DIGITAL DATA CARVING

CHAPTER 10

- Introduction
- Windows, MAC, Linux File Format
- File System
- File & Magic Number
- Carving Concept
- Case Study: Recovering Image File.

Introduction

- File Carving, or sometimes simply Carving, is the practice of searching an input for files or other kinds of objects based on content, rather than on metadata.
- File carving is a powerful tool for recovering files and fragments of files when directory entries are corrupt or missing, as may be the case with old files that have been deleted or when performing an analysis on damaged media
- Most file carvers operate by looking for file headers and/or footers, and then "carving out" the blocks between these two boundaries. Semantic Carving performs carving based on an analysis of the contents of the proposed files.
- File carving should be done on a disk image, rather than on the original disk

KNOWING FILE FORMAT FOR WIN/MAC/UNIX/LINUX

Understanding Disk

- Composed of one or more platters
- Elements of a disk:
  - Geometry
  - Head
  - Tracks
  - Cylinders
  - Sectors

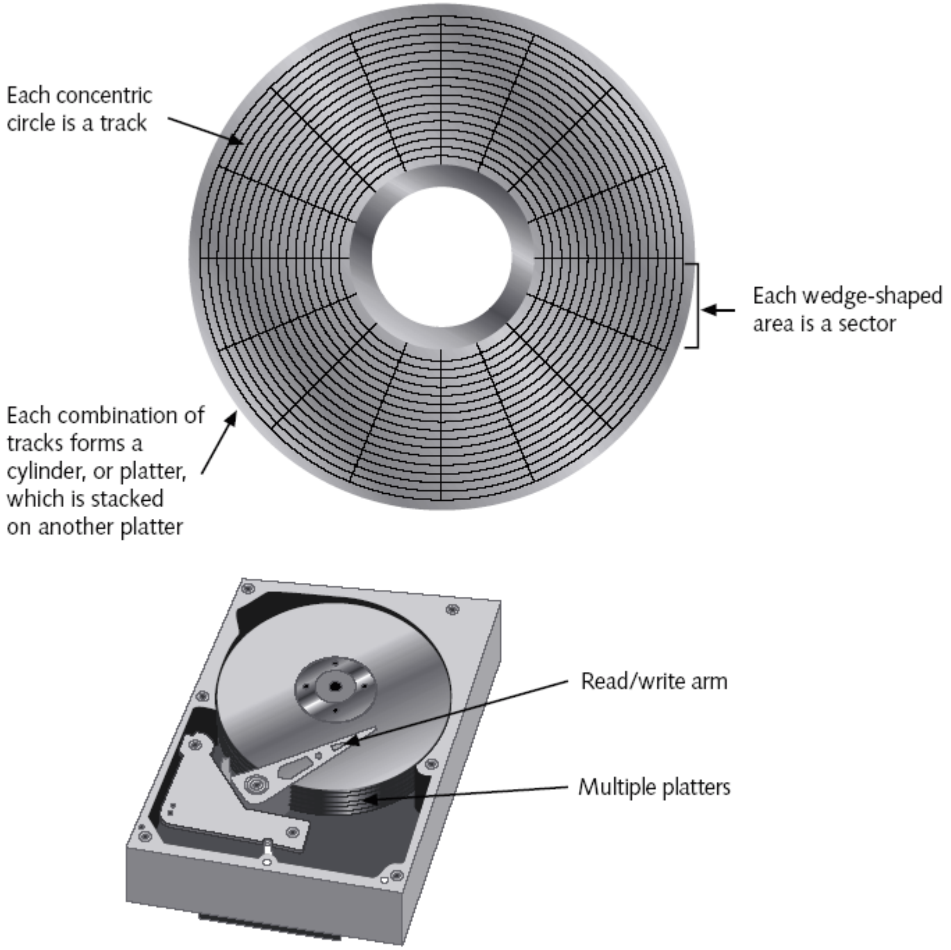

Each concentric circle is a track
Each wedge-shaped area is a sector
Each combination of tracks forms a cylinder, or platter, which is stacked on another platter
Read/write arm
Multiple platters

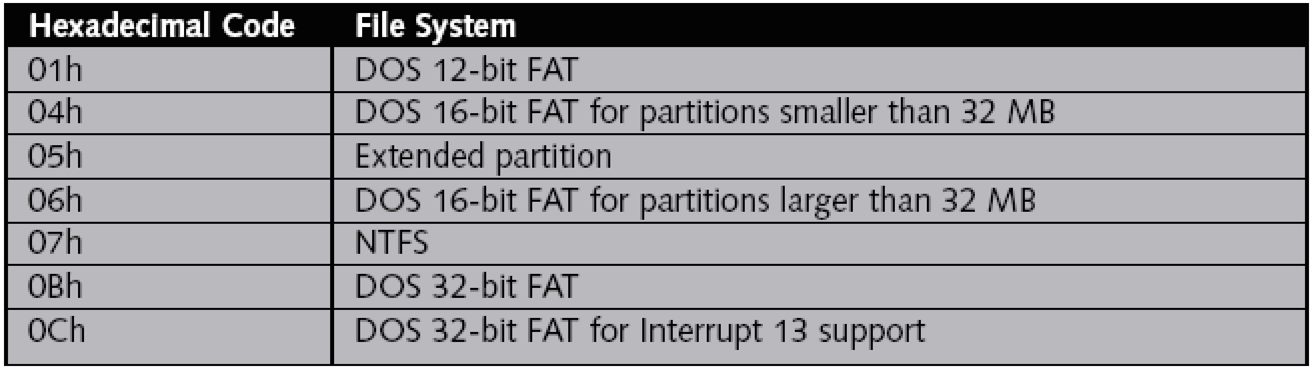
Figure 7-2 Disk drive structure

- Cylinder, head, sector (CHS) calculation
  - 512 bytes per sector
  - Tracks contain sectors
  - Number of bytes on a disk
    - Cylinders (platters) x Heads (tracks) x sectors
- First track is track 0
- Zoned bit recording (ZBR)
  - Platter’s inner tracks are smaller than outer tracks
  - Group tracks by zone
- Track density
  - Space between each track
- Areal density
  - Number of bits on one square inch of a platter

Exploring Microsoft File Structures

- Need to understand
  - FAT
  - NTFS
- Sectors are grouped on clusters
  - Storage allocation units of at least 512 bytes
  - Minimize read and write overhead
- Clusters are referred to as logical addresses
- Sectors are referred to as physical addresses

Table 7-1 Hexadecimal Codes in the Partition Table



Hexadecimal Code	File System
01h	DOS 12-bit FAT
04h	DOS 16-bit FAT for partitions smaller than 32 MB
05h	Extended partition
06h	DOS 16-bit FAT for partitions larger than 32 MB
07h	NTFS
0Bh	DOS 32-bit FAT
0Ch	DOS 32-bit FAT for Interrupt 13 support

Master Boot Record

- Stores information about partitions
  - Location
  - Size
  - Others
- Software can replace master boot record (MBR)
  - PartitionMagic
  - Can interfere with forensics tasks
  - Use more than one tool

Examining FAT Disks

- FAT was originally developed for floppy disks
  - Filenames, directory names, date and time stamps, starting cluster, attributes
- Typically written to the outermost track
- Evolution
  - FAT12
  - FAT16
  - FAT32

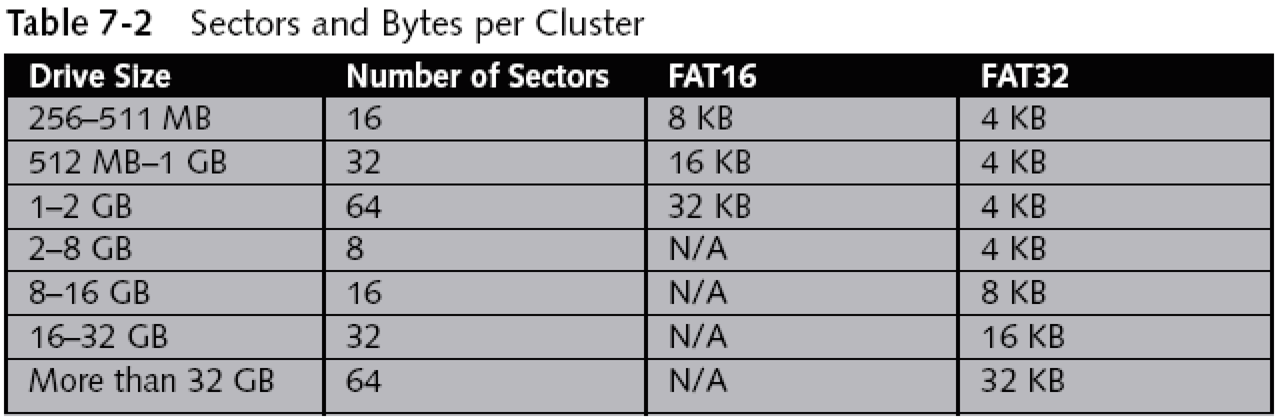


Table 7-2 Sectors and Bytes per Cluster

Drive Size	Number of Sectors	FAT16	FAT32
256–511 MB	16	8 KB	4 KB
512 MB–1 GB	32	16 KB	4 KB
1–2 GB	64	32 KB	4 KB
2–8 GB	8	N/A	4 KB
8–16 GB	16	N/A	8 KB
16–32 GB	32	N/A	16 KB
More than 32 GB	64	N/A	32 KB

Deleting file in Fat

- Filename in FAT database starts with HEX E5
- FAT chain for that file is set to zero
- Free disk space is incremented
- Actual data remains on disk
- Can be recovered with computer forensics tools

Examining NTFS Disks

- First introduced with Windows NT
- Spin off HPFS
- Provides improvements over FAT file systems
  - Stores more information about a file
- Microsoft’s move toward a journaling file system
  - Keep track of transactions
- Partition Boot Sector starts at sector 0
- Master File Table (MFT)
  - First file on disk
  - Contains information about all files on disk (meta-data)
- Reduces slack space
- NTFS uses Unicode
  - UTF-8, UTF-16, UTF-32

- Deleting file is similar to FAT
- NTFS is more efficient than FAT
  - Reclaiming deleted space
  - Deleted files are overwritten more quickly
- Hexadecimal codes identify OSs and file types
- NTFS uses inodes to link file attribute records
  - Resident and nonresident
- NTFS compressed files
- NTFS encrypted files (EFS)

Understanding the Macintosh File Structure and Boot Process

- Mac OS X version 10.4
  - Darwin core
  - **BSD UNIX** application layer
- **Hierarchical File System (HFS)**
  - Files stored in nested directories (folders)
- **Extended Format File System (HFS+)**
  - Introduced with Mac OS 8.1
  - Supports smaller file sizes on larger volumes, resulting in more efficient disk use

- **File Manager** utility
  - Reading, writing, and storing data to physical media
- **Finder**
  - Keeps track of files and maintain users' desktops
- In older Mac OSs, a file consists of two parts:
  - **Data fork** and **resource fork**
  - Stores file metadata and application information

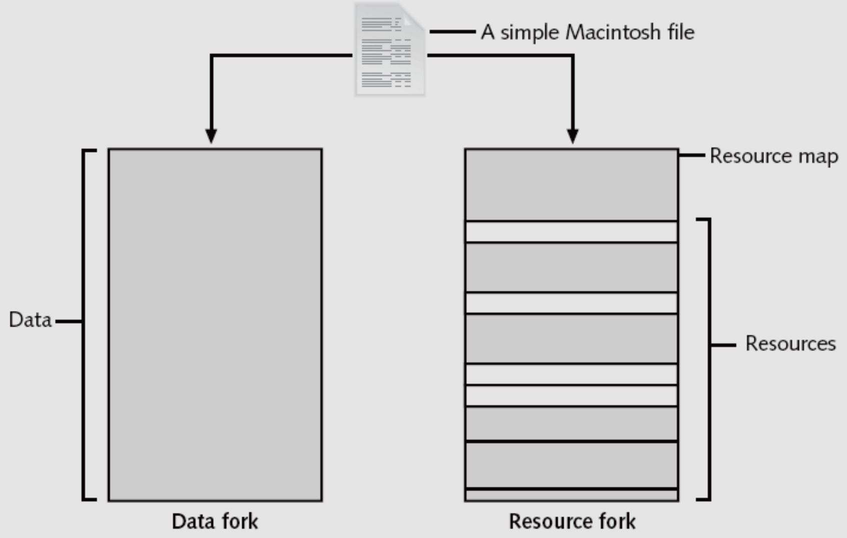


Figure 8-1 The resource fork and data fork in a Mac OS file

Understanding Macintosh OS 9 Volumes

- A volume is any storage medium used to store files
  - Can be all or part of a hard disk
  - On a floppy disk is always the entire disk
- **Allocation** and **logical blocks**
  - Logical blocks cannot exceed 512 bytes
  - Allocation blocks are a set of consecutive logical blocks

Understanding Macintosh OS 9 Volumes (continued)

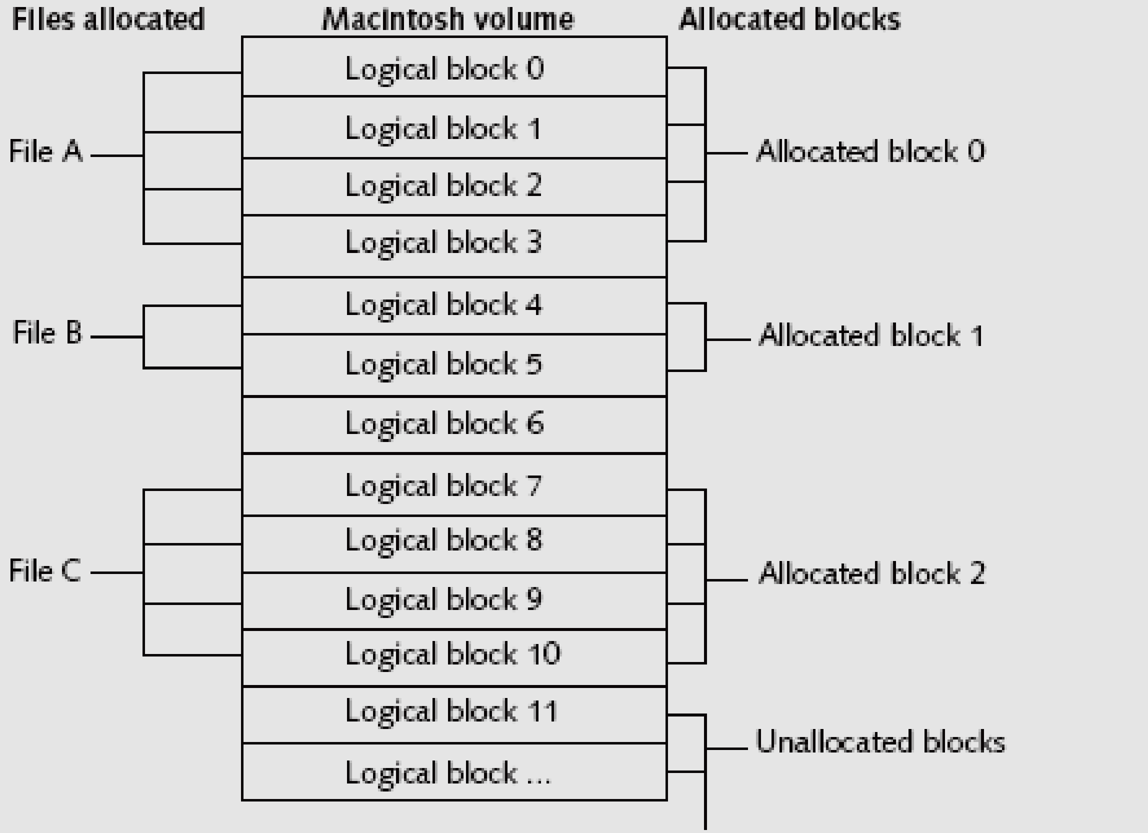


Figure 8-2 Logical and allocation block structures

Understanding Macintosh OS 9 Volumes (continued)

- Two EOF descriptors
  - **Logical EOF**
    - Actual size of the file
  - **Physical EOF**
    - The number of allocation blocks for that file
- **Clumps**
  - Groups of contiguous allocation blocks
  - Reduce fragmentation

Understanding Macintosh OS 9 Volumes (continued)

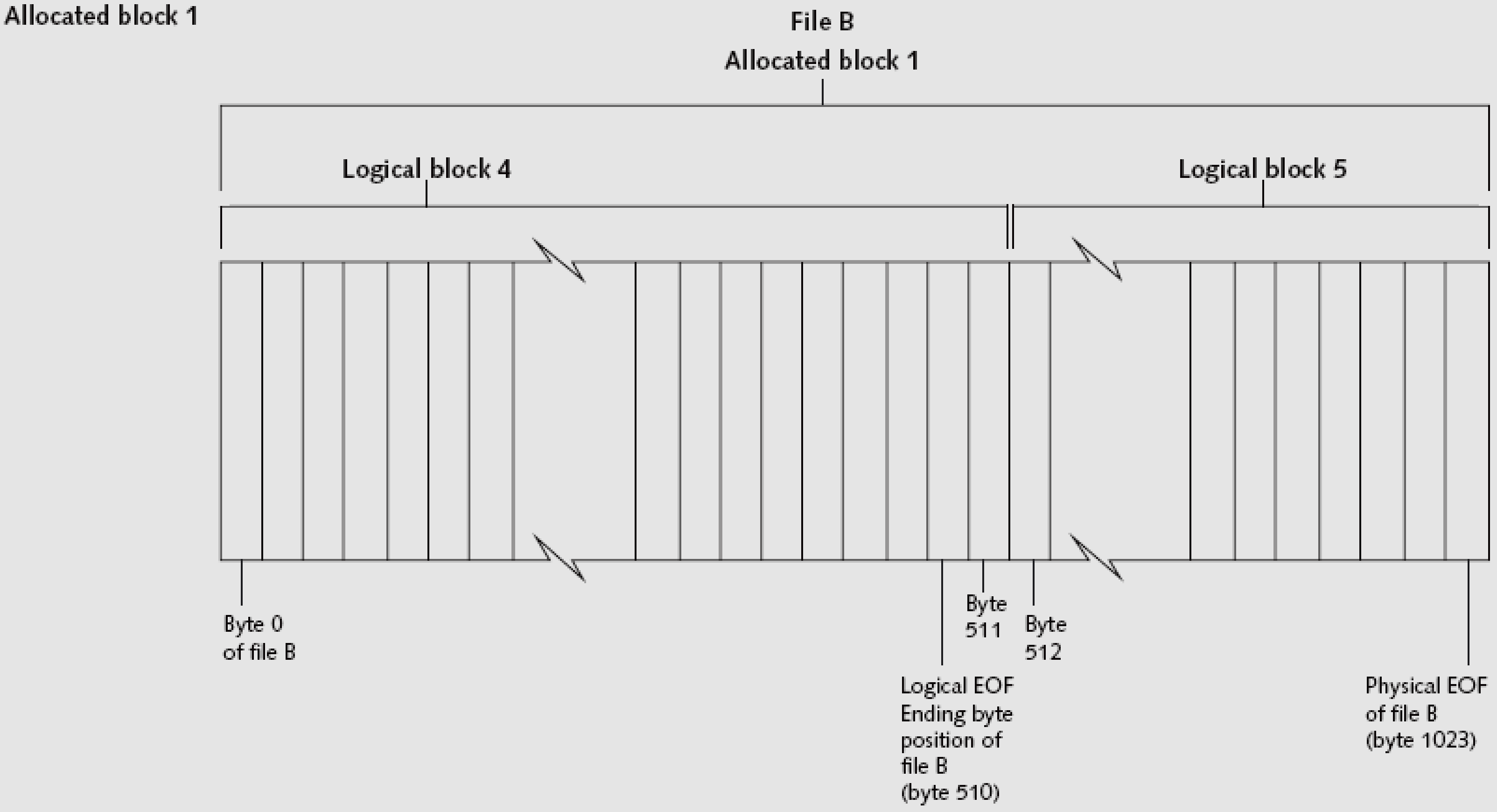


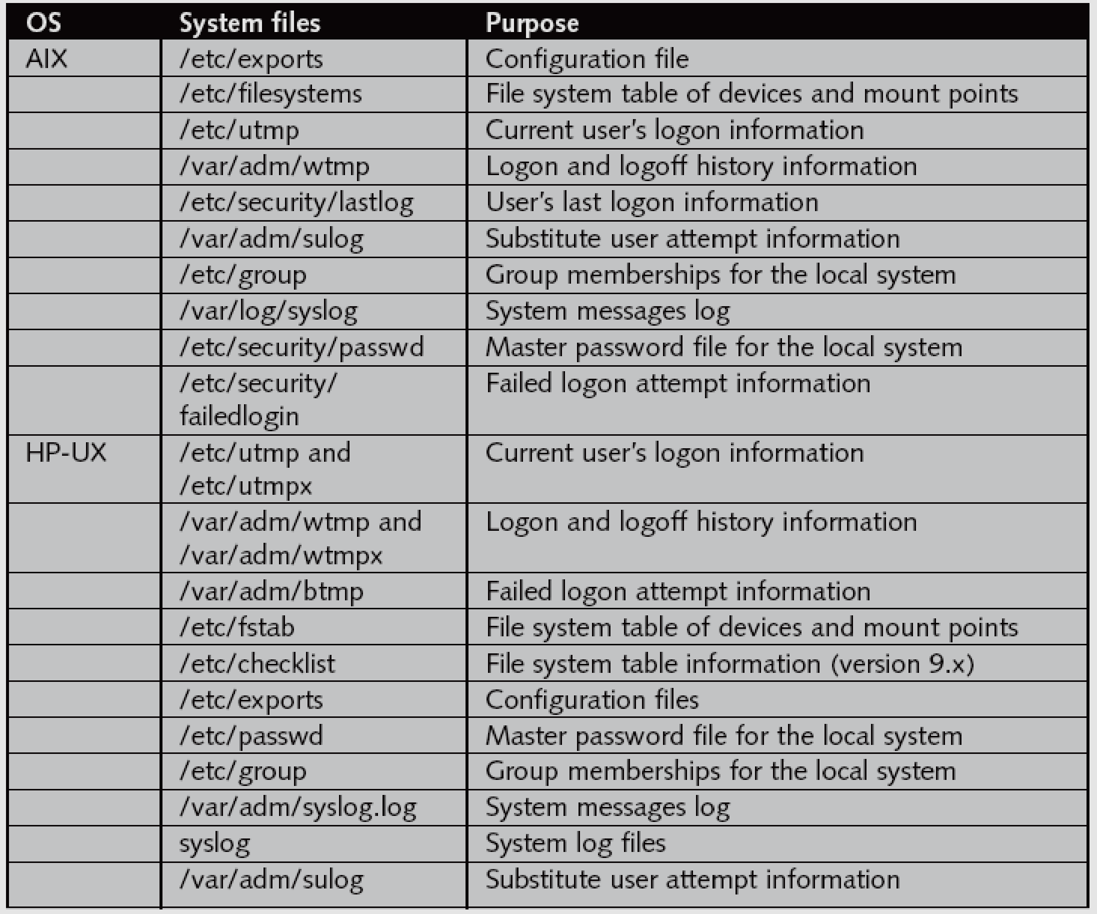
Figure 8-3 Logical EOF and physical EOF

Examining UNIX and Linux Disk Structures and Boot Processes

- UNIX flavors
  - System V variants, Sun Solaris, IBM AIX, and HP-UX
  - BSD, FreeBSD, OpenBSD, and NetBSD
- Linux distributions
  - Red Hat, Fedora, Ubuntu, and Debian
  - Most consistent UNIX-like OSs
- Linux kernel is regulated under the **GNU General Public License (GPL)** agreement

- BSD license is similar to the GPL
  - But makes no requirements for derivative works
- Some useful Linux commands to find information about your Linux system
  - uname –a
  - ls –l
  - ls –ul *filename*
  - netstat -s

Table 8-4 UNIX system files



OS	System files	Purpose
AIX	/etc/exports	Configuration file
	/etc/filesystems	File system table of devices and mount points
	/etc/utmp	Current user's logon information
	/var/adm/wtmp	Logon and logoff history information
	/etc/security/lastlog	User's last logon information
	/var/adm/sulog	Substitute user attempt information
	/etc/group	Group memberships for the local system
	/var/log/syslog	System messages log
	/etc/security/passwd	Master password file for the local system
	/etc/security/failedlogin	Failed logon attempt information
HP-UX	/etc/utmp and /etc/utmpx	Current user's logon information
	/var/adm/wtmp and /var/adm/wtmpx	Logon and logoff history information
	/var/adm/btmp	Failed logon attempt information
	/etc/fstab	File system table of devices and mount points
	/etc/checklist	File system table information (version 9.x)
	/etc/exports	Configuration files
	/etc/passwd	Master password file for the local system
	/etc/group	Group memberships for the local system
	/var/adm/syslog.log	System messages log
	syslog	System log files
	/var/adm/sulog	Substitute user attempt information

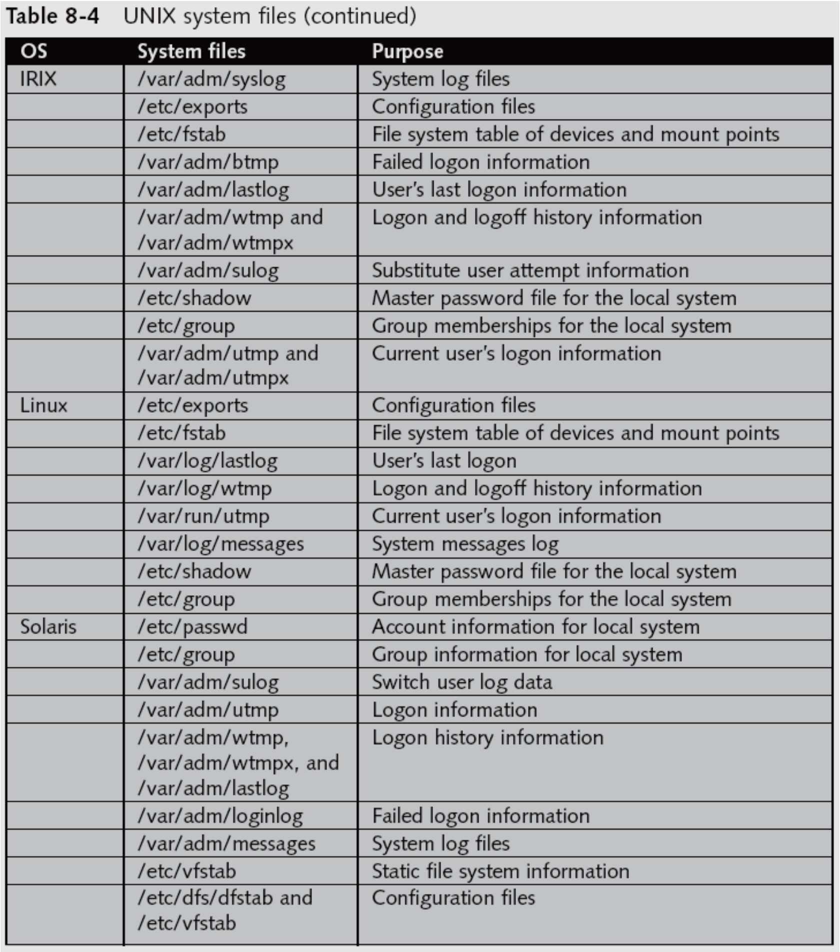


Table 8-4 UNIX system files (continued)

OS	System files	Purpose
IRIX	/var/adm/syslog	System log files
	/etc/exports	Configuration files
	/etc/fstab	File system table of devices and mount points
	/var/adm/btmp	Failed logon information
	/var/adm/lastlog	User's last logon information
	/var/adm/wtmp and /var/adm/wtmpx	Logon and logoff history information
	/var/adm/sulog	Substitute user attempt information
	/etc/shadow	Master password file for the local system
	/etc/group	Group memberships for the local system
	/var/adm/utmp and /var/adm/utmpx	Current user's logon information
Linux	/etc/exports	Configuration files
	/etc/fstab	File system table of devices and mount points
	/var/log/lastlog	User's last logon
	/var/log/wtmp	Logon and logoff history information
	/var/run/utmp	Current user's logon information
	/var/log/messages	System messages log
	/etc/shadow	Master password file for the local system
	/etc/group	Group memberships for the local system
Solaris	/etc/passwd	Account information for local system
	/etc/group	Group information for local system
	/var/adm/sulog	Switch user log data
	/var/adm/utmp	Logon information
	/var/adm/wtmp, /var/adm/wtmpx, and /var/adm/lastlog	Logon history information
	/var/adm/loginlog	Failed logon information
	/var/adm/messages	System log files
	/etc/vfstab	Static file system information
	/etc/dfs/dfstab and /etc/vfstab	Configuration files

- Linux file systems
  - **Second Extended File System (Ext2fs)**
  - Ext3fs, journaling version of Ext2fs
- Employs **inodes**
  - Contain information about each file or directory
  - Pointer to other inodes or blocks
  - Keep internal link count
    - Deleted inodes have count value 0

- Everything is a file
  - Files are objects with properties and methods
- UNIX consists of four components
- Boot block
  - Block is a disk allocation unit of at least 512 bytes
  - Contains the bootstrap code
  - UNIX/Linux computer has only one boot block, located on the main hard disk
- Superblock
  - Indicates disk geometry, available space, and location of the first inode
  - Manages the file system
- Inode blocks
  - First data after the superblock
  - Assigned to every file allocation unit
- Data blocks
  - Where directories and files are stored
  - This location is linked directly to inodes

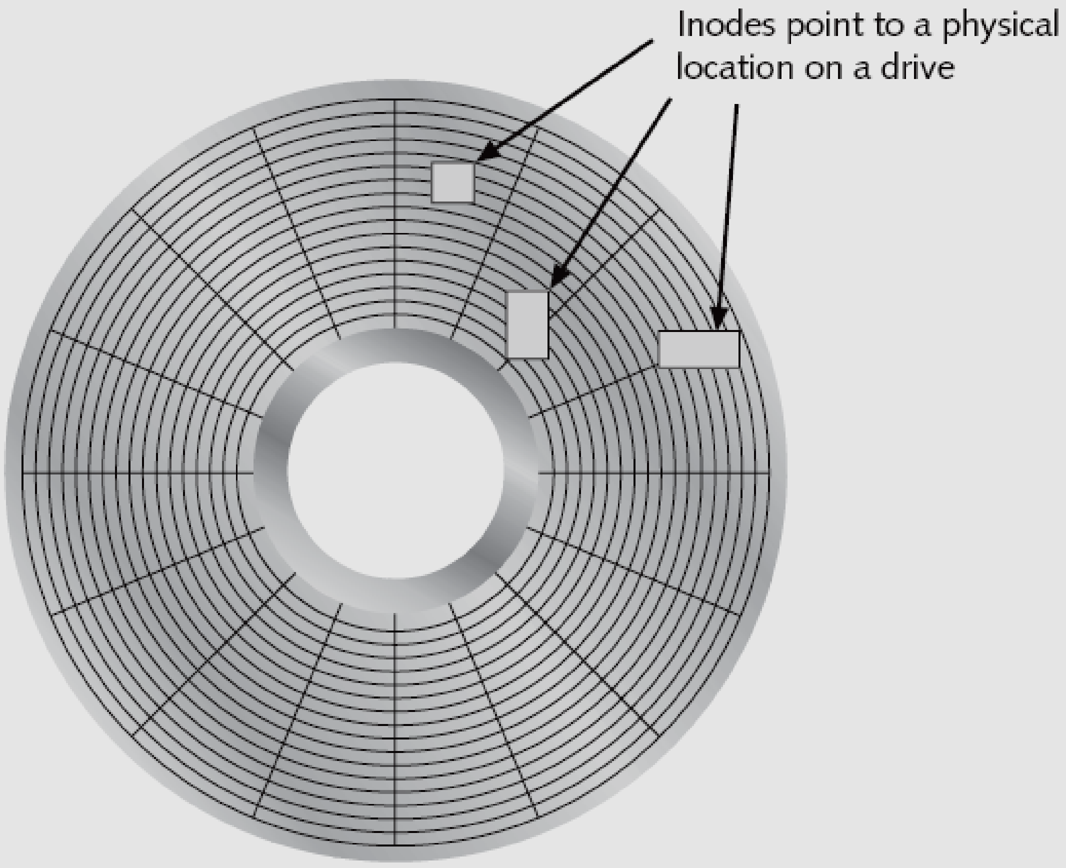


Figure 8-10 Clustering data blocks to save a file in Linux

- **Bad block inode**
  - Keeps track of disk’s bad sectors
  - Commands: badblocks, mke2fs, and e2fsck/
- Linux ls command displays information about files and directories
- **Continuation inode**
  - Provides information about a file or directory
    - Mode and file type, the quantity of links in the file or directory, the file or directory status flag

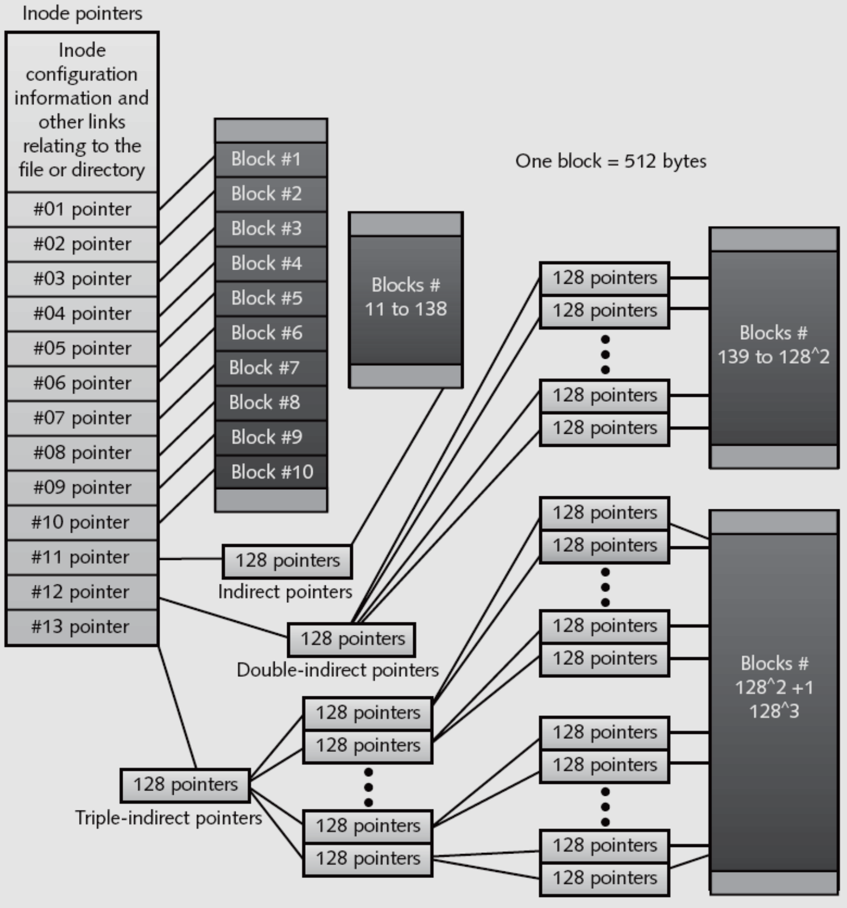
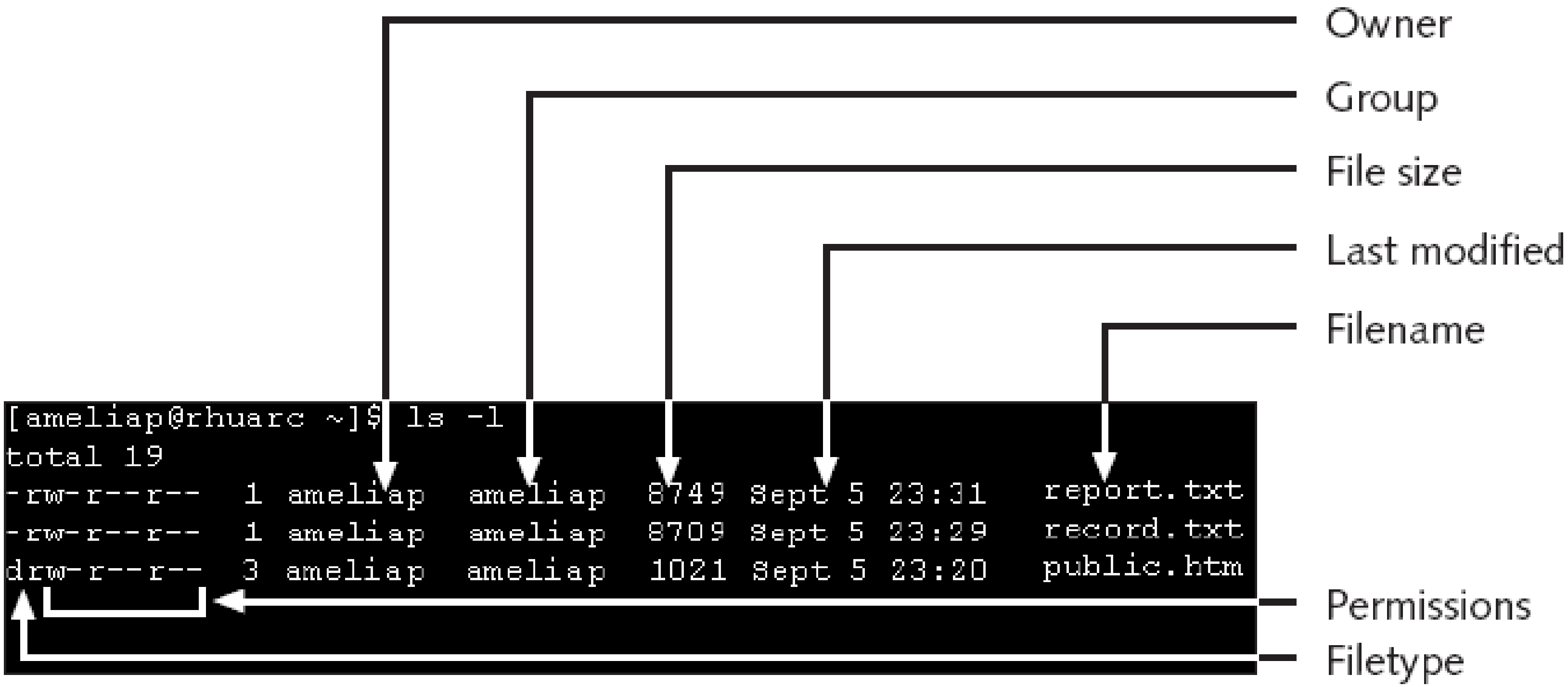


Figure 8-12 Inode pointers in the Linux file system

- Link data stored in data blocks
- Ext2fs and Ext3fs are improvements over Ext
  - Data recovery easier on Ext3fs than on Ext2fs
- First inode has 13 pointers
  - Pointers 1 to 10 are direct pointers to data storage blocks
  - Pointer 11 is an **indirect pointer**
  - Pointer 12 is a **double-indirect pointer**
  - Pointer 13 is a **triple-indirect pointer**



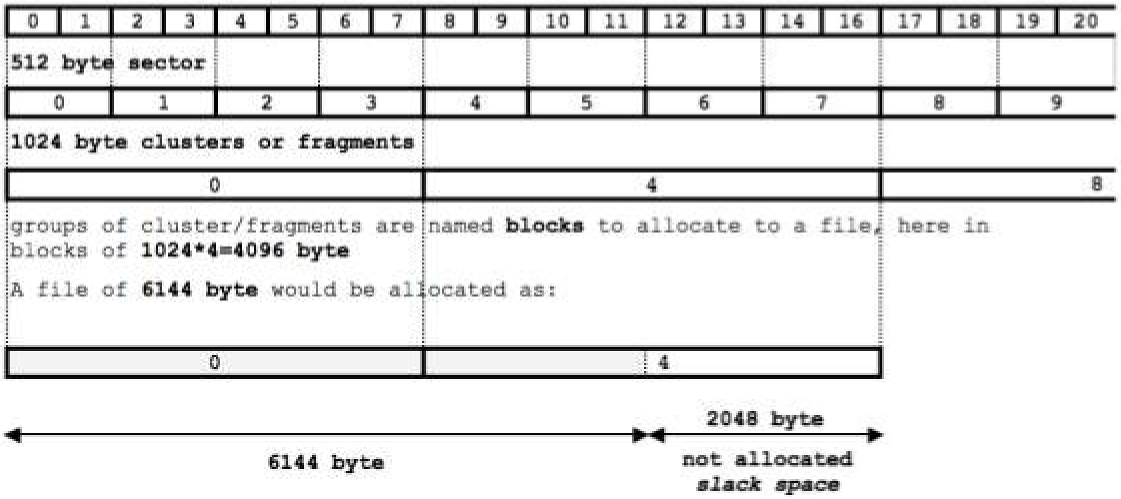
```
[ameliap@rhuarc ~]$ ls -l
total 19
-rw-r--r--  1 ameliap  ameliap  8749 Sept 5 23:31   report.txt
-rw-r--r--  1 ameliap  ameliap  8709 Sept 5 23:29   record.txt
drw-r--r--  3 ameliap  ameliap  1021 Sept 5 23:20   public.htm
```

Figure 8-11 Finding information about a file

FILE SYSTEM

File System (FS)

- A FS is a structure for storing and organizing computer files and the data they contain to make it easy to access and find them.
- Some of the common file systems are:
  - FAT (File Allocation Table) / NTFS
  - UFS/JFS on Unix Systems
- The FS software, is responsible for organizing disk sectors (typically 512 bytes each) into files and directories
- keeping track of which sectors belong to which file (allocated) and which are not being used (unallocated).
- FS typical have directories that associate file names with files, usually by connecting the file name to an index into a file allocation table, such as the FAT, or an inode for Unix-like file system.



groups of cluster/fragments are named **blocks** to allocate to a file, here in blocks of **1024*4=4096 byte**

A file of **6144 byte** would be allocated as:

Figure 1 – a tipical file allocation

- system fragmentation typically occurs when data is not contiguously stored, due to:
  - low free space;
  - deletion, truncation or extension of files.

FILE & MAGIC NUMBER

Files

- File is a term used in the Computer World to indicate a block of stored information (*binary digits) such as a document* in a doc file, an image in a *jpg file or a program in an exe* file
- Almost every computer systems use extensions in file names to help identify what they contain (*the file type)*
- It just being introduced to help OS to correctly address files or rather to identify a program the file is associated with
- nowadays programs analyze the structure of a file rather than extension, this structure refer to **magic number**

Magic Number

- The term magic number has different meanings, however here we are focusing on file, hence the magic number is a constant used to identify a file format (Kessler, 2008)
- basically every file has an header and a footer in order to get correctly recognized
- for example
  - a pdf file starts with “%PDF” and ends with “%EOF”
  - a jpeg image file begins with “0xFFD8” and ends with “0xFFD9”.
- http://www.garykessler.net/library/file_sigs.html

CARVING CONCEPT

Carving Concepts

- Data carving might be classified as basic and advanced, with basic data carving it is assumed that:
  - the beginning of file is not overwritten
  - the file is not fragmented
  - the file is not compressed (i.e. NTFS compressed)
- basically this type of carving is made with header and footer, while advanced data carving occurs even to fragmented files, where fragments are:
  - not sequential
  - out of order
  - Missing
- Having deep knowledge of internal file‟s structure could result in less false positive, this is the reason of why new algorithm also relies on "internal file structure"

For instance, the first sector of an office file contains a CDH header that must contain the hex value FE as the 29th character and the value FF as the30 th character, these values might be verified in order to recognize the file

Unallocated data recovery and slack space

- Sometimes, where it is needed to recover deleted data, might be useful to run the tool *dls against the system device before* carves data,
- This in order to extract all information from the unallocated data, *dls is part of the Sleuth Kit.*

Tool

- *Foremost is a well-known tool, originally developed at the* US AirForce
  - (http://foremost.sourceforge.net/)
  - it works on image files, such as those enerated by *dd, Safeback, Encase, etc. or directly on* a drive
- *Scalpel* is a complete rewrite of *foremost 0.69 done by Golden G. Richard III*
  - *http://www.digitalforensicssolutions.com/Scalpel/*
  - *enhance performance and decrease memory usage.*
  - *It is a* fast and filesystem-independent file carver that reads a database of header and footer definitions and extracts matching files from a set of image files or raw device files.

Other Tool

- **tcpxtract**
  - *Tcpxtract at http://tcpxtract.sourceforge.net/*
  - is a freeware tool written by *Nick Harbour for extracting files from network* traffic, based on file signatures (headers and footers) it uses the same techniques used by *foremost, but specifically for* the application of intercepting files transmitted across a network.
  - *Tool uses libpcap, a popular* portable and stable library for network data capture and moreover can be used against a live network or a *tcpdump* formatted capture file.

- **chaosreader**
- *http://chaosreader.sourceforge.net/*
- *Chaosreader* is a freeware tool written by *Brendan Gregg, it can trace TCP/UDP/etc.*
- *sessions and fetch application data from tcpdump or snoop logs.*
- *It fetches telnet sessions, FTP files, HTTP* transfers (*HTML, GIF, JPEG, etc.), SMTP emails, etc. from the* captured network traffic.

- *Msramdmp*
- *http://www.mcgrewsecurity.com/projects/msramdmp/*
- a freeware tool written by Wesley McGrew,
- since RAM is a volatile storage, it is erased when power is removed.
- Well, the research just demonstrate that this assumption might be incorrect! however going beyond this goal, the tool can be used to carves out data from memory
- this result can be obtained with the well known tool *dd against the mem device*

CASE STUDY :

RECOVERING GRAPHICS FILES

Recognizing a Graphics File

- Contains digital photographs, line art, three-dimensional images, and scanned replicas of printed pictures
  - **Bitmap images**: collection of dots
  - **Vector graphics**: based on mathematical instructions
  - **Metafile graphics**: combination of bitmap and vector
- Types of programs
  - Graphics editors
  - Image viewers

Understanding Bitmap and Raster Images

- Bitmap images
  - Grids of individual **pixels**
- **Raster images**
  - Pixels are stored in rows
  - Better for printing
- Image quality
  - Screen **resolution**
  - Software
  - Number of color bits used per pixel

Understanding Vector Graphics

- Characteristics
  - Lines instead of dots
  - Store only the calculations for drawing lines and shapes
  - Smaller size
  - Preserve quality when image is enlarged
- CorelDraw, Adobe Illustrator

Understanding Metafile Graphics

- Combine raster and vector graphics
- Example
  - Scanned photo (bitmap) with text (vector)
- Share advantages and disadvantages of both types
  - When enlarged, bitmap part loses quality

Understanding Graphics File Formats

- Standard bitmap file formats
  - Graphic Interchange Format (.gif)
  - Joint Photographic Experts Group (.jpeg, .jpg)
  - Tagged Image File Format (.tiff, .tif)
  - Window Bitmap (.bmp)
- Standard vector file formats
  - Hewlett Packard Graphics Language (.hpgl)
  - Autocad (.dxf)

Understanding Graphics File Formats (continued)

- Nonstandard graphics file formats
  - Targa (.tga)
  - Raster Transfer Language (.rtl)
  - Adobe Photoshop (.psd) and Illustrator (.ai)
  - Freehand (.fh9)
  - Scalable Vector Graphics (.svg)
  - Paintbrush (.pcx)
- Search the Web for software to manipulate unknown image formats

Understanding Digital Camera File Formats

- Witnesses or suspects can create their own digital photos
- Examining the raw file format
  - **Raw file format**
    - Referred to as a digital negative
    - Typically found on many higher-end digital cameras
  - Sensors in the digital camera simply record pixels on the camera’s memory card
  - Raw format maintains the best picture quality

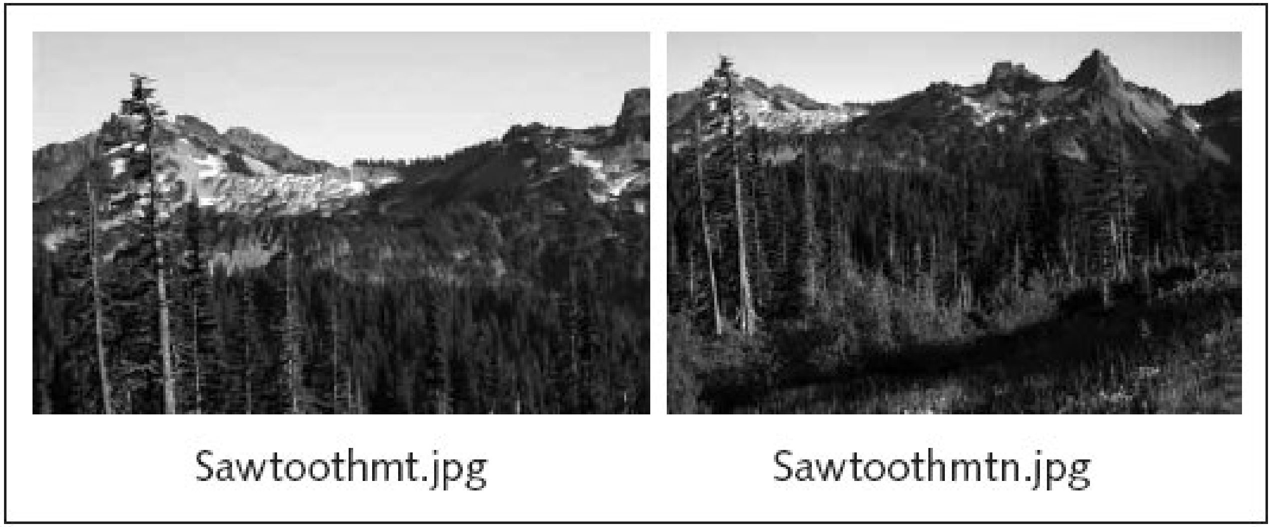
Understanding Digital Camera File Formats (continued)

- Examining the raw file format (continued)
  - The biggest disadvantage is that it’s proprietary
    - And not all image viewers can display these formats
  - The process of converting raw picture data to another format is referred to as **demosaicing**
- Examining the Exchangeable Image File format
  - **Exchangeable Image File (EXIF)** format
    - Commonly used to store digital pictures
    - Developed by JEIDA as a standard for storing metadata in JPEG and TIFF files

Understanding Digital Camera File Formats (continued)

- Examining the Exchangeable Image File format (continued)
  - EXIF format collects metadata
    - Investigators can learn more about the type of digital camera and the environment in which pictures were taken
  - EXIF file stores metadata at the beginning of the file

Understanding Digital Camera File Formats (continued)

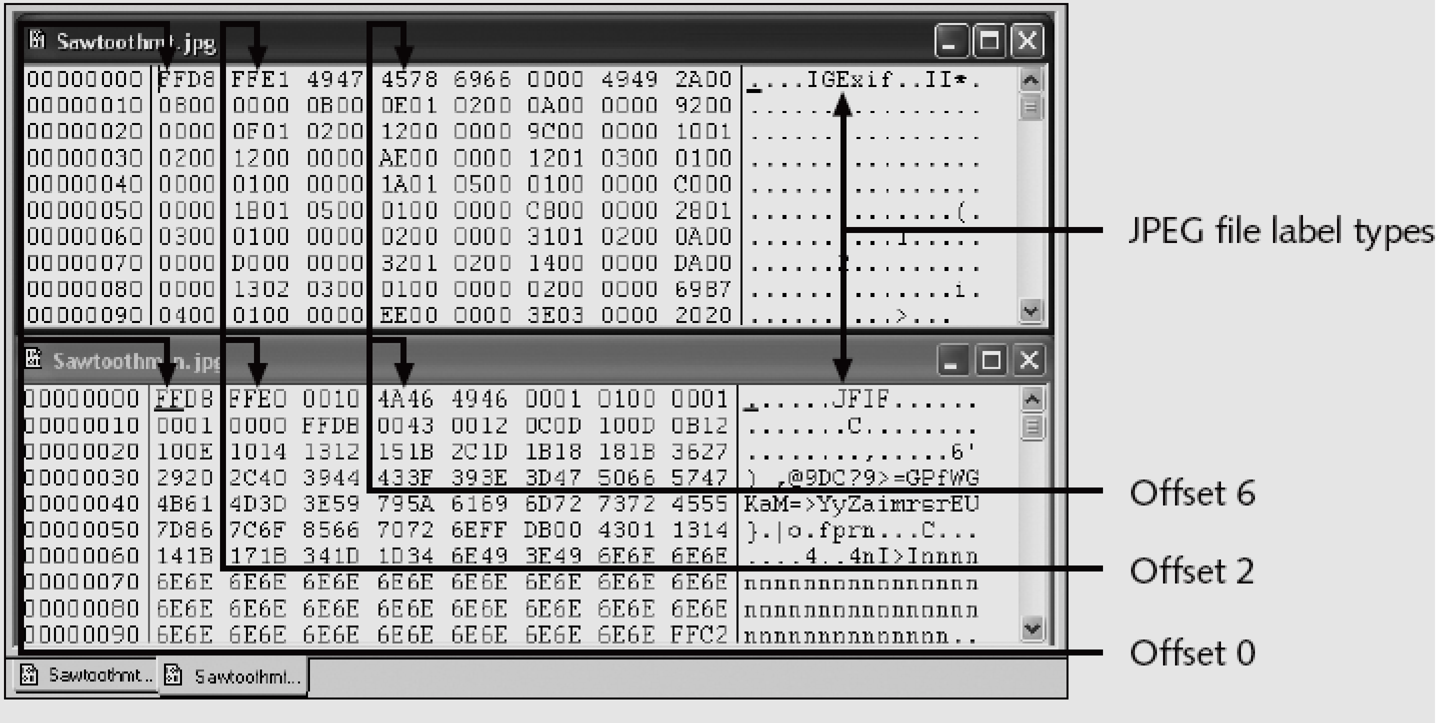


Sawtoothmt.jpg

Sawtoothmtn.jpg

Figure 10-1 Identical EXIF and JPEG pictures

Understanding Digital Camera File Formats (continued)



```
00000000 FFD8 FFE1 4947 4578 6966 0000 4949 2A00 ....IGExif..II*.
00000010 0800 0000 0B00 0E01 0200 0A00 0000 9200 ................
00000020 0000 0F01 0200 1200 0000 9C00 0000 1001 ................
00000030 0200 1200 0000 AE00 0000 1201 0300 0100 ................
00000040 0000 0100 0000 1A01 0500 0100 0000 C000 ................
00000050 0000 1B01 0500 0100 0000 C800 0000 2801 ..............(.
00000060 0300 0100 0000 0200 0000 3101 0200 0A00 ..........1.....
00000070 0000 D000 0000 3201 0200 1400 0000 DA00 ......2.........
00000080 0000 1302 0300 0100 0000 0200 0000 6987 ..............i.
00000090 0400 0100 0000 EE00 0000 3E03 0000 2020 ..........>...
```

```
00000000 FFD8 FFE0 0010 4A46 4946 0001 0100 0001 ......JFIF......
00000010 0001 0000 FFDB 0043 0012 0C0D 100D 0B12 .......C........
00000020 100E 1014 1312 151B 2C1D 1B18 181B 3627 ..........,.....6'
00000030 2920 2C40 3944 433F 393E 3D47 5066 5747 ) ,@9DC?9>=GPfWG
00000040 4B61 4D3D 3E59 795A 6169 6D72 7372 4555 KaM=>YyZaimrsrEU
00000050 7D86 7C6F 8566 7072 6EFF DB00 4301 1314 }.|o.fprn...C...
00000060 141B 171B 341D 1D34 6E49 3E49 6E6E 6E6E ....4..4nI>Innnn
00000070 6E6E 6E6E 6E6E 6E6E 6E6E 6E6E 6E6E 6E6E nnnnnnnnnnnnnnnn
00000080 6E6E 6E6E 6E6E 6E6E 6E6E 6E6E 6E6E 6E6E nnnnnnnnnnnnnnnn
00000090 6E6E 6E6E 6E6E 6E6E 6E6E 6E6E 6E6E FFC2 nnnnnnnnnnnnnn..
```

JPEG file label types

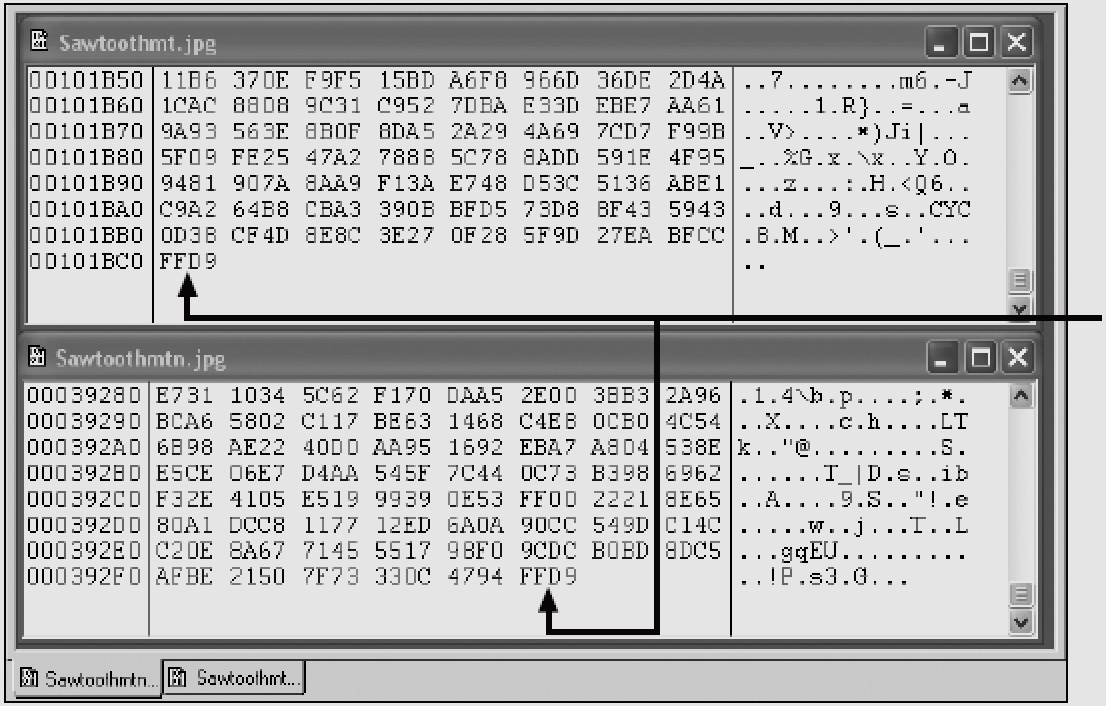
Offset 6

Offset 2

Offset 0

Figure 10-2 Differences in EXIF and JPEG file header information

Understanding Digital Camera File Formats (continued)



```
00101B50 11B6 370E F9F5 15BD A6F8 966D 36DE 2D4A ..7........m6.-J
00101B60 1CAC 8808 9C31 C952 7DBA E33D EBE7 AA61 .....1.R}..=...a
00101B70 9A93 563E 8B0F 8DA5 2A29 4A69 7CD7 F99B ..V>....*)Ji|...
00101B80 5F09 FE25 47A2 788B 5C78 8ADD 591E 4F95 _..%G.x.\x..Y.O.
00101B90 9481 907A 8AA9 F13A E748 D53C 5136 ABE1 ...z...:.H.<Q6..
00101BA0 C9A2 64B8 CBA3 390B BFD5 73D8 BF43 5943 ..d...9...s..CYC
00101BB0 0D38 CF4D 8E8C 3E27 0F28 5F9D 27EA BFCC .8.M..>'.(_.'...
00101BC0 FFD9                                    ..
```

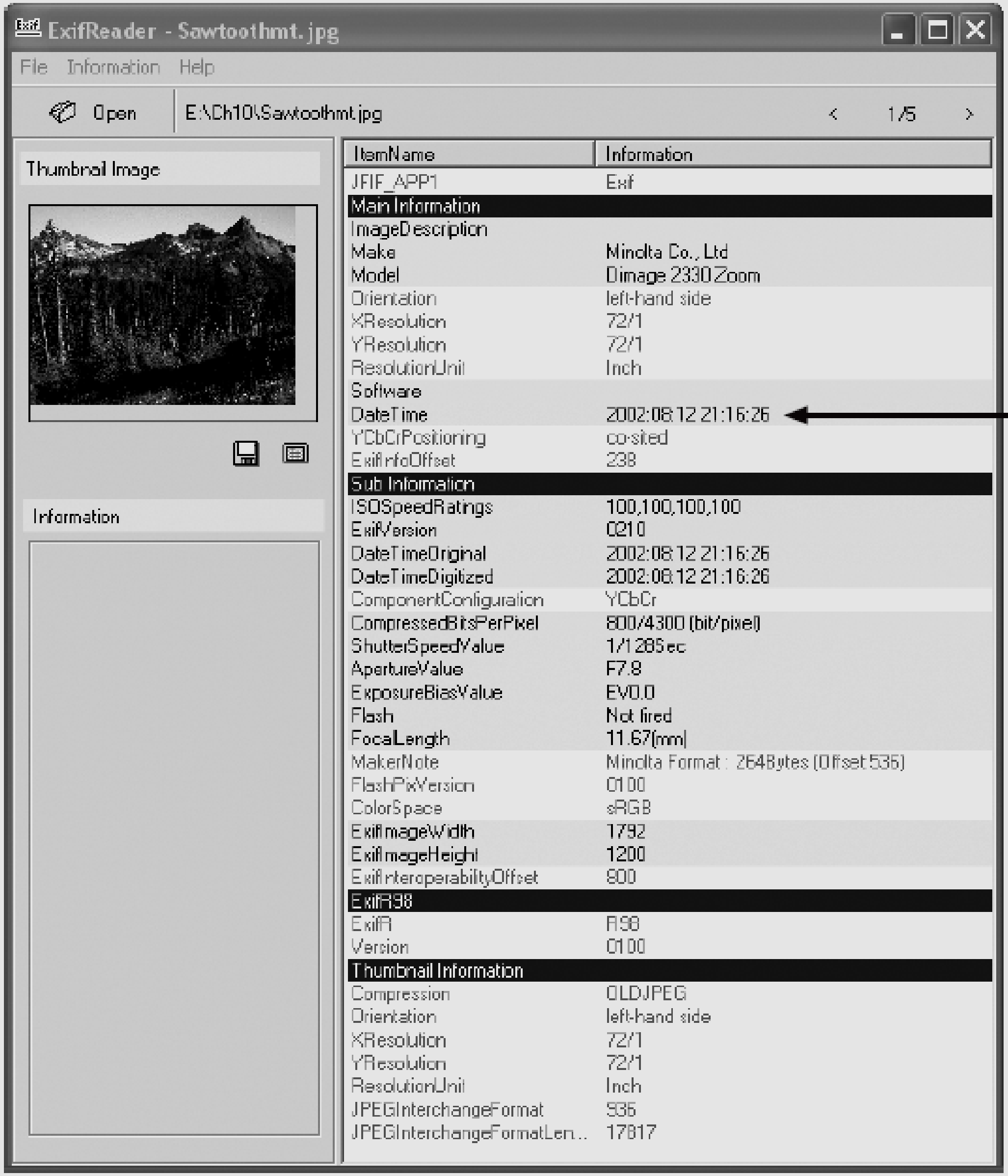
```
00039280 E731 1034 5C62 F170 DAA5 2E00 3BB3 2A96 .1.4\b.p....;.*.
00039290 BCA6 5802 C117 BE63 1468 C4E8 0CB0 4C54 ..X....c.h....LT
000392A0 6B98 AE22 400D AA95 1692 EBA7 A804 538E k.."@.........S.
000392B0 E5CE 06E7 D4AA 545F 7C44 0C73 B398 6962 ......T_|D.s..ib
000392C0 F32E 4105 E519 9939 0E53 FF00 2221 8E65 ..A....9.S.."!.e
000392D0 80A1 DCC8 1177 12ED 6A0A 90CC 549D C14C .....w..j...T..L
000392E0 C20E 8A67 7145 5517 98F0 9CDC B0BD 8DC5 ...gqEU.........
000392F0 AFBE 2150 7F73 330C 4794 FFD9           ..!P.s3.G...
```

JPEG file EOI marker

Figure 10-3 EOI marker FFD9 for all JPEG files

Understanding Digital Camera File Formats (continued)

- Examining the Exchangeable Image File format (continued)
  - With tools such as ProDiscover and Exif Reader
    - You can extract metadata as evidence for your case



ExifReader - Sawtoothmt.jpg

E:\Ch10\Sawtoothmt.jpg

ItemName	Information
JFIF_APP1	Exif
Main Information	
ImageDescription	
Make	Minolta Co., Ltd
Model	Dimage 2330 Zoom
Orientation	left-hand side
XResolution	72/1
YResolution	72/1
ResolutionUnit	Inch
Software	
DateTime	2002:08:12 21:16:26
YCbCrPositioning	co-sited
ExifInfoOffset	238
Sub Information	
ISOSpeedRatings	100,100,100,100
ExifVersion	0210
DateTimeOriginal	2002:08:12 21:16:26
DateTimeDigitized	2002:08:12 21:16:26
ComponentConfiguration	YCbCr
CompressedBitsPerPixel	800/4300 (bit/pixel)
ShutterSpeedValue	1/128Sec
ApertureValue	F7.8
ExposureBiasValue	EV0.0
Flash	Not fired
FocalLength	11.67[mm]
MakerNote	Minolta Format : 264Bytes (Offset 536)
FlashPixVersion	0100
ColorSpace	sRGB
ExifImageWidth	1792
ExifImageHeight	1200
ExifInteroperabiltyOffset	800
ExifR98	
ExifR	R98
Version	0100
Thumbnail Information	
Compression	OLDJPEG
Orientation	left-hand side
XResolution	72/1
YResolution	72/1
ResolutionUnit	Inch
JPEGInterchangeFormat	936
JPEGInterchangeFormatLen...	17817

Camera-recorded date and time of photograph

Figure 10-4 Exif Reader displaying metadata from an EXIF JPEG file

Understanding Data Compression

- Some image formats compress their data
  - GIF, JPEG, PNG
- Others, like BMP, do not compress their data
  - Use data compression tools for those formats
- **Data compression**
  - Coding of data from a larger to a smaller form
  - Types
    - Lossless compression and lossy compression

Lossless and Lossy Compression

- **Lossless compression**
  - Reduces file size without removing data
  - Based on Huffman or Lempel-Ziv-Welch coding
    - For redundant bits of data
  - Utilities: WinZip, PKZip, StuffIt, and FreeZip
- **Lossy compression**
  - Permanently discards bits of information
  - **Vector quantization (VQ)**
    - Determines what data to discard based on vectors in the graphics file
  - Utility: Lzip

Locating and Recovering Graphics Files

- Operating system tools
  - Time consuming
  - Results are difficult to verify
- Computer forensics tools
  - Image headers
    - Compare them with good header samples
    - Use header information to create a baseline analysis
  - Reconstruct fragmented image files
    - Identify data patterns and modified headers

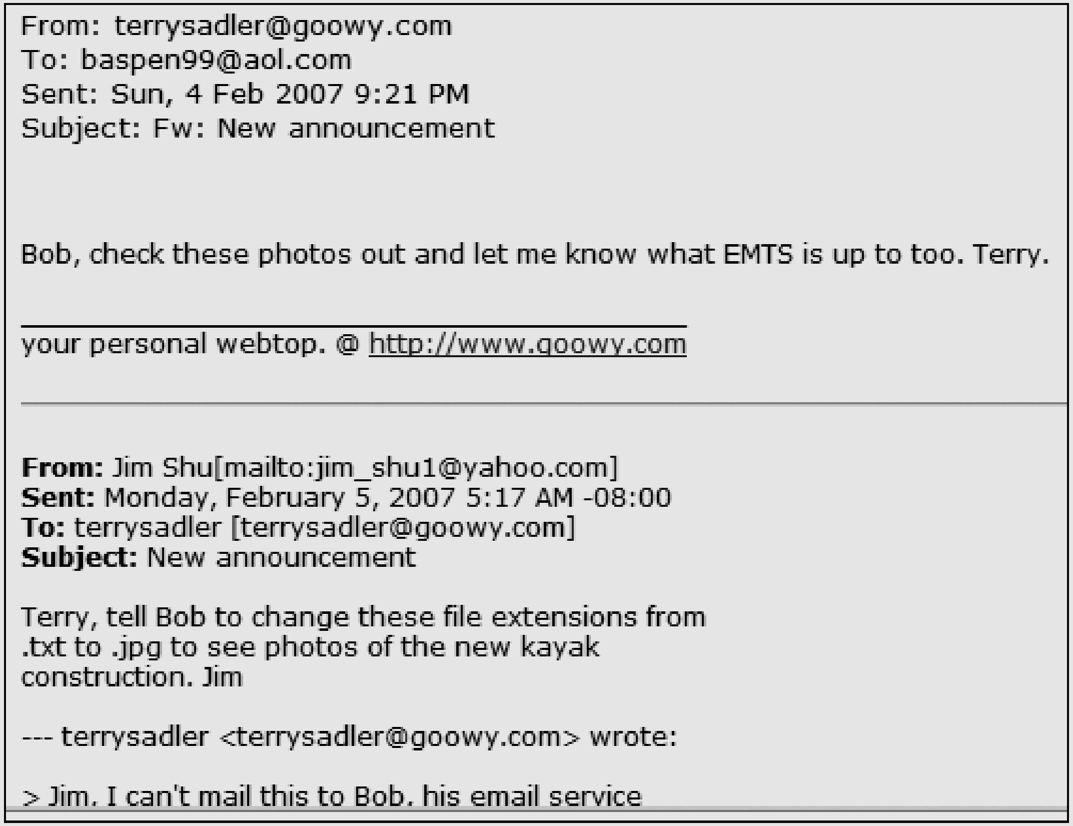
Identifying Graphics File Fragments

- Carving or salvaging
  - Recovering all file fragments
- Computer forensics tools
  - Carve from slack and free space
  - Help identify image files fragments and put them together

Repairing Damage Headers

- Use good header samples
- Each image file has a unique file header
  - JPEG: FF D8 FF E0 00 10
  - Most JPEG files also include JFIF string
- Exercise:
  - Investigate a possible intellectual property theft by a contract employee of Exotic Mountain Tour Service (EMTS)

Searching for and Carving Data from Unallocated Space



From: terrysadler@goowy.com
To: baspen99@aol.com
Sent: Sun, 4 Feb 2007 9:21 PM
Subject: Fw: New announcement

Bob, check these photos out and let me know what EMTS is up to too. Terry.

your personal webtop. @ http://www.goowy.com

From: Jim Shu[mailto:jim_shu1@yahoo.com]
Sent: Monday, February 5, 2007 5:17 AM -08:00
To: terrysadler [terrysadler@goowy.com]
Subject: New announcement

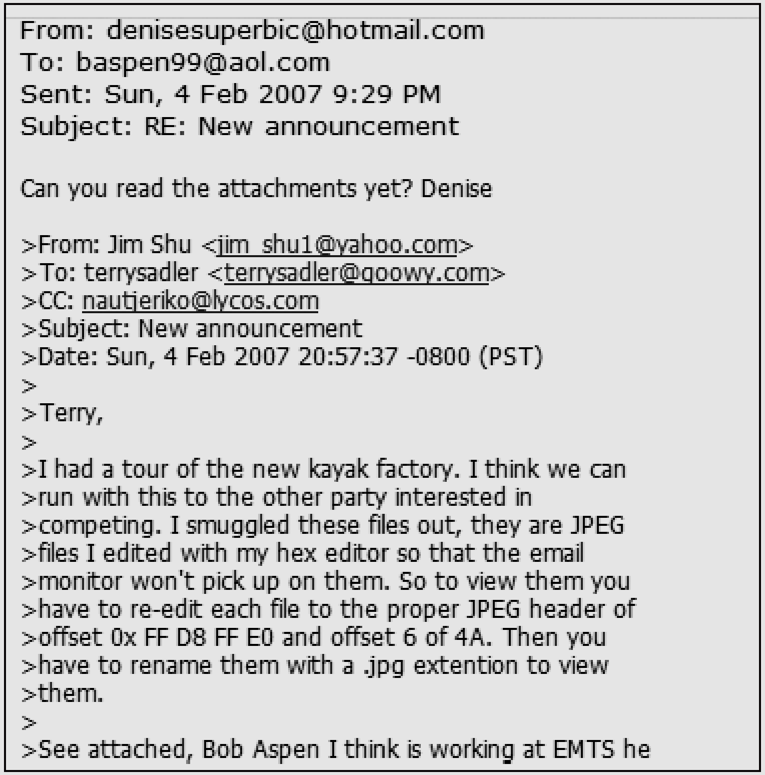
Terry, tell Bob to change these file extensions from
.txt to .jpg to see photos of the new kayak
construction. Jim

--- terrysadler <terrysadler@goowy.com> wrote:

> Jim. I can't mail this to Bob. his email service

Figure 10-5 First intercepted capture of an e-mail from Terry Sadler

Searching for and Carving Data from Unallocated Space (continued)

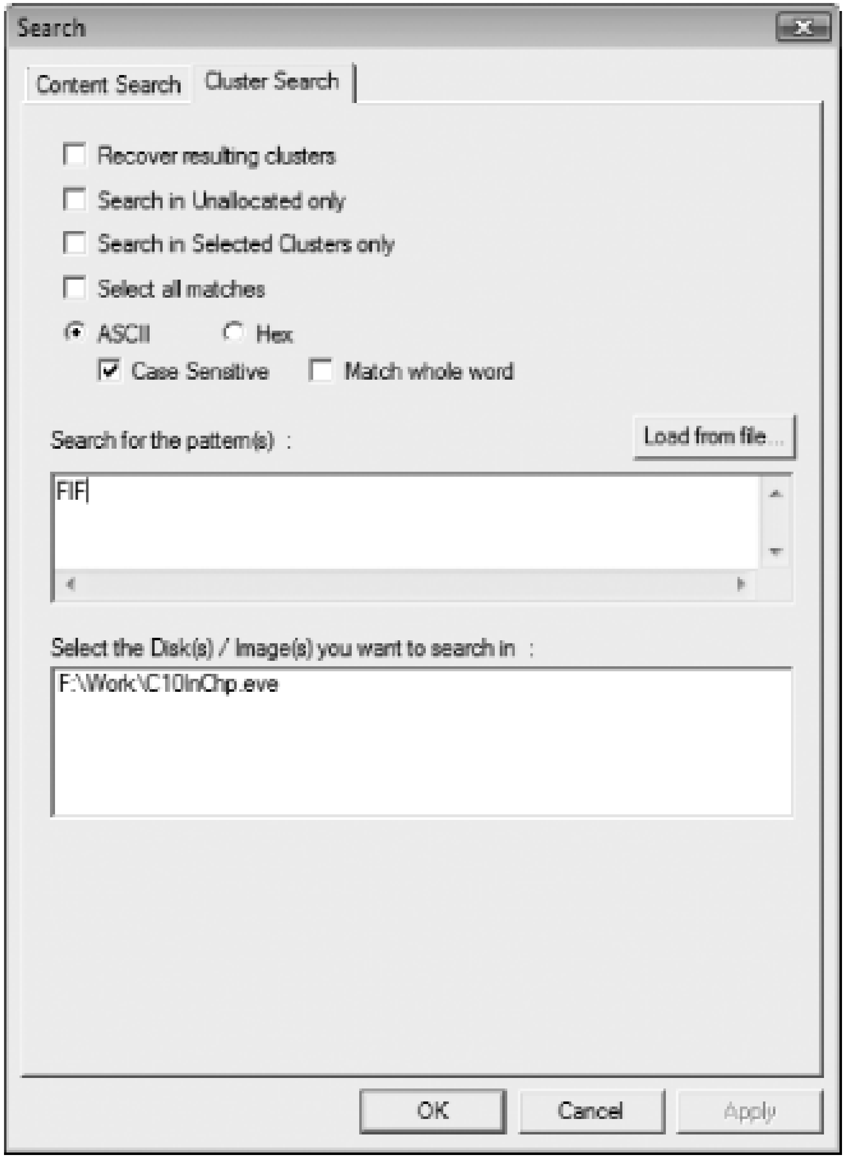


```
From: denisesuperbic@hotmail.com
To: baspen99@aol.com
Sent: Sun, 4 Feb 2007 9:29 PM
Subject: RE: New announcement

Can you read the attachments yet? Denise

>From: Jim Shu <jim_shu1@yahoo.com>
>To: terrysadler <terrysadler@goowy.com>
>CC: nautjeriko@lycos.com
>Subject: New announcement
>Date: Sun, 4 Feb 2007 20:57:37 -0800 (PST)
>
>Terry,
>
>I had a tour of the new kayak factory. I think we can
>run with this to the other party interested in
>competing. I smuggled these files out, they are JPEG
>files I edited with my hex editor so that the email
>monitor won't pick up on them. So to view them you
>have to re-edit each file to the proper JPEG header of
>offset 0x FF D8 FF E0 and offset 6 of 4A. Then you
>have to rename them with a .jpg extention to view
>them.
>
>See attached, Bob Aspen I think is working at EMTS he
```

Figure 10-6 Second intercepted capture of an e-mail from denisesuperbic@hotmail.com

Searching for and Carving Data from Unallocated Space (continued)

- Steps
  - Planning your examination
  - Searching for and recovering digital photograph evidence
    - Use ProDiscover to search for and extract (recover) possible evidence of JPEG files
    - False hits are referred to as **false positives**



□ Recover resulting clusters
□ Search in Unallocated only
□ Search in Selected Clusters only
□ Select all matches
● ASCII ○ Hex
☑ Case Sensitive □ Match whole word

Search for the pattern(s) :

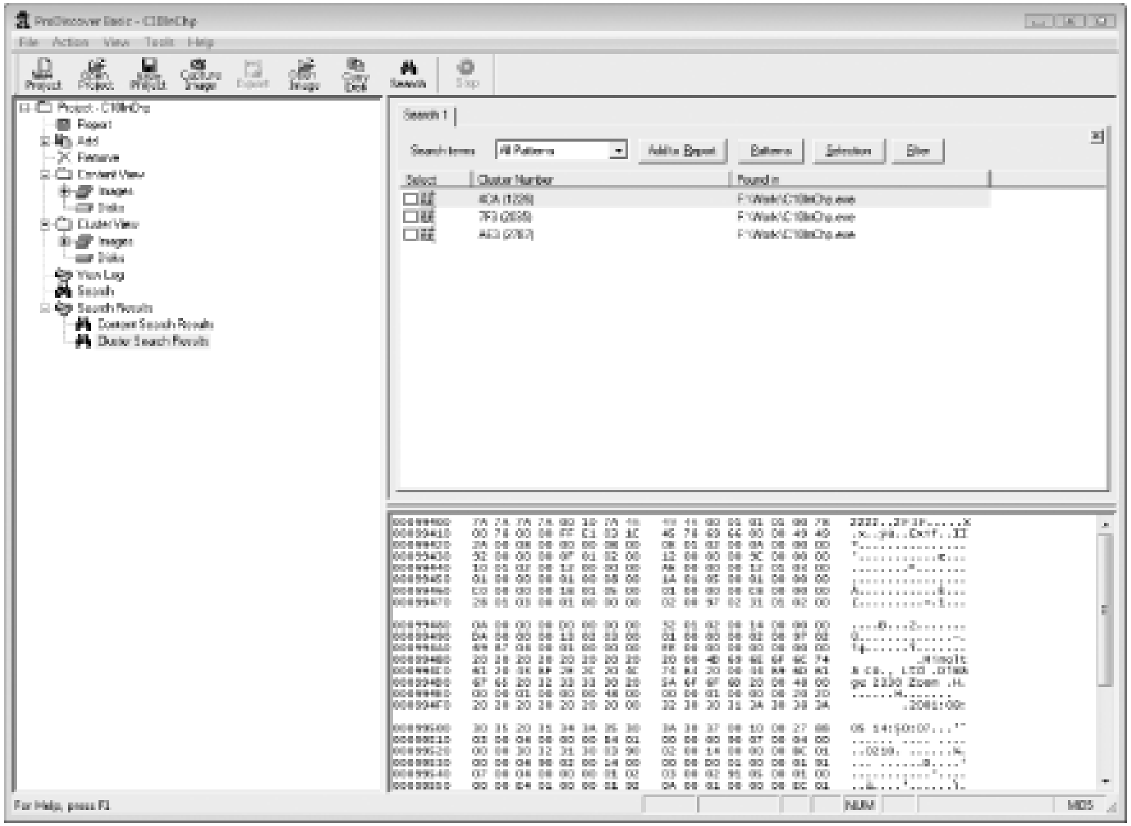
FIF

Select the Disk(s) / Image(s) you want to search in :

F:\Work\C10InChp.eve

Figure 10-7 Searching clusters in ProDiscover

Searching for and Carving Data from Unallocated Space (continued)



Search terms: All Patterns

Select	Cluster Number	Found in
□	4CA (1226)	F:\Work\C10InChp.exe
□	7F3 (2035)	F:\Work\C10InChp.exe
□	AE3 (2787)	F:\Work\C10InChp.exe

Figure 10-8 Completed cluster search for FIF

Searching for and Carving Data from Unallocated Space (continued)

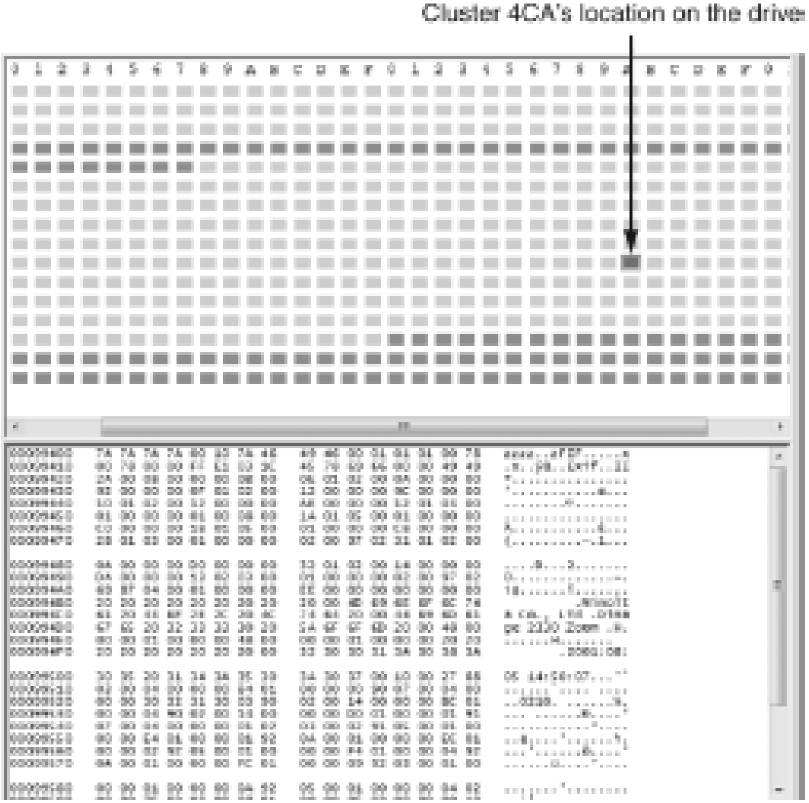
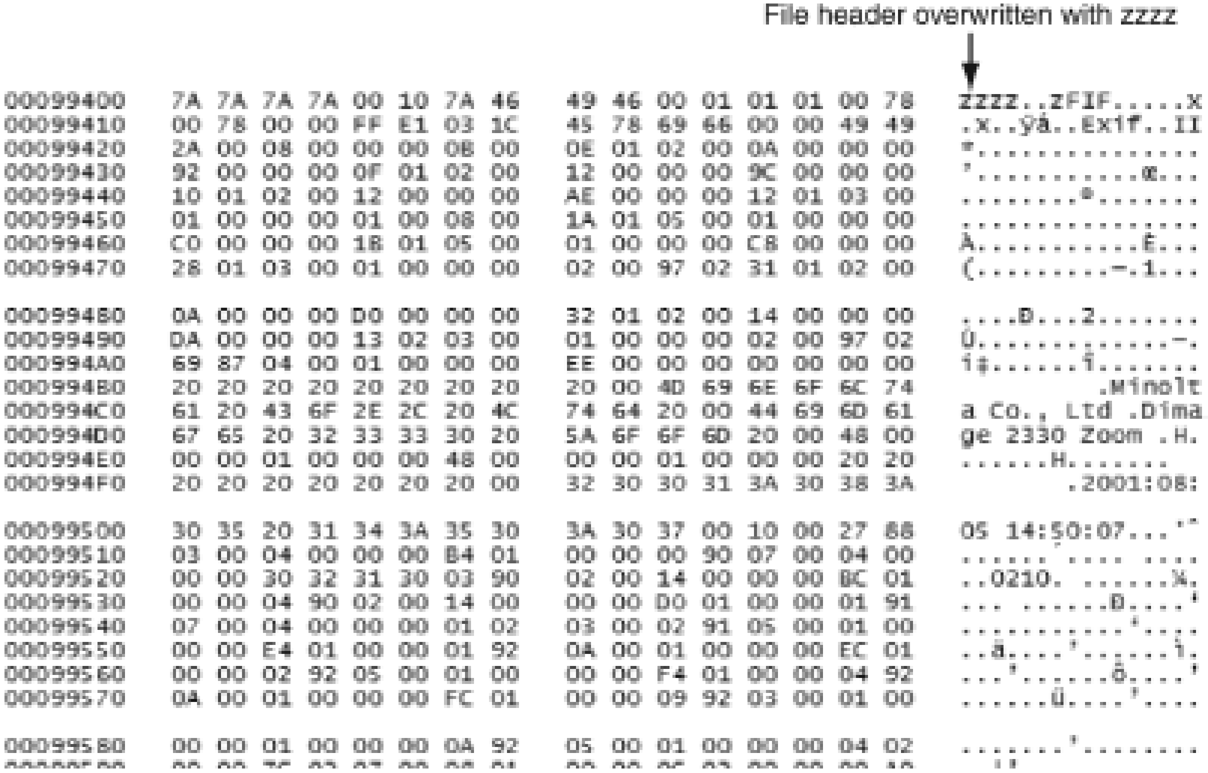


Figure 10-9 Viewing cluster use and location of search hit for 4CA(1226)

Searching for and Carving Data from Unallocated Space (continued)

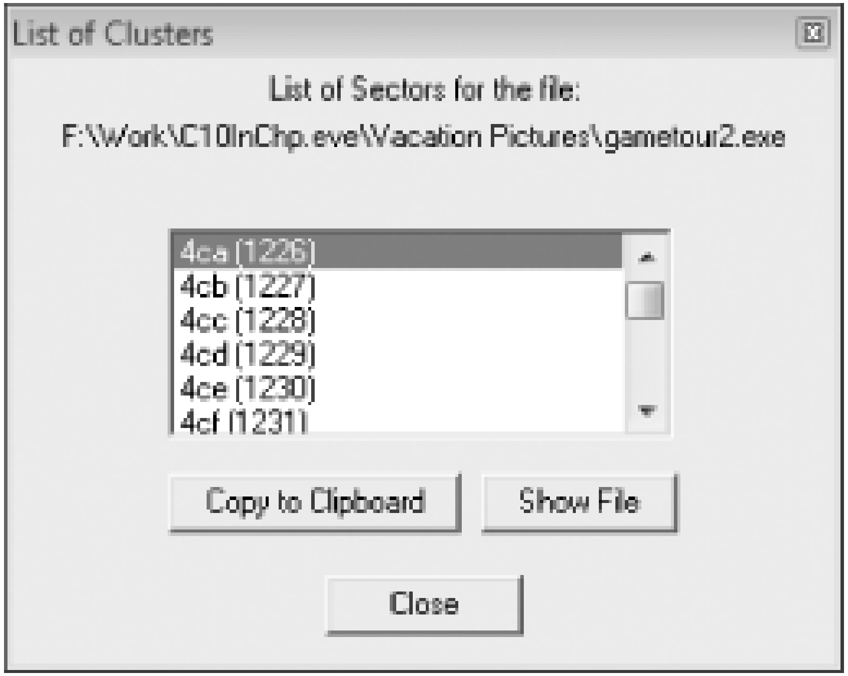


File header overwritten with zzzz

```
00099400   7A 7A 7A 7A 00 10 7A 46   49 46 00 01 01 01 00 78   zzzz..zFIF.....x
00099410   00 78 00 00 FF E1 03 1C   45 78 69 66 00 00 49 49   .x..ÿá..Exif..II
00099420   2A 00 08 00 00 00 0B 00   0E 01 02 00 0A 00 00 00   *...............
00099430   92 00 00 00 0F 01 02 00   12 00 00 00 9C 00 00 00   '...........œ...
00099440   10 01 02 00 12 00 00 00   AE 00 00 00 12 01 03 00   ........®.......
00099450   01 00 00 00 01 00 08 00   1A 01 05 00 01 00 00 00   ................
00099460   C0 00 00 00 1B 01 05 00   01 00 00 00 C8 00 00 00   À...........È...
00099470   28 01 03 00 01 00 00 00   02 00 97 02 31 01 02 00   (.........—.1...

00099480   0A 00 00 00 D0 00 00 00   32 01 02 00 14 00 00 00   ....Ð...2.......
00099490   DA 00 00 00 13 02 03 00   01 00 00 00 02 00 97 02   Ú.............—.
000994A0   69 87 04 00 01 00 00 00   EE 00 00 00 00 00 00 00   i‡......î.......
000994B0   20 20 20 20 20 20 20 20   20 00 4D 69 6E 6F 6C 74            .Minolt
000994C0   61 20 43 6F 2E 2C 20 4C   74 64 20 00 44 69 6D 61   a Co., Ltd .Dima
000994D0   67 65 20 32 33 33 30 20   5A 6F 6F 6D 20 00 48 00   ge 2330 Zoom .H.
000994E0   00 00 01 00 00 00 48 00   00 00 01 00 00 00 20 20   ......H.......
000994F0   20 20 20 20 20 20 20 00   32 30 30 31 3A 30 38 3A           .2001:08:

00099500   30 35 20 31 34 3A 35 30   3A 30 37 00 10 00 27 88   05 14:50:07...'ˆ
00099510   03 00 04 00 00 00 B4 01   00 00 00 90 07 00 04 00   ................
00099520   00 00 30 32 31 30 03 90   02 00 14 00 00 00 BC 01   ..0210. ......¼.
00099530   00 00 04 90 02 00 14 00   00 00 D0 01 00 00 01 91   ... ......Ð....'
00099540   07 00 04 00 00 00 01 02   03 00 02 91 05 00 01 00   ...........'....
00099550   00 00 E4 01 00 00 01 92   0A 00 01 00 00 00 EC 01   ..ä....'......ì.
00099560   00 00 02 92 05 00 01 00   00 00 F4 01 00 00 04 92   ...'......ô....'
00099570   0A 00 01 00 00 00 FC 01   00 00 09 92 03 00 01 00   ......ü....'....

00099580   00 00 01 00 00 00 0A 92   05 00 01 00 00 00 04 02   .......'........
```

Figure 10-10 Content of cluster 4CA(1226)

Searching for and Carving Data from Unallocated Space (continued)



List of Clusters

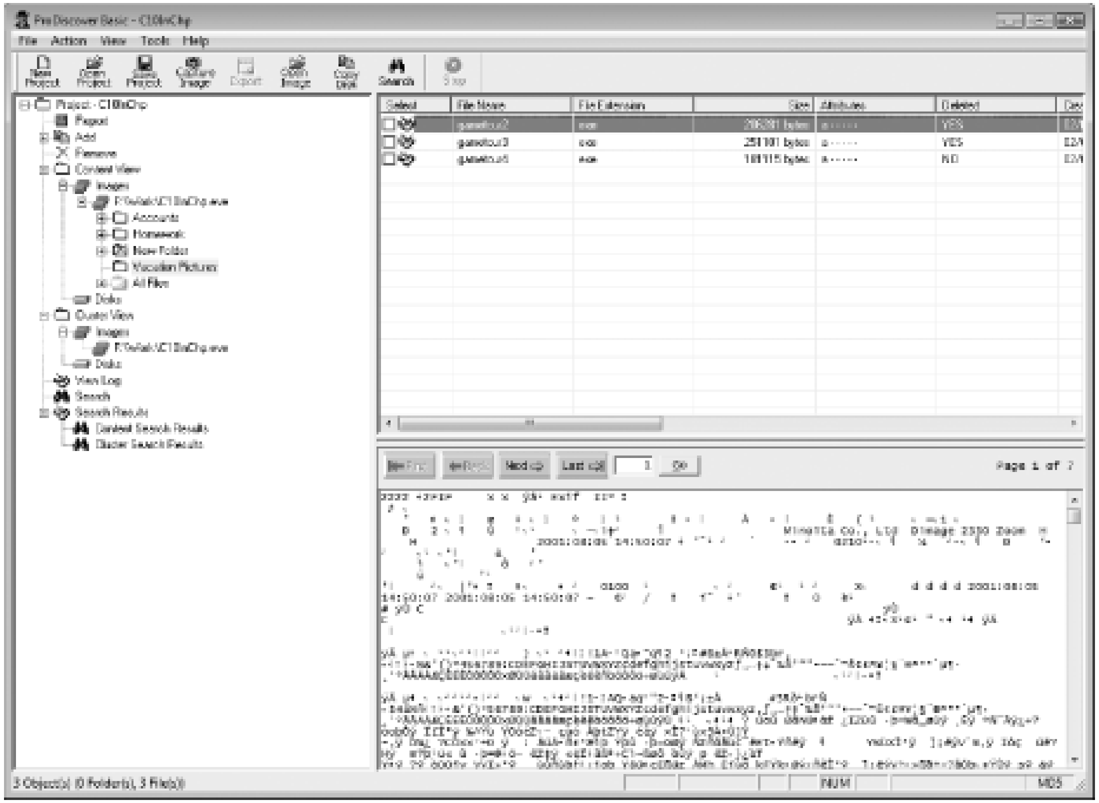
List of Sectors for the file:

F:\Work\C10InChp.eve\Vacation Pictures\gametour2.exe

```
4ca (1226)
4cb (1227)
4cc (1228)
4cd (1229)
4ce (1230)
4cf (1231)
```

Figure 10-11 Viewing all clusters used by the gametour2.exe file

Searching for and Carving Data from Unallocated Space (continued)



Select	File Name	File Extension	Size	Attributes	Deleted
□	gametour2	exe	206281 bytes	a------	YES
□	gametour3	exe	251101 bytes	a------	YES
□	gametour1	exe	181115 bytes	a------	NO

```
Minolta Co., Ltd.  DiMAGE 2300 Zoom
2001:08:05 14:50:07
2001:08:05 14:50:07 2001:08:05 14:50:07
```

Figure 10-12 Mislabeled file that appears to be altered intentionally

Rebuilding File Headers

- Try to open the file first and follow steps if you can’t see its content
- Steps
  - Recover more pieces of file if needed
  - Examine file header
    - Compare with a good header sample
    - Manually insert correct hexadecimal values
  - Test corrected file

Rebuilding File Headers (continued)

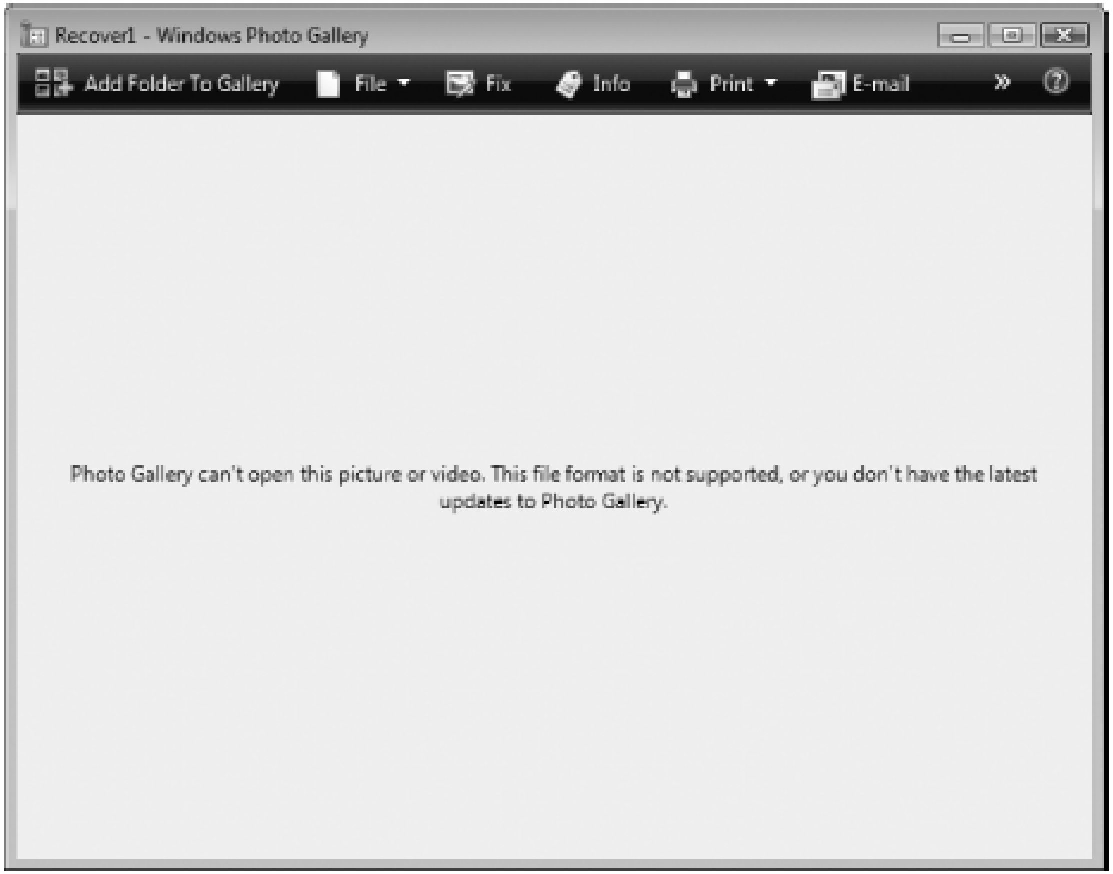
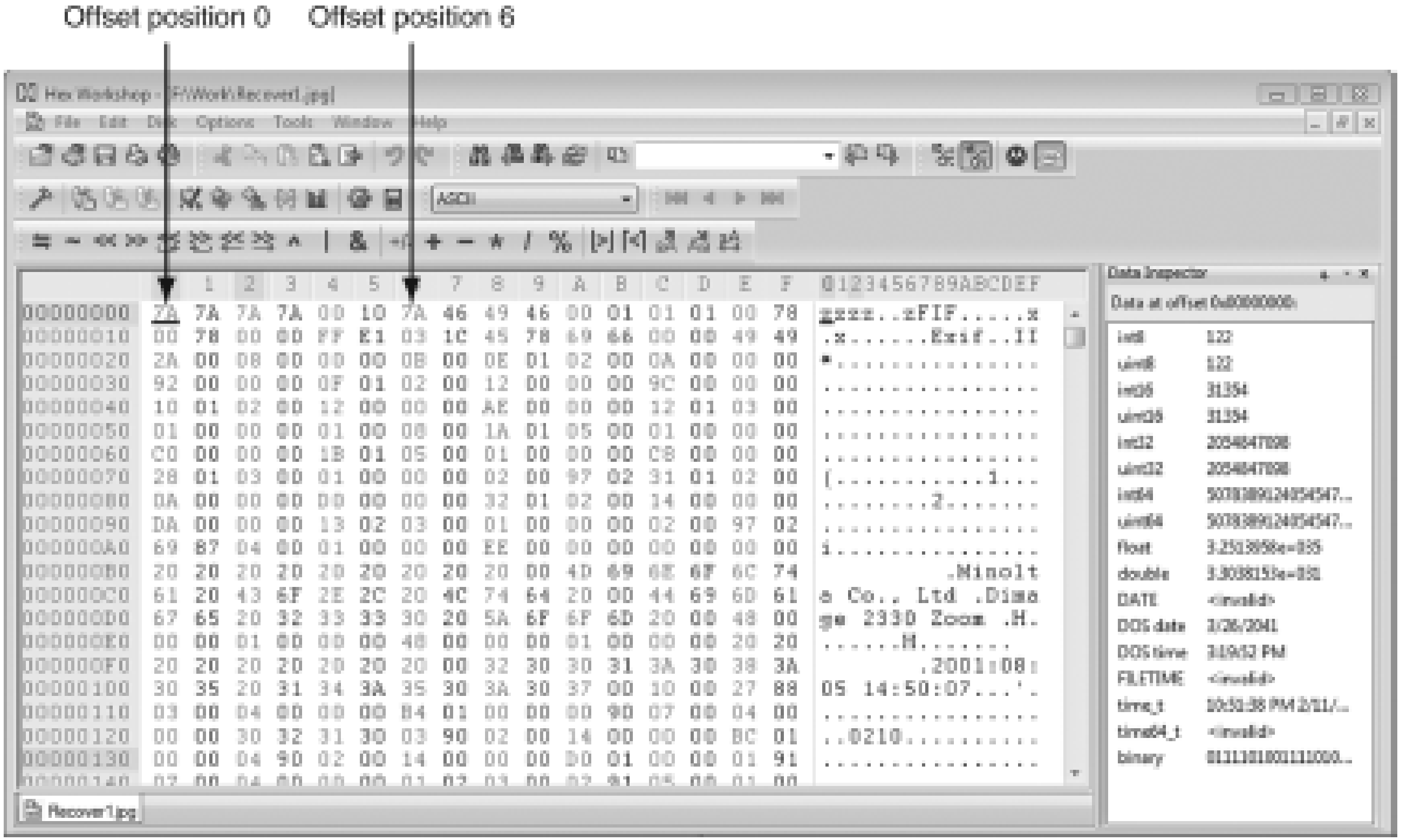


Photo Gallery can't open this picture or video. This file format is not supported, or you don't have the latest updates to Photo Gallery.

Figure 10-13 Error message indicating a damaged or an altered graphics file


Offset position 0
Offset position 6
Hex Workshop - [F:\Work\Recover1.jpg]
File Edit Disk Options Tools Window Help
ASCII
Data Inspector
Recover1.jpg

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
00000000  7A 7A 7A 7A 00 10 7A 46 49 46 00 01 01 01 00 78  zzzz..zFIF.....x
00000010  00 78 00 00 FF E1 03 1C 45 78 69 66 00 00 49 49  .x......Exif..II
00000020  2A 00 08 00 00 00 0B 00 0E 01 02 00 0A 00 00 00  *...............
00000030  92 00 00 00 0F 01 02 00 12 00 00 00 9C 00 00 00  ................
00000040  10 01 02 00 12 00 00 00 AE 00 00 00 12 01 03 00  ................
00000050  01 00 00 00 01 00 00 00 1A 01 05 00 01 00 00 00  ................
00000060  C0 00 00 00 1B 01 05 00 01 00 00 00 C8 00 00 00  ................
00000070  28 01 03 00 01 00 00 00 02 00 97 02 31 01 02 00  (...........1...
00000080  0A 00 00 00 00 00 00 00 32 01 02 00 14 00 00 00  ........2.......
00000090  DA 00 00 00 13 02 03 00 01 00 00 00 02 00 97 02  ................
000000A0  69 87 04 00 01 00 00 00 EE 00 00 00 00 00 00 00  i...............
000000B0  20 20 20 20 20 20 20 20 20 00 4D 69 6E 6F 6C 74           .Minolt
000000C0  61 20 43 6F 2E 2C 20 4C 74 64 20 00 44 69 6D 61  a Co., Ltd .Dima
000000D0  67 65 20 32 33 33 30 20 5A 6F 6F 6D 20 00 48 00  ge 2330 Zoom .H.
000000E0  00 00 01 00 00 00 48 00 00 00 01 00 00 00 20 20  ......H.......
000000F0  20 20 20 20 20 20 20 00 32 30 30 31 3A 30 38 3A         .2001:08:
00000100  30 35 20 31 34 3A 35 30 3A 30 37 00 10 00 27 88  05 14:50:07...'.
00000110  03 00 04 00 00 00 84 01 00 00 00 90 07 00 04 00  ................
00000120  00 00 30 32 31 30 03 90 02 00 14 00 00 00 BC 01  ..0210..........
00000130  00 00 04 90 02 00 14 00 00 00 D0 01 00 00 01 91  ................
00000140  02 00 04 00 00 00 01 02 03 00 02 91 05 00 01 00
```

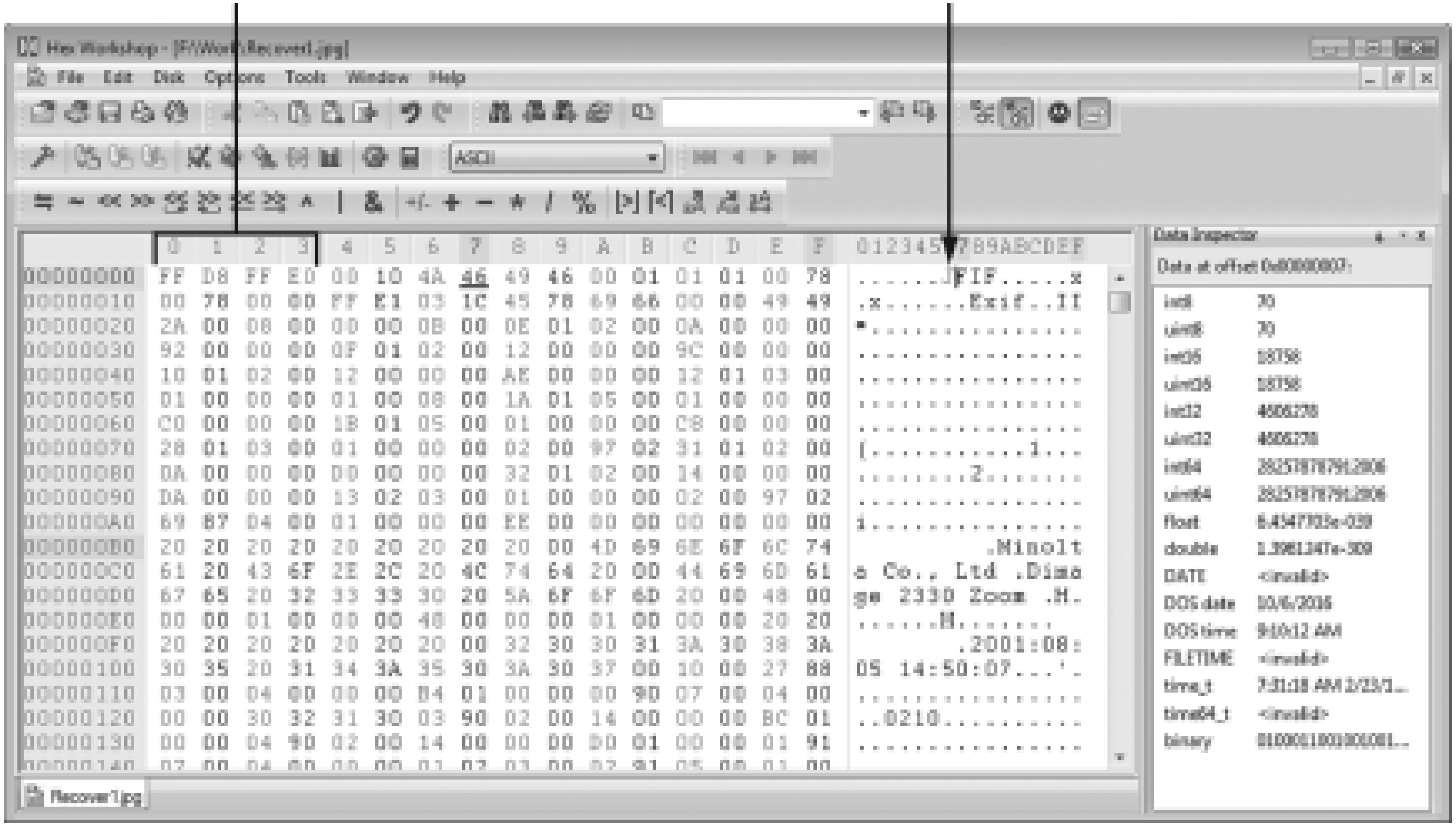
Data at offset 0x00000000:

```
int8        122
uint8       122
int16       31354
uint16      31354
int32       2054847098
uint32      2054847098
int64       5078389124054547...
uint64      5078389124054547...
float       3.2513058e+035
double      3.3038153e+281
DATE        <invalid>
DOS date    3/26/2041
DOS time    3:19:52 PM
FILETIME    <invalid>
time_t      10:51:38 PM 2/11/...
time64_t    <invalid>
binary      0111101001111010...
```

Figure 10-14 Recover1.jpg open in Hex Workshop

Insert FF D8 FF E0 starting at offset 0

Insert an uppercase J here


Hex Workshop - [F:\Work\Recover1.jpg]
File Edit Disk Options Tools Window Help
ASCII
Data Inspector
Recover1.jpg

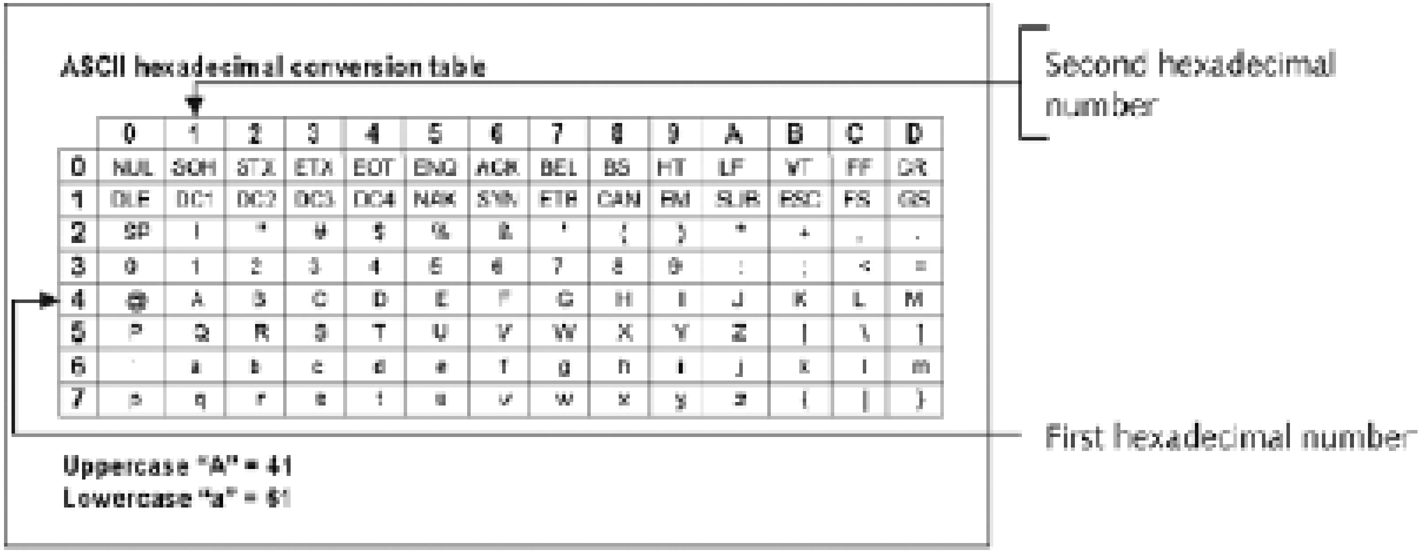
```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
00000000  FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 78  ......JFIF.....x
00000010  00 78 00 00 FF E1 03 1C 45 78 69 66 00 00 49 49  .x......Exif..II
00000020  2A 00 08 00 00 00 0B 00 0E 01 02 00 0A 00 00 00  *...............
00000030  92 00 00 00 0F 01 02 00 12 00 00 00 9C 00 00 00  ................
00000040  10 01 02 00 12 00 00 00 AE 00 00 00 12 01 03 00  ................
00000050  01 00 00 00 01 00 08 00 1A 01 05 00 01 00 00 00  ................
00000060  C0 00 00 00 1B 01 05 00 01 00 00 00 C8 00 00 00  ................
00000070  28 01 03 00 01 00 00 00 02 00 97 02 31 01 02 00  (...........1...
00000080  0A 00 00 00 D0 00 00 00 32 01 02 00 14 00 00 00  ........2.......
00000090  DA 00 00 00 13 02 03 00 01 00 00 00 02 00 97 02  ................
000000A0  69 87 04 00 01 00 00 00 EE 00 00 00 00 00 00 00  i...............
000000B0  20 20 20 20 20 20 20 20 20 00 4D 69 6E 6F 6C 74            .Minolt
000000C0  61 20 43 6F 2E 2C 20 4C 74 64 20 00 44 69 6D 61  a Co., Ltd .Dima
000000D0  67 65 20 32 33 33 30 20 5A 6F 6F 6D 20 00 48 00  ge 2330 Zoom .H.
000000E0  00 00 01 00 00 00 48 00 00 00 01 00 00 00 20 20  ......H.......
000000F0  20 20 20 20 20 20 20 00 32 30 30 31 3A 30 38 3A         .2001:08:
00000100  30 35 20 31 34 3A 35 30 3A 30 37 00 10 00 27 88  05 14:50:07...'.
00000110  03 00 04 00 00 00 B4 01 00 00 00 90 07 00 04 00  ................
00000120  00 00 30 32 31 30 03 90 02 00 14 00 00 00 BC 01  ..0210..........
00000130  00 00 04 90 02 00 14 00 00 00 D0 01 00 00 01 91  ................
00000140  07 00 04 00 00 00 01 02 03 00 02 91 05 00 01 00
```

Data at offset 0x00000007:

```
int8        70
uint8       70
int16       18758
uint16      18758
int32       4606278
uint32      4606278
int64       28257878791,2806
uint64      28257878791,2806
float       6.4547703e-039
double      1.3981347e-309
DATE        <invalid>
DOS date    10/6/2016
DOS time    9:10:12 AM
FILETIME    <invalid>
time_t      7:31:18 AM 2/23/1...
time64_t    <invalid>
binary      0100011001001001...
```

Figure 10-15 Inserting correct hexadecimal values for a JPEG file

Rebuilding File Headers (continued)



ASCII hexadecimal conversion table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}

Uppercase "A" = 41
Lowercase "a" = 61

Second hexadecimal number
First hexadecimal number

Figure 10-16 ASCII equivalents of hexadecimal values

Rebuilding File Headers (continued)

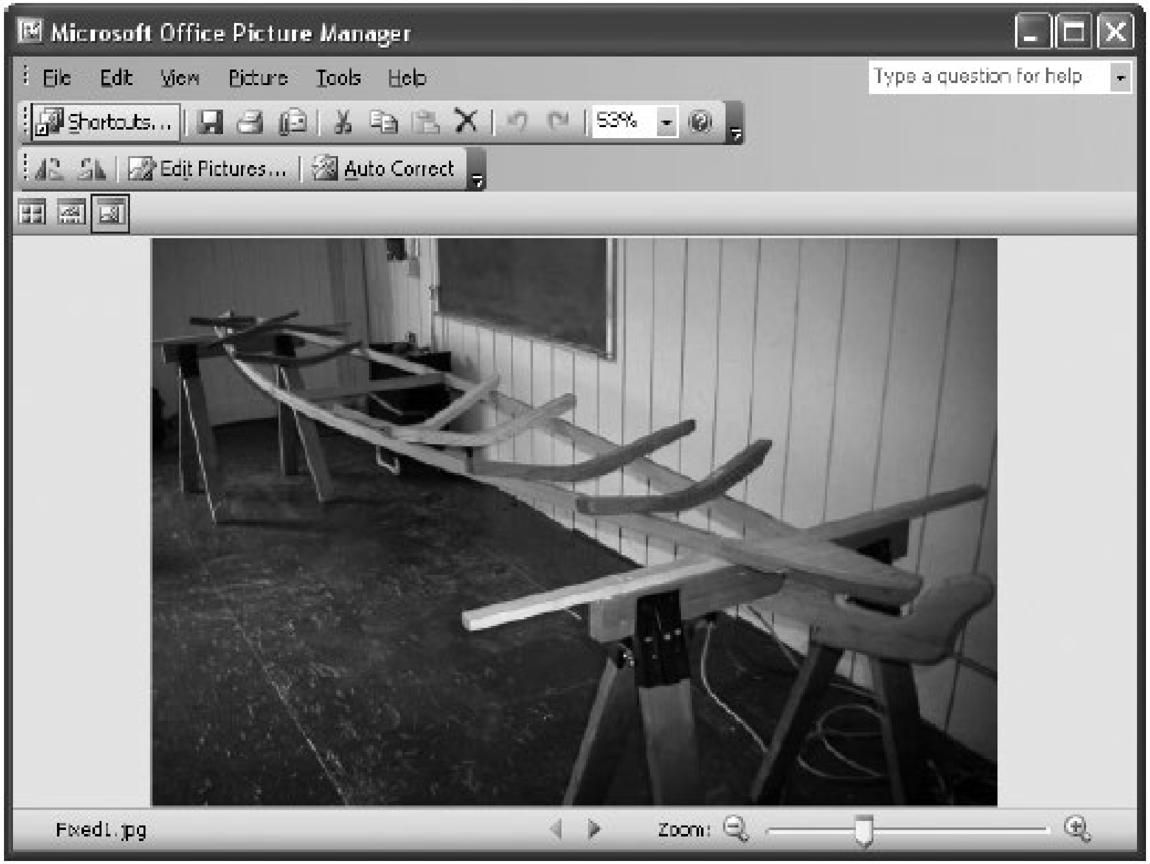
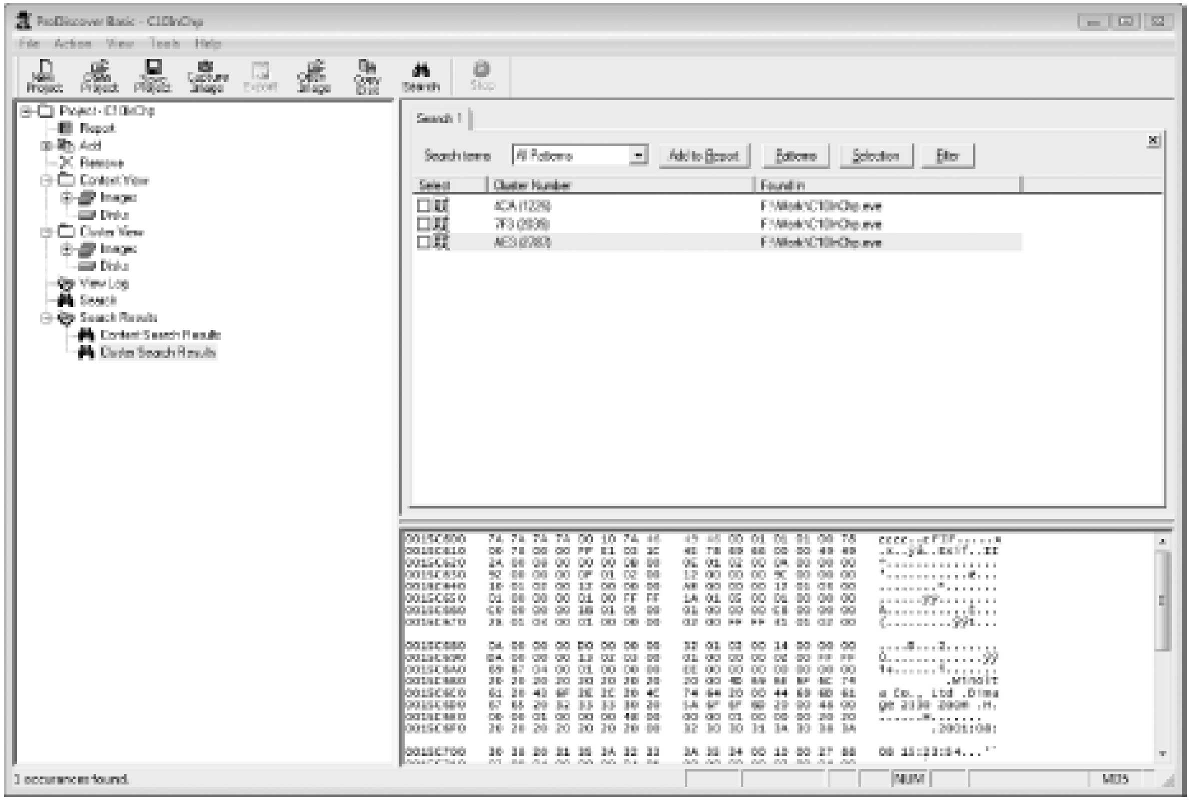


Figure 10-17 Fixed1.jpg open in Microsoft Office Picture Manager

Reconstructing File Fragments

- Locate the starting and ending clusters
  - For each fragmented group of clusters in the file
- Steps
  - Locate and export all clusters of the fragmented file
  - Determine the starting and ending cluster numbers for each fragmented group of clusters
  - Copy each fragmented group of clusters in their proper sequence to a recovery file
  - Rebuild the corrupted file’s header to make it readable in a graphics viewer

Reconstructing File Fragments (continued)



Search items: All Patterns

Select	Cluster Number	Found in
□	4CA (1226)	F:\Work\C10InChp.eve
□	7F3 (2035)	F:\Work\C10InChp.eve
□	AE3 (2787)	F:\Work\C10InChp.eve

1 occurrences found.

Figure 10-18 Cluster search results for the AE3(2787) cluster

Reconstructing File Fragments (continued)

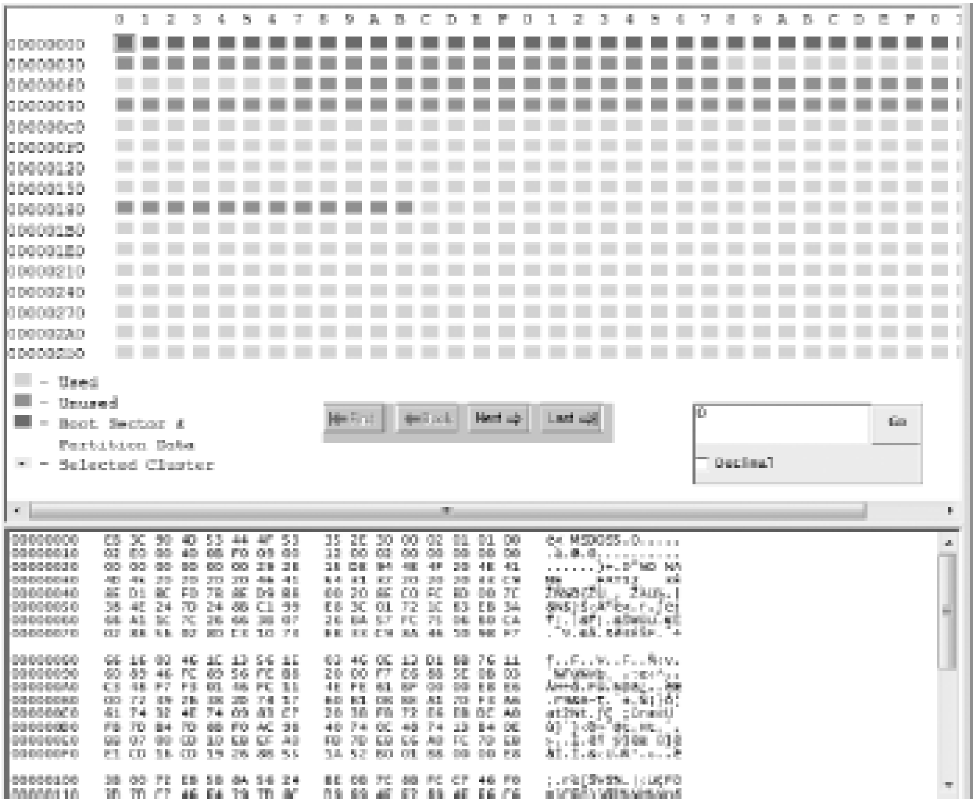


Figure 10-19 Cluster view of C10InChp.eve

Reconstructing File Fragments (continued)

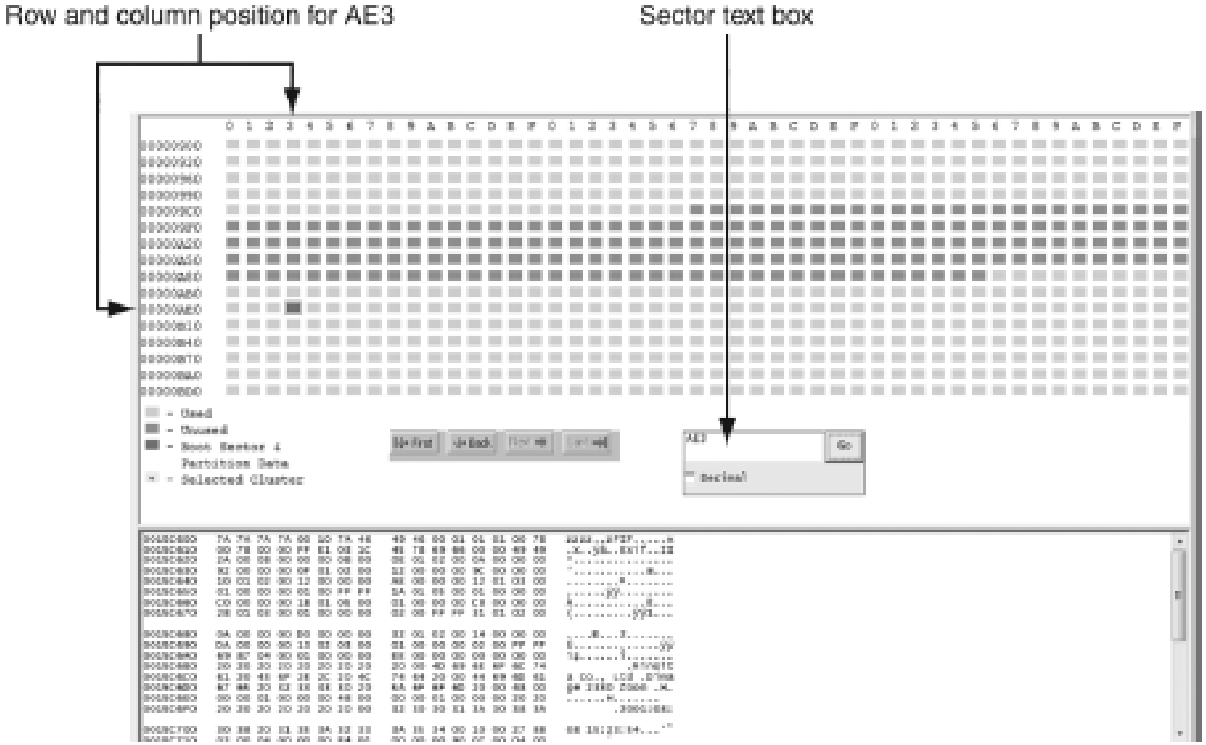
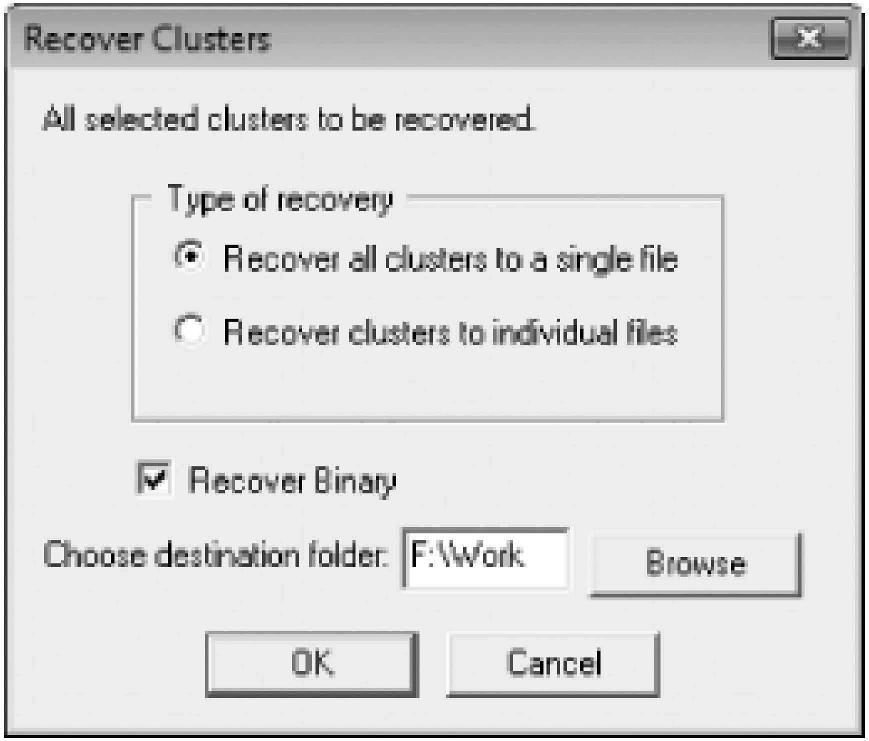


Figure 10-20 Cluster view of sector AE3

Reconstructing File Fragments (continued)


Recover Clusters

All selected clusters to be recovered.

Type of recovery

● Recover all clusters to a single file

○ Recover clusters to individual files

☑ Recover Binary

Choose destination folder: F:\Work

Figure 10-22 Copying all selected clusters or sectors to a file

Reconstructing File Fragments (continued)

- Remember to save the updated recovered data with a .jpg extension
- Sometimes suspects intentionally corrupt cluster links in a disk's FAT
  - Bad clusters appear with a zero value on a disk editor

Reconstructing File Fragments (continued)

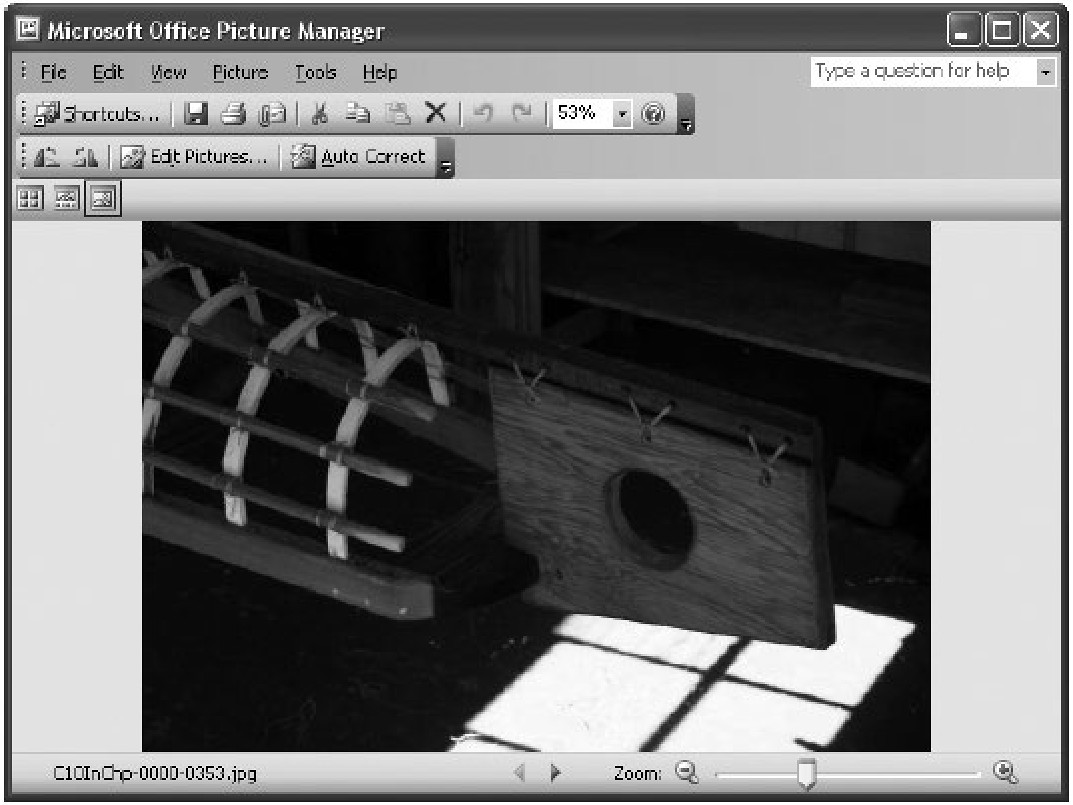
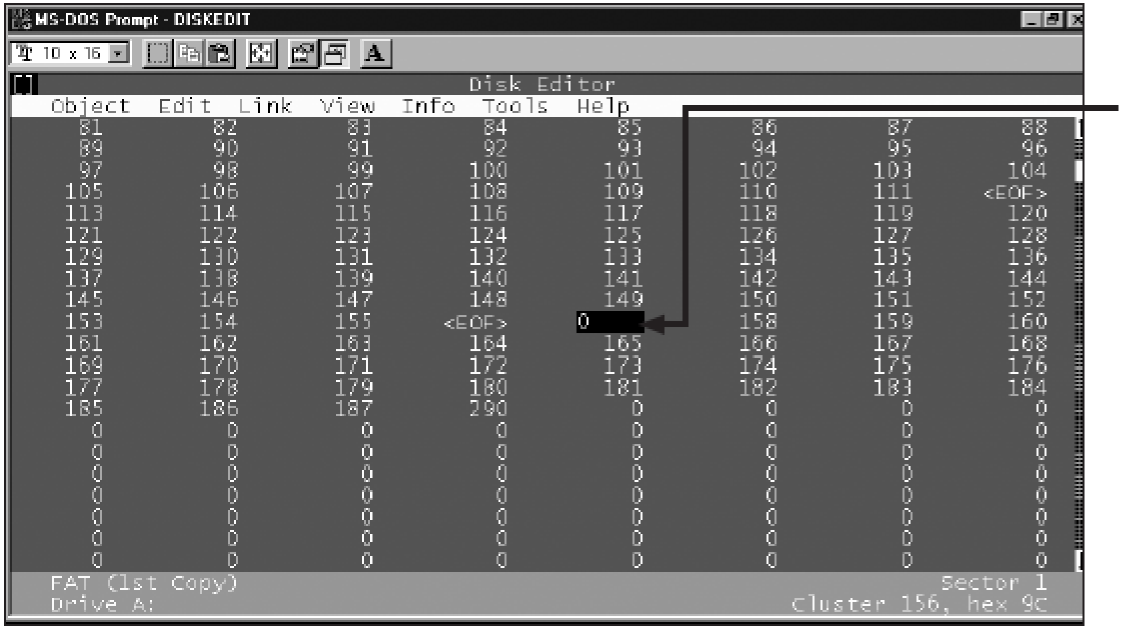


Figure 10-23 Recovered data from starting sector AE3 after Hex Workshop corrects the header

Reconstructing File Fragments (continued)



```
     81        82        83        84        85        86        87        88
     89        90        91        92        93        94        95        96
     97        98        99       100       101       102       103       104
    105       106       107       108       109       110       111     <EOF>
    113       114       115       116       117       118       119       120
    121       122       123       124       125       126       127       128
    129       130       131       132       133       134       135       136
    137       138       139       140       141       142       143       144
    145       146       147       148       149       150       151       152
    153       154       155     <EOF>   0             158       159       160
    161       162       163       164       165       166       167       168
    169       170       171       172       173       174       175       176
    177       178       179       180       181       182       183       184
    185       186       187       290         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
FAT (1st Copy)                                                      Sector 1
Drive A:                                                Cluster 156, hex 9C
```

Cluster marked as bad

Figure 10-24 Bad cluster appearing as 0 in Norton DiskEdit

Identifying Unknown File Formats

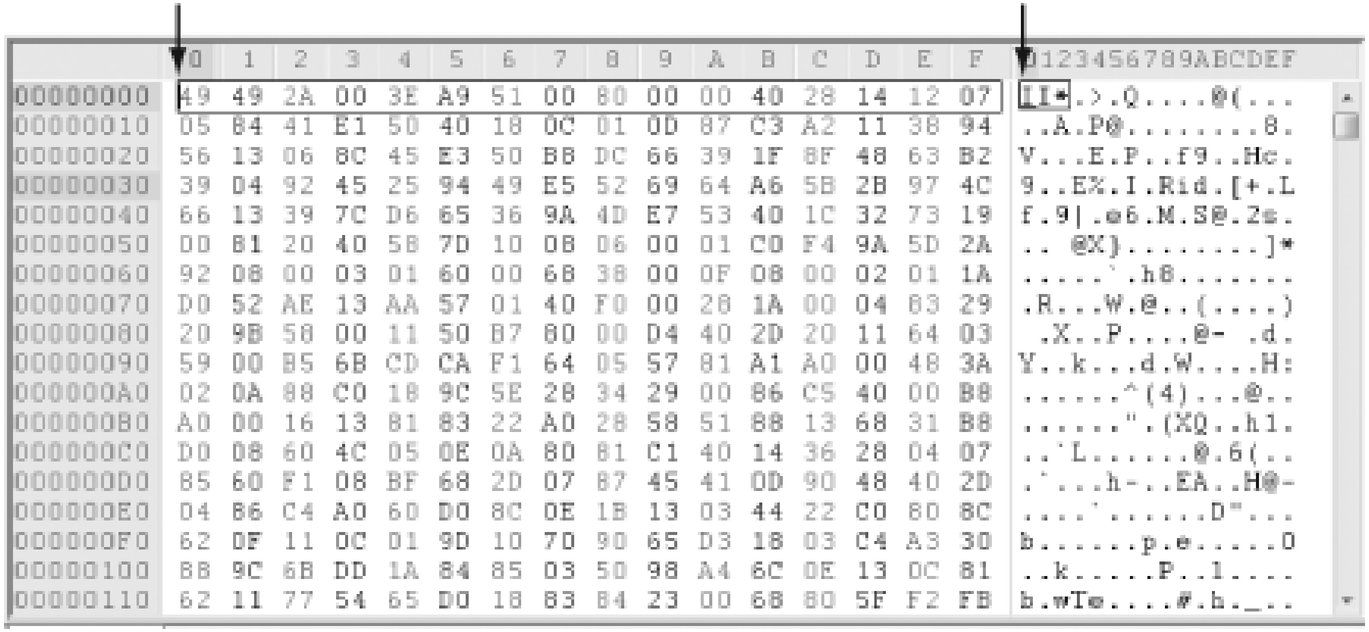
- The Internet is the best source
  - Search engines like Google
  - Find explanations and viewers
- Popular Web sites
  - www.digitek-asi.com/file_formats.html
  - www.wotsit.org
  - http://whatis.techtarget.com

Analyzing Graphics File Headers

- Necessary when you find files your tools do not recognize
- Use hex editor such as Hex Workshop
  - Record hexadecimal values on header
- Use good header samples

Analyzing Graphics File Headers (continued)

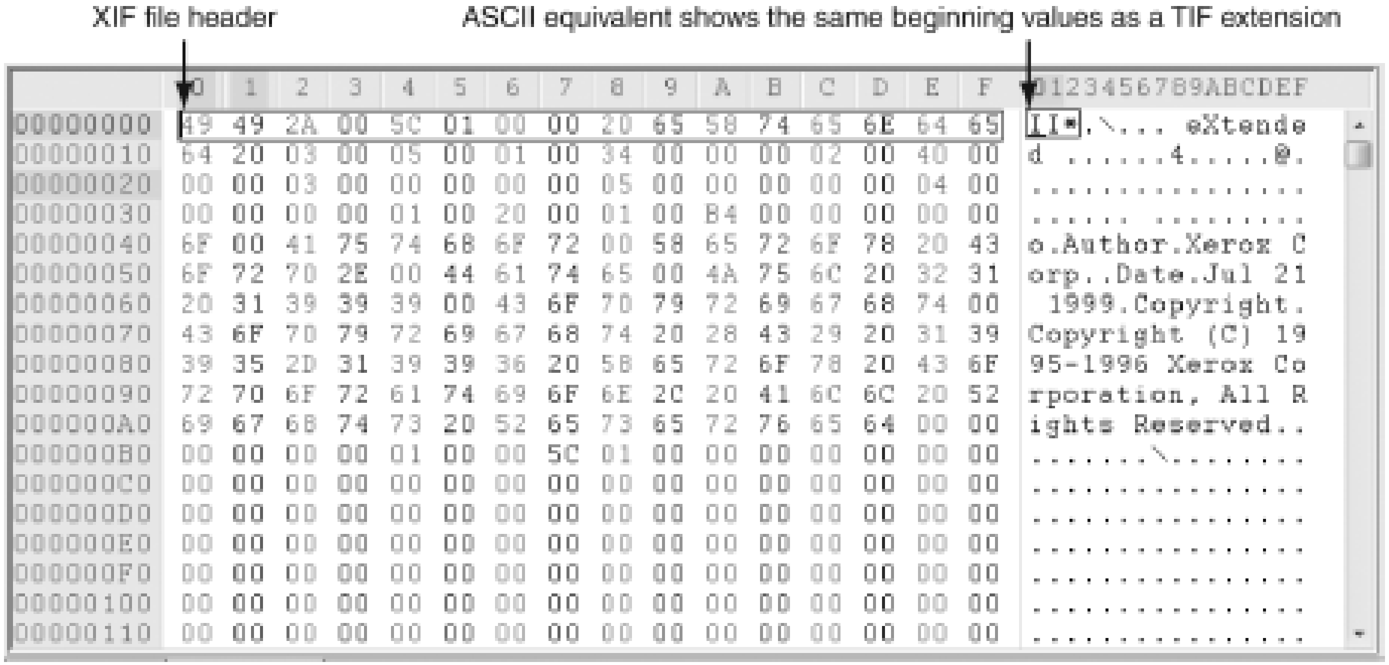
TIF file headers start with hexadecimal 49 49 2A, equivalent to ASCII II



```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
00000000  49 49 2A 00 3E A9 51 00 80 00 00 40 28 14 12 07  II*.>.Q....@(...
00000010  05 B4 41 E1 50 40 18 0C 01 0D 87 C3 A2 11 38 94  ..A.P@........8.
00000020  56 13 06 8C 45 E3 50 BB DC 66 39 1F 8F 48 63 B2  V...E.P..f9..Hc.
00000030  39 D4 92 45 25 94 49 E5 52 69 64 A6 5B 2B 97 4C  9..E%.I.Rid.[+.L
00000040  66 13 39 7C D6 65 36 9A 4D E7 53 40 1C 32 73 19  f.9|.e6.M.S@.2s.
00000050  00 B1 20 40 5B 7D 10 0B 06 00 01 CD F4 9A 5D 2A  .. @X}........]*
00000060  92 08 00 03 01 60 00 6B 38 00 0F 08 00 02 01 1A  .....`.h8.......
00000070  D0 52 AE 13 AA 57 01 40 F0 00 28 1A 00 04 83 29  .R...W.@..(....)
00000080  20 9B 58 00 11 50 B7 80 00 D4 40 2D 20 11 64 03   .X..P....@- .d.
00000090  59 00 B5 6B CD CA F1 64 05 57 81 A1 A0 00 48 3A  Y..k...d.W....H:
000000A0  02 0A 88 C0 18 9C 5E 28 34 29 00 86 C5 40 00 B8  ......^(4)...@..
000000B0  A0 00 16 13 81 83 22 A0 28 58 51 88 13 68 31 B8  ......".(XQ..h1.
000000C0  D0 08 60 4C 05 0E 0A 80 81 C1 40 14 36 28 04 07  ..`L......@.6(..
000000D0  85 60 F1 08 BF 68 2D 07 87 45 41 0D 90 48 40 2D  .`...h-..EA..H@-
000000E0  04 B6 C4 A0 60 D0 8C 0E 1B 13 03 44 22 C0 80 8C  ....`......D"...
000000F0  62 DF 11 0C 01 9D 10 70 90 65 D3 18 03 C4 A3 30  b......p.e.....0
00000100  88 9C 6B DD 1A 84 85 03 50 98 A4 6C 0E 13 0C 81  ..k.....P..l....
00000110  62 11 77 54 65 D0 18 83 84 23 00 68 80 5F F2 FB  b.wTe....#.h._..
```

Figure 10-25 A TIF file open in Hex Workshop

Analyzing Graphics File Headers (continued)



XIF file header

ASCII equivalent shows the same beginning values as a TIF extension

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
00000000  49 49 2A 00 5C 01 00 00 20 65 58 74 65 6E 64 65  II*.\... eXtende
00000010  64 20 03 00 05 00 01 00 34 00 00 00 02 00 40 00  d .......4.....@.
00000020  00 00 03 00 00 00 00 00 05 00 00 00 00 00 04 00  ................
00000030  00 00 00 00 01 00 20 00 01 00 B4 00 00 00 00 00  ...... .........
00000040  6F 00 41 75 74 68 6F 72 00 58 65 72 6F 78 20 43  o.Author.Xerox C
00000050  6F 72 70 2E 00 44 61 74 65 00 4A 75 6C 20 32 31  orp..Date.Jul 21
00000060  20 31 39 39 39 00 43 6F 70 79 72 69 67 68 74 00   1999.Copyright.
00000070  43 6F 70 79 72 69 67 68 74 20 28 43 29 20 31 39  Copyright (C) 19
00000080  39 35 2D 31 39 39 36 20 58 65 72 6F 78 20 43 6F  95-1996 Xerox Co
00000090  72 70 6F 72 61 74 69 6F 6E 2C 20 41 6C 6C 20 52  rporation, All R
000000A0  69 67 68 74 73 20 52 65 73 65 72 76 65 64 00 00  ights Reserved..
000000B0  00 00 00 00 01 00 00 5C 01 00 00 00 00 00 00 00  .......\........
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 10-26 An XIF file open in Hex Workshop

Tools for Viewing Images

- Use several viewers
  - ThumbsPlus
  - ACDSee
  - QuickView
  - IrfanView
- GUI forensics tools include image viewers
  - ProDiscover
  - EnCase
  - FTK
  - X-Ways Forensics
  - iLook

Summary

- Introduction
- Windows, MAC, Linux File Format
- File System
- File & Magic Number
- Carving Concept
- Case Study: Recovering Image File.