



AFF: The Advanced Forensics Format



AFF®
Advanced Forensic Format

Basis Technology Corporation

P 617.386.2000
800.697.2062 (toll-free)
F 617.386.2020
W info@basistech.com
www.basistech.com



What is AFF?

- The Advanced Forensic Format (AFF) is an open and extensible file format.
 - Open: Published specification and open source software
 - Extensible: You can store any type of data and metadata
- AFF can store any type of forensic data
 - Disk images
 - Exported files
 - etc.
- Designed and developed by Simson Garfinkel and Basis Technology
 - Some of these slides were created by Simson



Today there are many disk image formats

- Raw (dd)
- Raw with external hash values (dcfldd)
- EnCase Forensic Evidence File (EWF/EVF)
- ILook Investigator IDIF, IRBF and IEIF

- DIBS USA Rapid Action Imaging Device (RAID)
- ProDiscover Image File Format
- PyFlag Seekable GZIP (sgzip)
- SafeBack
- Turner's Digital Evidence Bags (DEB)



Limitations of Popular Formats

Raw:

- No compression
 - Can compress entire file for transport, but not efficient during analysis
- No standardized metadata
 - Metadata must be stored in an external file
 - Format and structure will vary by investigator or tool being used
- No integrity checks
 - MD5/SHA-1 hash values are not included - unless dcfldd is used

EnCase:

- Proprietary
 - Design is not published and changes between versions
- Limited storage of metadata
 - Limited types of metadata can be stored



Benefits of AFF

- Stores arbitrary metadata
- Stores arbitrary types of data
- Wider range of compression levels
 - Can produce images $\frac{1}{4}$ to $\frac{1}{2}$ the size of EnCase images using LZMA compression
- No practical size limits (64-bit clean)
- Open Source



AFF Schema

- Everything can be defined as a name and value pair:
 - md5 = "6acf1e36e671b926b8d66f974042356e"
 - sha1 = "8231e522bfd63a9c7b832fdc1e6050c7ecf515bb"
 - page1 =
b82084a7e161d233b22b0ea7b048afada8d40138a2740d94be9bb0c206d5
7c616c8fa5e64734eaca31cd14e6cd1dbb5c1d73fc936df49243689476b5e
caacdee457307a7afae71d4737615954a35340ba610198717c827d27356a
79ec611ec13d5f814d1ca0e5fd9bedcc7d6b240390188367933c9cd90...
- The AFF Schema defines what names are used when storing disk images and exported files.
- You can also define your own name and value pairs:
 - gps = "42.394543, -71.144256"
 - classification = "secret"



Example Disk Image Pairs

imagesize	# bytes in image
page0	First data page
page0_md5	MD5 hash for page0
page0_md5_sig	MD5 digital signature for page0
page1	Second data page
pagesize	Data page size
md5	per-disk MD5 hash
sha1	per-disk SHA1 hash

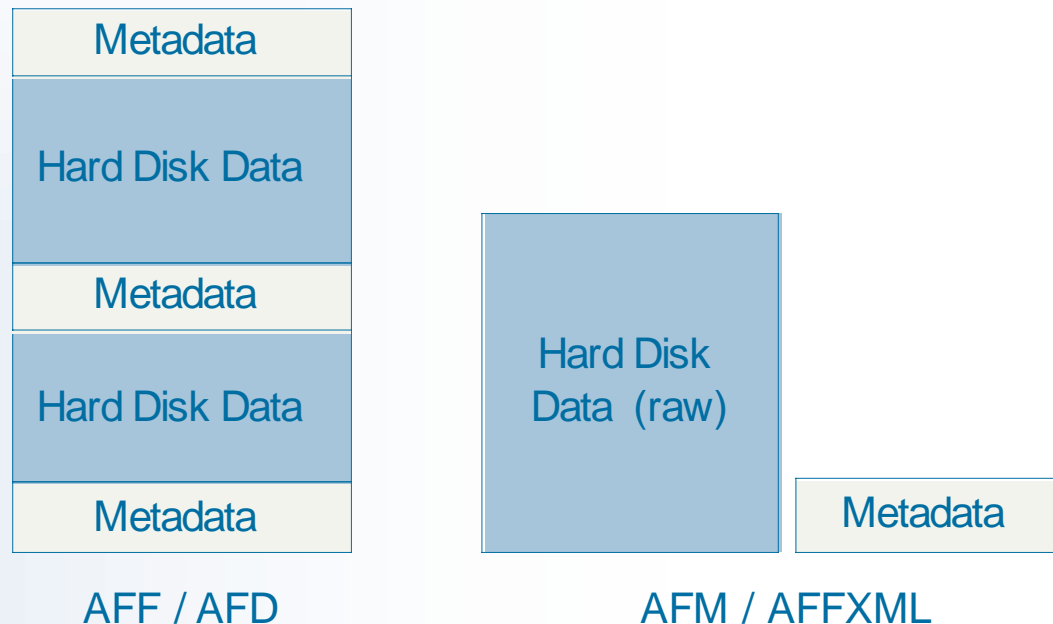
case_num	Case number
image_gid	Unique 128-bit identifier
device_sn	Drive serial number
imaging_date	Date imaged

... ..



AFF Storage Formats

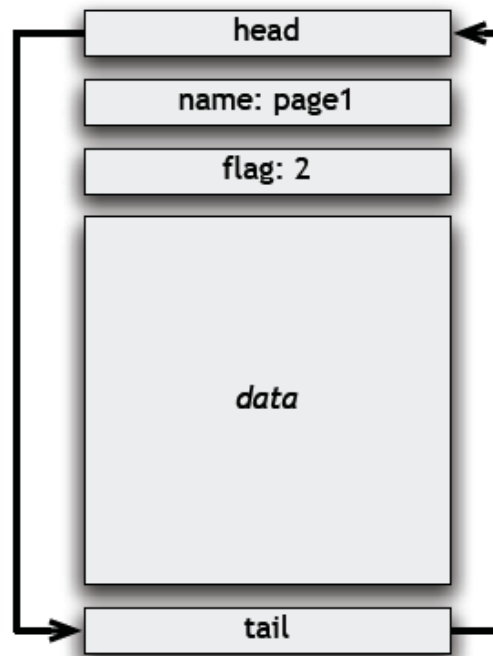
- There are four AFF storage formats for the schema
- AFF and AFD are embedded formats where the metadata is embedded with the forensic data.
- AFM and AFFXML store the metadata in a file separate from the forensic data (which is stored as a raw file).





AFF segment structure

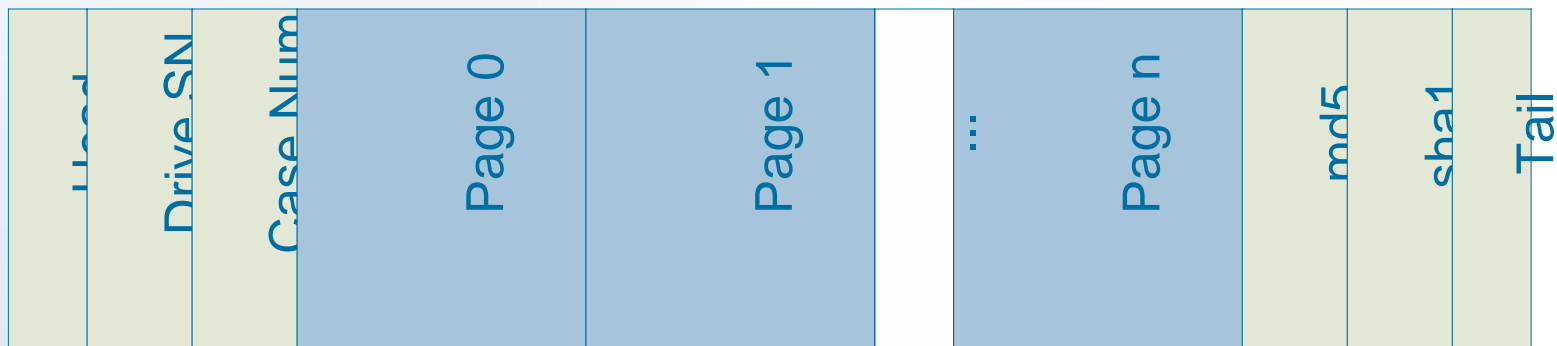
- AFF, AFD, and AFM use the same data structures to store name and value pairs (called segments)
 - AFFXML uses XML
- Each AFF segment has a head, name, flag, data, and tail.
 - The flag identifies if the data is compressed, etc.





Segment storage

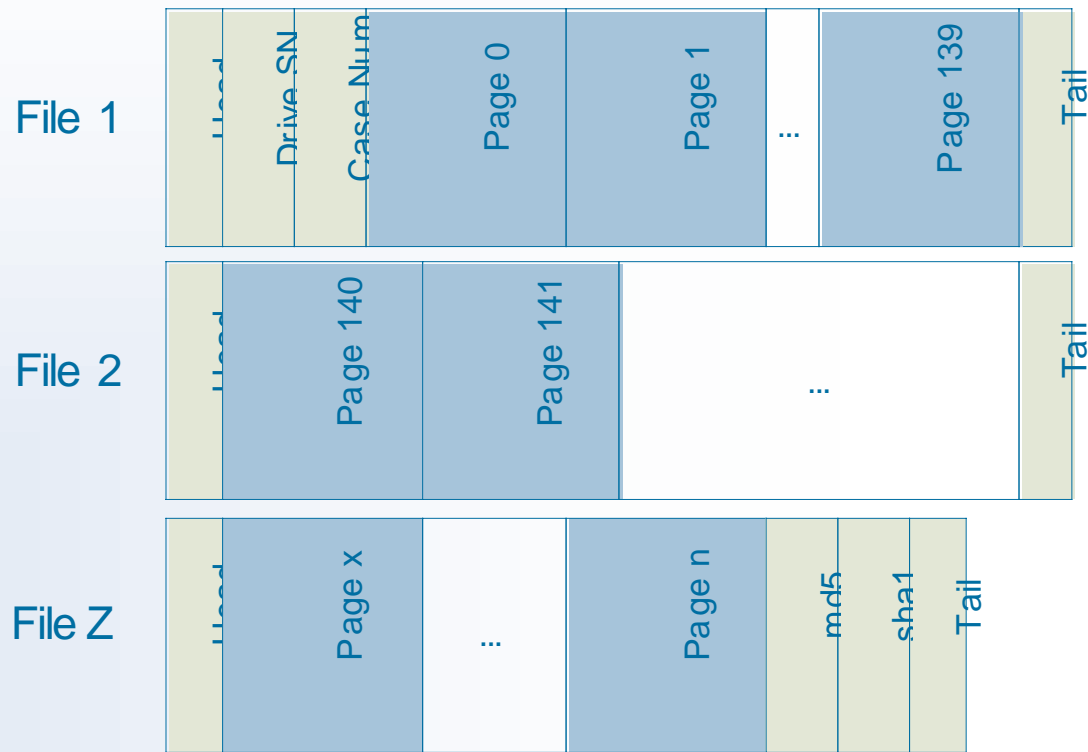
- Hard disk data is broken up into “pages”
 - Typically 16MB each
 - Each page is stored as a segment (name and value pair)
 - Each page can be compressed (or encrypted)
 - Each page can have its own cryptographic hash values
- AFF stores all segments in one large file





AFD Segment Storage

- AFD stores the segments in multiple AFF files in a single directory
 - 2GB files for FAT32
 - 650MB files for CD
- Each file knows where it fits in the set





AFM / AFFXML Segment Storage

- AFM stores the metadata segments in an AFF file and the hard disk data in a raw file
- AFFXML uses XML to store the metadata segments
- AFM and AFFXML allow you to use any analysis tool that supports raw formats



```
<affinfo name='/project/affs/551.aff'>
  <segsz arg='16777216' />
  <imagesize arg='2'>4290600960</imagesize>
  <md5>5M/87t7f6N4N5koMKb4QPA==</md5>
  <sha1>2XzW2A+kfKIGtmDloxiovqXjWcg=</sha1>
  <badsectors>AAAAAAAAAAAA=</badsectors>
  <blanksectors>AASGTQAAAAA=</blanksectors>
</affinfo>
```



AFF example

- Example: AFF file of a good disk (converted from raw).

```
% ainfo -a /project1/affs/47.aff
```

Segment	file	arg	data	data preview
=====	=====	=====	=====	=====
pagesize	585	16777216	0	
imagesize	616	0	8	= 1629831168 (64-bit value)
md5	657	0	16	.U[...'L....ng..
sha1	700	0	20	.8..1..).(.....kMcI
page0	748	1	10488912	x.... \u.7..4...^...E.....
page1	10489688	1	16398437	x...@..=6Mh.jz...A...
page2	26888153	1	16305513	x..}%E.v.T....'W.t.D...(
page3	43193694	1	16665964	x...@....E.A..8...05.x..'.
page4	59859686	1	16742440	x.DyS.0.....m..m{..m..m..m
page5	76602154	1	16726198	x...@..4.....,}....x.0.M..
page6	93328380	1	16768092	x...@..9..Vd. 3...NF..u..
...				



AFFLib and Tools

- AFFLib is an open source library for reading and writing AFF files
- AFF Tools are a collection of tools built using the library:
 - aimage Hard disk acquisition tool that writes AFF files
 - afinfo Print information about an AFF file
 - afcat Print data in AFF file to stdout
 - afcompare Compares two files
 - afconvert Convert AFF\$raw or AFF\$AFF
 - afxml Output AFF metadata as XML
 - aftest Validate the AFF Library
- These tools work under FreeBSD, Linux, OS X.
- Available at: <http://www.afflib.org>



AFF support

- AFFLIB stream abstraction makes it easy to add AFF support to existing programs.

```
AFFILE *af;  
  
af = af_open(filename, O_RDONLY, 0666);  
af_seek(af, offset, whence);  
af_read(af, buf, count);  
af_write(af, buf, count);  
af_close(af);
```

- Support has been added to The Sleuth Kit.



Storing metadata in AFF

- AFFLIB put/get routines make it easy for programs to read and write metadata.

```
AFFILE *af;  
char *name;  
int arg;  
  
int af_get_seg(af,name,arg,data,datalen);  
int af_get_next_seg(af,name,arg,data,datalen);  
int af_rewind_seg(af);  
int af_update_seg(af,name,arg,value,vallen,add);  
int af_del_seg(af,name);
```




Almage

- Almage is the Advanced Disk Imager tool in AFFLIB.

```
Terminal — ssh — 80x24
IMAGING Thu Nov 10 10:53:27 2005
Source device: /dev/ad2 AFF Output: /project/junk.aff
Model #: QUANTUM FIREBALL ST3.2A
firmware: A0F.0000 Sector size: 512 bytes
S/N: 153718340531 Total sectors:6,306,048

[-----]
Currently reading sector: 97,792 (512 sectors at once)
Sectors read: 98,304 ( 1.56%) # blank: 1,026

Time spent reading: 00:00:05 Estimated total time left: 00:21:34
Total bytes read: 50,331,648

Compressed bytes written: 25,735,396
Time spent compressing: 00:00:09
Overall compression ratio: 48.87% (0% is none; 100% is perfect)
Free space on 192.168.1.1:/project: 68,937 MB (12.44%)
```



Almage features

- Writes AFF and/or raw (dd)
- Automatically reads drive SN and other drive metadata
- Compress during imaging or afterwards
- MACs & signatures for segments and/or image
- Intelligent error recovery
- Bad Block Markup
 - Bad sectors are stored as “BAD SECTOR” and random data.
- Works on Linux and FreeBSD



Bad sector example

- Example: AFF file of a bad disk (acquired with aimage):

Segment	file offset	arg	data length	data preview
=====	=====	=====	=====	=====
badflag	8	0	512	BAD SECTOR.%.p.l.r.P1W(e6....
badsectors	551	2	8	= 6144 (64-bit value)
pagesize	593	16777216	0	
imagesize	624	2	8	= 853622784 (64-bit value)
device_model	665	0	11	WDC AC2850F
device_sn	712	0	15	WD-WT3130627676
device_firmware	760	0	8	28.25E40
cylinders	807	1654	0	
heads	840	16	0	
sectors/track	869	63	0	
device_capabilities	906	0	758	.Protocol ATA/ATAP
imaging_commandline	1707	0	37	aimage -z ata1 /project3/affs/9
imaging_device	1787	0	8	/dev/ad2
image_gid	1833	0	16	..-..I2s6..0...P
imaging_date	1882	0	19	2006-01-06 11:45:14
page0	2028	1	81994	x...N...a.iT:Sd....R.bR.T ..
blanksectors	84050	2	8	= 11 (64-bit value)
page50	84094	1	82003	x...N...P.IT:Sd..I..... ..\2



Future: Sector Annotations

- Motivation:
 - Many acquisition tools write all zeros when bad sectors are encountered
 - Makes it difficult to know a sector is all zeros because it was wiped or it was bad...
 - Data on a disk may need to be redacted before it is given to someone else:
 - *Clearance levels*
 - *Contraband*
- AFF will maintain annotations for every sector:
 - Not read (did not get a chance to read before acquisition was aborted)
 - Read
 - Bad
 - Redacted



Future: AFF Consortium

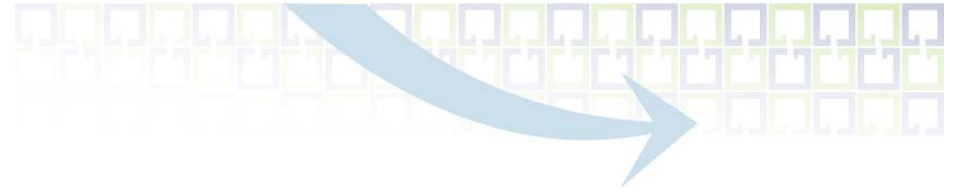
- Goal is to involve others with the design and development
- Consortium membership will help to define the AFF schema and standards
- E-mail info@basistech.com for more information





Summary

- AFF is an open and extensible format for storing digital evidence
 - Arbitrary metadata can be stored
 - Several storage formats exist for different needs
- LZMA compression allows smaller disk images to be created for transport
- AFM and AFFXML allow acquired data to be used with tools that support raw images
- AFFLIB allows AFF to be easily incorporated into existing tools



Questions?

www.afflib.org

www.basistech.com/digital-forensics/aff.html