

# METADATA

Image source:  
<https://www.redbubble.com/people/zombiecumny/>

In this section we're going to discuss the different information and reports one can collect while creating disk images, and the tools for creating and interacting with that information

## sidecars



NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Some of the most valuable metadata about a disk image is thankfully automatically captured by disk imaging software. All of the disk imaging software we discussed create a “sidecar file” (hence this slide with a picture of a dog in a motorcycle sidecar). Different applications contain different information, but almost all of them contain some key items, such as when and how the disk image was created.

## ddrescue sidecar

```
# Mapfile. Created by GNU ddrescue version 1.23
# Command line: ddrescue -b 512 -r4 -v /dev/disk2 /Volumes/TBMA Drobo/sharedVM/
Samsung_32gb_test_USB_raw_blocker3_ddrescue.dd /Volumes/TBMA Drobo/sharedVM/
Samsung_32gb_test_USB_raw_blocker3_ddrescue.log
# Start time: 2019-01-29 10:01:06
# Current time: 2019-01-29 10:51:30
# Finished
# current_pos  current_status  current_pass
0x778800000    +                1
#      pos      size  status
0x000000000  0x778800000  +
```

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Ddrescue sadly does not include the md5 checksum of the disk image in the sidecar file. The software does however provide certain helpful technical metadata like the block size of the file system, or specific errors encountered.

# ewfacquire sidecar

- Not automatically generated!
  - The user must specify the log output path
- Just the md5 hash of the checksum
- Also lists errors if they are encountered

```
bcadmin@bitcurator: ~  
File Edit View Search Terminal Help  
Number of bytes to acquire: 2.0 GiB (2206984192 bytes)  
Evidence segment file size: 1.4 GiB (1572864000 bytes)  
Bytes per sector: 512  
Block size: 64 sectors  
Error granularity: 64 sectors  
Retries on read error: 2  
Zero sectors on read error: no  
  
Continue acquiry with these values (yes, no) [yes]:  
Acquiry started at: Jul 01, 2021 20:24:52  
This could take a while.  
  
Status: at 18%.  
  acquired 383 MiB (402554880 bytes) of total 2.0 GiB (2206984192 bytes).  
  completion in 18 second(s) with 95 MiB/s (100317463 bytes/second).  
  
Status: at 36%.  
  acquired 777 MiB (815693824 bytes) of total 2.0 GiB (2206984192 bytes).  
  completion in 14 second(s) with 95 MiB/s (100317463 bytes/second).  
  
Status: at 50%.  
  acquired 1.2 GiB (1309081600 bytes) of total 2.0 GiB (2206984192 bytes).  
  completion in 8 second(s) with 105 MiB/s (110349209 bytes/second).  
  
Status: at 72%.  
  acquired 1.4 GiB (1592131504 bytes) of total 2.0 GiB (2206984192 bytes).  
  completion in 6 second(s) with 95 MiB/s (100317463 bytes/second).  
  
Status: at 91%.  
  acquired 1.9 GiB (2026110976 bytes) of total 2.0 GiB (2206984192 bytes).  
  completion in 1 second(s) with 100 MiB/s (105094485 bytes/second).  
  
Acquiry completed at: Jul 01, 2021 20:25:14  
  
Written: 2.0 GiB (2206984380 bytes) in 22 second(s) with 95 MiB/s (100317471 bytes/second).  
MD5 hash calculated over data: 634c9336d2dcb3d3a670071cd032952f  
ewfacquire: SUCCESS  
bcadmin@bitcurator:~$
```

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Ewfacquire is the disk imaging command built into libewf. The log file created by ewfacquire is very bare bones, the least detailed of any disk imaging tool I've seen. It's not even automatically created, and even then, it just includes the md5 hash of the disk image, and nothing else.



# Metadata Tools

- Disktype (pre-installed on BitCurator)

- Sample input:

```
disktype ~/path/to/disk.img
```

- Sample output:

```
--- /Volumes/TBMA Drobo/Time Based Media Artwork/2018-027_Giorno/2018-027-b-1_FTK.001  
Regular file, size 3.637 GiB (3904897024 bytes)  
No type and creator code  
DOS/MBR partition map  
Partition 1: 3.633 GiB (3900702720 bytes, 7618560 sectors from 8192)  
Type 0x0B (Win95 FAT32)  
FAT32 file system (hints score 4 of 5)  
Volume size 3.629 GiB (3896508416 bytes, 118912 clusters of 32 KiB)
```

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

I'm now going to run through a big list of tools that I think are helpful when collecting metadata on disk images.

Disktype detects and outputs the file systems and partition tables of a volume

You can run disktype on disk images or on a disk itself, but you cannot run disktype on a mounted disk. So, in the event you want to run disktype on a mounted disk, you need to unmount from the command line, either using the diskutil command in macOS or the umount command in Linux

# Metadata Tools

- mmls (part of SleuthKit, pre-installed on BitCurator)
  - Sample output:

```
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors
```

	Slot	Start	End	Length	Description
000:	Meta	0000000000	0000000000	0000000001	Primary Table (#0)
001:	-----	0000000000	0000008191	0000008192	Unallocated
002:	000:000	0000008192	0007626751	0007618560	Win95 FAT32 (0x0b)

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

mmls is similar to disktype, in that it also displays the contents of a file system. In general, this is used to list the partition table contents so that you can determine where each partition starts. These are the “offsets” we talked about on Day 1. The output identifies the type of partition and its length, which makes it possible to use tools to extract the partitions.

# Metadata Tools

- SleuthKit
  - Open source digital forensics software
  - Library of command line tools (pre-installed on BitCurator)
- Autopsy
  - GUI that bundles these tools, runs on Windows
- Command line tools include
  - Mmls
  - Tsk\_recover
  - A bunch I don't use

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

SleuthKit is an open source suite of tools for processing disk images. I've predominantly interacted with sleuthkit as a library of command line tools pre-installed in BitCurator, but I have also used the Windows dependent GUI Autopsy, which bundles the tools in sleuth kit. I only tested Autopsy a bit before the pandemic, but at the time I was struggling with exporting reports from the software. It presented information nicely and visually, but from what I could tell, wouldn't be ideal for creating metadata reports for future reference.

We've already talked about mmls. The other commonly used sleuth kit tool in cultural heritage is tsk\_recover. "Tsk" stands for "The Sleuth Kit." Tsk\_recover is used to carve files from a file system.

File carving is a process used in computer forensics to extract data from a disk drive or other storage device without the assistance of the file system that originality created the file. Essentially looking at the byte offset of a file based on it's metadata, and copying the raw bits to a new location, and rewrapping those bits with the original file name.

For our purposes, this isn't terribly different from mounting a disk image and copying the files out of it, it just skips a step. With damaged file systems, or data that is obfuscated from the file system, carving can allow you to extract files you otherwise couldn't.



# Metadata Tools

- siegfried (pre-installed on BitCurator)
  - Sample output

```
siegfried : 1.9.1
scandate  : 2021-07-01T14:45:31-06:00
signature : default.sig
created   : 2020-10-06T19:13:40+02:00
identifiers :
- name    : 'pronom'
  details : 'DROID_SignatureFile_V97.xml;
container-signature-20201001.xml'
---
filename  :
'/Users/eddy/Downloads/SantaClausConquersTheMartians.iso'
filesize  : 2206984192
modified  : 2021-06-13T20:38:26-06:00
errors    :
matches   :

- ns      : 'pronom'
id       : 'fmt/468'
format   : 'ISO Disk Image File'
version  :
mime     :
basis    : 'extension match iso; byte
match at [[32769 5] [34816 6]]'
warning  :
```

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Siegfried is a file format identification tool which examines a file and attempts to match it to the PRONOM registry. Siegfried is not a disk image specific tool, it's just a metadata tool I really like.

The PRONOM id is listed in the id field, in this example it is fmt/468.

# Metadata Tools

- tree (pre-installed on BitCurator)
  - Sample output:

```
/Volumes/Santa Claus\ Conquers\ The\ Martians
├── AUDIO_TS
└── VIDEO_TS
    ├── VIDEO_TS.BUP
    ├── VIDEO_TS.IFO
    ├── VTS_01_0.BUP
    ├── VTS_01_0.IFO
    ├── VTS_01_0.VOB
    ├── VTS_01_1.VOB
    ├── VTS_01_2.VOB
    └── VTS_01_3.VOB

2 directories, 8 files
```

Tree is also not a disk imaging specific tool, but again can be helpful when looking to automate description of disk images. Tree is a command line tool for creating a text output of the directory structure of a volume or directory

# Metadata Tools

```
fiwalk -X ~/path/to/fiwalk_output.xml  
'/path/to/disk_image.e01'
```

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

A fiwalk report can be really helpful in developing a more thorough understanding of every byte in a disk image. Fiwalk, short for “file and inode walk,” is an open source command line tool. The tool creates detailed technical information about a disk image, including a checksum for each file in the file system.

As the name suggests, fiwalk “walks” the file tree and collects information (metadata) about each of the files along the way, including the date the file was last accessed, the date it was last modified, the file type, the user who created the file, and more.

# DFXML

```
<fileobject>
  <parent_object>
    <inode>66569</inode>
  </parent_object>
  <filename>Stills/Full TIFFs/HNTBS-720p25-master-v61191.tif</filename>
  <partition>1</partition>
  <id>327</id>
  <name_type>r</name_type>
  <filesize>3692302</filesize>
  <alloc>1</alloc>
  <used>1</used>
  <inode>71429</inode>
  <meta_type>1</meta_type>
  <mode>511</mode>
  <nlink>1</nlink>
  <uid>0</uid>
  <gid>0</gid>
  <mtime prec="2">2015-11-04T01:56:20</mtime>
  <atime prec="86400">2019-02-08T01:13:24</atime>
  <ctime prec="2">2015-11-04T01:56:20</ctime>
  <byte_runs>
    <byte_run file_offset='0' fs_offset='713424896' img_offset='713425920' len='
  </byte_runs>
  <hashdigest type='md5'>f529a76b9df60158927c2e08c1cebc3b</hashdigest>
  <hashdigest type='sha1'>f0552fe049c11e75737a1597b5c2a49ff07ff989</hashdigest>
</fileobject>
<fileobject>
  <parent_object>
    <inode>66569</inode>
  </parent_object>
  <filename>Stills/Full TIFFs/HNTBS-720p25-master-v61192.tif</filename>
  <partition>1</partition>
  <id>328</id>
  <name_type>r</name_type>
```

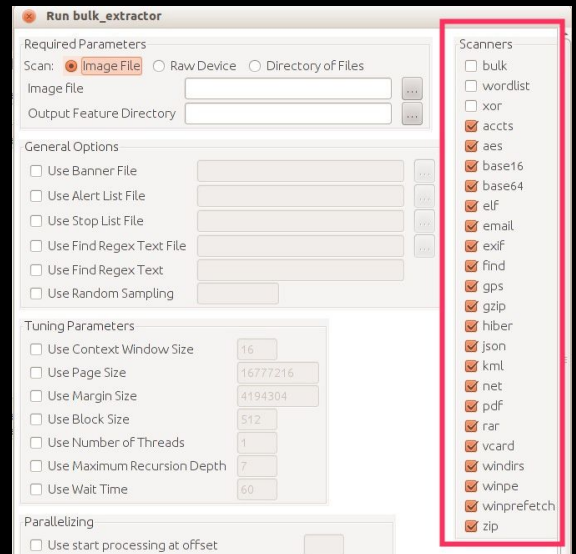
NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Fiwalk exports this information in a standardized and interoperable format, Digital forensics XML. You can see in the slide it includes the md5 and sha1 checksums of each file, along with the files sector offset.

# Bulk\_extractor Scanners

## BEviewer

- Helps protect the materials' donor by discovering potentially sensitive information before making a disk public (e.g. social security numbers), and
- explore materials by finding specific types of information (e.g. search for email addresses in order to locate correspondence between an author and her editors)



NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Another tool we want to introduce is Bulk Extractor, which I have honestly only interacted with using the GUI version, which is called The Bulk Extractor Viewer (BEviewer). Bulk extractor scans a disk image, or a directory of files and looking for personally identifiable information (PII) that a donor may want redacted before their materials are made publicly available. The output of the tool is a series of text files. The text file named "emails.txt" for instance, will have any emails encountered during Bulk\_extractor's scans. Another text file might be named url.txt and contain and urls discovered, another telephone.txt, etc. etc.

# Brunnhilde

- Open source tool by Tessa Walsh
  - Pre-installed in BitCurator
  - Also available on GitHub
- Python tool, easy to download with “pip” outside of BitCurator
  - But with many dependencies, especially for working with disk images
- Automatically runs `tsk_recover`, `fiwalk`, `siegfried`, `bulk_extractor`, and `tree`
  - “Carves” files from disk image
  - Compiles metadata in CSV format, and creates html reports you can read in your browser

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Brunnhilde is a tool that is used for processing digital collections, not just disk images. But it can be a great tool for automating disk imaging workflows. Brunnhilde combines many of the tools we’ve talked about already, including: Siegfried, tree, bulk\_extractor, tsk\_recover and fiwalk.

It also includes ClamAV (a virus scanning software).  
(read bullet points)

The most cumbersome Brunnhilde dependency is libewf, which we’ll walk through how to download

# Brunnhilde

- Usage:

```
brunnhilde.py -d disk_image.E?? /output/directory/to/create [basename]
```

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

This is the command for running brunnhilde from the command line. There is a new GUI for brunnhilde but I've never used it. You need to use the "d" flag when running brunnhilde on disk images. If you're just pointing brunnhilde to a directory or a mounted volume, omit the -d flag. As you can see from the example onscreen, the tool works with both raw images and ewf disk images. The [basename] bit there is just "some number." I think it was intended for acquisition or accession numbers. The basename requirement is deprecated, and only necessary with older versions of brunnhilde (less than 1.9), but the most recent version of BitCurator is still running v1.8

# Brunnhilde

Name	Size	Modified
carved_files	1 item	00:53
csv_reports	8 items	00:54
logs	1 item	00:54
dfxml.xml	14.0 kB	00:54
report.html	8.9 kB	00:54
siegfried.csv	1.9 kB	00:54
tree.txt	493 bytes	00:54

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Brunnhilde outputs its reports to a directory, which looks like this. The “carved files” directory contains the identified files from the file system. The other files and directories contain reports. The CSV files are used to populate the very helpful html report.



# Brunnhilde

Brunnhilde [Provenance](#) [Statistics](#) [File formats](#) [Versions](#) [MIME types](#) [Dates](#) [Unidentified](#) [Errors](#) [Duplicates](#)

## Brunnhilde HTML report

### Provenance

**Input source (directory or disk image):** /home/bcadmin/Desktop/SantaClausConquersTheMartians.iso  
**Accession/Identifier:** 007  
**Brunnhilde version:** brunnhilde 1.8.1  
**Siegfried version:** siegfried 1.8.0 /usr/share/siegfried/default.sig (2020-01-21T23:30:42+01:00) identifiers: - pronom: DROID\_SignatureFile\_V96.xml; container-signature-20200121.xml  
**Siegfried command:** sf -csv -hash md5 "/home/bcadmin/Desktop/Santa/007/carved\_files" -> "/home/bcadmin/Desktop/Santa/007/siegfried.csv"  
**Scan started:** 2021-07-01 00:54:27.215143

### Statistics

#### Overview

**Total files:** 8  
**Total size:** 2 GB  
**Years (last modified):** 2021 - 2021  
**Earliest date:** 2021-07-01T00:53:34Z  
**Latest date:** 2021-07-01T00:53:52Z

#### File counts and contents

Calculated by hash value. Empty files are not counted in first three categories. Total files = distinct + duplicate + empty files.

**Distinct files:** 6

The Brunnhilde html reports are very human friendly, and contain the information collected by the different tools incorporated into Brunnhilde

# Brunnhilde

Brunnhilde Provenance Statistics File Formats Versions MIME types Dates Unidentified Errors Duplicates

None found.

**Errors**  
None found.

**Duplicates**  
Duplicates are grouped by hash value.

Files matching checksum **0fa247951857779f36b3dc86dc1aa521**:

Filename	Filesize	Date modified	Errors	Checksum	Namespace	ID	Format	Format version	MIME type	Basis for ID	Warning
/home/bcadmin/Desktop/Santa/007/carved_files/VIDEO_TS/VIDEO_TS.BUP	6144	2021-07-01T00:53:34Z		0fa247951857779f36b3dc86dc1aa521	pronom	x-fmt/419	DVD data file and backup data file			extension match bup; byte match at 0, 12	
/home/bcadmin/Desktop/Santa/007/carved_files/VIDEO_TS/VIDEO_TS.IFO	6144	2021-07-01T00:53:34Z		0fa247951857779f36b3dc86dc1aa521	pronom	x-fmt/419	DVD data file and backup data file			extension match ifo; byte match at 0, 12	

Files matching checksum **4be3059ae4b10df4e00cf6916823386**:

Filename	Filesize	Date modified	Errors	Checksum	Namespace	ID	Format	Format version	MIME type	Basis for ID	Warning
/home/bcadmin/Desktop/Santa/007/carved_files/NTS_01_0.BUP	57344	2021-07-01T00:53:34Z		4be3059ae4b10df4e00cf6916823386	pronom	x-fmt/419	DVD data file and backup data file			extension match bup; byte match at 0, 12	
/home/bcadmin/Desktop/Santa/007/carved_files/VIDEO_TS/NTS_01_0.IFO	57344	2021-07-01T00:53:34Z		4be3059ae4b10df4e00cf6916823386	pronom	x-fmt/419	DVD data file and backup data file			extension match ifo; byte match at 0, 12	

Here you can see (maybe) brunnhilde listing out the checksums of individual files from within the disk image, along with other technical metadata like their file size and data modified.

# Disk Image Differencing

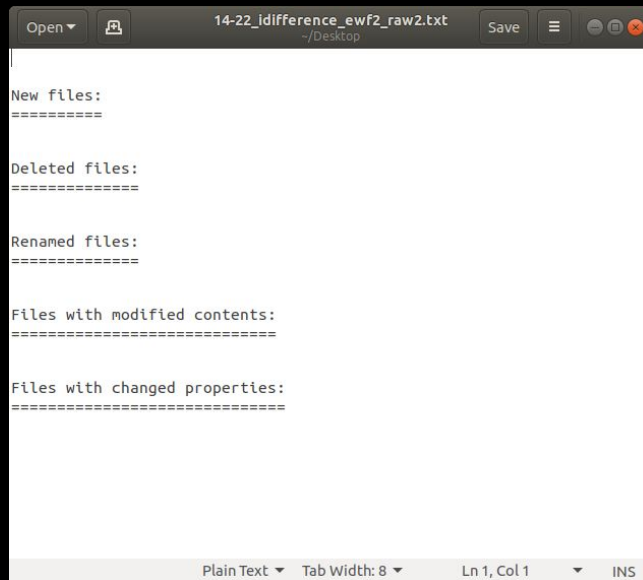
- Download python scripts from the dFXML github
- Run them on DFXML data using python

```
bcadmin@ubuntu:~$ python3.6 /dfxml-master/python/idifference2.py  
fiwalk_output1.xml fiwalk_output2.xml > differences.txt
```

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Lastly I want to point out a python script that you can download from the DFXML github page (located on the resources document at the bottom of the agenda). The script is called “idifference.” When this python script is run against 2 Fiwalk reports it creates a document that points out the specific difference between 2 disk images.

# Disk Image Differencing



The screenshot shows a text editor window titled "14-22\_idifference\_ewf2\_raw2.txt" with a menu bar containing "Open", "Save", and window control buttons. The main content area displays a differencing report with the following sections, each followed by a line of equals signs: "New files:", "Deleted files:", "Renamed files:", "Files with modified contents:", and "Files with changed properties:". The status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS".

```
Open  14-22_idifference_ewf2_raw2.txt  Save  [Window Control Icons]
|
New files:
=====

Deleted files:
=====

Renamed files:
=====

Files with modified contents:
=====

Files with changed properties:
=====

Plain Text  Tab Width: 8  Ln 1, Col 1  INS
```

NMAAHC Workshop,  
Caroline Gil and Eddy Colloton,  
2021

Here's an example output of a idifference report run on two identical disk images, so you can see how the report is organized.