encountered as an entity</td>
</tr>
<tr>
<td>7</td>
<td>Object or package</td>
<td>Object composed of multiple files, each of which could also be encountered as individual files</td>
</tr>
<tr>
<td>6</td>
<td>In-application rendering</td>
<td>As rendered and encountered within a specific application</td>
</tr>
<tr>
<td>5</td>
<td>File through filesystem</td>
<td>Files encountered as discrete set of items with associate paths and file names</td>
</tr>
<tr>
<td>4</td>
<td>File as “raw” bitstream</td>
<td>Bitstream encountered as a continuous series of binary values</td>
</tr>
<tr>
<td>3</td>
<td>Sub-file data structure</td>
<td>Discrete “chunk” of data that is part of a larger file</td>
</tr>
<tr>
<td>2</td>
<td>Bitstream through I/O equipment</td>
<td>Series of 1s and 0s as accessed from the storage media using input/output hardware and software (e.g. controllers, drivers, ports, connectors)</td>
</tr>
<tr>
<td>1</td>
<td>Raw signal stream through I/O equipment</td>
<td>Stream of magnetic flux transitions or other analog electronic output read from the drive without yet interpreting the signal stream as a set of discrete values (i.e. not treated as a digital bitstream that can be directly read by the host computer)</td>
</tr>
<tr>
<td>0</td>
<td>Bitstream on physical medium</td>
<td>Physical properties of the storage medium that are interpreted as bitstreams at Level 1</td>
</tr>
</tbody>
</table>
Interaction Examples

Level

Aggregation of objects

Browse Gazette issues

- 2014


+ No. 123 Special: 2014 General Election Amended Official Result for the Te Tai Tokerau Electoral District.

+ No. 122 Principal Edition, 9 October 2014


+ No. 120 Special: 2014 General Election Results of the Official Count, 4 October 2014.

+ No. 119 Principal Edition, 2 October 2014

+ No. 118 Supplement: Financial Markets Authority: Authorised Futures Dealers Notices, 30 September 2014
Interaction Examples

Level

Aggregation of objects

Object or package

In-application rendering

File through filesystem

File as “raw” bitstream

Sub-file data structure

Bitstream through I/O equipment

Raw signal stream through I/O equipment

Bitstream on physical medium

Browse Gazette issues
- 2014
+ No. 124 Special: 2014 General Election - Election of List Candidates. PDF (99kb)
+ No. 123 Special: 2014 General Election Amended Official Result for the Te Tai Tokerau Electoral District. PDF (99kb)
+ No. 122 Principal Edition, 9 October 2014 PDF (291kb)
+ No. 121 Customs Edition, 7 October 2014 PDF (55kb)
- No. 120 Special: 2014 General Election Results of the Official Count, 4 October 2014. PDF (296kb)

<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Type</th>
<th>Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 OCT 2014</td>
<td>2014 General Election</td>
<td>Authorities/Other Agencies of State</td>
<td>Electoral Act</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
+ No. 119 Principal Edition, 2 October 2014 PDF (414kb)
## Interaction Examples

### Level

**Aggregation of objects**

**Object or package**

### In-application rendering

**File through filesystem**

**File as “raw” bitstream**

**Sub-file data structure**

**Bitstream through I/O equipment**

**Raw signal stream through I/O equipment**

**Bitstream on physical medium**
Interaction Examples

Level

Aggregation of objects

Object or package

In-application rendering

File through filesystem

File as “raw” bitstream

Sub-file data structure

Bitstream through I/O equipment

Raw signal stream through I/O equipment

Bitstream on physical medium
File System

- Access controls
- File names & identifiers
- File size (length)
- Where to find files in storage (sectors and clusters)
- MAC times
  - Modified – when the content was last changed
  - Accessed – time file was last accessed (by person or software)
  - Changed – last time metadata changed
  - Created – (implemented inconsistently, if at all, across different file systems)
This is HFS+
# File System Examples

<table>
<thead>
<tr>
<th>Name</th>
<th>Operating System(s) Using it as Native File System [often other OSs can also recognize it]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ext, ext2, ext3 (Extended File System)</td>
<td>Linux</td>
</tr>
<tr>
<td>FAT16</td>
<td>MS-DOS</td>
</tr>
<tr>
<td>FAT32 (VFAT)</td>
<td>Windows 95, 98</td>
</tr>
<tr>
<td>HFS (Hierarchical File System)</td>
<td>Macintosh System 4-8</td>
</tr>
<tr>
<td>HFS+</td>
<td>Macintosh System 8.1-X</td>
</tr>
<tr>
<td>HPFS (High Performance File System)</td>
<td>OS/2</td>
</tr>
<tr>
<td>ISOFS (ISO 9660)</td>
<td>Any OS that reads data from a CD</td>
</tr>
<tr>
<td>JFS1 (Journaled File System)</td>
<td>AIX (IBM)</td>
</tr>
<tr>
<td>MFS (Macintosh File System)</td>
<td>Macintosh System 1-3</td>
</tr>
<tr>
<td>ReiserFS</td>
<td>Several Linux distributions</td>
</tr>
<tr>
<td>UFS (Unix File System) aka FFS (Fast File System)</td>
<td>Various flavors of Unix</td>
</tr>
</tbody>
</table>
### File System Examples

<table>
<thead>
<tr>
<th>Name</th>
<th>Operating System(s) Using it as Native File System [often other OSs can also recognize it]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ext, ext2, ext3 (Extended File System)</td>
<td></td>
</tr>
<tr>
<td>FAT16</td>
<td></td>
</tr>
<tr>
<td>FAT32 (VFAT)</td>
<td></td>
</tr>
<tr>
<td>HFS (Hierarchical File System)</td>
<td></td>
</tr>
<tr>
<td>HFS+</td>
<td></td>
</tr>
<tr>
<td>HPFS (High Performance File System)</td>
<td></td>
</tr>
<tr>
<td>ISOES (ISO 9660)</td>
<td></td>
</tr>
<tr>
<td>JFS1 (Journaled File System)</td>
<td></td>
</tr>
<tr>
<td>MFS (Macintosh File System)</td>
<td></td>
</tr>
<tr>
<td>NTFS</td>
<td></td>
</tr>
<tr>
<td>ReiserFS</td>
<td></td>
</tr>
<tr>
<td>UFS (Unix File System) aka FFS (Fast File System)</td>
<td></td>
</tr>
</tbody>
</table>

The filesystems you’re most likely to encounter within archival collections.
Interaction Examples

<table>
<thead>
<tr>
<th>Level</th>
<th>Aggregation of objects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Object or package</td>
</tr>
<tr>
<td></td>
<td>In-application rendering</td>
</tr>
<tr>
<td></td>
<td>File through filesystem</td>
</tr>
<tr>
<td>File as “raw” bitstream</td>
<td>Sub-file data structure</td>
</tr>
<tr>
<td></td>
<td>Bitstream through I/O equipment</td>
</tr>
<tr>
<td></td>
<td>Raw signal stream through I/O equipment</td>
</tr>
<tr>
<td></td>
<td>Bitstream on physical medium</td>
</tr>
</tbody>
</table>
Hex Dump

• A more compact and more humanly readable way of conveying a stream of bits

• Uses hexadecimal notation
  – Each character represents one of 16 possible values (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)
  – Conveniently, a series of two characters represented in hexadecimal can represent exactly one byte ($2^8 = 256$ possible values) of data, because $16^2 = 256$

• Hex dumps from computer’s memory often used for debugging or reverse engineering software and for data recovery
In the BitCurator environment:
Interaction Examples

Level

Aggregation of objects

Object or package

In-application rendering

File through filesystem

File as “raw” bitstream

Sub-file data structure

Bitstream through I/O equipment

Raw signal stream through equipment

Bitstream on physical medium
Interaction Examples

Level

Aggregation of objects

Object or package

In-application rendering

File through filesystem

File as “raw” bitstream

Sub-file data structure

Bitstream through I/O equipment

Raw signal stream through I/O equipment

Bitstream on physical medium
Level

Aggregation of objects

Object or package

In-application rendering

File through filesystem

File as "raw" bitstream

Sub-file data structure

Bitstream through I/O equipment

Raw signal stream through I/O equipment

Bitstream on physical medium
## Interaction Examples

**Level**

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation of objects</td>
<td></td>
</tr>
<tr>
<td>Object or package</td>
<td></td>
</tr>
<tr>
<td>In-application rendering</td>
<td></td>
</tr>
<tr>
<td>File through filesystem</td>
<td></td>
</tr>
<tr>
<td>File as “raw” bitstream</td>
<td></td>
</tr>
<tr>
<td>Sub-file data structure</td>
<td></td>
</tr>
<tr>
<td>Bitstream through I/O equipment</td>
<td></td>
</tr>
<tr>
<td>Raw signal stream through I/O equipment</td>
<td></td>
</tr>
<tr>
<td><strong>Bitstream on physical medium</strong></td>
<td></td>
</tr>
</tbody>
</table>

---

Current Situation with Born-Digital Materials

• Many institutions with substantial “backlogs,” including at-risk media within their holdings
• More born-digital materials will increasingly be coming in
• Initial processing efforts can involve substantial human time and attention
Immediate Priorities

• Getting data off media so data can be stored safely on their original state (create disk images)
• Basic characterization (from files and filesystems) of what’s on the media
• Flagging of information that’s most likely to be sensitive and need of removal, closure, redaction or filtering
• Triage activities:
  – Select files to export for further processing
  – Identify duplicates
More Product, More (Machine) Process

• Collecting institutions need to apply many more processes to born-digital records (e.g. integrity checks, metadata extraction, audit trails, characterization)

• The good news is that most of these processes can be performed by software

• Human attention is always quite limited and should be focused on higher-level tasks to the extent possible
Some potential tools for your toolbox:

AFFLIB
Open Source Computer Forensics Software
UNC School of Information and Library Science
Digital Forensics Can Help Archivists to Fulfill their Principles

Provenance
- Identify, extract and save essential information about context of creation

Original Order
- Reflect original folder structures, files associations, related applications and user accounts

Chain of Custody
- Documentation of how records were acquired and any transformations to them
- Use well-established hardware and software mechanisms to ensure that data haven’t been changed inadvertently

Identifying Sensitive Information
- Identify personally identifying information, regardless of where it appears
- Flag for removal, redaction, closure or restriction
Building Workflows: Two Resources with Examples


Funded by Andrew W. Mellon Foundation
– Phase 1: October 1, 2011 – September 30, 2013
– Phase 2 – October 1, 2013 – September 30, 2014
Partners: SILS at UNC and Maryland Institute for Technology in the Humanities (MITH)
BitCurator Goals

• Develop a system for collecting professionals that incorporates the functionality of open-source digital forensics tools

• Address two fundamental needs not usually addressed by the digital forensics industry:
  – incorporation into the workflow of archives/library ingest and collection management environments
  – provision of public access to the data
Core BitCurator Team

- Cal Lee, PI
- Matt Kirschenbaum, Co-PI
- Kam Woods, Technical Lead
- Porter Olsen, Community Lead
- Alex Chassanoff, Project Manager
- Sunitha Misra, Software Developer (UNC)
- Kyle Bickoff, GA (MITH)
- Amanda Visconti, GA (MITH)
# Two Groups of Advisors

<table>
<thead>
<tr>
<th>Professional Experts Panel</th>
<th>Development Advisory Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bradley Daigle, University of Virginia Library</td>
<td>Barbara Guttman, National Institute of Standards and Technology</td>
</tr>
<tr>
<td>Erika Farr, Emory University</td>
<td>Jerome McDonough, University of Illinois</td>
</tr>
<tr>
<td>Jennie Levine Knies, University of Maryland</td>
<td>Mark Matienzo, Digital Public Library of America</td>
</tr>
<tr>
<td>Jeremy Leighton John, British Library</td>
<td>Courtney Mumma, Artefactual Systems</td>
</tr>
<tr>
<td>Leslie Johnston, US National Archives and Records Administration</td>
<td>David Pearson, National Library of Australia</td>
</tr>
<tr>
<td>Naomi Nelson, Duke University</td>
<td>Doug Reside, New York Public Library</td>
</tr>
<tr>
<td>Erin O’Meara, Gates Archive</td>
<td>Seth Shaw, University Archives, Duke University</td>
</tr>
<tr>
<td>Michael Olson, Stanford University Libraries</td>
<td>William Underwood, Georgia Tech</td>
</tr>
<tr>
<td>Gabriela Redwine, Beinecke, Yale University</td>
<td></td>
</tr>
<tr>
<td>Susan Thomas, Bodleian Library, University of Oxford</td>
<td></td>
</tr>
</tbody>
</table>
BitCurator Environment*

- Bundles, integrates and extends functionality (primarily data capture and reporting) of open source software
- Can be run as:
  - Self-contained environment (based on Ubuntu Linux) running directly on a computer (download installation ISO)
  - Self-contained Linux environment in a virtual machine using e.g. Virtual Box or VMWare
  - As individual components run directly in your own Linux environment or (whenever possible) Windows environment

*To read about and download the environment, see: http://wiki.bitcurator.net/*
BitCurator-Supported Workflow Elements

- Acquisition
- Reporting
- Redaction
- Metadata Export

See: http://bitcurator.net
Checksums – Compact Representations of Bitstreams

• A given bitstream, fed into an algorithm, will generate a short string of characters that is extremely unlikely to be generated by a different bitstream fed into that same algorithm
• Most common = MD5, SHA-1
• Can determine:
  – If bits have changed after a transfer
  – If bits have flipped within a storage environment
  – Whether two different files are identical bitstreams
• A library of hash values can identify “known and notable” (EnCase terminology) files
  – Known – files that can be ignored (e.g. software listed in National Software Reference Library)
  – Notable – specific bitstreams that you’re trying to find
In BitCurator environment: Right Click on File or Directory and Calculate MD5
Calculate MD5 (Files and Directories)

Please choose the way you want the MD5 hash to be presented:
(1 file(s) selected)

Handling

○ Display on screen

○ Save to file (the selected filename + .md5 extension)

Cancel  OK

"bitcurator-grub.png" selected (43.3 kB)
The MD5 hash of the selected file:

```
keb2622125be1231b0fc9babee27942d /home/bcadmin/Pictures/bitcurator-grub.png
```

"bitcurator-grub.png" selected (43.3 kB)
Creating a Disk Image in Guymager

- Select the file format:
  - Linux dd raw image (file extension .dd or .xxx)
  - Expert Witness Format, sub-format Guymager (file extension .exx)
  - Advanced forensic image (file extension .aff)

- Case number: 1
- Evidence number: 1
- Examiner: BitCurator User
- Description: A sample floppy disk image
- Notes: Additional notes go here

- Destination:
  - Image directory: /home/bcadmin/Desktop/SampleData/
  - Image filename (without extension): sampleimage
  - Info filename (without extension): sampleimage

- Hash calculation / verification:
  - Calculate MD5
  - Calculate SHA-1
  - Calculate SHA-256
  - Re-read source after acquisition for verification (takes twice as long)
  - Verify image after acquisition (takes twice as long)

- Split image files: Split size 2047 MiB

- Cancel | Duplicate image... | Start
Building Blocks for a Business Case: Why Create Disk Images?

- **Simplify and compartmentalize** electronic records processing tasks – don’t need to solve all technical challenges at the same time
- Make sure full set of bits is safe – e.g. have the disk but not depend on fragile physical medium
- Surprises about how things were structured within the file system
- You could inadvertently change something in the act of examining or dealing with the files
- Proof of file integrity and chain of custody
- Corrupted files and viruses - to determine what subset of the bitstream can be recovered
- Recovery of traces of online activity
- Avoid irreversible transformations
- Changes in preservation strategy over time
- Supports widest range of potential access scenarios (see BitCurator Access project)
Bulk Extractor Scanning Options

See: http://www.forensicswiki.org/wiki/Bulk_extractor
Histogram of Email Addresses (Specific Instances in Context on Right)
BitCurator Reporting Tool
<fileobject>
    <filename>Documents and Settings/All Users/Documents/My Pictures/Sample Pictures/Blue hills.jpg</filename>
    ...
    <filesize>28521</filesize>
    <alloc>1</alloc>
    <used>1</used>
    <inode>6245</inode>
    ...
    <uid>0</uid>
    <gid>0</gid>
    <mtime>1208174400</mtime>
    <ctime>1257729636</ctime>
    <atime>1257729636</atime>
    <crttime>1257729636</crttime>
    <seq>2</seq>
    <libmagic>JPEG image data, JFIF standard 1.02</libmagic>
    <byte_runs>
        <run file_offset='0' fs_offset='0' img_offset='363200512'
             len='0'/>
    </byte_runs>
    <hashdigest type='MD5'>
        6fb2a38dc107eacb41cf1656e899cf70
    </hashdigest>
    <hashdigest type='SHA1'>
        4eee44b18576e84de7b163142b537d2fe6231845
    </hashdigest>
</fileobject>
Identifying (and Possibly Deleting) Duplicate Files
Customization and Scripts to Expedite Processing
Calculate MD5 of File or Directory
The MD5 hash of the selected file: 

```
keb2622125be1231b0fc9babe27942d /home/bcadmin/Pictures/bitcurator-grub.png
```

"bitcurator-grub.png" selected (43.3 kB)
View File in a Hex Editor
Show Metadata Embedded in Forensically-Packaged Disk Image
ewfinfo 20130416

Acquire information
  Acquisition date: Wed Jan 19 12:09:18 2011
  System date: Wed Jan 19 12:09:18 2011
  Operating system used: Linux
  Software version used: 20100226
  Password: N/A

EWF information
  File format: EnCase 6
  Sectors per chunk: 64
  Error granularity: 64
  Compression method: deflate
  Compression level: best compression
  Set identifier: 4eb6701d-6cf0-2f4a-a0c6-0cb5d5e20959

Media information
  Media type: fixed disk
  Is physical: yes
  Bytes per sector: 512
  Number of sectors: 2068480
  Media size: 1010 MiB (1059061760 bytes)

Digest hash information
  MD5: 9c0de6c85327a66ddcf01861dfb6535
Other Functionality:

<table>
<thead>
<tr>
<th>Function</th>
<th>Tool(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify duplicate files</td>
<td>FSLint</td>
</tr>
<tr>
<td>Characterize files</td>
<td>FITS</td>
</tr>
<tr>
<td>Scan for viruses</td>
<td>ClamTK</td>
</tr>
<tr>
<td>Examine, copy, extract information from old Mac disks</td>
<td>HFSEExplorer</td>
</tr>
<tr>
<td>Capture AV file metadata</td>
<td>MediaInfo, FFProbe</td>
</tr>
<tr>
<td>Extract text from older binary (.doc) Word files</td>
<td>antiword</td>
</tr>
<tr>
<td>Read contents of Microsoft Outlook PST files</td>
<td>readpst</td>
</tr>
<tr>
<td>Examine embedded header information in images</td>
<td>pyExifToolGUI</td>
</tr>
<tr>
<td>Generate images of problematic disks or particular disk types</td>
<td>dd, dcfldd, ddrescue, cdrdao (in addition to Guymager)</td>
</tr>
<tr>
<td>Identify files that are partially similar but not identical</td>
<td>sdhash, ssdeep</td>
</tr>
<tr>
<td>Package files for storage and/or transfer</td>
<td>BagIt (Java) library, Bagger</td>
</tr>
</tbody>
</table>
Quick Start Guide
Most recent version always available at:
http://wiki.bitcurator.net/

BitCurator
Quickstart Guide

Last updated: September 23, 2014
Release coverage: 1.0.0
BitCurator Access Project

• Funded by Andrew W. Mellon Foundation – October 1, 2014 – September 30, 2016

• Ultimate goal is enabling new forms of research and discovery based on born-digital materials.

• Pursuing three main approaches to providing access:
  – setting up a server that holds full disk images and lets end users dynamically walk the directory tree to access folders and files
  – exporting files and metadata (DFXML) and loading files and DFXML into collection access environments for search and navigation
  – access to disk image content through emulation.

• Will also develop tools to redact files, file system metadata, and targeted bitstreams within disks or directories.
BitCurator Access Core Team

• Cal Lee, PI
• Kam Woods, Technical Lead and Co-PI
• Alex Chassanoff, Project Manager
• Sunitha Misra, Software Developer
DIMAC (Disk Image Access for the Web)

- Developed by Sunitha Misra and Kam Woods
- To dynamically navigate and download contents of a disk image, without having to download or mount the full image
- See: [https://github.com/kamwoods/dimac](https://github.com/kamwoods/dimac)
- Demo at: [http://www.youtube.com/watch?v=BwiWFqxYzQ8](http://www.youtube.com/watch?v=BwiWFqxYzQ8)

BitCurator Consortium

• Continuing home for hosting, stewardship and support of BitCurator tools and associated user engagement
• Administrative home: Educopia Institute
• Funding based on membership dues
• Institutions as members, with two categories of membership: Charter and General
• Software continues to be free and open-source, but member institutions get additional benefits

http://www.bitcurator.net/bitcurator-consortium/
Get the software
Documentation and technical specifications
Screencasts
Google Group
http://wiki.bitcurator.net/

People
Project overview
Publications
News
http://www.bitcurator.net/

Twitter: @bitcurator
Conclusions

• There are many tools available to do routine tasks
• This does involve a lot more “product” (data and metadata) than with analog materials
• The trick(s):
  – Identifying those tasks that are fit for automation
  – Tame existing tools to do our bidding
  – Focus human attention on things that require judgment and professional decisionmaking