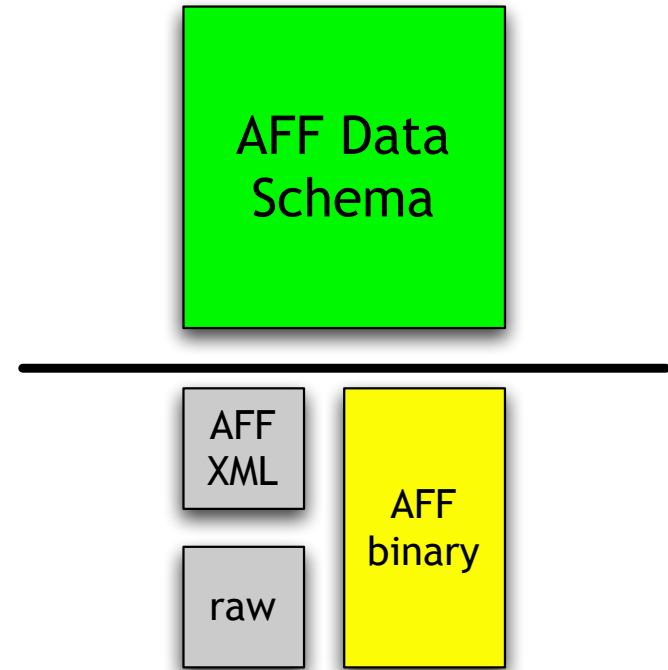


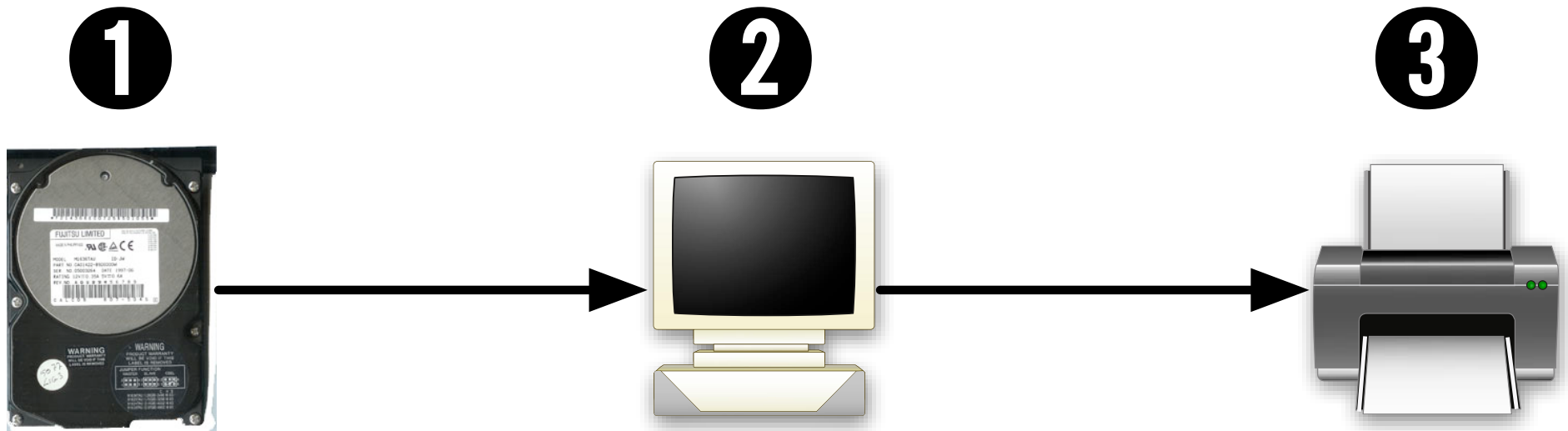
The Advanced Forensics Format Library and Tools

Simson L. Garfinkel
David J. Malan
Karl-Alexander Dubec
Christopher C. Stevens
Cecile Pham



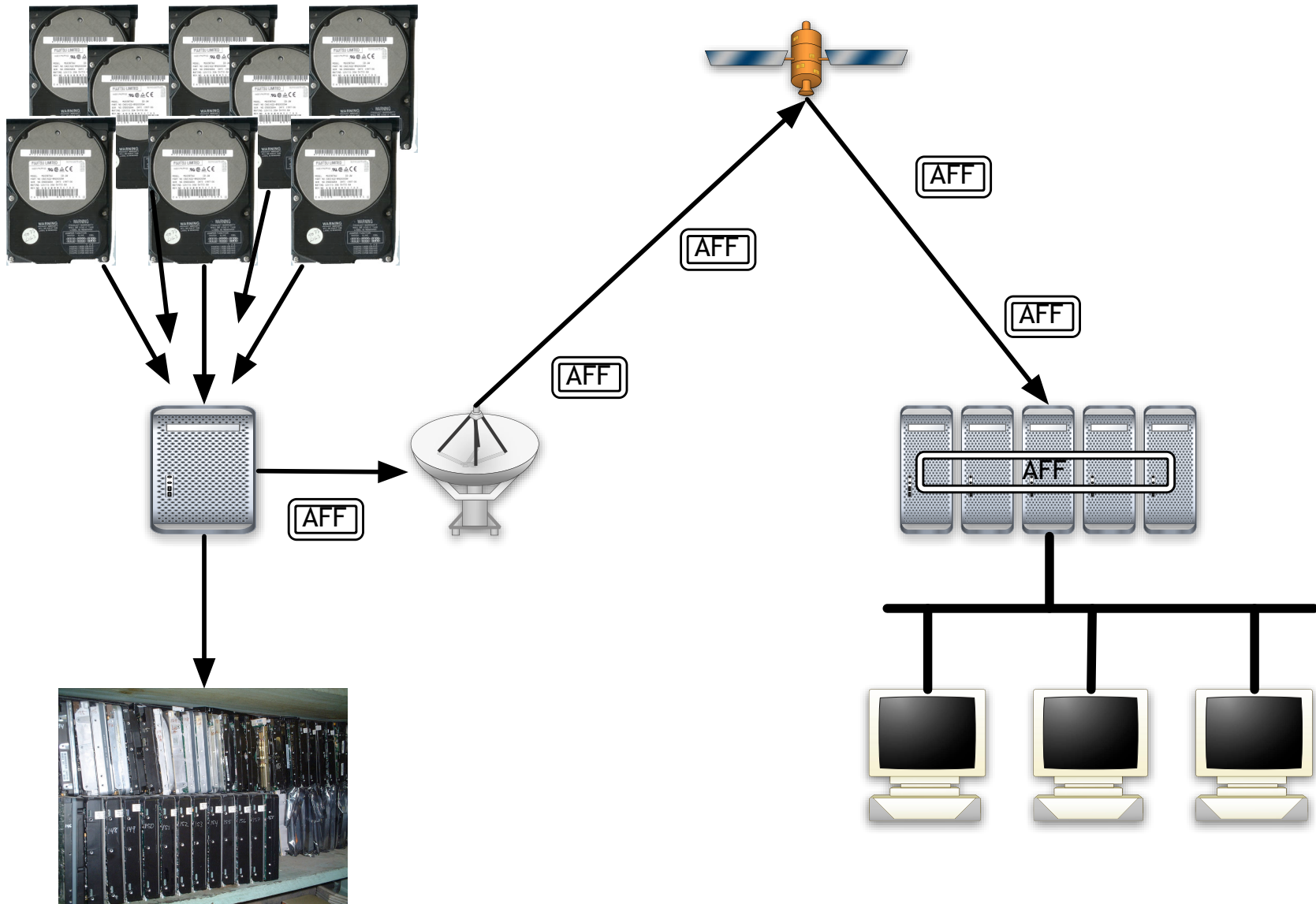
**Second Annual IFIP WG 11.9
International Conference on Digital Forensics
National Center for Forensic Science, Orlando, FL
29 January 2006**

Today's forensic tools and file formats are designed to analyze a single drive at a time.



This is no longer adequate for today's forensics challenge.

**AFF is a simple, compact, self-describing,
and open way to capture and move around disk images.**



There are many disk imaging formats.

Raw (dd)

EnCase Forensic Evidence File (EVF)

DIBS USA Rapid Action Imaging Device (RAID)

ILook Investigator IDIF, IRBF and IEIF

ProDiscover Image File Format

PyFlag Seekable GZIP (sgzip)

SafeBack

Turner's Digital Evidence Bags (DEB)

AFF is superior to existing disk imaging formats.

Raw:

- No compression
- No metadata
- No integrity checks

EnCase EVF:

- Proprietary
- Limited storage of metadata
- 2GB file size limit

AFF:

- No filesize limits
(64-bit clean)
- Stores arbitrary metadata
- Tunable compression
- Extensible
- Indicates “bad” sectors
- Smaller images than EVF

AFF consists of four parts:

AFF specification — Defines schema and data storage.

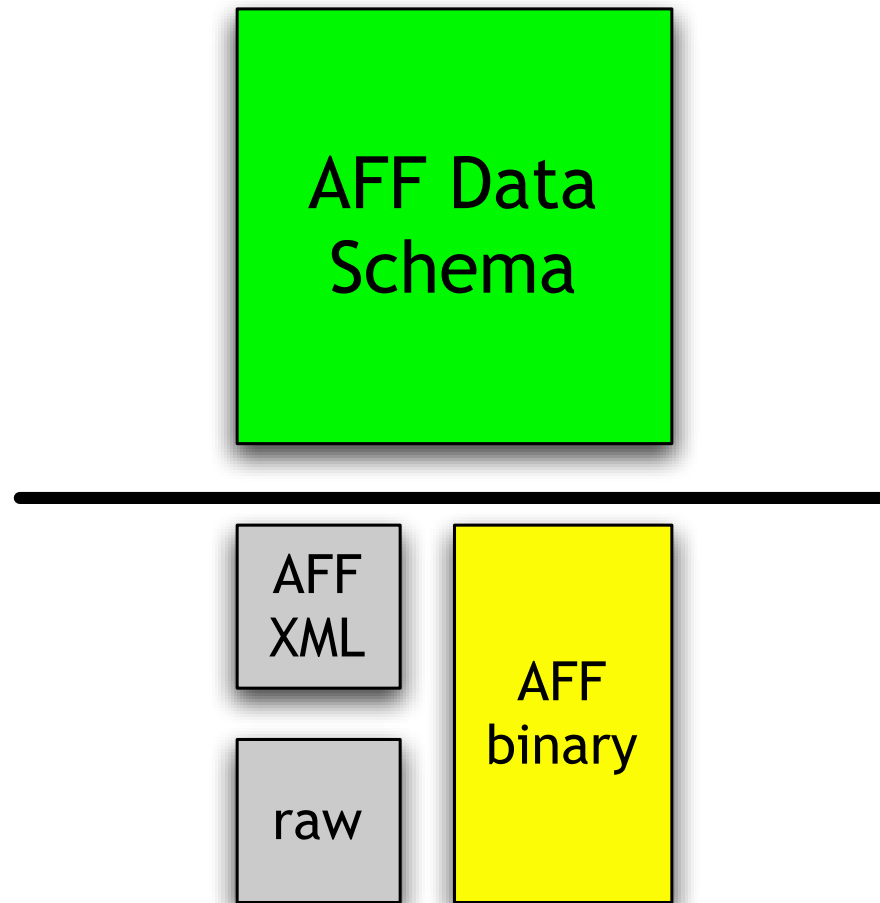
AFFLIB — Open Source C/C++ implementation.

AFF Tools — Utilities for working with AFF files.

Almage — The Advanced disk imager.

All can be downloaded today from <http://www.afflib.org>.

The AFF specification is divided into two layers.

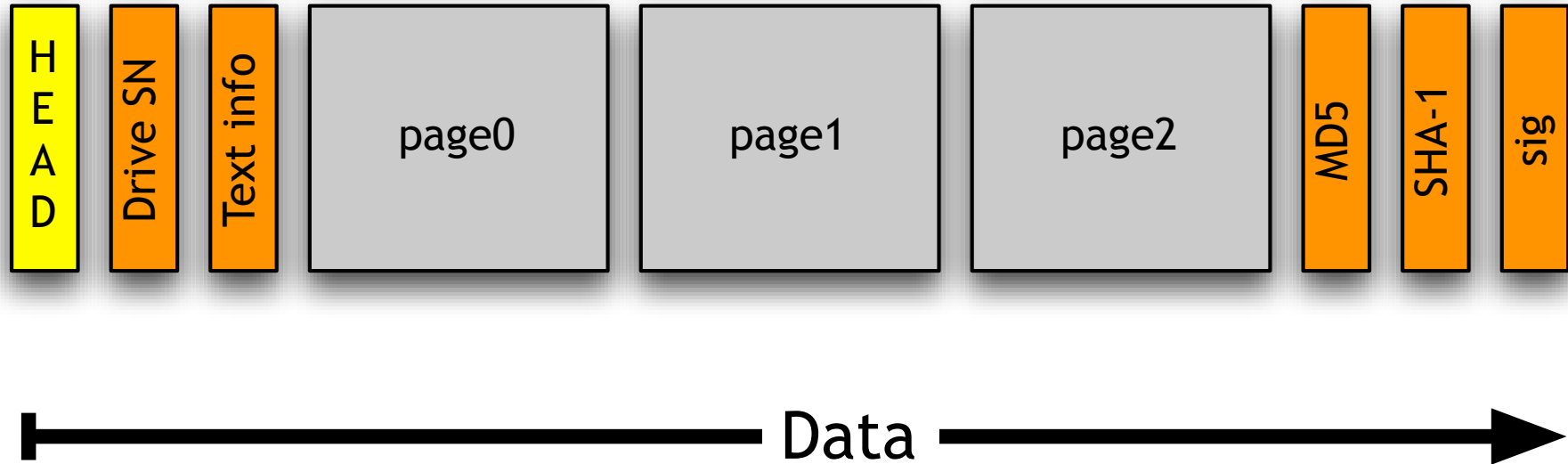


The AFF schema defines name/value pairs for disk imaging.

imagesize	# bytes in image
page0	First data page
page0_md5	per-page MD5 hash
page0_md5_sig	per-page digital signature
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
...	...

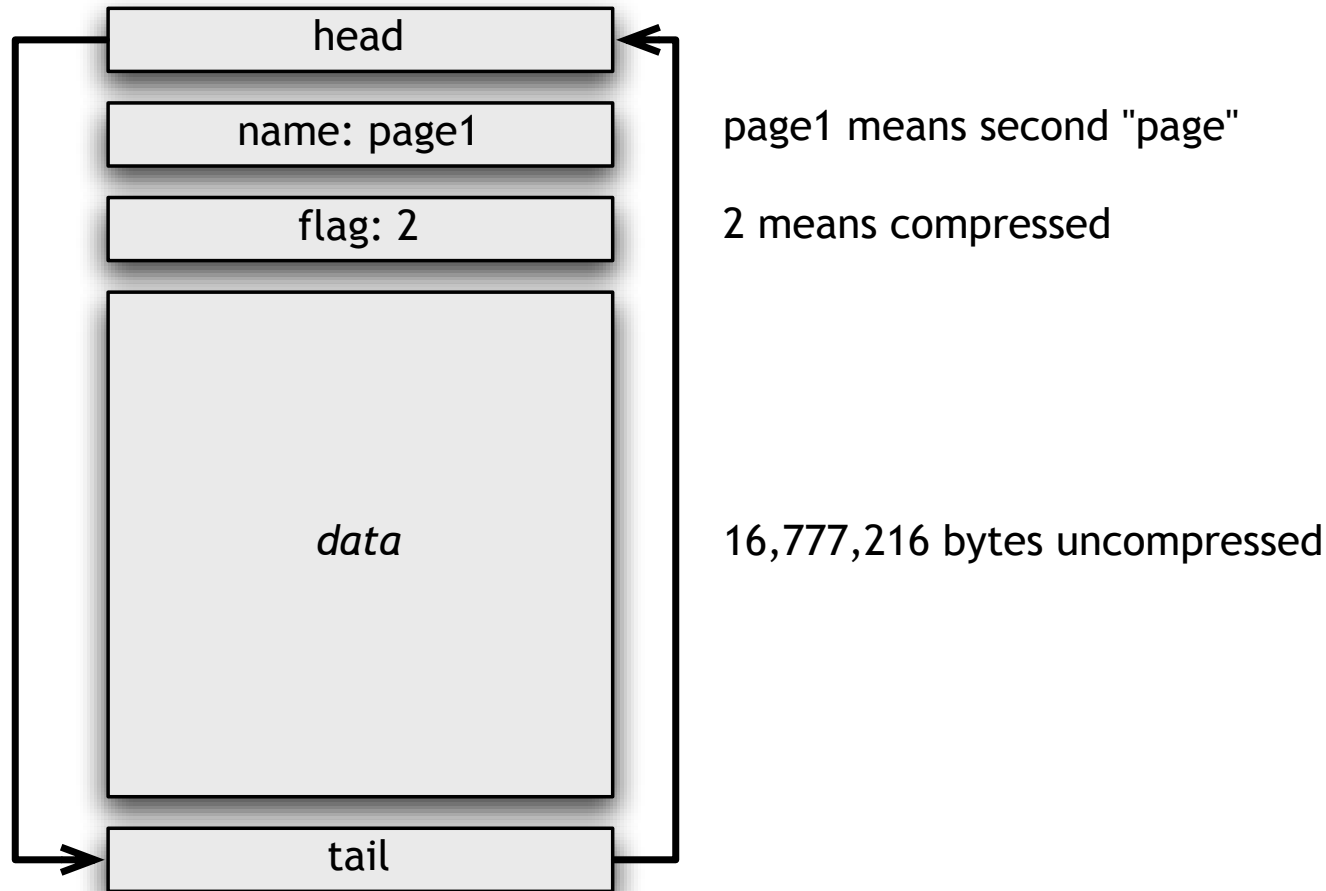
New segment names can be added as need arises.

AFF stores named segments in one or more AFF files.



**Put the entire disk image in one file,
or automatically acquire into multiple files.**

Each AFF segment has a head, name, flag, data, and tail.



flag is 32 bits.

data can be $0 - 2^{32} - 1$ bytes.

Head and tail allow for rapid seeking through file.

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

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

	file		data	
Segment	offset	arg	length	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.O.M..
page6	93328380	1	16768092	x...@..9..Vd. 3...NF..u..
...				

AFFXML is an alternative representation for the AFF schema.

AFF
XML

raw

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

Use to annotate raw files.

Move data between AFF and another programs.

AFF TOOLS make it easy to work with AFF files.

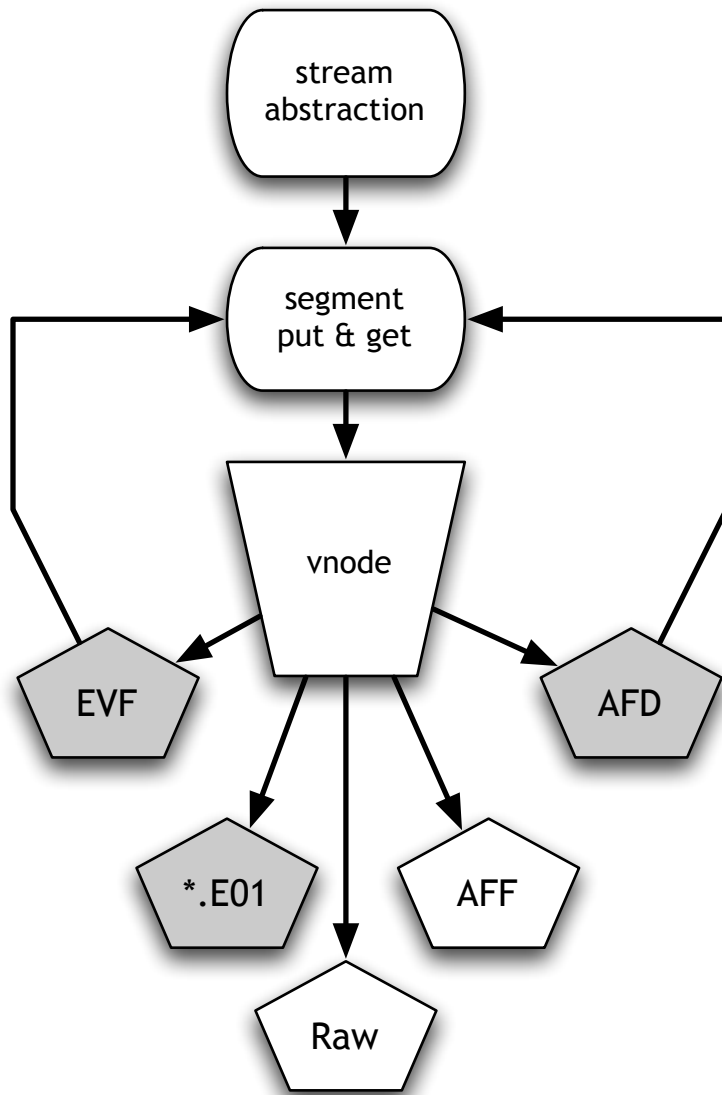
afcat	Pipe AFF to <i>stdout</i>
afcompare	Compares two files
afconvert	Convert AFF $\leftrightarrow$ raw or AFF $\leftrightarrow$ AFF
afinfo	Print information about an AFF
afxml	Output AFF metadata as XML

aftest	Validate the AFF Library
--------	--------------------------

aimage	The Advanced Disk Imager
--------	--------------------------

These tools work under FreeBSD, Linux, and Windows.*

AFFLIB is an Open Source multi-format implementation.



Layered implementation.

Supports AFF today.

Will soon support EVF.

More formats can be easily added.

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 Brian Carrier's Sleuth Kit.

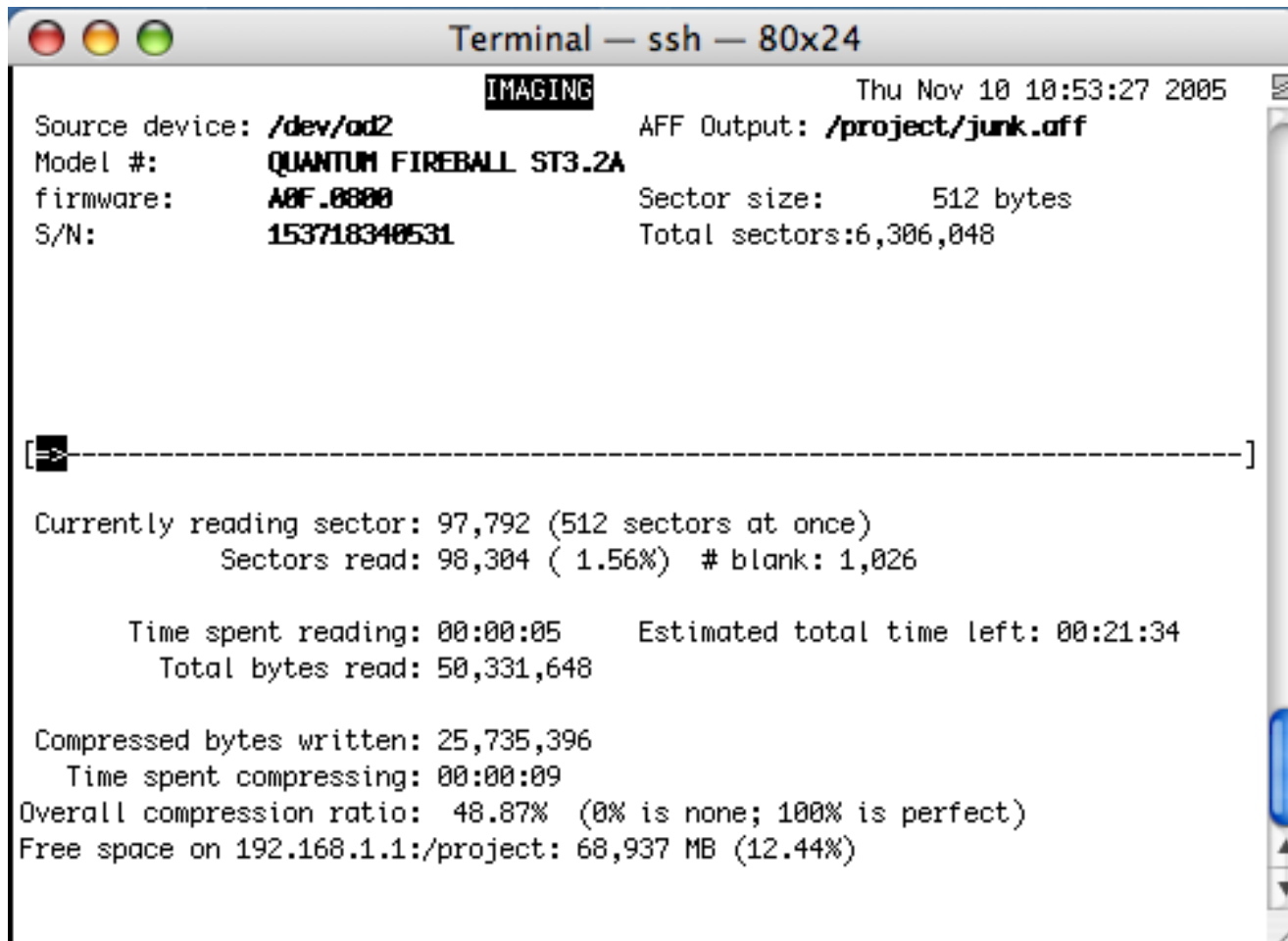
AFFLIB put/get routines make it easy to read and write metadata.

```
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);
```


Adding new formats requires implementing just 8 functions.

<code>x_identify()</code>	— Is this my file?
<code>x_open()</code>	— Open image file
<code>x_close()</code>	— Close image file
<code>x_stat()</code>	— Return image and page sizes
<code>x_get_seg()</code>	— Get a segment
<code>x_get_next_seg()</code>	— Get the “next” segment
<code>x_rewind_seg()</code>	— Rewinds segment pointer
<code>x_update_seg()</code>	— Writes a segment back
<code>x_del_seg()</code>	— Delete a segment

Almage is the Advanced Disk Imager.

A screenshot of a terminal window titled "Terminal — ssh — 80x24". The window shows the output of the Almage disk imaging tool. At the top, the word "IMAGING" is displayed in a black box. The date and time "Thu Nov 10 10:53:27 2005" are shown in the top right. The main output displays source device information, disk model, and progress statistics. A dashed line separates the header information from the progress section. The progress section shows the current sector being read, sectors read so far, and the estimated time left. The bottom section shows compression statistics and free space on the target drive.

```
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:

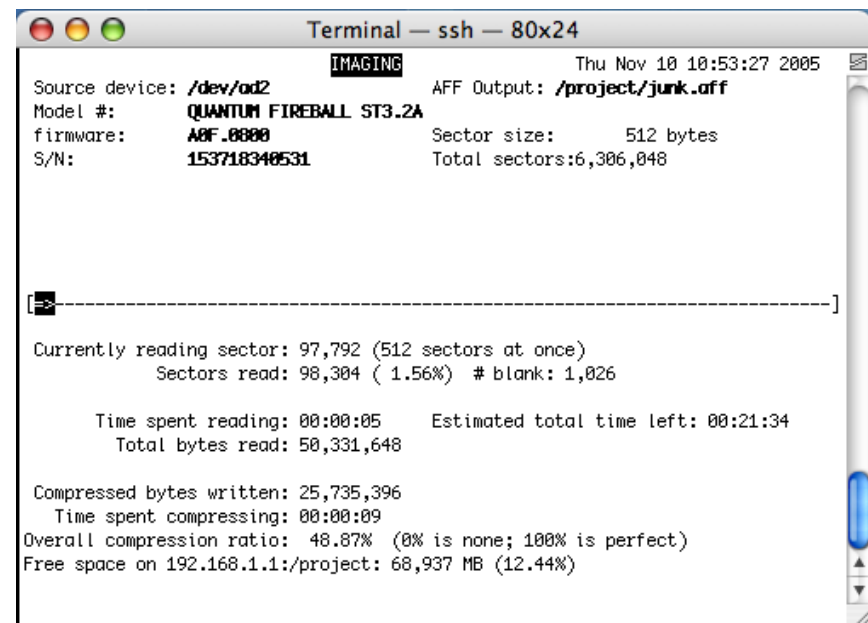
Writes AFF and/or raw (dd)

Automatically reads drive SN and other drive metadata

Compress during imaging or afterwards

MACs & signatures for segments and/or image

Intelligent error recovery

A terminal window titled "Terminal — ssh — 80x24" showing the output of the Almage imaging process. The window has a standard macOS-style title bar with red, yellow, and green window control buttons. The text inside the terminal is as follows:

```
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%)
```

The 512-byte “badflag” indicates data could not be read.

“BAD SECTOR” NULL {501 bytes of random data}

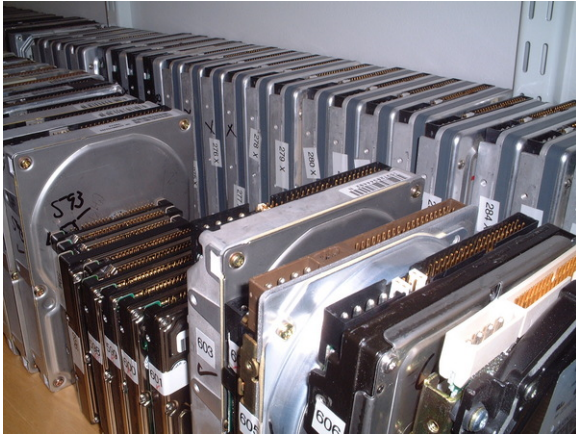
```
BAD SECTOR^@R<8A><8D><FD><F4><F6><A7>0^F<94>' 0<AC><E6> aq<F0><9B>|<C8>|<C8><93>A<80>
<AF>Is^X[<84><EC>"0<AD>]b<AC>0<FA>0<F6>f<81>N<D2>^Z^M<C2><A5><97><EB><A7><85>Z<DF><A2>
<FE>p0<9F><91><8C>S<EC>P<C3>|<BF><D0><D4>^W3^D#<B3>zDp4g<A3><8E><93><BE><CA>5'<84>Qw
<C1><FF><EB>^Af^G<A0>0<EF><FB>X^CZ<FA><E5>i^A[[<A6><AE>^P;=?<ED><FC>1^Y<F6><9D><EC>^X
<8A><CE><F0>|<CC><BB><D7>^T<EF><E2><AF>eQJ<FE><B2><BD>8^O<D2>3<FE><D3>W3<AB><9E>Mm?S!
)<BA>^D<BB><B5>^A<84><A6><BF><95>eDH^]<BD><8D><D9><87><90>^DY^R<D9>|<F3>1q2<D8><FA>J
<BB>^E<EA>*<F9><E0>q<B1><B4><F1><E6><8D>$',<F7>^W<DE><FA>v<AA>^];S^\\<9A><F8><DB>"em
<82>|<C3>E|<LS><D5><AE><DD><9F><D0>^?<C4><9E>^F<F7><EF><ESC><9E>+7<A4>pM<E3><24>W<8D><A1>
#<C0><FA>[<8A><CF><88><B9><F6><DE><CB>R<FF>{x|0AT^AP<z<D0>^BAY<E4>
<D3><81><80><CB><F5>^Y<DC><98><B5>9<AA><BE>
<89><AB><8D><FF><F7><AA>J<B2><B4>|<99>|<EF><B2><FF><CD><F7>6<E1><A4><CF>|<B7><D2><BE>b
[<C7>^Zv<C4>|<C1>^D<98>q^Vl#<E4><F9>^\\<A5>^?<BF><C9><CF><A5><F1><90><B8><9E><CB><98>
<B9><EF><A9>^\\<91>J^Z3<AC><AD><F3>q~<98><D9><A8>A<8B><C4><A6>fR<EA>^P?^X<CA>D+X<A4><8C>J
,<^>]<FA><D9>7B<ED>lh<C1><E2>q0<C3>?<C5>l<C2>jgm.<92><AE><CE><E9>4<BC><E2><B0><CA>3<E2>5
o^<B6><8F>|<CB><96><F2><83><BD><FF><F7><ESC><E9>^R[9e3fIn<98>7<B1><A1>x<94><BB>|<9B>
<C1><9A><F6><EB><94>W<E1>lv<DB><CA>?Eh<D5>b
\\n^A<99>|<^N><E9>j<B6>|<9B>0b<A8><93>^GS1<E1><A5><D6><D5><DF>pR
```

Unique for each AFF file.
Stored in “badflag” segment.
Obvious upon inspection.

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

Almage has been used to image more than 800 hard drives.



Automatic capture of metadata is exceedingly important!

Advanced Forensics Format 1.0 in summary:

- AFF Specification: Simple, self-describing, robust
- AFF Library: open source AFF implementation
- AFF Tools: for working with AFF files
- Almage: easy-to-use imager

Download from <http://www.afflib.org>