Automated Analysis and Visualization of Disk Images and File Systems for Preservation

Kam Woods, Christopher A. Lee, Sunitha Misra; University of North Carolina at Chapel Hill, School of Information and Library Science; Chapel Hill, NC

Abstract

We present work on the analysis and visualization of disk images and associated filesystems using open source digital forensics software as part of the BitCurator project. We describe methods and software designed to assist in the acquisition of forensically-packaged disk images, analysis of the filesystems they contain, and associated triage tasks in archival workflows. We use open source forensics tools including fiwalk*,* bulk extractor*, and* The Sleuth Kit *to produce technical metadata. These metadata are then reprocessed to aid in triage tasks, serve as documentation of digital collections, and to support a range of preservation decisions.*

Introduction

As media and filesystems become larger and more complex, effective preservation decisions depend upon not only improved automation, but also an ability to visualize the data effectively; to provide practitioners with the tools that allow them to make inferences and decisions based on factors that are not apparent when browsing a live filesystem. Archives and other collecting institutions are increasingly in need of better tools and methods to enable simple, effective, and timely decisions about born-digital information acquired on fixed and removable media such as floppy and optical disks, hard disks, and solid-state devices.

Disk images pose significant preservation opportunities and challenges for collecting institutions. Images may be preserved in their entirety, preserved with some redaction of the original bitstream, or discarded after specific contents (individual files and metadata) have been extracted from the filesystem. Although current repository software systems provide mechanisms to search, analyze, and visualize the contents of curated collections, pre-ingest assessment and triage of disk images is often performed by exploring the filesystem manually. Both digital curation professionals and end users can benefit from simplified and more informative characterizations of a disk's content.

This paper describes work on the analysis and visualization of disk images and associated filesystems using open source digital forensics software as part of the BitCurator project.

BitCurator incorporates software designed to improve coverage and efficiency when analyzing disk images, and reduce the potential for error when handling these materials in archival workflows. We use and expand on tools including Simson Garfinkel's *fiwalk* and *bulk extractor* and Basis Technology's *The Sleuth Kit* to produce human-readable reports using technical metadata extracted from raw and forensically-packaged images. The data generated by these tools can be used to improve triage of and access to digital collections, and to support a range of preservation decisions.

In this approach, bitstreams are acquired from digital media and packaged in a forensic disk image format. They are subsequently processed to produce two sets of data: (1) a detailed report – based on Digital Forensics XML (DFXML) – on data from the filesystem, and (2) sets of features corresponding to information within the filesystem that may be private, sensitive, individually identifying, or indicative of specific actions on the part of a user. The initial disk image filesystem report (produced by *fiwalk*) details the following filesystem hierarchy information in a single XML file using the current set of Digital Forensics XML tags: volume structure, permissions, timestamps, file hashes, and information on data that has been orphaned or deleted. With this metadata, one can rapidly produce informative, human-readable reports, including file format distribution, histograms and timelines of modification, access, and change (MAC) times, location and contents of user accounts, and identification of compressed, encrypted, and "hidden" data.

We describe how these customizable reports can assist in rapidly assessing the contents of a disk in order to answer questions about usage of the original device, relevance of file-level objects within the image, and the technical context in which these objects were produced, modified, or copied.

In the remainder of the paper, we describe experimental application of these tools and methods to an approximately 6TB corpus of disk images provided by real-world collecting institutions. We discuss how the disk image acquisition and reporting techniques can be added to existing workflows and the potential for integration with existing software systems used for collection management.

Use of Digital Forensics in Archives

The relevance of digital forensics methods and tools to archives and other collecting institutions has been widely discussed in the professional literature over the past several years [1,2,4].

Forensically packaged disk images incorporate information that can be used to support complex preservation decisions. Forensic image formats such as the Advanced Forensic Format (AFF) and Guidance Software's Evidence File (E01) format include metadata that document the acquisition process, indicate whether areas of the source media are damaged, and record manufacturer data associated with the media. This information may be used to support technically consistent workflows, improve records of provenance, and assess issues associated with authenticity and duplication [3,13].

Both commercial (AccessData's *Forensic Tool Kit*, Guidance Software's *EnCase*) and open source (Basis Technology's *The Sleuth Kit*, Simson Garfinkel's *bulk extractor* and DFXML tools) forensic analysis tools can further assist in making informed

preservation and access decisions. Users can capture a rich body of contextual information associated with files, identify the presence of deleted and damaged files (and determine whether such "lost" materials have archival value or require removal), and isolate potentially private and sensitive information to be protected or redacted.

BitCurator Project

The BitCurator Project, a collaborative effort led by the School of Information and Library Science (SILS) at the University of North Carolina at Chapel Hill and Maryland Institute for Technology in the Humanities (MITH) at the University of Maryland, builds on previous work by addressing two fundamental needs and opportunities for collecting institutions: (1) integrating digital forensics tools and methods into the workflows and collection management environments of libraries, archives and museums (LAMs) and (2) supporting properly mediated public access to forensically acquired data [11]. The project team has developed a set of requirements documents, which we are iteratively revising based on input from our expert advisors.

We are developing and disseminating a suite of open source tools. These tools are currently being developed and tested in a Linux environment; the software on which they depend can readily be compiled for Windows environments (and in most cases are currently distributed as both source code and Windows binaries). We intend the majority of the development for BitCurator to support cross-platform use of the software.

We are freely disseminating the software under an open source (GPL, Version 3) license. BitCurator provides users with two primary paths to integrate digital forensics tools and techniques into archival and library workflows.

First, the BitCurator software can be run as a ready-to-run Linux (Ubuntu 12.04LTS) environment that can be used either as a virtual machine (VM) or installed as a host operating system. This environment is customized to provide users with graphic user interface (GUI)-based scripts that provide simplified access to common functions associated with handling media, including facilities to prevent inadvertent write-enabled mounting (software write-blocking).

Second, the BitCurator software can be run as a set of individual software tools, packages, support scripts, and documentation to reproduce full or partial functionality of the ready-to-run BitCurator environment. These include a software metapackage (.deb) file that replicates the software dependency tree on which software sources built for BitCurator rely; a set of software sources and supporting environmental scripts developed by the BitCurator team and made publicly available at via our GitHub repository (links at http://wiki.bitcurator.net); and all other third-party open source digital forensics software included in the BitCurator environment.

Disk Imaging

Forensic disk imaging techniques use write-blocking hardware (or software) to support the extraction of unmodified raw or logical images from source media. Write-blocking is used in concert with a software package capable of reading the source media and writing one or more forensic packaging formats. Common formats include the aforementioned AFF and E01, Access Data's AD1 logical container to package files and folders only, and raw bitstreams such as those produced by the UNIX *dd* (disk duplication) tool.

The extraction of disk images from source media can significantly reduce risk during triage and analysis tasks. Legacy media arriving at a collecting institution in aged or degraded condition may have a limited number of read cycles prior to failure or further damage. Forensic imaging programs typically identify damaged and unreadable sectors on a disk. Mounting the resulting images in a sandboxed virtual machine or accessing the contained filesystems via a dedicated forensic environment widens the range of recovery options and reduces the risk of host contamination (in cases when the source media contains malware) and the possibility of inadvertent changes to the original filesystem.

The preconfigured BitCurator environment incorporates the open source GUI-based forensic imaging tool Guymager, along with forensically-enhanced versions of *dd* including *dc3dd* and *dcfldd*. A sample disk image acquisition is shown in Figure 1.

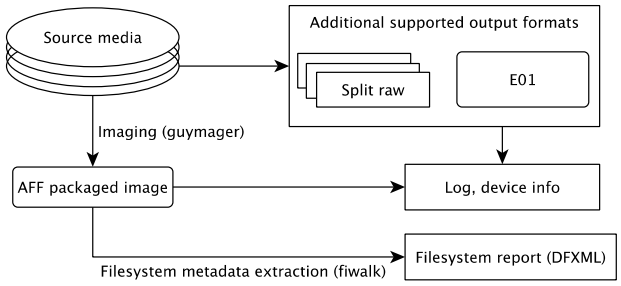

***Figure 1**. Forensically imaging a disk and extracting filesystem metadata prior to analysis.*

Disk Image Analysis and Reporting

In this section, we discuss an approach to automatically extracting and semi-automatically analyzing information from the filesystem(s) contained on a piece of digital media. In previous publications, we have discussed some of the interactive disk image and file analysis tools available to users of the BitCurator environment [11,12]. Here we focus specifically on a set of software modules that interoperate primarily via the production and consumption of Digital Forensics XML (DFXML).

Following the creation of a disk image (forensically-packaged or otherwise), a user tasked with analyzing, recording, and reporting on the contents of that disk image has a number of options. The disk image can be mounted on a host system or in a virtual machine (using the appropriate mount command or third-party tool depending on the operating system used by the host or VM), and any readable filesystems can be explored manually. Alternatively, both the filesystem(s) and unallocated spaces can be identified and explored using a freely available commercial tool such as AccessData's FTK Imager.

There are several issues with these approaches. Manual examination of mounted file systems is error-prone and scales poorly in terms of hours of human effort as file format complexity and the size of the filesystem increases [9,10]. The commonly used FTK Imager provides a powerful disk image browsing interface, but has a limited set of filesystem analysis utilities. Full

commercial forensics packages such as the Forensic Tool Kit (FTK) and EnCase present financial and technical barriers to smaller collecting institutions.

Forensic Analysis Workflow

For the BitCurator project, we have developed a forensic analysis workflow for collecting institutions based entirely on open source and public domain software and application programming interfaces (APIs). An overview of this workflow is shown in Figure 2, and described in the remainder of this section.

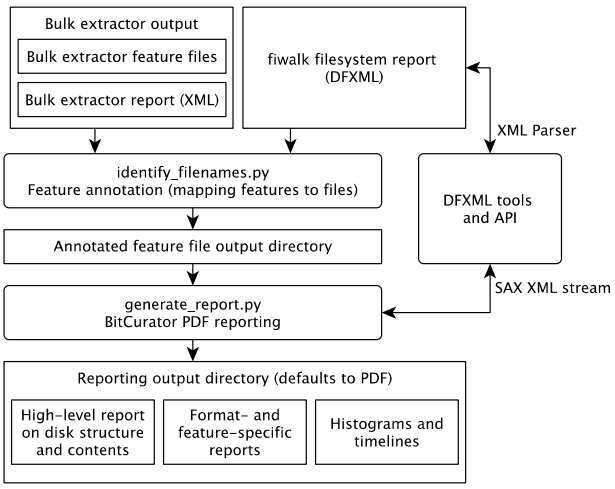

***Figure 2**. A semi-automated report generation workflow. BitCurator relies on the output of a number of open source tools including fiwalk, bulk extractor, the feature annotation script, and the DFXML toolset). BitCurator maintains a modified (open source) branch of the DFXML tools for use with our reporting facilities.*

The disk image analysis workflow in BitCurator begins with the execution of two forensics tools produced by Simson Garfinkel: *fiwalk* and *bulk extractor*. The *fiwalk* tool identifies and interprets the contents of filesystems contained in disk images using the filesystem access API in Basis Technology's *The Sleuth Kit* (TSK). Currently, TSK is capable of identifying and extracting the structure and contents of the NTFS, FAT, Ext3, HFS+, and UFS filesystems. In situations when the filesystem encountered cannot be recognized, *fiwalk* reports the associated TSK error string, and attempts to process the remainder of the image. The *fiwalk* tool can produce both XML (as Digital Forensics XML) and simple text reports on the processed media: filesystem(s) and volume(s) encountered, file objects and associated metadata within a given filesystem, and information on byte runs associated with file fragments.

The *bulk extractor* tool is employed to identify potentially private and sensitive information, and to search for relevant patterns in the bitstream specified by the user. *Bulk extractor* does not parse filesystems, but instead reads the raw contents of the disk image. It applies a set of fast lexical analyzer (flex) scanners to identify and report on 'features' (e.g., email addresses, telephone numbers, and uniform resource locators (URLs) such as those appearing in search history cache files) present in the bitstream; these may include potentially private and individually identifying information (PII) requiring redaction [7,8]. This tool recursively reparses common types of container files (e.g., zip files and tar archives) and may also be configured to execute regular expression scans over the image file.

The following section describes how the output produced by these tools is read and processed by the BitCurator reporting tool.

Reporting on Forensic Tool Output

The BitCurator reporting tool (written in Python 3) reads and reprocesses forensically-extracted metadata, reporting output from the above tools in one of two ways.

First, it reads an XML stream from the *fiwalk* DFXML output using a SAX (Simple API for XML) parser and the DFXML Python API [7]. The *fiwalk* XML output file can be quite large; for example, processing a 10GB Windows XP image in our test corpus resulted in a 63MB XML file. The use of a SAX parser reduces memory overhead and is sufficient for a task in which one is simply reading technical metadata on each file object reported within the DFXML independently.

Second, it processes feature output produced by *bulk extractor*. Building reports from potential personally identifying information (PII) and other features identified within disk images by *bulk extractor* requires an additional step prior to running the BitCurator report generator. Because *bulk extractor* ignores filesystem structure, the features it reports are identified only by their absolute byte offset into the disk image. In order to produce reports on which files contain potential PII and other features of interest, another tool distributed with *bulk extractor* must be run (*identify_filenames.py*). This script generates a directory of annotated feature files that indicate both the absolute offset and relevant filename (if the feature appears within a file). The BitCurator reporting tool is then run over the directory of annotated files, producing (by default) PDF reports for each feature type; for example, a PDF file including a table with each email address identified within a file, along with a graphical histogram representation of the most commonly identified email addresses.

For disk images containing large numbers of files, or those on which hundreds or thousands of individual features are identified by the forensics tools, producing a static document such as a PDF becomes impractical, both in terms of document size and usefulness. At the same time, it is undesirable to require the user to independently configure and run the tool multiple times to generate individual reports. There are two ways in which we provide users with options to overcome these issues.

First, using the existing SAX parsing facilities and the open source OpenPyXL tool (distributed with BitCurator), we provide users with a simple Python tool capable of reading *fiwalk* XML output and writing the file object metadata in tabular form in an Excel spreadsheet.

Second, the output of the BitCurator reporting tool can be manipulated using a plain text configuration file. This allows the user to adjust the maximum number of entries in any given report (for batch disk analysis tasks when it is unclear ahead of time how many entries may be encountered), produce reports only for specific file types and feature types, and enable or disable reporting for "hidden" and deleted files. By default, the reporting tool also produces high-level reports on the contents of the disk

images that can be used to aid in triage tasks and to retain as documentation of a disk's content (for use by digital curation professionals or end users). These include:

- Volume(s), partition(s), filesystem(s) encountered
- Total number of files, deleted files, and empty files
- File format by count (currently identified by UNIX *file*)
- Number of features identified by *bulk extractor*, including counts for features found in files and unallocated areas

Technical Metadata

Disk Image: image_filename: charlie-work-usb-2009-12-11.aff

Feature	Value
SECTORSIZE	1024
FTYPE STR	ntfs
PARTITION OFFSET	512
BLOCK SIZE	4096
ACQUISITION SECONDS	73
FIRST BLOCK	0
BLOCK COUNT	258559
LAST BLOCK	258558
PAGESIZE	16777216
FTYPE	1
IMAGE FILENAME	charlie-work-usb-2009-12-11.aff
Number of Files	128
Total Directories	23
Total Deleted Files	0
Total Unused Files	0
Files with Nlinks > 1	0
Empty Files	9
Big Files(> 1 MB)	5

***Figure 3**. A partial technical report on a disk image extracted from a removable USB disk. The first ten lines of the report include metadata on the structure and size of the NTFS file system along with the time required to acquire the image. The remaining entries provide an overview of the contents of the file system, including a tunable parameter to identify files over a certain size (set in this example to files larger than 1MB).*

A partial view of a report produced by the BitCurator software is shown in Figure 3. For smaller disk images (those containing less than a few hundred files), this report may be combined with a tabular listing of the contents of the filesystem, including paths and filenames, whether or not a particular file has been deleted, and file size.

BitCurator uses a Python 3 adaptation of the *fpdf* package to produce PDF reports on demand, and the *matplotlib* library to output the corresponding graphs and histograms.

Disk Image Test Corpus

Testing of the tools incorporated into BitCurator (along with the workflow outlined here) provides substantial data on how they might be used in collecting institutions and how they might be improved.

In order to address potential issues encountered by collecting institutions when handling digital media, we have constructed a 6TB test corpus composed of real-world archival disk images and other data from collecting institutions that have participated in the advisory groups for BitCurator, including the National Institute for Standards and Technology (NIST), the Maryland Institute for Technology in the Humanities, Duke University, iBiblio, the National Library of Australia, and the City of Vancouver Archives.

***Figure 4**. A selection of legacy digital media imaged for the BitCurator test corpus.*

A primary goal for this corpus is to cover a wide range of use cases relevant both to the preservation activities currently being performed in institutions with legacy media holdings, and those that are not yet commonplace. The corpus includes 400K MFS, 800K HFS, and 1.4MB HFS floppy images; 720K and 1.44MB FAT floppy images; HFS and HFS+ formatted hard disk images in various sizes; FAT and NTFS formatted hard disk images in various sizes; 100MB and 250MB FAT formatted Zip disk images; Ext3 and Ext4 formatted hard drive images; and ISO9660 format CD-ROMs.

Corpus Processing

The time required to process a given disk image with *fiwalk*, *bulk extractor*, the annotation tool, and the BitCurator reporting module is a function not only of the processing and disk speed of the workstation, but also on the composition of the disk images. For example, a compressed 10GB NTFS-formatted Windows XP image (packaged as an Advanced Forensic Format file) containing more than 40,000 file objects processed on our reference workstation (Intel Core i7, 8GB RAM, 3Gb/s SATA disks) requires approximately 14 minutes to process with *bulk extractor*, and approximately 20 minutes to process with *fiwalk*. In contrast, an AFF image of a 16GB NTFS-formatted USB drive (similarly compressed) containing fewer than 200 files is processed by *bulk extractor* in under three minutes, and by *fiwalk* in a few seconds.

Smaller legacy media items such as 720K and 1.44MB floppy disks can be processed extremely quickly. In our corpus, a set of 470 such floppies was processed as a batch in less than 10 minutes by both *fiwalk* and *bulk extractor*, with an additional 10 minutes required for the report generation tool.

The limiting factor in terms of time required to process a collection of media items is generally the BitCurator report generation tool, which may have to process extremely large text feature reports and XML file system reports as produced by *bulk*

extractor and *fiwalk*. Following the initial configuration, however, each of these tools runs non-interactively, making it possible to process collections of arbitrary size as background tasks.

Integration with Archival Environments

The BitCurator project packages the software described in the previous sections as both a virtual machine and an installable operating system (both currently built using Ubuntu 12.04LTS). All of the software used in this environment is open source or public domain, and may be freely distributed. This packaging reduces barriers to entry in working with digital forensics tools, particularly for smaller collecting institutions with limited technical and hardware resources.

Both the individual tools and the BitCurator environment may be run alongside existing collection management, repository or preservation environment software. We are continuing to refine and modularize the BitCurator-specific software components to simplify access to the reports and metadata as microservices.

There is currently no common method for integrating metadata produced as Digital Forensics XML into established archival metadata standards. DFXML is continuing to evolve and currently has no fixed schema, as it is designed to accommodate and encourage interoperability between a variety of forensics tools that emit and consume XML. The structure and tags associated with the DFXML filesystem metadata produced by *fiwalk* are relatively stable and well documented, and as part of the BitCurator project we provide a tag library that includes definitions, structural properties, and examples for each element associated with filesystem and file object output. The most current version of the tag library can be found at our development portal, http://wiki.bitcurator.net/.

Future Work

Tools that generate compact, expressive reports describing the structure and content of disk images address an ongoing need in collecting institutions. Currently, the reporting tools developed for the BitCurator project produce a range of independently formatted (as PDF, text, and Office documents) reports, primarily using the DFXML output produced by *fiwalk* and the low-level feature reports produced by *bulk extractor*. The output of these tools can be adjusted to the needs of specific institutions and collections, but deciding what is useful in a report can still be a significant barrier for users with little prior experience with forensics tools

As part of our ongoing work, we envision improvements to these reports that build both on our internal corpus testing and feedback from users of our current releases. These may include high-level visualizations of disk structure (in the form of treemaps or simple graphs), filesystem activity histograms over timelines that visually highlight date ranges indicating significant read/write activity on the media, and the use of additional tools to identify and flag copy-protection schemes and encrypted materials.

Processing of additional legacy filesystems is another area of interest. At the time of writing, there is no support for legacy HFS filesystems (used by early Macintosh computers) in *The Sleuth Kit*. Because our reporting system depends both on *TSK* and *fiwalk*, we are currently unable to generate reports for media containing HFS volumes.

Conclusion

The disk image analysis and reporting techniques described in this paper have been implemented as modular, incremental changes to established open source and public domain digital forensics tools. These changes focus on improving accessibility and the relevance of these tools in preservation workflows used in collecting institutions.

The BitCurator project packages and freely distributes these tools in a number of ways to allow practitioners and researchers to apply them on their own collections: as a virtual machine preconfigured to support software write-blocking, forensic imaging, filesystem analysis, and forensic metadata extraction and reporting; as an installable version of the same environment; and as an open source repository including code specifically developed for BitCurator. Packages, downloads, and associated documentation can be downloaded at our development portal, http://wiki.bitcurator.net/.

Acknowledgements

The BitCurator project is supported by a grant from the Andrew W. Mellon Foundation. In addition to the authors, members of the BitCurator team are Alexandra Chassanoff, Matthew Kirschenbaum (Co-PI), and Porter Olsen. We would also like to acknowledge the valuable contributions of the project's two advisory boards: the Development Advisory Group (DAG) and Professional Experts Panel (PEP).

References

[1] AIMS Working Group. "AIMS Born-Digital Collections: An Inter-Institutional Model for Stewardship." 2012.

[2] J. L. John, "Digital Forensics and Preservation", Digital Preservation Coalition, 2012.

[3] M. J. Gengenbach, "The Way We Do it Here" Mapping Digital Forensics Workflows in Collecting Institutions. Masters Paper for the M.S. in L.S degree. August, 2012.

[4] M. G. Kirschenbaum, R. Ovenden, and G. Redwine, "Digital Forensics and Born-Digital Content in Cultural Heritage Collections." (Council on Library and Information Resources, Washington, DC, 2010).

[5] M. Cohen, S. L. Garfinkel, and B. Schatz, Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow, Proceedings of DFRWS 2009, Montreal, Canada, 2009.

[6] S. L. Garfinkel, AFF: A New Format for Storing Hard Drive Images, *Communications of the ACM* 49, no. 2, 2006), pg 85-87.

[7] S. L. Garfinkel, Digital Forensics XML and the DFXML Toolset, Digital Investigation 8, 2012, pg. 161-174

[8] S. L. Garfinkel, "Providing Cryptographic Security and Evidentiary Chain-of-Custody with the Advanced Forensic Format, Library, and Tools." (International Journal of Digital Crime and Forensics 1, no. 1, 2009), pg. 1-28.

[9] S. L. Garfinkel, Lessons Learned Writing Digital Forensics Tools and Managing a 30TB Digital Evidence Corpus, Digital Investigation 9, 2012, pg. S80-S89.

[10] S. L. Garfinkel, Digital Forensics Research: The Next 10 Years, Proceedings of DFRWS 2010, Portland, OR, August 2010

[11] C.A. Lee, M. G. Kirschenbaum, A. Chassanoff, P. Olsen, and K. Woods, BitCurator: Tools and Techniques for Digital Forensics in Collecting Institutions, D-Lib Magazine 18, No. 5/6, May/June 2012.

[12] K. Woods and C. A. Lee, Acquisition and Processing of Disk Images to Further Archival Goals, Proceedings of Archiving 2012,

Springfield, VA, Society for Imaging Science and Technology, pg. 147-152.

[13] K. Woods, C. A. Lee, and S. L. Garfinkel, Extending Digital Repository Architectures to Support Disk Image Preservation and Access, JCDL '11: Proceeding of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries, New York, NY, 2011, ACM Press, pg 57-66.

Author Biographies

Kam Woods is currently a Postdoctoral Research Associate in the School of Information and Library Science at the University of North Carolina at Chapel Hill. His research interests include long-term digital preservation, digital archiving, and the application of digital forensics tools and techniques to archival and preservation data analysis and management.

Christopher (Cal) Lee is an Associate Professor at the School of Information and Library Science at the University of North Carolina at Chapel Hill. His primary area of research is the long-term curation of digital collections. He is Principal Investigator for the BitCurator project and editor of I, Digital: Personal Collections in the Digital Era *published by the Society of American Archivists.*

Sunitha Misra is a Masters student in the School of Information and Library Science at the University of North Carolina at Chapel Hill. She is currently a software developer on the BitCurator project.