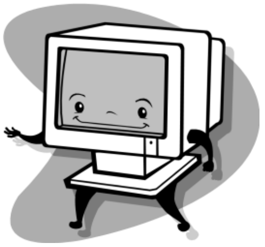
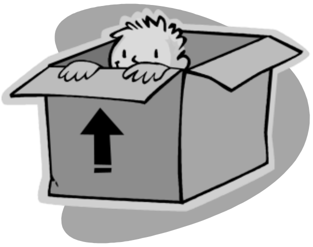


Computer



Metadata

File

Electronic Evidence Workbook

CLASS 1-8

2022

CLASS 1-8

University of Texas Schools of Computer Science, Information Science and Law

LAW386N, INF385T, CS395T: Digital Evidence & Discovery

Spring 2022 Ver. 22.0112A © Craig Ball, All Rights Reserved

Name:

THIS PAGE INTENTIONALLY BLANK

Course Workbook Reading Assignments: Classes 1-8

NOTE: This table is for your convenience; but the timing and scope of your responsibilities in this course are established by the latest Syllabus, not this table. ***Always* consult the latest Syllabus!! The** following Workbook exercises must be submitted, and readings completed, **prior to the start of class** on the stated dates. Late submissions will not be credited without advance permission.

Clas 1: Wednesday, January 19
Workbook: Read pp. 1-58; Complete Exercise 1, p. 58; Once you complete your Workbook reading, please watch the ten-minute video here: https://www.youtube.com/watch?v=oyWsdS1h-TM and the four-minute video here: https://d34zi8nxray0rk.cloudfront.net/102695601-240.mp4

Class 2: Wednesday, February 26
Workbook: Read pp. 59-94; Complete Exercises 2, 3 and 4
Canvas: *Green v. Blitz* w/ Preface (14 pp.); THE SEDONA PRINCIPLES: THIRD EDITION, *pages 51-54*

Class 3: Wednesday, February 2
Workbook: Read pp. 95-158; Review TRCP Rule 196.4 p. 442; Complete Exercises 5 and 6
Canvas: *In Re: Weekley Homes*

Class 4: Wednesday, February 9
Workbook: Read pp. 159-218**;** Complete Exercises 7, 8 and 9

Class 5: Wednesday, February 16
Workbook: Read pp. 219-249; Complete Exercises 10 and 11
Canvas: *Zubulake* Cases I-V with Preface, *focus on decisions I and III*

Class 6: Wednesday, February 23
Workbook: Read pp. 250-307**;** Complete Exercise 12

Class 7: Wednesday, March 2
Workbook: Read pp. 308-366**;** Complete Exercises 13-15
Watch the 5-minute video here: https://www.youtube.com/watch?v=enQ-zrNSSM4
Read this animated graphic: https://animagraffs.com/hard-disk-drive/

Class 8: Wednesday, March 9
Workbook: Read pp. 367-394; Complete Exercise 16;
Watch this 18-minute video: https://www.youtube.com/watch?v=5Mh3o886qpg

FURTHER ASSIGNMENTS AND A REVISED WORKBOOK TO FOLLOW AFTER SPRING BREAK

Contents

Goals for this Workbook

The goal of this course is to change the way you think about electronically stored information and digital evidence. Despite its daunting complexity, all digital content—photos, music, documents, spreadsheets, databases, social media and communications—exist in one common and mind-boggling form: *Almost all the information in the world exists as faint electric charges or impossibly tiny reversals of magnetic polarity.* Most of these minute polar fluctuations are read by a detector flying above the surface of a spinning disk *on a cushion of air one-thousandth the width of a human hair* in an operation akin to a jet fighter flying around the world at more than 800 times the speed of sound, less than a millimeter above the ground…*and precisely counting every blade of grass it passes!*

That's astonishing, but what should astound you more is that there are *no pages, paragraphs or markers of any kind* to define the data stream. It's the history, knowledge and creativity of humankind distilled to two different states (on/off…one/zero) as a continuous, featureless expanse. It's a data stream that carries not only the information we store but all the *instructions* needed to make sense of the data as well--all the information *about* the data required to play it, display it, transmit it or otherwise put it to work. It's a reductive feat that'll make your head spin…and make you want to buy a computer scientist a drink.

Yet, it should comfort you to know that no matter the volume or variety of digital electronic evidence, electronic evidence is more alike than different. Digital evidence is daunting, but it's far easier to identify, preserve, collect, search, process, review, authenticate, challenge, and produce once you see its common threads. That's why information technologists are prone to dismiss overblown claims of burden with the observation, "It's just data."

These exercises and readings seek to get you thinking about the fantastic journey data takes from its seamless notation as ones and zeroes to the vast variety of documents, communications, records, recordings, and formats that confront lawyers in litigation. You are encouraged to ask "how" and "why?" Especially, "*How does it work*" and "*Why will knowing how benefit me?*"

These exercises and readings will help you answer such questions as:

1. How do computers store data?
2. What are the differences between common storage media?
3. What's the difference between system and application metadata?
4. What's cryptographic hashing and how do I use it?
5. How do computers encode data?
6. How do computers use binary signatures and file extensions to distinguish file types?
7. How are foreign languages encoded differently from English?
8. What are the seen and unseen elements of an e-mail message?

9. Where does deleted data go if it's not really gone?
10. How are deleted data recovered?
11. How are data preserved, processed and presented for attorney review?
12. How do we cull and filter to reduce data volumes and isolate relevant evidence?
13. How do search tools and review platforms operate?
14. What is forensic imaging?
15. What are load files, how are they used and what challenges do they present?
16. What impact do alternate forms of production have upon cost and utility?

Some of these questions may seem removed from the day-to-day practice of law; but they are essential building blocks in a solid foundation for development of practical skills. In any discipline, the absence of a solid foundation ultimately limits how high you can go.

This is exciting stuff! Consider how much of our lives are lived online via digital devices! Consider the extent we are instrumented and networked and tracked. We are more telemetered today than the Apollo astronauts of a generation ago! Never in human history has there been so much probative and reliable evidence to draw on to help us find truth. *We are so lucky!*

Thank you for enrolling in this class. Together, I hope we can make it a course of study you will value and enjoy.

Craig Ball, January 12, 2022

About the Author

I'm a Texas trial lawyer and certified computer forensic examiner. I was lead trial counsel in a personal injury and products liability practice for twenty-five years. I've taught digital evidence for eleven years. My professional interests include digital forensics, emerging technologies, visual persuasion, e-discovery, and trial tactics. I limit my law practice to service as a court-appointed Special Master and consultant in Electronically Stored Information. I've spoken thousands of times at educational programs for the bench and bar and served as an instructor in computer forensics and electronic evidence to multiple law enforcement and security agencies around the world. For nine years, I wrote a syndicated column on computer forensics and e-discovery for American Lawyer Media called "*Ball in your Court*" and my writing still appear in the national media. You'll find my blog posts at ballinyourcourt.com, other writings at craigball.com and me in the flesh whistling a happy tune in crazy/wonderful New Orleans when I'm not flying somewhere to teach or explore or, lately, hunkering down for COVID. You can reach me as craig@ball.net or call/text me at 713-320-6066. ***I'm here to help you, so reach out!***

Why E-Discovery and Digital Evidence?

The passing mention made of discovery during first year civil procedure classes cannot prepare law students to grasp the extent to which discovery devours litigators' lives. For every hour spent in trial, attorneys and trial teams devote hundreds or thousands of hours to discovery and its attendant disputes.

Too, discovery is a trial lawyer's most daunting ethical challenge. It demands lawyers seek and surrender information providing aid and comfort to the enemy—over the objections of clients, irrespective of the merits of the case, and no matter how much they distrust or detest the other side. Is there a corollary duty to act against interest in any other profession?

Discovery is hard because it runs counter to human nature, and electronic discovery is harder because it demands a specialized knowledge and experience few lawyers possess and far afield of conventional legal scholarship. E-discovery skills, as much as they've been key to lawyer competency for decades, are yet apt to be denigrated or delegated.

Civil discovery is a high-stakes game of "Simon Says." Counsel must phrase demands for information with sufficient precision to implicate what's relevant, yet with adequate breadth to forestall evasion. It's as confounding as it sounds, making it miraculous that discovery works as well as it does. The key factors making it work are counsel's professional integrity and judges' enforcement of the rules.

Counsel's professional integrity isn't mere altruism; the failure to protect and produce relevant evidence carries consequences ranging from damaged professional reputations to costly remedial actions to so-called "death penalty" sanctions, where a discovery cheater forfeits the right to pursue or defend a claim. Lawyers may face monetary sanctions and referral to disciplinary authorities.

The American system of civil discovery embodies the principle that just outcomes are more likely when parties to litigation have access to facts established by relevant evidence. Since relevant evidence often lies within the exclusive province of those whose interests are not served by disclosure, justice necessitates a means to compel disclosure, subject to exceptions grounded on claims of privilege, privacy, and proportionality.

As noted, the U.S. Federal Rules of Civil Procedure articulate the scope of discovery as, "Parties may obtain discovery regarding any nonprivileged matter that is relevant to any party's claim or defense and proportional to the needs of the case…." Adding, "Information within this scope of discovery need not be admissible in evidence to be discoverable." Rule 401 of the Federal Rules of Evidence defines evidence as relevant if it has any tendency to make a fact more or less probable than it would be without the evidence and the fact is of consequence in determining the action (*i.e.,* the fact is *material*).

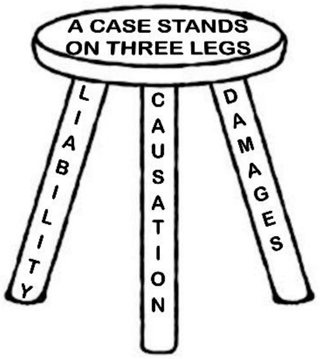
Relevant. Proportional. Nonprivileged. Commit these touchstones to memory as we will return to them often.

The discovery of an opponent's electronically stored information begins with a request for production under Rule 34 of the Federal Rules of Civil Procedure or a similar state rule of procedure. Rule 34 lets a party request any other party produce any designated documents or electronically stored information—including writings, drawings, graphs, charts, photographs, sound recordings, images, and other data or data compilations—in the responding party's possession, custody, or control. The responding party must respond to the request in writing within 30 days and may lodge specific objections and withhold production pursuant to those objections.

The simplicity of the rule hardly hints at its complexity in practice. A multibillion-dollar industry of litigation service providers and consultants exists to support discovery, and a crazy quilt of court rulings lays bare the ignorance, obstinance, guile, and ingenuity of lawyers and clients grappling with the preservation and exchange of electronic evidence.

To appreciate what competent counsel must know about digital discovery, consider the everyday case where a customer slips and falls in a grocery store. A store employee witnesses the fall, helps the customer up and escorts her to the store manager, who prepares a written incident report. The customer claims the fall was caused by a pool of grease on the floor alongside a display of roasted chickens. The customer returns home but feels enough pain to visit an emergency room the next day. After months of medication and therapy, doctors diagnose a spinal injury necessitating surgery. When the grocery store refuses to pay for medical care, the customer hires a lawyer to seek compensation.

From the standpoint of relevance in discovery, the case will stand on three legs: **liability, causation** and **damages**.



To establish liability, tort law requires the plaintiff demonstrate duty and a breach of that duty. The store owes customers a duty to furnish reasonably safe premises and to act reasonably to correct or warn of an unsafe condition like slippery chicken fat on the floor. Yet, the store's personnel must be *aware* of the condition to be obliged to correct or warn of the hazard *or* the defect must be present for a sufficient time that a reasonable store *should* have become aware of the hazard and protected its customers.

The store defends against liability by asserting that there was no grease on the floor and, alternatively, that any grease on the floor was spilt by another customer and, despite exercising reasonable care, the store lacked the opportunity to find and clean up the spill before the fall. The store also asserts the plaintiff failed to watch where she was walking, contributing to cause her injuries. Finally, the store contests causation and damages, arguing that the plaintiff exaggerates the extent of her injuries and something other than the fall—perhaps a pre-existing condition or an unrelated trauma—is the true cause of plaintiff's complaints.

As plaintiff's counsel ponders the potentially relevant evidence in the store's control, he wonders:

1. Who might have witnessed the fall or the conditions?
2. Were witness statements obtained?
3. How did the store clean up after the fall?
4. Were photographs taken?
5. Were video cameras monitoring the premises?
6. Is there a history of other falls?
7. Did the roasted chicken display leak?
8. How frequently are the floors inspected and cleaned?

Defense counsel has her own questions:

1. Did the plaintiff stage the fall to profit from a claim?
2. Did the plaintiff suffer from a pre-existing condition?
3. Has the plaintiff made other claims?
4. Was the plaintiff impaired by drink, drugs or disability?
5. Has the plaintiff behaved inconsistently with her claimed infirmities?

Both sides worry whether the other side acted diligently to preserve relevant evidence and if anyone has altered or destroyed probative material. In gauging proportionality, comparable cases have prompted damage awards ranging from one-half million to two million dollars.

The store is part of a national chain, so there are detailed policies and procedures setting out how to police and document the premises for hazards and deal with injuries on the property. There's an extensive network of digital video cameras throughout the store, warehouse, and parking lot. A database logs register sales, and all self-checkout scanners incorporate cameras. Employees clock in and out of their shifts digitally. Multiple suppliers and subcontractors come and go daily. Virtually everyone carries a cell phone or other device tracking geolocation and exertion. A corporate database serves to manage claims, investigations, and dispositions. Even a simple fall on schmaltz casts a long shadow of electronic artifacts.

Video of the fall and the area where it occurred is crucial evidence. Store policy required a manager review and preserve video of the event before recordings overwrite every 14 days. The manager reviewed the store video and, from one of the deli-area feeds, kept footage beginning one minute before the fall until five minutes afterward, when a store employee led the plaintiff away, but before cleanup occurred. In the video, another kiosk obstructs the view of the floor. The manager also preserved video of the plaintiff arriving and leaving the premises. In one, plaintiff is looking at her phone. The surveillance system overwrote other video recordings two weeks later.

The manager photographed the area showing the condition of the floor but arrived after employees mopped and placed yellow caution cones. The store's counsel claims staff mopped because the plaintiff dropped a chicken she'd selected, spilling grease when she fell, not because there was any grease already on the floor.

The parties engage in discovery seeking the customary complement of medical records and expenses, lost earnings documentation, store policies and procedures, similar prior incidents, and incident investigations.

Seeking to identify eyewitnesses or others who may have spilled grease buying roast chicken, plaintiff requests the store "produce for a period one hour before and after the fall, any photographic or transaction record (including credit- and loyalty-card identifying data) of any persons on the premises." Plaintiff makes the same request for "any persons who purchased roast chicken." Plaintiff also demands the names, addresses, and phone numbers of employees or contractors on the premises within one hour on either side of her fall.

In its discovery, the store asks that plaintiff "produce any texts, call records, application data or other evidence of phone usage for one hour before and after the alleged fall and the contents of any social networking posts for six months prior to the alleged injury to the present where any content, comment, or imagery in the post touches or concerns the Plaintiff's state of mind, physical activity, or consumption of drugs or alcohol." The store also demands that plaintiff produce "data from any devices (including, but not limited to, phones, apps, fitness equipment, fitness monitors, and smart watches) that record or report information about the plaintiff's sleep, vital signs, activity, location, movement, or exertion from six months prior to the alleged fall to the present date."

Chances are both sides will balk at production of the electronically stored data, and it will eventually emerge that neither side considered the data sought when obliged to preserve potentially relevant evidence in anticipation of litigation. The parties will meet and confer, seeking to resolve the dispute; but when they don't arrive at a compromise narrowing the scope of the requests, both sides will file Motions to Compel asking the Court to order their opponent to hand over the information sought.

The parties will object on various grounds, alleging that the information isn't relevant, doesn't exist, or is not reasonably accessible. Lawyers will point to undue burden and cost, oppression, excessive inroads into private matters, and even claims the data requested is privileged or a trade secret. Requests will be challenged as "disproportionate to the needs of the case."

One side assures the judge it's just a few clicks to gather the data sought. With equal certainty, the other side counters that the task requires teams of expensive experts and months of programming and review.

Plaintiff's counsel points out that every roast chicken sold the day of the fall bore a Universal Product Code (UPC) scanned at a register to establish its price and update the store's inventory control system. Thus, every roast chicken sale was logged and the name of every buyer who used a credit, debit, loyalty, or EBT/SNAP assistance card was likewise recorded. "It's right there on the register receipts," counsel argues, "Just print them out." "It's the same for every employee," he adds, "they scan people in and out like roast chickens."

Plaintiff is less sanguine about the defense's demand for phone, social networking, and fitness monitor evidence, uncertain how to collect, review, and produce whatever's not been lost to the passage of time. “It’s going to take forever to look at it all,” she protests, “and who knows if there’s anything relevant? It’s disproportional!”

The defendant concedes it tracks purchases and card usage, but not in the same system. The store claims it can’t pair the transactions and, if they produce the names, will those buyers prove to be eyewitnesses? Defense counsel cries, “Judge, it’s a fishing expedition!”

As both sides dodge and dither, the information sought in discovery vanishes as, *e.g.,* the store purges old records or plaintiff upgrades her digital devices. All but a minute of video leading up to the fall has been overwritten by the time the first discovery request is served. When that scant minute proves too short to establish how long the grease was on the floor, the plaintiff is prejudiced and files a Motion for Sanctions seeking to punish the defendant for the failure to preserve crucial evidence. When it’s learned the plaintiff closed her Facebook account after the fall and her posts are gone, the defendant files its own Motion for Sanctions.

The defendant will argue that it shouldn’t be punished because it didn’t intend to deprive the plaintiff of the video; “it just seemed like a minute was enough.” Defendant will claim harm occasioned by the loss of plaintiff’s Facebook posts, positing the lost posts would have shown the plaintiff to be physically active and happy, undermining plaintiff’s claims of disability and lost enjoyment of life.

This is just a run-of-the-mill slip and fall case, but the outcome depends upon the exchange of an assortment of relevant and probative sources of electronic evidence.

Now, consider the far-flung volume and variety of electronic evidence in a class action brought for 100,000 employees, for a million injured by a massive data breach or a bet-the-company patent fight between technology titans. We cannot throw up our hands and say, “It’s too much! It’s too hard! It’s too expensive!”

Instead, we must balance the need to afford access to information enabling resolution of disputes based on relevant evidence against denying that access because costs and burdens outweigh benefits. Competency is key because disparity breeds distrust. Most would agree that the better a lawyer’s grasp of information systems and electronic evidence, the greater the potential for consensus with a knowledgeable opponent acting in good faith.

But, when it comes to competency in e-discovery, there’s little agreement. Must lawyers comprehend the discovery tasks they delegate to others? Where is the line between delegating discovery to laypersons and the unauthorized practice of law? How does a lawyer counsel a client to preserve and produce what the lawyer does not understand and cannot articulate?

We can define literacy and measure reading proficiency; but there is no measure of literacy when it comes to electronic evidence and e-discovery. How can one become literate in the conventional sense without knowing an alphabet, possessing a vocabulary, and understanding the concepts of words and phrases? A gift for pattern recognition might let a savant fake it for a time; but genuine literacy entails mastering fundamentals, like awareness of speech sounds (phonology), spelling patterns (orthography), word meaning (semantics), grammar, (syntax), and patterns of word formation (morphology). One in eight adult Americans cannot read. Do we expect any of them are lawyers?

Electronic evidence and e-discovery literacy demands more than what’s required for computer literacy (the ability to use computers and related technology efficiently) or digital literacy (the ability to find, evaluate, and communicate information via digital platforms). Computer and digital literacy are just a start: necessary but insufficient.

Competence in e-discovery and digital evidence encompasses a working knowledge of matters touching evidence integrity and being equipped to support and challenge the authenticity and admissibility of electronic evidence. Competence requires that one understand, *inter alia*, what electronically stored information is, where it resides, the forms it takes, and the metadata it implicates. What makes it trustworthy? How is it forged and manipulated? What constitutes a chain of custody sufficient to counter attacks on your handling of evidence? How do you properly preserve data without altering it? How do you communicate technical obligations to technical personnel without understanding the language they speak and the environment in which they work? How do you seek, cull, search, sort, review, and produce electronically stored information? What does it cost? How long does it take?

We expect banking attorneys to understand banking and real estate attorneys to understand real estate. Shouldn’t we expect trial lawyers to understand electronic evidence and discovery? If so, do we start by teaching them the alphabet or do we hope they can learn to fake it without fundamentals?

This course reflects my sense that, while one can surely become a fine physician without it, I want my doctor to have taken biochemistry…and passed. Likewise, I believe students of electronic evidence and e-discovery must not be strangers to data storage, collection, encoding, processing, metadata, search, forms of production, and the vocabulary of information technology (IT) and computer forensics.

If you believe that all a trial lawyer needs to know is the law, this is not the course for you. Here, we celebrate the “e” in e-discovery and e-evidence. You’ll get your hands dirty with data, use modern tools and learn to speak geek. We strive together toward competence and confidence, so that you may emerge, not as ill-equipped computer scientists, but poised to be tech-savvy litigators.

Introduction to Discovery in U.S. Civil Litigation

Until the mid-20th century, the trial of a civil lawsuit was an exercise in ambush. Parties to litigation knew little about an opponent's claims or defenses until aired in open court. A lawyer's only means to know what witnesses would say was to locate them before trial and persuade them to talk. Witnesses weren't obliged to speak with counsel, and what they volunteered out of court might change markedly under oath in court. Too, at law, there was no right to see documentary evidence before trial.

John Henry Wigmore, nicely summed up the situation in his seminal, *A Treatise on the System of Evidence in Trial at Common Law* (1904). Citing the Latin maxim, *nemo tenetur armare adversarium suum contra se* ("no one is bound to arm his adversary against himself"), Wigmore explained:

> To require the disclosure to an adversary of the evidence that is to be produced would be repugnant to all sportsmanlike instincts. Rather permit you to preserve the secret of your tactics, to lock up your documents in the vault, to send your witness to board in some obscure village, and then, reserving your evidential resources until the final moment, to marshal them at the trial before your surprised and dismayed antagonist, and thus overwhelm him. Such was the spirit of the common law; and such in part it still is. It did not defend or condone trickery and deception; but it did regard the concealment of one's evidential resources and the preservation of the opponent's defenseless ignorance as a fair and irreproachable accompaniment of the game of litigation.
> Id. At Vol. III, §1845, p. 2402.

Our forebears at common law[1] feared that disclosure of evidence would facilitate unscrupulous efforts to tamper with witnesses and promote the forging of false evidence. The element of surprise was thought to promote integrity of process.

Legal reformers hated "trial by ambush" and, in the late-1930's, they sought to eliminate surprise and chicanery in U.S. courts by letting litigants obtain information about an opponent's case before trial in a process dubbed "discovery."[2] The reformer's goal was to streamline the trial process and

[1] "Common law" refers to the law as declared by judges in judicial decisions ("precedent") rather than rules established in statutes enacted by legislative bodies.

[2] That is not to say that discovery was unknown. Many jurisdictions offered a mechanism for a Bill of Discovery, essentially a separate suit in equity geared to obtaining testimony or documents in support of one's own position. However, Bills of Discovery typically made no provision for obtaining information about *an opponent's* claims, defenses or evidence—which is, of course, what one would most desire. As well, some states experimented with procedural codes that allowed for discovery of documents and taking of testimony (*e.g.*, David Dudley Field II's model code). For a comprehensive treatment of the topic, see, Ragland, George, Jr., Discovery Before Trial, 1932.

enable litigants to better assess the merits of the dispute and settle their differences without need of a trial.

After three years of drafting and debate, the first Federal Rules of Civil Procedure went into effect on September 16, 1938. Though amended many times since, the tools of discovery contained in those nascent Rules endure to this day:

- Oral and written depositions (Rules 30 and 31)
- Interrogatories (Rule 33)
- Requests to inspect and copy documents and to inspect tangible and real property (Rule 34)
- Physical and mental examinations of persons (Rule 35)
- Requests for admissions (Rule 36)
- Subpoena of witnesses and records (Rule 45)

Tools of Discovery Defined

Depositions

A deposition is an interrogation of a party or witness ("deponent") under oath, where both the questions and responses are recorded for later use in hearings or at trial. Testimony may be elicited face-to-face ("oral deposition") or by presenting a list of questions to be posed to the witness ("written deposition"). Deposition testimony may be used in lieu of a witness' testimony when a witness is not present or to impeach the witness in a proceeding when a witness offers inconsistent testimony. Deposition testimony is typically memorialized as a "transcript" made by an official court reporter but may also be a video obtained by a videographer.

Interrogatories

Interrogatories are written questions posed by one party to another to be answered under oath. Although the responses bind the responding party much like a deposition on written questions, there is no testimony elicited nor any court reporter or videographer involved.

Requests for Production

Parties use Requests for Production to demand to inspect or obtain copies of tangible evidence and documents. Requests for Production are the chief means by which parties pursue **electronically stored information (ESI)**. Requests may also seek access to places and things.

Requests for Physical and Mental Examination

When the physical or mental status of a party is in issue (such as when damages are sought for personal injury or disability), an opposing party may seek to compel the claimant to submit to examination by a physician or other qualified examiner.

Requests for Admission

These are used to require parties to concede, under oath, that particular facts and matters are true or that a document is genuine.

Subpoena
A subpoena is a directive requiring the recipient to take some action, typically to appear and give testimony or hand over or permit inspection of specified documents or tangible evidence. Subpoenas are commonly used to obtain evidence from persons and entities who are not parties to the lawsuit.

Strictly speaking, the Federal Rules of Civil Procedure do not characterize subpoenas as a discovery mechanism because their use is ancillary to depositions and proceedings. Still, they are employed so frequently and powerfully in discovery as to warrant mention here.

Scope of Discovery Defined

Rule 26(b)(1) of the Federal Rules of Civil Procedure defines the scope of discovery this way:

> Parties may obtain discovery regarding any nonprivileged matter that is relevant to any party's claim or defense and proportional to the needs of the case, considering the importance of the issues at stake in the action, the amount in controversy, the parties' relative access to relevant information, the parties' resources, the importance of the discovery in resolving the issues, and whether the burden or expense of the proposed discovery outweighs its likely benefit. Information within this scope of discovery need not be admissible in evidence to be discoverable.

The Federal Rules don't define what is "relevant," but the generally accepted definition is that a matter is relevant when it has any tendency to make a fact more (or less) probable. Information may be relevant even when not admissible as competent evidence, such as hearsay or documents of questionable authenticity.

The requirement that the scope of discovery be proportional to the needs of the case was added to the Rules effective December 1, 2015, although it has long been feasible for a party to object to discovery efforts as being disproportionate and seek protection from the Court.

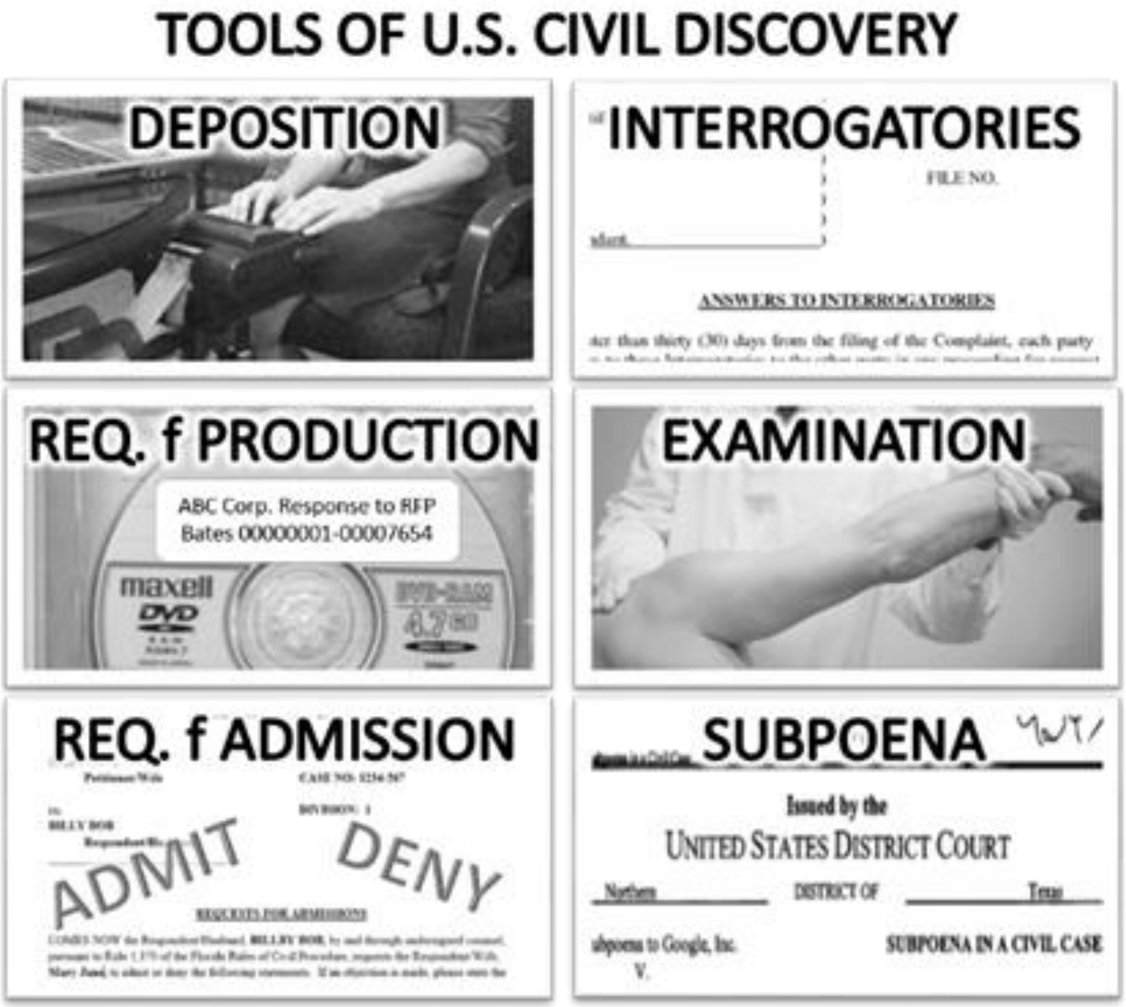
Certain matters fall beyond the proper scope of discovery because they enjoy a privilege from disclosure. The most common examples of these privileged matters are confidential attorney-client communications and attorney trial preparation materials (also called "attorney work product"). Other privileged communications include confidential communications between spouses, between priest and penitent and communications protected by the Fifth Amendment of the U.S. Constitution.

Protection from Abuse and Oppression

The discovery provisions of the Federal Rules of Civil Procedure are both sword and shield. They contain tools by which litigants may resist abusive or oppressive discovery efforts. Parties have the right to object to requests and refrain from production on the strength of those objections. Parties may also seek **Protective Orders** from the court. Rule 26(c) provides:

“The court may, for good cause, issue an order to protect a party or person from annoyance, embarrassment, oppression, or undue burden or expense, including one or more of the following:

> (A) forbidding the disclosure or discovery;
>
> (B) specifying terms, including time and place or the allocation of expenses, for the disclosure or discovery;
>
> (C) prescribing a discovery method other than the one selected by the party seeking discovery;
>
> (D) forbidding inquiry into certain matters, or limiting the scope of disclosure or discovery to certain matters;
>
> (E) designating the persons who may be present while the discovery is conducted;
>
> (F) requiring that a deposition be sealed and opened only on court order;
>
> (G) requiring that a trade secret or other confidential research, development, or commercial information not be revealed or be revealed only in a specified way; and
>
> (H) requiring that the parties simultaneously file specified documents or information in sealed envelopes, to be opened as the court directs.”



Electronic Discovery and Digital Evidence: What Every Lawyer Should Know

Discovery is the legal process governing the right to obtain and the obligation to tender non-privileged matter relevant to any party's claims or defenses in litigation. Though discovery sometimes entails gaining access to physical objects like real estate, defective products or people (*e.g.*, medical exams), most discovery efforts are directed to information existing as human recollection elicited by testimony or recorded either as ink on paper or stored electronically, often as magnetized regions of spinning disks. Discovery is e-discovery when the relevant "matter" consists of **electronically stored information** (**ESI**).

Born of simpler times, requests for production were conceived to operate simply. A party to a lawsuit asks another party or a third party to furnish information, either by specifically or generically specifying the documents or records of interest or by describing topics about which information is sought. The party responding to the request had about a month to locate responsive items and make them available for inspection and copying or supply copies of the responsive items. The responding party could withhold or redact items containing privileged information, such as confidential communications between lawyer and client, but must furnish a log describing items withheld or redacted. The court served as a referee, affording protection to litigants for whom the process proved unduly burdensome and compelling production when responses proved insufficient.

At the dawn of civil discovery, people had been recording information on textual media for thousands of years, and the second half of the twentieth century was the apex of document-centric recordkeeping. Until mass adoption of personal computing and the internet in the 1990's, virtually all personal and most business communications took place as ink on paper or via ephemeral discussion, literally invisible vibration of the air.

The halcyon days of paper discovery were rife with quarrels about vague requests and obstructive responses. Paper discovery was expensive and time-consuming; but paper discovery was manageable, principally because we were schooled from childhood in how to understand and organize paper documents.

Then everything changed.

Today, virtually all personal and business communications entail the movement of electrons. Ephemeral phone conversations are now tangible texts. What once was ink on paper are now pixels on screens, often guised to mimic familiar experiences with paper. More, electronic transactions and communications come coupled with information that describes the Who, What, When, Where and How of the transaction or communication. Such data-about-data, called **metadata**, may convey more useful information than the transaction or communication it describes.

Thus, things civilization had done one way for millennia stopped being done that way in the course of a generation, spawning digital whiplash in the world of civil discovery.

Many lawyers hearken back to the exchange of information on paper. It is their sole, enduring context for discovery. Paper discovery was simpler. Being tangible, paper had to be more aggressively managed and organized to be useful; and being tangible, paper delivered its information payload right on the page as, *e.g.,* letterhead data, content, dates and circulation lists. Finally, being tangible, paper felt finite. There might be a lot of it, but you could see how much and gauge the workload.

Electronically stored information feels infinite. Indeed, there is a lot of it—replicated, distributed and fragmented. Being intangible and taking many forms, it's hard to know what you're dealing with. Being intangible, people store ESI wherever they wish, without undertaking to manage it--imagining that when the time came to deal with ESI, the skills used for paper would suffice.

E-discovery is more complex than paper discovery; but then, electric lighting is more complex than candles, and cars more complex than wagons. It's a complexity borne to useful ends.

For its challenges, ESI has advantages the legal system has yet to fully harness. ESI is inherently electronically searchable and structured to allow it to be culled, categorized and analyzed more effectively than paper records. Metadata afford us ways to assess the origins, integrity and import of evidence. The variety, ubiquity and richness of ESI blazes new trails to the truth of an event or transaction. Even the much-lamented loss of personal privacy attendant to modern digital life reveals a silver lining when it serves as reliable, probative evidence in support of just outcomes.

More information does not inevitably lead to better information and may serve to obscure the best information. So, the skills needed in e-discovery are not only those that can ferret out relevant information but also those that can manage the signal-to-noise ratio of that information.

Character and Competence in Discovery

In this golden age of evidence, ushered in by the monumental growth of data, all who access electronically stored information and use digital devices generate and acquire vast volumes of digital evidence. Never during human history have we had so much probative evidence, and never has that evidence been so objective and precise. Yet, lawyers are like farmers complaining of oil on their property; they bemoan electronic evidence because they haven't awoken to its value.

That's not surprising. What lawyer in practice received practical instruction in electronic evidence? Few law schools offer courses in electronic evidence, and those teaching e-discovery rarely tackle the essential "e" that sets e-discovery apart. Continuing legal education courses shy away from the nuts and bolts of information technology needed to competently manage and marshal digital evidence. Law graduates are expected to acquire trade skills by apprenticeship; yet experienced counsel haven't technical savvy to share. Competence in electronic evidence is exceptionally rare, and there is little afoot to change that save the vain expectation that lawyers will miraculously gain competence without education or effort.

As sources of digital evidence proliferate in the cloud, on mobile devices and tablets and within the burgeoning Internet of Things, the gap between competent and incompetent counsel grows. We suffer most when standard setters decline to define competence in ways that might exclude them. Vague pronouncements of a duty to stay abreast of "relevant technology" are noble but do not help lawyers know what they must know.[3]

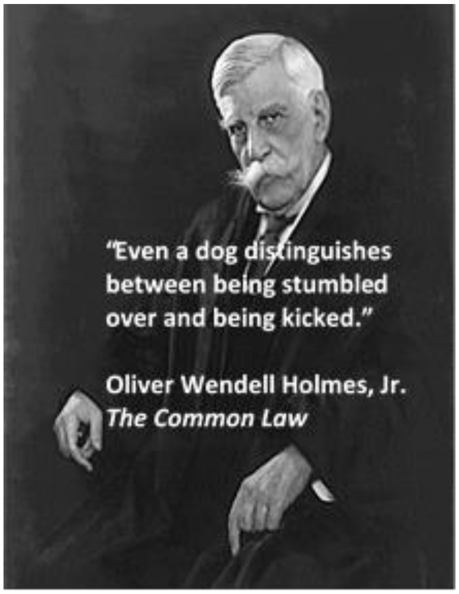
Discovery is much maligned as a too costly, too burdensome, too intrusive fishing expedition.[4] Discovery is effective and affordable when deployed with character and competence, but there's sufficient lack of both in the world to ensure that discovery abuse and obstruction remain commonplace.

Character is hard to instill and harder still to measure; but competence is not. We can require that lawyers master the fundamentals of modern information—particularly those needed for electronic discovery, where so many lag—and we can objectively assess their ken. When you can establish competence, you can more easily discern character or, as Oliver Wendell Holmes, Jr. aptly observed, you can know what any dog knows; that is, the difference between being stumbled over and being kicked.

[3] Rule 1.1 of the American Bar Association Model Rules of Professional Conduct provides that, "[a] lawyer shall provide competent representation to a client. Competent representation requires the legal knowledge, skill, thoroughness and preparation reasonably necessary for the representation." Comment 8 to Rule 1.1 adds, "[t]o maintain the requisite knowledge and skill, a lawyer should keep abreast of changes in the law and its practice, *including the benefits and risks associated with relevant technology….*" Emphasis added. Many states have added the same language to their disciplinary rules.

[4] Such concerns are not new. Well before the original Rules went into effect, the Chairman of the Rules Advisory Committee exclaimed, "We are going to have an outburst against this discovery business unless we can hedge it about with some appearance of safety against fishing expeditions." Proceedings of the Advisory Committee (Feb. 22, 1935), at CI-209-60-0-209.61. Many still curse "this discovery business," particularly those most likely to benefit from the return of trial by ambush and those who would more-or-less do away with trials altogether.

To leap the competence chasm, lawyers must recognize the value and necessity of acquiring a solid foundation in the technical and legal aspects of electronic evidence, and bar associations, law schools and continuing education providers must supply the accessible and affordable educational opportunities and resources needed to help lawyers across.


"Even a dog distinguishes between being stumbled over and being kicked."
Oliver Wendell Holmes, Jr.
The Common Law

So, it is heartening when the state with the second largest number of practicing lawyers in America takes a strong, clear stand on what lawyers must know about discovery of electronic evidence. The State Bar of California Standing Committee on Professional Responsibility and Conduct issued an advisory opinion in which the Committee sets out the level of skill and familiarity required when, acting alone or with assistance, counsel undertakes to represent a client in a matter implicating electronic discovery.[5]

The Committee wrote:

> We start with the premise that "competent" handling of e-discovery has many dimensions, depending upon the complexity of e-discovery in a particular case. The ethical duty of competence requires an attorney to assess at the outset of each case what electronic discovery issues, if any, might arise during the litigation, including the likelihood that e-discovery will or should be sought by either side. If it is likely that e-discovery will be sought, the duty of competence requires an attorney to assess his or her own e-discovery skills and resources as part of the attorney's duty to provide the client with competent representation. If an attorney lacks such skills and/or resources, the attorney must take steps to acquire sufficient learning and skill, or associate or consult with someone with appropriate expertise to assist. … Taken together generally, and under current technological standards, attorneys handling e-discovery should have the requisite level of familiarity and skill to, among other things, be able to perform (either by themselves or in association with competent co-counsel or expert consultants) the following:
>
> 1. initially assess e-discovery needs and issues, if any;

[5] The State Bar of California Standing Committee on Professional Responsibility and Conduct Formal Opinion Interim No. 11-0004 (2014).

2. implement appropriate ESI preservation procedures, including the obligation to advise a client of the legal requirement to take actions to preserve evidence, like electronic information, potentially relevant to the issues raised in the litigation;
3. analyze and understand a client's ESI systems and storage;
4. identify custodians of relevant ESI;[6]
5. perform appropriate searches;
6. collect responsive ESI in a manner that preserves the integrity of that ESI;
7. advise the client as to available options for collection and preservation of ESI;
8. engage in competent and meaningful meet and confer with opposing counsel concerning an e-discovery plan; and
9. produce responsive ESI in a recognized and appropriate manner.[7]

ATTORNEY ESI COMPETENCE

When it comes to handling cases with ESI, *learn it, get help or get out*

1 Assessment

2 Preservation

3 Sources

4 Custodians

5 Search

6 Collection

7 Counsel

8 Conference

9 Production

State Bar of California Standing Committee on Professional Responsibility and Conduct, Formal Opinion No. 2015-193 (2015)

[6] Though the term “records custodian” is customarily defined as the person responsible for, or the person with administrative control over, granting access to an organization's documents or electronic files while protecting the data as defined by the organization's security policy or its standard IT practices, the term tends to be accorded a less precise definition in e-discovery and is best thought of as anyone with possession, custody or control of ESI, including a legal right or *practical ability* to access same. *See*, *e.g.*, *In re NTL, Inc. Securities Litigation*, 244 F.R.D. 179, 195 (S.D.N.Y. 2007).

[7] Id.

Thus, California lawyers face a simple mandate when it comes to e-discovery, and one that should take hold everywhere: ***Learn it, get help*** **or** ***get out.*** Declining the representation may be the only ethical response when the lawyer lacks enough time to acquire the requisite skills and the case won't bear the cost of associating competent co-counsel or expert consultants. Most cases aren't big enough to feed two mouths when only one is competent.

Each of the nine tasks implicate a broad range of technical and tactical skills. The interplay between technical and tactical suggests that just asking Information Technology (IT) personnel some questions won't suffice. Both efficiency and effectiveness demand that, if the lawyer is to serve as decision maker and advocate, the *lawyer* needs to do more than parrot a few phrases. The lawyer needs to *understand* what the technologists are talking about, then follow the evidence.

A lawyer must be capable of recognizing the digital evidence obligations and issues that arise. This requires experience and a working knowledge of the case law and professional literature. Certainly, a lawyer's first step toward competence begins with reading the Rules and the leading cases, but the lawyer must then explore the argot of information technology. When we come across an unfamiliar technical term in an opinion or article, we can't elide over it. We must l*ook it up!* Google and Wikipedia are our friends!

Implementing appropriate ESI preservation procedures means knowing how to scope, communicate and implement a defensible legal hold. You can't be competent to scope a hold without understanding the tools and software your client uses. You can't help your client avoid data loss and spoliation if you have no idea what data is robust and tenacious and what is fragile and transitory. How do you preserve relevant data and metadata without the barest notion of what data and metadata exist and where it resides?

At first blush, identifying custodians of relevant ESI seems to require no special skills; but behind the scenes, a cadre of IT specialists administer and maintain the complex and dynamic server, database and Cloud environments businesses use. We can't expect people unschooled in information technology to preserve backup media or suspend programs purging data our clients must preserve. *These are tasks for IT specialists*. Competence in counsel includes the ability to pose the right questions to the right people, including information technologists.

Performing appropriate searches entails more than just guessing what search terms seem suitable. *Search is a science.* Search tools vary widely, and counsel must understand what these tools can and cannot do. Queries must be tested to assess precision and recall. Small mistakes in search prompt big downstream consequences, where timely tweaks engender big savings. How do you negotiate culling and filtering criteria if you don't understand the ways ESI is culled and filtered?

Some ESI can be preserved in place with little cost and burden and may even be safely and reliably searched *in situ* to save money. Other ESI requires data be collected and processed to be amenable to search. Understanding which is which is crucial to being competent to advise clients about available options and forthcoming costs.

Lawyers lacking digital evidence skills can mount a successful meet and confer on ESI issues by getting technically astute personnel together to 'dance geek-to-geek.' But that's costly and risky. Cautious, competent counsel will want to *understand* the risks and costs, not just trust the technologists to know what's relevant and how and when to protect privileged and sensitive data.

Competent counsel understands that there is no single form suited to production of every item of ESI and recognizes the costs and burdens associated with alternate forms of production. Competent counsel knows that converting native electronic formats to TIFF images increases the size of the files many times and inflates the cost of ingestion and hosting by vendors. Competent counsel knows when it's essential to demand native forms of production to guard against data loss and preserve utility. Conversely, competent counsel knows how to make the case for TIFF production to handicap an opponent or when needed for redaction.

Clearly, there's a lot more to e-discovery than many imagine, and much of it must fall within counsel's ambit. Virtually all evidence today is born digitally. It's data, and only a fraction takes forms we've traditionally called documents. Lawyers ignored ESI for decades while information technologies changed the world. Is it any wonder there's a competence chasm? Few excel at all of the skills that trial work requires; but every trial lawyer must be minimally competent in them all. Today, the most demanding of these skills is e-discovery.

Is it fair to deem practicing attorneys incompetent, or even unethical, because they don't possess skills not taught to them in law school? It may not feel fair to lawyers trained for a vanished world of paper documents; but to the courts and clients ill-served by those old ways, it's not just fair—it's right.

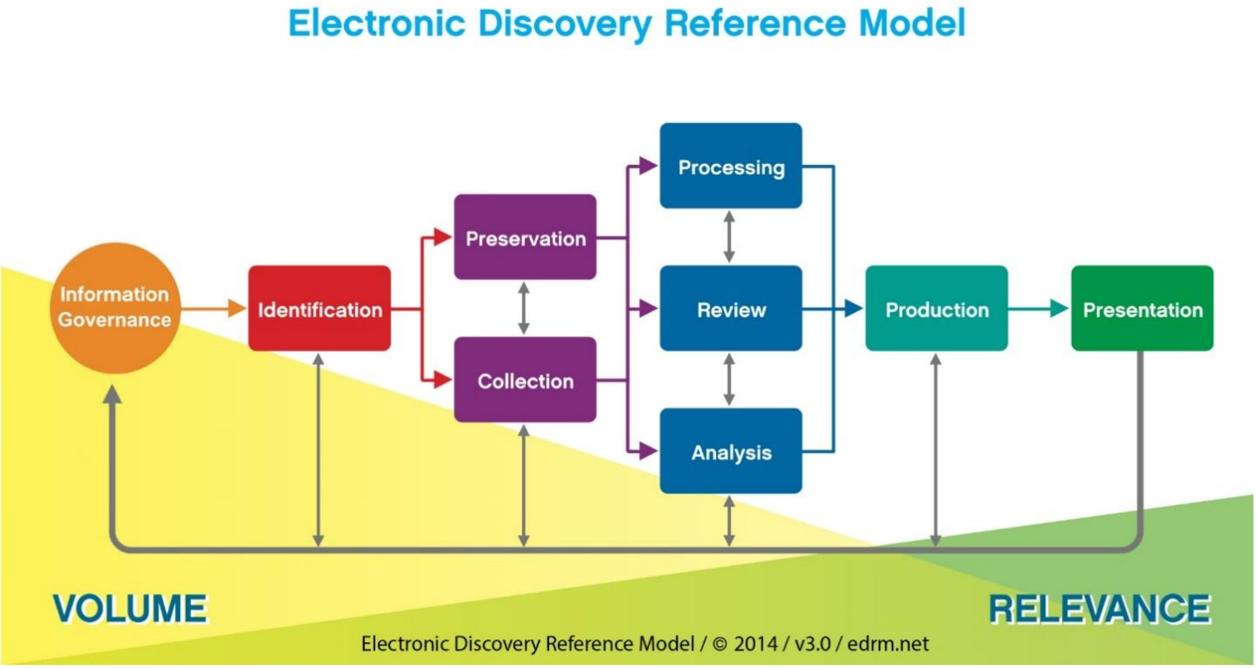
The development of e-evidence skills by the legal profession is hampered by a delusion that if lawyers can just keep electronic evidence at bay, there will be a way to turn back the digital deluge. Many lawyers, judges and litigants still mistakenly believe that the paper-centric methods that served so well in the past will work in a digital world and indulge their delusion (at horrific expense) by converting electronic data into clumsy, paper- or paper like formats. Big firm lawyers and corporations have grown comfortable outsourcing electronic discovery to service providers whose expanding roles blur the boundary between technical support and substantive law practice.

Delegating e-discovery duties to vendors has allowed lawyers to muddle through, but at great expense and peril of error, delay and miscommunication.

Vendors have won the battle, but have they won the war? Lawyers have been willing dupes to the idea that e-discovery is a service they must buy instead of a competence they must possess. If trial lawyers hope to remain the captains of litigation, they must be competent in doing, not just buying, e-discovery. If you want to take the helm, it’s wise to know how to steer.

The Electronic Discovery Reference Model (EDRM)

In 2005, when e-discovery had hardly entered lawyer lexicons, Minnesotans George Socha and Tom Gelbmann saw we weren't speaking the same language. George and Tom proposed a conceptual view of the electronic discovery process, expressed as a diagram to serve as a basis for discussion and analysis. The now-iconic EDRM diagram helped the legal and vendor communities communicate more clearly and reinforced the iterative nature of a successful e-discovery model; one that vanquishes volume and enriches relevance.



Information Governance

Getting your electronic house in order to mitigate risk & expenses should e-discovery become an issue, from initial creation of ESI (electronically stored information) through its final disposition.

Identification

Locating potential sources of ESI & determining its scope, breadth & depth.

Preservation

Ensuring that ESI is protected against inappropriate alteration or destruction.

Collection

Gathering ESI for further use in the e-discovery process (processing, review, etc.).

Processing

Reducing the volume of ESI and converting it, if necessary, to forms more suitable for review & analysis.

Review

Evaluating ESI for relevance & privilege.

Analysis

Evaluating ESI for content & context, including key patterns, topics, people & discussion.

Production

Delivering ESI to others in appropriate forms & using appropriate delivery mechanisms.

Presentation

Displaying ESI before audiences (at depositions, hearings, trials, etc.), especially in native & near-native forms, to elicit further information, validate existing facts or positions, or persuade an audience.

In this course, the EDRM serves as our conceptual framework for understanding the functional components of electronic discovery from identification through presentation.

Too, we must peer past the colorful confines of the EDRM and weigh the evidentiary and societal aspects of electronic evidence. In a world without notarized pen-and-ink signatures--where simply flipping a bit value changes everything--*how do we prove the authenticity or attack the integrity of digital evidence? How will we trust, train and perhaps restrain Artificial Intelligence (AI) systems serving as proxies for human judgment? How do we balance personal privacy and the need to safeguard trade secrets against the obligation to bring the best evidence into court?*

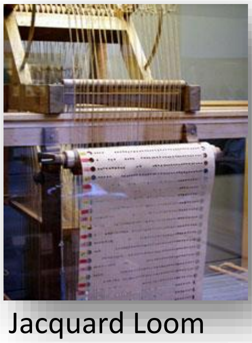
Introduction to Data Storage Media

Mankind has been storing data for thousands of years, on stone, bone, clay, wood, metal, glass, skin, papyrus, wax, paper, plastic and film. In fact, people were storing data in binary formats long before the emergence of modern digital computers. Records from 9th century Persia describe an organ playing interchangeable cylinders. Eighteenth century textile manufacturers employed perforated rolls of paper to control Jacquard looms, and Swiss and German music box makers used metal drums or platters to store tunes. At the dawn of the Jazz Age, no self-respecting American family of means lacked a player piano capable (more-or-less) of reproducing the works of the world's greatest pianists.

Whether you store data as a perforation or a pin, you're storing binary data. That is, there are two data states: hole or no hole, pin or no pin, one or zero.

VITAL VOCABULARY

Analog
Areal Density
Binary
Actuator Arm
Bates Numbering
Behind the Firewall
Cloud
Clusters
Cylinders
De-Duplication
De-NISTing
Electromagnetic
Encoding
Form Factor
Hashing
Interface
IT
Master File Table
Network Share
NTFS
RAID Array
Read-Write Head
SAN and NAS
Sectors
Solid State
Storage Media
Tracks


Jacquard Loom

Music Box

Player Piano

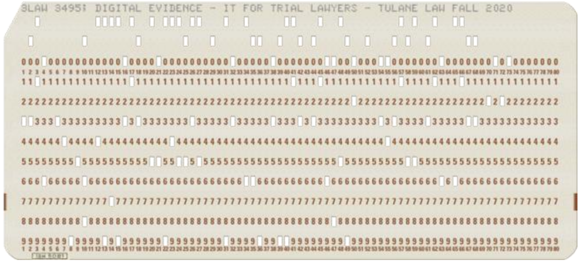
Punched Cards

In 1889, U.S. inventor Herman Hollerith (1860-1929) was granted a patent for his system for storing data on perforated paper cards that revolutionized the 1890 U.S. census. Using 43 Hollerith machines and 62 million punched cards, the time to tabulate the census dropped from eight years to three.[8] In the 1930's, demand for *electronic* data storage led to widespread adoption of Hollerith cards as a fast, practical and cost-effective binary storage media. These punched cards, initially made in a variety of sizes and formats, were ultimately standardized by

[8] Hollerith founded The Tabulating Machine Company, based in the Georgetown neighborhood of Washington, D.C. Hollerith's company was later merged with others and renamed International Business Machines Company, now IBM.

IBM as the 80 column, 12 row (7.375" by 3.25") format (right) that dominated computing well into the 1970's. In the mid-1950s, punched card sales accounted for 20 percent of IBM's revenues. From 1975-79, this author spent many a night in the basement of a computer center at Rice University typing program instructions onto these unforgiving punch cards, cousins to the oily, yellow perforated paper tape that Bill Gates and this author used on opposite coasts to program mainframe computers via a teletype terminal in the early 1970s.

IBM 5081-style 80-column card


3LAW 3495T DIGITAL EVIDENCE - IT FOR TRIAL LAWYERS - TULANE LAW FALL 2020

In the punch card era, storing programs and data sets meant storing stacks of IBM cards, often in motorized "tub files" as seen at right.

The encoding schemes of these obsolete media differ from those we use today principally in speed and scale; but the binary fundamentals are still…fundamental and connect our toil in e-discovery and computer forensics to the likes of Charles Babbage, Alan Turing, Ada Lovelace, John von Neumann, Robert Noyce and both Steves (Wozniak and Jobs).

In the space of a single generation, we have come far.

The IBM punched cards held 80 columns of 12 punch positions or 960 bits. Nominally, that's 120 eight-bit bytes, but because eight columns weren't always used for data storage, the storage capacity was closer to 864 bits or 108 bytes–and not that much in fact, because each column was typically dedicated to just one 7- or 8-bit ASCII character, so the practical capacity of a punch card was 80 characters/80 bytes or less.[9]

[9] After hours researching the capacity question, I couldn't arrive at a definitive answer because capacity varied according, *inter alia*, to the type of information being stored (binary versus ASCII) and a reluctance to punch out too many adjacent perforations lest it become a "lace card" too fragile to use.

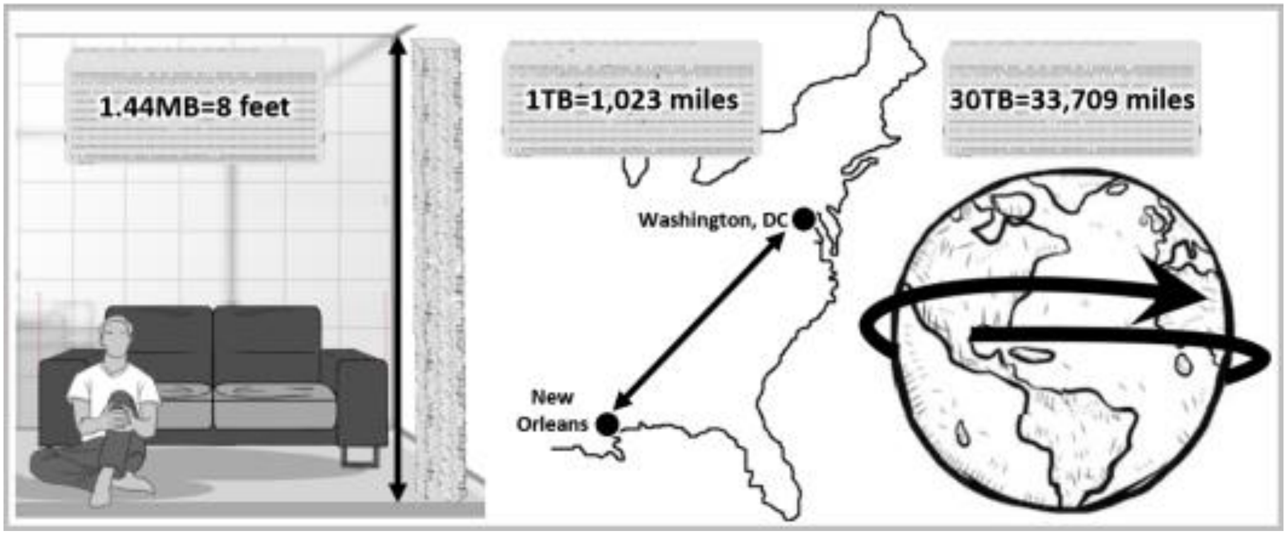
Using the 108-byte value, the formatted “1.44mb” 3.5-inch floppy disks commonly used from the mid-1980s to early 2000s held 1.4 million bytes (megabytes), so a floppy disk could store the same amount of data as about 13,653 IBM cards, *i.e.,* seven 2,000 card boxes of cards or, at 143 cards to the inch, an eight-foot stack. That’s a common ceiling height and taller than anyone who ever played for the NBA.

The prim Eisenhower-era programmer in the photo below steadies 62,500 punched cards said to hold the five megabytes of program instructions for the massive SAGE (Semi-Automatic Ground Environment) military computing network (an 80 byte capacity for each card).

Fast forward to today’s capacious hard drives, a fifty-dollar terabyte drive holds 1,099,511,627,776 bytes. That’s over *ten billion* IBM cards (10,180,663,220 to be precise). Now, our stack of cards is 1,123 *miles,* or roughly the driving distance between Washington, D.C. and New Orleans.

So, the 30TB (compressed) capacity of an LTO-8 backup tape cartridge starts to equal something like ***305 billion IBM cards***—a stack spanning 33,709 miles that would handily circle the globe at the Equator.

These are hypothetical extrapolations, not real-world metrics because much storage capacity is lost to file system overhead. If you used a warehouse for physical storage, you’d



need to sacrifice space for shelving and aisles, and you'd likely find that not everything you store perfectly fits wall-to-wall and floor-to-ceiling. Similarly, digital storage sacrifices capacity to file tables and wastes space by using fixed cluster sizes. If a file is smaller than the clusters (i.e., storage blocks) allocated to its storage, then the bytes between the end of the file and the end of the cluster is wasted "slack space."

The 1950's saw the emergence of **electromagnetic storage** as the dominant medium for electronic data storage. Although solid-state storage will ultimately eclipse electromagnetic media for local storage, electromagnetic storage will continue to dominate network and cloud storage well into the 2020s, if not long after, because it's considerably more economical and capacious storage versus solid state media.

Magnetic Tape

The earliest popular form of electromagnetic data storage was magnetic tape.

Spinning reels of tape were a clichéd visual metaphor for computing in films and television shows from the 1950s through 1970's. Though the miles of tape on those reels now resides in cartridges and cassettes, tapes remain an enduring medium for backup and archival of electronically stored information. Compact cassette tape was the earliest data storage medium for personal computers including the pioneering Radio Shack TRS-80 and the very first IBM personal computer, the model XT.

The LTO-8 format tapes introduced in 2017 house 3,150 feet of half inch tape in a cartridge just four inches square and less than an inch thick; yet each cartridge natively hold 12.0 terabytes of uncompressed data and up to 30 TB of compressed data[10] delivered at a transfer rate of 360 megabytes per second. LTO tapes use a back-and-forth or linear serpentine recording scheme. "Linear" because it stores data in parallel tracks running the length of the tape, and "serpentine" because its path snakes back-and forth, reversing direction on each pass. Thirty-two of the LTO-8

[10] Since most data stored on backup tape is compressed, the actual volume of ESI on tape may be 2+ times greater than the native capacity of the tape.

cartridge's 3,584 tracks are read or written as the tape moves past the recording heads, so it takes 112 back-and-forth passes or "wraps" to read or write the full contents of a single LTO-8 cartridge.

That's *67 miles* of tape passing the heads, roughly the distance between Austin and San Antonio! So, it takes *hours* to read or write each tape; *e.g.*, more than nine hours to write a full tape at maximum uncompressed speed. While tape isn't as fast as hard drives, it's proven to be more durable and less costly for long term storage; that is, so long as the data is being *stored,* not *re*stored.

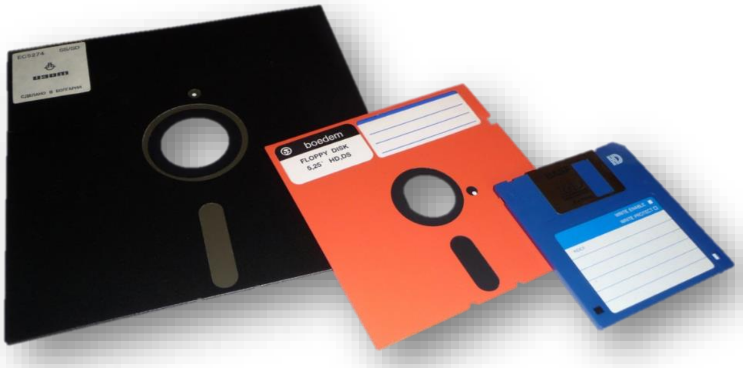
LTO-8 Ultrium Tape

Sony AIT-3 Tape

SDLT-II Tape

Floppy Disks

Today, the only place a computer user is likely to see a floppy disk is as the menu icon for storage on the menu bar of Microsft Office applications. But, floppy disks played a central role in software distribution and data storage for personal computing for thirty years..



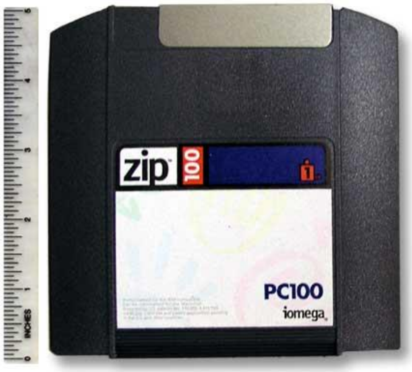
8", 5.25" and 3.5" Floppy Disks

Floppy disks are another form of **electromagnetic storage**. All floppy disks have a spinning, flexible plastic disk coated with a magnetic oxide (e.g., rust). The disk is essentially the same composition as magnetic tape in disk form. Disks were **formatted** (either by the user or pre-formatted by the manufacturer) so as to divide the disk into various concentric rings of data called **tracks**, with tracks further subdivided into tiny arcs called **sectors**. Formatting enables systems to locate data on physical storage media much as platting a subdivision into streets and house numbers enable us to locate homes in a neighborhood.

8" Floppy Disk in Use

Though many competing floppy disk sizes and formats have been introduced since 1971, only five formats are likely to be encountered in e-discovery. These are the 8”, 5.25”, 3.5 standard, 3.5 high density and Zip formats and, of these, the 3.5HD format 1.44 megabyte capacity floppy is by far the most prevalent legacy floppy disk format.

Zip Disk



The Zip Disk was one of several proprietary “super floppy” products that enjoyed brief success before the high capacity and low cost of recordable (non-magnetic) optical media (CD-R and DVD-R) and flash drives rendered them obsolete.

Optical Media

The most common forms of *optical* media for data storage are the CD, DVD and Blu-ray disks in read only, recordable or rewritable formats. Each typically exists as a 4.75” plastic disk with a metalized reflective coating and/or dye layer that can be distorted by a focused laser beam to induce pits and lands in the media. These pits and lands, in turn, interrupt a laser reflected off the surface of the disk to generate the ones and zeroes of digital data storage. The practical difference between the three prevailing forms of optical media are their native data storage capacities and the speed (“throughput”) at which they can deliver data. In contrast to tape, floppies and mechanical hard drives, optical storage media do *not* use electromagnetism to store and retrieve data.

A **CD** (for **C**ompact **D**isk) or **CD-ROM** (for **CD R**ead **O**nly **M**edia) is read only and not recordable by the end user. It’s typically fabricated in factory to carry music or software. A **CD-R** is recordable by the end user, but once a recording session is closed, it cannot be altered in normal use. A **CD-RW** is a re-

recordable format that can be erased and written to multple times. The native data storage capacity of a standard-size CD is about **700 megabytes.**

A **DVD** (for **D**igital **V**ersitile **D**isk) also comes in read only, recordable (**DVD±R**) and rewritable (**DVD±RW**) iterations and the most common form of the disk has a native data storage capacity of approximately **4.7 gigabytes**. So, one DVD holds the same amount of data as six and one-half CDs.

By employing the narrower wavelength of a blue laser to read and write disks, a dual layer **Blu-ray** disk can hold up to about 50 gigabytes of data, equalling the capacity of about ten and one-half DVDs. Like their predecessors, Blu-ray disks are available in recordable (BD-R) and rewritable (CD-RE) formats.

"I should have had him put into a more manageable format years ago."

Computers, Hard Drives and Servers

Though ESI resides on a dizzying array of media and devices, by far the largest complement of same occurs within three closely related species of computing hardware: *computers, hard drives* and *servers.* A server is essentially a computer dedicated to a specialized task or tasks, and both servers and computers routinely employ hard drives for program and data storage.

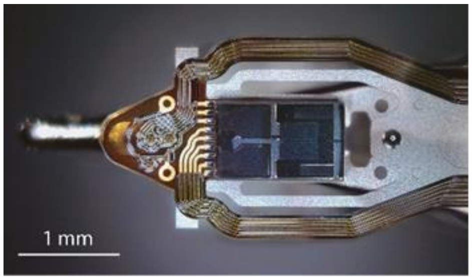
Electromagnetic Hard Drives

As noted, mankind has long stored information by translating it into physical manifestations: cave drawings, Gutenberg bibles, musical notes, Braille dots or undulating grooves in a phonograph record. Because it's simply a long sequence of ones and zeros, binary data can be memorialized by many physical phenomena. You could build a computer that stored data as a row of beads (the abacus), holes punched in paper (a piano roll), black and white vertical lines (bar codes) or bottles of beer on the wall (still waiting for this one!).

If we build our computer to store data using bottles of beer on the wall, we'd better be thirsty because we'll need a load of bottles and a whole lot of time to set them up, count them and replace them as data changes. Too, we will need something like the Great Wall of China to hold them. So, our beer bottle data storage system isn't practical. Instead, we need something compact, lightweight and efficient --in short, a refrigerator magnet and some paper clips.

Okay, maybe not a refrigerator magnet *per se*, but the principles are the same. If you take a magnet off your refrigerator and rub it against a metal paperclip, you will transfer some magnetic properties to the paperclip. Suppose you lined up about a zillion paper clips and magnetized some but not others. You could go down the row with a piece of ferrous metal (or, better yet, a compass) and distinguish the magnetized clips from the non-magnetized clips. If you call the magnetized clips “ones” and the non-magnetized clips “zeroes,” you’ve got yourself a system that can record binary data. Were you to glue all those paper clips in concentric circles onto a spinning phonograph record and substitute an electromagnet for the refrigerator magnet, you wouldn’t be too far afield of what goes on inside the hard and floppy disk drives of a computer, albeit at a much smaller scale. In case you wondered, this is also how we recorded sound on magnetic tape, except that instead of determining that a spot on the tape is magnetized or not as it rolls by, we gauge varying degrees of magnetism which corresponding to variations in the recorded sounds. This is **analog** recording—the variations in the recording are analogous to the variations in the music.

Since computers process electrical signals much more effectively than magnetized paper clips jumping onto a knife blade, what is needed is a device that transforms magnetic signals to electrical signals and vice-versa—an energy converter. Inside every floppy and hard disk drive is a gadget called a read/write head. The read/write head is a tiny electromagnet that perform the conversion from electrical information to magnetic and back again. Each bit of data is written to the disk using an encoding method that translates zeros and ones into patterns of magnetic flux reversals. Don’t be put off by Star Wars lingo like “magnetic flux reversal” --it just means flipping the magnet around to the other side or “pole.”



Hard Drive Read-Write Head

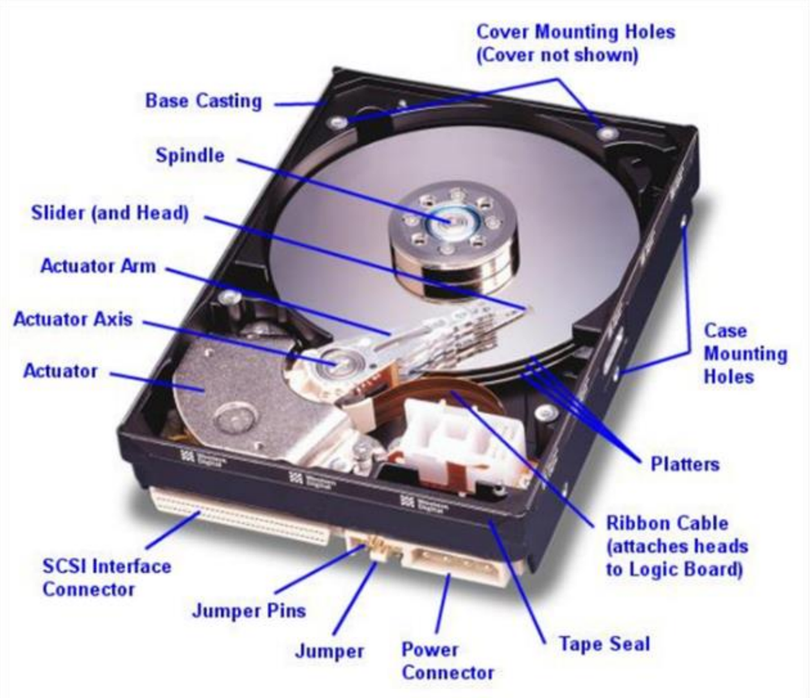
Older hard disk heads make use of the two main principles of electromagnetic force. The first is that applying an electrical current through a coil produces a magnetic field; this magnet field imparts magnetic properties—writes--to the disk. The direction of the magnetic field produced depends on the direction that the current is flowing through the coil. The converse principle is that moving a magnetic field alongside a coil of wire induces an electrical current to flow through the coil. That current corresponds to previously written magnetic information and so serves to “read” the disk. Newer disk heads use different physics and are more efficient, but the basic approach hasn’t changed: electricity to magnetism and magnetism to electricity.

A hard drive is an immensely complex data storage device engineered to appear deceptively simple. When you connect a hard drive to your machine, and the operating system detects the drive, assigns it a drive letter and—presto! —you’ve got trillions of bytes of new storage! Microprocessor chips garner the glory, but the humdrum hard drive is every bit the paragon of ingenuity and technical prowess.

A conventional electromagnetic hard drive is a sealed aluminum box measuring (for a desktop system) roughly 4" x 6" x 1" in height. A hard drive can be located almost anywhere within the computer's case, customarily secured by several screws attached to any of ten pre-threaded mounting holes along the edges and base of the case. One face of the case is labeled to reflect the drive specifications, while a printed circuit board containing logic and controller circuits will cover the opposite face.

A conventional electromagnetic hard disk contains round, flat discs called **platters**, coated on both sides with a special material able to store data as magnetic patterns. Much like a record player, the platters have a hole in the center allowing multiple platters to be stacked on a spindle for greater storage capacity.

The platters rotate at high speed—typically 5,400, 7,200 or 10,000 rotations per minute—driven by an electric motor. Data is written to and read from the platters by **read/write heads** mounted on the end of a pivoting extension called an **actuator arm** that functions similarly to the tone arm that carried a phonograph cartridge and needle across the face of a vinyl record. Each platter has two read/write heads, one on the top of the platter and another on the bottom. So, a conventional hard disk with three platters typically sports six surfaces and six read/write heads.



Unlike a record player, the read/write head never touches the spinning platter. Instead, when the platters spin up to operating speed, their rapid rotation causes a cushion of air to flow under the read/write heads and lift them off the surface of the disk—the same principle of lift that operates on aircraft wings and enables them to fly. The head then reads the magnetic patterns on the disc while flying just .5 millionths of an inch above the surface. At this speed, if the head bounces against the surface, there is a good chance that the head will burrow into the surface of the platter, obliterating data, destroying both read/write heads and rendering the hard drive inoperable—a so-called "head crash."

The hard disk drive has been around for more than 50 years, but it was not until the 1980's that the physical size and cost of hard drives fell sufficiently for their use to be commonplace.

IBM 350 Disk Storage Unit

Introduced in 1956, the IBM 350 Disk Storage Unit pictured was the first commercial hard drive. It was 60 inches long, 68 inches high and 29 inches deep (so it could fit through a door). Called the **RAMAC** (for Random Access Method of Accounting and Control), it held fifty 24" magnetic disks of 50,000 sectors, each storing 100 alphanumeric (7-bit) characters. Thus, it held about **3.75 megabytes**, or one or two cellphone snapshots today. It weighed a ton (literally), and users paid $3,200.00 per month to *rent* it. That's about $30,000.00 in today's dollars.

Now, you can buy a ten terrabyte hard drive storing *two million times* more information for a fraction of that monthly rental. That 10TB drive weighs less than two pounds, can hide behind a paperback book and costs less than $200.00.

Over time, hard drives took various shapes and sizes (or "form factors" as the standard dimensions of key system components are called in geek speak). Two electromagnetic drive form factors are still in use: 3.5" (desktop drive) and 2.5" (laptop drive). A third, the 1.8" (iPod and microsystem drive, is wholly supplanted by solid state storage.

Hard drives connect to computers by various mechanisms called "interfaces" that describe both how devices "talk" to one-another as well as the physical plugs and cabling required. The five most common hard drive interfaces are:

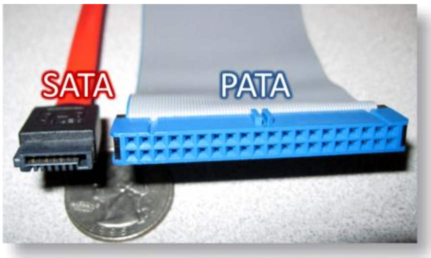
PATA for **P**arallel **A**dvanced **T**echnology **A**ttachment (sometimes called EIDE for **E**xtended **I**ntegrated **D**rive **E**lectronics) [obsolete]

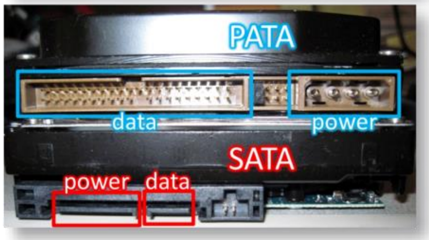
SATA for **S**erial **A**dvanced **T**echnology **A**ttachment [most common]

SCSI for **S**mall **C**omputer **S**ystem Interface

SAS for **S**erial **A**ttached **S**CSI

FC for **F**ibre **C**hannel





Though once dominant in personal computers, PATA drives largely disappeared after 2006. Today, virtually all laptop and desktop computers employ SATA drives for local storage. SCSI, SAS and FC drives tend to be seen exclusively in servers and other applications demanding high performance and reliability.

From the user's perspective, PATA, SATA, SCSI, SAS and FC drives are indistinguishable; however, from the point of view of the technician tasked to connect to and image the contents of the drive, the difference implicates different tools and connectors.

The five drive interfaces divide into two employing parallel data paths (PATA and SCSI) and three employing serial data paths (SATA, SAS and FC). Parallel ATA interfaces route data over multiple simultaneous channels necessitating 40 wires where serial ATA interfaces route data through a single, high- speed data channel requiring only 7 wires. Accordingly, SATA cabling and connectors are smaller than their PATA counterparts (see photos, right).

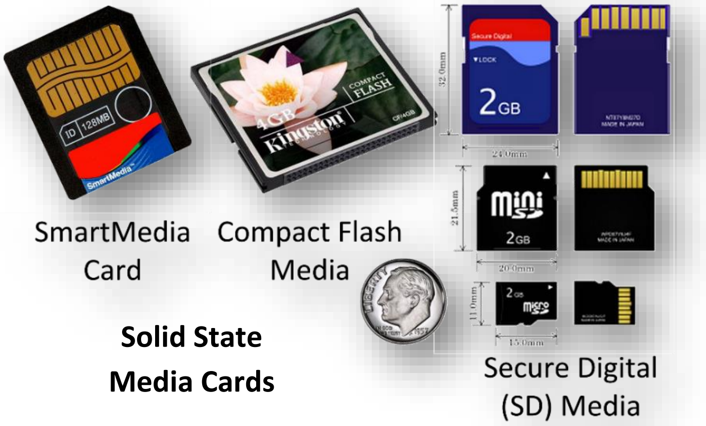
Fibre Channel employs optical fiber (the spelling difference is intentional) and light waves to carry data at impressive speeds. The premium hardware required by FC dictates that it will be found in enterprise computing environments, typically in conjunction with a high capacity/high demand storage device called a **SAN** (for Storage Attached Network) or a **NAS** (for Network Attached Storage).

It's easy to become confused between hard drive interfaces and external data transfer interfaces like USB, Thunderbolt or (the now obsolete) FireWire seen on external hard drives. The drive within the external hard drive housing will employ one of the interfaces described above (except FC); however, to facilitate external connection to a computer, a device called a **bridge** will convert data written to and from the hard drive to a form that can traverse a USB or Thunderbolt connection. In some compact, low-cost external drives, manufacturers dispense with the external bridge board altogether and build the USB interface right on the hard drive's circuit board.

Flash Drives, Memory Cards, SIMs and Solid-State Drives

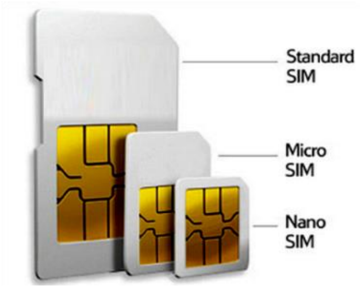
Many late-model laptops and nearly all portable computing devices and phones employ data storage devices with no moving parts where the data resides entirely within the solid semiconductor material which comprise the memory chips, hence the term, "solid-state."

Historically, rewritable solid-state storage was volatile (in the sense that data disappeared when power was withdrawn) and expensive.



Solid State Media Cards

Beginning around 1995, a type of non-volatile memory called NAND flash became inexpensive enough to be used for removable storage in emerging applications like digital photography. Further leaps in the capacity and dips in the cost of NAND flash led to the near-eradication of film for photography and the extinction of the floppy disk, replaced by simple, inexpensive and reusable USB storage devices called, variously, Smart Media, Compact Flash media, SD cards, flash drives, thumb drives, pen drives and memory sticks or keys.



SIM Cards

A specialized form of solid-state memory seen in cell phones is the **Subscriber Identification Module** or **SIM card**. SIM cards serve both to authenticate and identify a communications device on a cellular network and to store SMS messages and phone book contacts.

USB Flash Drives

As the storage capacity of NAND flash has gone up and its cost has come down, the conventional electromagnetic hard drive is rapidly being replaced by **solid-state drives** in standard hard drive form factors. Solid-state drives are significantly faster, lighter and more energy efficient than conventional drives, but they currently cost substantially more per gigabyte stored than their mechanical counterparts. All signs point to the ultimate obsolescence of mechanical drives by solid-state drives, and some products (notably tablets like the iPad and Microsoft Surface and premium laptops) have eliminated hard drives altogether in favor of solid-state storage.

Until recently, solid state drives assumed the size and shape of mechanical drives to facilitate compatibility with existing devices. However, the size and shape of mechanical hard drives was driven by the size and operation of the platter they contain. Because solid state storage devices have no moving parts, they can assume virtually any shape. So, slavish adherence to 2.5" and 3.5" rectangular form factors is fading in favor of shapes and sizes uniquely suited to the devices that employ them.

Today, it's common for solid state drives to take one of three forms. The first is the familiar 2.5" enclosure with a conventional SATA interface (below left). Ultraportable laptops employ either NVMe SSD cards like the one lower right top or M.2 SSD "sticks" seen at lower right bottom. M.2 cards are 22mm wide but come in lengths from 42mm to 110mm, with 80mm (3.1") most common.

With respect to e-discovery, the shift from electromagnetic to solid-state drives is inconsequential. However, the move to solid-state drives will significantly impact matters necessitating computer forensic analysis. Because the NAND memory cells that comprise solid-state drives wear out rapidly with use, solid-state drive controllers must constantly reposition data to ensure usage is distributed across all cells. Such "wear leveling" hampers techniques that forensic examiners have long employed to recover deleted data from conventional hard drives.

RAID Arrays (Hard Drives Working Together)

Whether local to a user or in the Cloud, hard drives account for nearly all the electronically stored information attendant to e-discovery. In network server and Cloud applications, hard drives rarely operate singly. Instead, hard drives are ganged together to achieve greater capacity, speed and reliability in so-called **R**edundant **A**rrays of **I**ndependent **D**isks or **RAID**s. In the Storage Area

Network (SAN) device pictured at left, the 16 hard drives housed in trays may be accessed as **J**ust a **B**unch **o**f **D**isks or **JBOD**, but it's far more likely they are working together as a RAID

RAIDs have two advantages over single drives: redundancy and performance. The redundancy aspect is obvious—two drives holding identical data safeguard against data loss due to mechanical failure of either drive—but how do multiple drives improve **performance**? The answer lies in dividing the data across multiple drives using a technique called **striping**. Physical movement of disks and heads in a mechanical hard drive is a glacially-slow process compared to the lightspeed pace of electrons in a circuit; however, you can speed the mechanical transfer by reading and writing data to and from several drives at the same time

A RAID improves performance by allocating data across more than one physical drive, supporting such simultaneous reads. Each split swath of data in an array is called a "stripe" and the method of depositing the data across drives is called "striping." If you imagine the drives lined up alongside one-another, you can see why moving back-and-forth between them to store data is akin to painting a stripe across the drives. By striping data, each drive can deliver its share of the data simultaneously, increasing the amount of information handed off to the computer's microprocessor, *i.e.*, faster *throughput.*

But, when you stripe data across drives, you lose Information if *any* drive holding striped data fails. You gain performance at the expense of security.

This type of RAID configuration is called a **RAID 0**. It wrings maximum performance from a storage system; but it's risky.

If RAID 0 is for gamblers, **RAID 1** is for the risk averse. A RAID 1 configuration duplicates everything from one drive to an identical twin, so that a failure of one drive won't lead to data loss. RAID 1 doesn't improve performance, and it requires twice the hardware to store the same information.

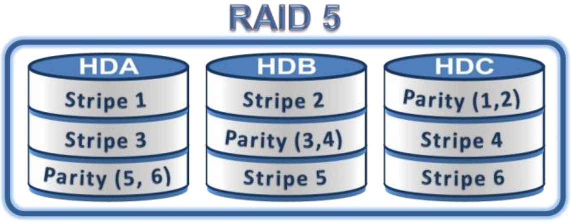
A helpful way to remember which RAID is which: *When a drive fails using RAID 1, you still got one copy of the data; when a drive fails using RAID 0, you've got nothing.*

Other RAID configurations blend the *performance* features of RAID 0 with the *protection* of RAID 1.

Thus, a "RAID 0+1" mirrors two striped drives, but demands four hard drives delivering only half their total storage capacity. Safe and fast, but not cost-efficient. The safety flows from a concept called ***parity,*** key to a range of other sequentially numbered RAID configurations. Of those other configurations, the ones most often seen are **RAID 5 and RAID 7.**

To understand parity, consider the simple equation 5 + 2 = 7. If you didn't know one of the three values in this equation, you could easily solve for the missing value, *i.e.,* presented with "5 + __ = 7," you know the missing value is 2. In this example, "7" is the ***parity value*** or **checksum** for "5" and "2."

The same process is used in RAID configurations to gain increased performance by striping data across multiple drives while using parity values to permit the calculation of any missing values lost to drive failure. In a three-drive array, any one of the drives can fail, and we can use the remaining two to recreate the third (just as we solved for 2 in the equation above).

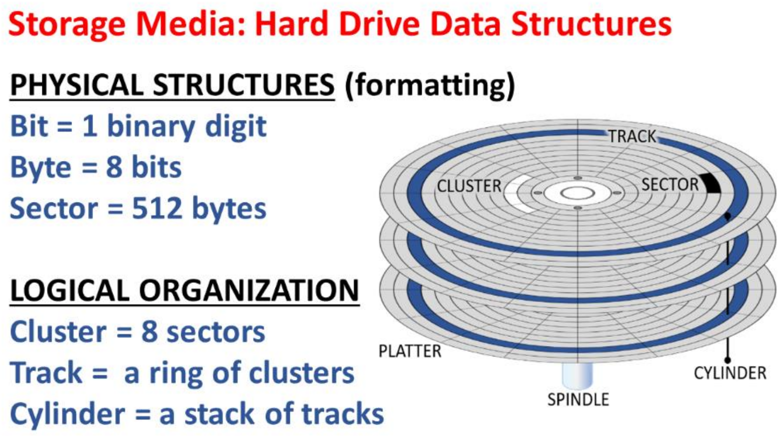


In this illustration, data is striped across three hard drives, HDA, HDB and HDC. HDC holds the parity values for data stripe 1 on HDA and stripe 2 on HDB. It's shown as "Parity (1, 2)." The parity values for the other stripes are distributed on the other drives. Again, any *one* of the three drives can fail, and *all* data is recoverable. This configuration is RAID 5 and, though it requires a minimum of three drives, it scales to dozens or hundreds of disks.

Knowing a little about RAID arrays helps lawyers gauge the burden and cost of preserving a server. Preservation may be as simple as swapping out a single drive from a RAID 1 or entail the duplication of 3 or more drives to preserve a RAID 5.

Sectors, and Clusters and Tracks, Oh My!

Now, we will shift gears and briefly touch on how data resides physically and logically on hard drives. Recall the earlier discussion of electromagnetic hard drives. At the factory, a hard drive’s platters are organized to enable the storage and retrieval of data. This is **low level formatting,** dividing each platter into tens of thousands of densely packed concentric circles called **tracks**. If you could see them (and you can’t because they are nothing more than microscopic magnetic traces), they would resemble the growth rings of the world’s oldest tree. It’s tempting to compare platter tracks to a phonograph record, but a phonograph record’s track is a single spiraling groove, not concentric circles. A track holds far too much information to serve as the smallest unit of storage on a disk, so each track is broken down into physical **sectors**. “Physical” because it resides in a fixed location on the media. A sector is normally the smallest individually addressable unit of information stored on a hard disk and historically held **512 bytes** of information (through about 2010). Today, sector sizes tend to be **4,096 bytes,** but emulate the 512-byte sector size for



Storage Media: Hard Drive Data Structures

PHYSICAL STRUCTURES (formatting)

Bit = 1 binary digit

Byte = 8 bits

Sector = 512 bytes

LOGICAL ORGANIZATION

Cluster = 8 sectors

Track = a ring of clusters

Cylinder = a stack of tracks

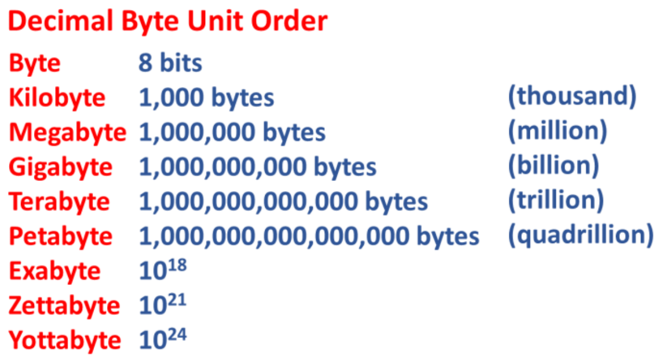
compatibility. The figure at right shows a simplified representation of three platters depicting tracks, cylinders and sectors. The number of tracks, cylinder and sectors is far, far greater that the illustration suggests. Each platter is formatted on both sides for double the information storage, much as a phonograph album was recorded on both sides. Each platter has two read/write heads, one on the top of the platter and another on the bottom. So, a conventional hard disk with three platters uses six surfaces and six read/write heads. Tracks aligned on both sides of the same platter and with tracks on other platters form so-called "cylinders" of data storage.

When electromagnetic hard drives held less data than they do now, file storage locations were based on the physical geometry of the platters, addressed by Cylinder, Head and Sector tuples, so-called **CHS addressing**. Any specific sector could be located by specifying its cylinder, read-write head and sector number. Early hard drives were limited to a maximum of 1024 cylinders, 16 heads (two sides of eight platters) and 63 sectors per track. That meant the maximum addressable capacity of a CHS hard drive formatted in 512-byte sectors was a measly 528 MB[11] (512 x 63 x 16 x 1024)—absurdly small by modern standards but vast thirty years ago. As drive capacities grew, computer companies resorted to a series of schemes to deploy larger drives using software running on the circuit boards of the drives. This "firmware" remapped fake disk geometries to real data locations. Over time, even these workarounds couldn't keep up with burgeoning drive sizes and were abandoned.

The challenge computer scientists faced wasn't increasing the areal density of the drives.[12] New materials and recording technologies ensured those drives were getting huge! Instead, the impediment was coming up with a way to address the vast volume of storage without making older systems obsolete. Today, hard drives employ **LBA** for **Logical Block Addressing**, numbering each sector sequentially and allocating many more bits to catalog the locations of the sectors. Modern computers can address up to 144 petabytes of 512-byte sectors. A petabyte is 1,000 terabytes or one million gigabytes. That should hold us for a while.

[11] 504 MiB

[12] Areal Density describes the quantity of data (in bits) that can be stored on a given surface area of a computer storage medium.



Decimal Byte Unit Order

Byte	8 bits	
Kilobyte	1,000 bytes	(thousand)
Megabyte	1,000,000 bytes	(million)
Gigabyte	1,000,000,000 bytes	(billion)
Terabyte	1,000,000,000,000 bytes	(trillion)
Petabyte	1,000,000,000,000,000 bytes	(quadrillion)
Exabyte	10^{18}	
Zettabyte	10^{21}	
Yottabyte	10^{24}	

To this point, we have described only *physical* units of storage. Platters, cylinders, tracks, sectors and even bits and bytes exist as discrete *physical* manifestations written to the media. Computers manage data not only physically but also *logically*. As it's impractical to manage and gather the data by assembling it from individual sectors, operating system speed the process by grouping sectors into contiguous chunks of data called **clusters**. Just as we don't buy single eggs at the grocery, clusters are akin to the 6-, 12- or 18- egg cartons that have become standard.

A cluster is the smallest amount of disk space that can be allocated to hold a file. Computers organize hard disks based on clusters, which consist of one or more contiguous sectors. *The smaller the cluster size, the more efficiently a disk stores information. Conversely, the fewer the number of clusters, the less space consumed by the table required to track their content and locations.*

To recap, data is stored in <u>logical</u> units called **clusters**, made up of multiple <u>physical</u> storage units termed **sectors**. A series of logical clusters, in turn, comprise **tracks** (concentric circles or "tree rings" of data) on **platters**, one or more disks of rotating electromagnetic storage media within the enclosure of a mechanical hard drive. Tracks that overlie one-another on both sides of a platter and across multiple platters is termed a **Cylinder** (although "cylinder" is an archaic term from the days when hard drive storage was tied to the physical geometry of the formatted disks)**.** In order of data capacity: **Bits > Bytes > Sectors > Clusters > Tracks > Cylinders > Platters > Drive>Array**

Operating Systems and File Systems

As hard disks have grown exponentially in size, using them efficiently is increasingly difficult. A library with thirty books operates much differently than one with 30 billion. The **file system** is the name given to the logical structures and software routines used to control access to the storage on a hard disk system and the overall structure in which files are named, stored and organized. An **operating system** is a large and complex collection of functions, including the user interface and control of peripherals like printers. Operating systems are built on file systems. If the operating system is the car, then the file system is its chassis. Operating systems are known by familiar household names, like **MS-DOS**, **Windows or MacOS**. In contrast, file systems go by obscure monikers like **FAT, FAT32 (DOS), ext2 (Linux), NTFS (Windows) and HFS+ and APFS (Apple)**.

NTFS File Systems

The Microsoft Windows environment, in particular the **NTFS file system** at the heart of Windows NT, 2000, XP, Vista and Windows 7-11, accounts for most personal computers in the world; however, there are many non-Microsoft operating systems out there, such as Unix, Linux and, MacOS. Though similarities abound, these other operating systems use different file systems, and the Unix or Linux operating systems often lie at the heart of corporate and web file servers—today's "big iron" systems and Cloud computing. As well, MacOS usage has grown markedly as Apple products have kicked down the door of business computing and captivated consumers.

NTFS uses a powerful and complex file system database called the **Master File Table** or **MFT** to manage file storage. Understanding the file system is key to appreciating the evidentiary potential of computer forensics*, viz.*, why deleted data doesn't necessarily go away and where probative artifacts reside. It's the file system that marks a file as deleted though it leaves the data on the drive. It's the file system that enables the creation of multiple partitions where data can be hidden from prying eyes. Finally, it's the file system that determines the size of a disk cluster with the attendant persistence of data within the slack space.

Formatting and Partitioning

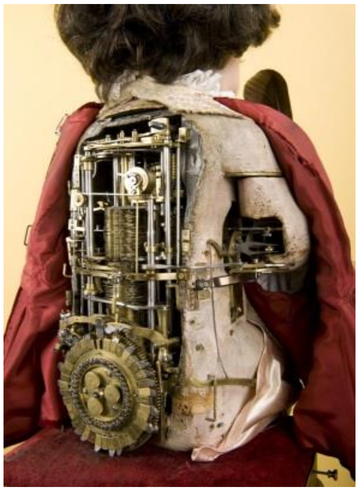
Partitioning divides drives into **volumes,** which users see as drive letters (*e.g.,* C:, E:, F: and so on). **Formatting** defines the logical structures on the partition and places necessary operating system files at the start of the disk to facilitate booting. For most users, their computer comes with their hard drive partitioned as a single volume (universally called C:). Windows machines may also come with a hidden recovery partition holding files needed to repair the operating system. Some users will find (or will cause) their hard drive to be partitioned into multiple volumes, each appearing to the user as if it were an independent disk drive. Partitions can be designated "**active"** and **"inactive.** Only one partition may be designated as active at any given time, and that partition is the one that boots the computer. The significance in computer forensics is that inactive partitions are invisible to anyone using the computer unless they know to look for them and how to find them. Inactive partitions are a place where users with something to conceal from prying eyes may choose to hide it.

Computers

Historically, all sorts of devices—even people—were "computers." During World War II, human computers—women for the most part—were instrumental in calculating artillery trajectories and assisting with the challenging number-crunching needed by the Manhattan Project.

Today, laptop and desktop personal computers spring to mind when we hear the term “computer;” yet smart phones, tablet devices, global positioning systems, video gaming platforms, televisions and a host of other intelligent tools and toys are also computers. More precisely, the **central processing unit** (CPU) or **microprocessor** of a system is the “computer,” and the various input and output devices that permit humans to interact with the processor are termed **peripherals.** The key distinction between a mere calculator and a computer is the latter’s ability to be programmed and its use of **memory** and **storage**. The physical electronic and mechanical components of a computer are its **hardware,** and the instruction sets used to program a computer are its **software.**

In 1774, a Swiss watchmaker named Pierre Jaquet-Droz built an ingenious mechanical doll resembling a barefoot boy. Constructed of 6,000 handcrafted parts and dubbed "L'Ecrivain” (“The Writer”), Jaquet-Droz’ automaton uses quill and ink to handwrite messages in cursive, up to 40 letters long, with the content controlled by interchangeable cams. The Writer is a charming example of an early programmable computer.

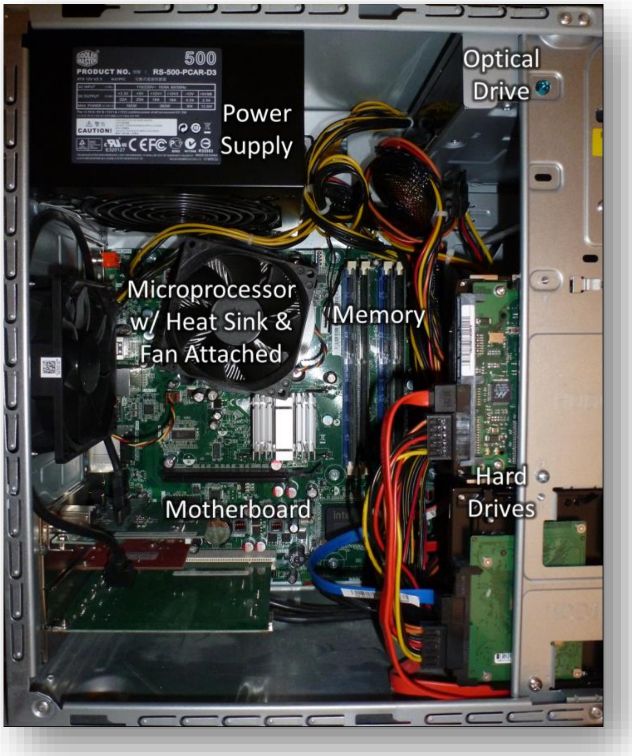


When you push the power button on your computer, you trigger an extraordinary, expedited education that takes the machine from insensible illiterate to worldly savant in seconds. The process starts with a snippet of data on a chip called the ROM BIOS storing just enough information in its **R**ead **O**nly **M**emory to grope around for the **B**asic **I**nput and **O**utput **S**ystem peripherals (like the keyboard, screen and, most importantly, the hard drive). The ROM BIOS also holds the instructions needed to permit the processor to access more and more data from the hard drive in a widening gyre, “teaching” itself to be a modern, capable computer.

Unlike the interchangeable cams of Pierre Jaquet-Droz’ mechanical doll, modern electronic computers receive their instructions in the form of data retrieved from the same electronic storage medium as the digital information upon which the computer performs its computational wizardry.

This rapid, self-sustaining self-education is as magical as if you lifted yourself into the air by pulling on the straps of your boots, which is truly why it’s called “bootstrapping” or just “booting” a computer.

Computer hardware shares certain common characteristics. Within the CPU, a microprocessor chip is the computational "brain" of system and resides in a socket on the **motherboard**, a rigid surface etched with metallic patterns serving as the wiring between the components on the board. The microprocessor generates considerable heat necessitating the attachment of a heat dissipation device called a **heat sink**, often abetted by a small fan. The motherboard also serves as the attachment point for memory boards (grouped as modules or "sticks") called **RAM** for Random Access Memory. RAM serves as the working memory of the processor while it performs calculations; accordingly, the more memory present, the more information can be processed at once, enhancing overall system performance.



Other chips comprise a Graphics Processor Unit (GPU) residing on the motherboard or on a separate expansion board called a **video card** or **graphics adapter**. The GPU supports the display of information from the processor onto a monitor or projector and has its own complement of memory dedicated to superior graphics performance. Likewise, specialized chips on the motherboard or an expansion board called a **sound card** support the reproduction of audio to speakers or a headphone. Video and sound processing capabilities may even be fully integrated into the microprocessor chip.

The processor communicates with networks through an interface device called a **network adapter** which connects to the network physically, through a Local Access Network or **LAN Port,** or wirelessly using a Wi-Fi or Bluetooth connection.

Users convey information and instructions to computers using tactile devices like a keyboard, mouse or track pad, but may also employ voice or gestural recognition mechanisms.

Persistent storage of data is a task delegated to other peripherals: previously **optical drives** (CD-ROM and DVD-ROM devices) and **floppy disk drives**, now **solid-state media** (*i.e.,* thumb drives) and, most commonly, **hard drives**.

All the components just described require electricity, supplied by batteries in portable devices or by a **power supply** converting AC current to the lower DC voltages required by electronics. To guard against data loss from power failure, computing systems employ battery-powered Uninterruptible Power Supplies (UPS) that supply electricity until power is restored.

From the standpoint of digital evidence, it's less important to define these devices than it is to comprehend the information they hold, the places it resides and the forms it takes. Parties and lawyers have been punished for their failure to inquire into and understand the roles computers, hard drives and servers play as repositories of electronic evidence. Moreover, much money spent on electronic discovery today is wasted through parties' efforts to convert ESI to paper-like forms instead of learning to work with ESI in the forms in which it customarily resides on computers, hard drives and servers.

Servers

We defined servers as computers dedicated to a specialized task or tasks. But that definition doesn't begin to encompass the profound impact upon society of the so-called **client-server** computing model. The ability to connect local "client" applications to servers via a network, particularly to **database servers**, is central to the operation of most businesses and to all telecommunications and social networking. Google and Facebook are just enormous groupings of servers, and the Internet merely a vast, global array or "cloud" of shared servers.

Local, Cloud and Peer-to-Peer Servers

For e-discovery, let's divide the world of servers into three realms: **Local**, **Cloud** and **Peer-to-Peer** server environments.

"**Local**" or "**on-prem**" servers employ hardware that's physically available to the party that owns or leases the servers. Local servers reside in a computer room on a business' premises or in leased equipment "lockers" accessed at a co-located data center where a lessor furnishes, *e.g.,* premises security, power and cooling. Local servers are often easier to deal with in e-discovery because physical access to the hardware supports more and faster options when it comes to preservation and collection of potentially responsive ESI. Because on-premises ("on prem") computers and networks exist within a party's physical dominion, protected by the party's security protocols, they're often termed "**behind-the-firewall**."[13] The worldwide pandemic forced corporations and e-discovery service providers to broaden use of remote acquisition tools to search and collect potentially responsive ESI and will doubtlessly prompt even more custodians to shift data to the Cloud.

[13] A firewall is a network security device facing the Internet that monitors incoming and outgoing network traffic in order to block or allow data transfers based on a set of rules for safe "white list" or unsafe "black list" transactions.

“**Cloud**” servers typically reside in facilities not physically accessible to persons using the servers, and servers are not typically dedicated to a single user. Instead, the Cloud computing consumer is buying services via the Internet that emulate the operation of a single machine or a room full of machines, all according to the needs of the Cloud consumer. Web mail is the most familiar form of Cloud computing, in a variant called SaaS (for Software as a Service). Webmail providers like Google, Yahoo and Microsoft make e-mail accounts available on their servers in massive data centers, and the data on those servers is available via the Internet, no user having the right to gain physical access to the machines storing their messaging.

“**Peer-to-Peer**” (P2P) networks exploit the fact that any computer connected to a network has the potential to serve data across the network. Accordingly, P2P networks are decentralized; that is, each computer or “node” on a P2P network acts as client and server, sharing storage space, communication bandwidth and/or processor time with other nodes. P2P networking may be employed to share a printer in the home, where the computer physically connected to the printer acts as a print server for other machines on the network. On a global scale, P2P networking is the technology behind file sharing applications like Bit Torrent that have garnered headlines for their facilitation of illegal sharing of copyrighted content. When users install P2P applications to gain access to shared files, they simultaneously (and often unwittingly) dedicate their machine to serving up such content to a multitude of other nodes.

Virtual Servers

Though we’ve so far spoken of server hardware, *i.e.,* physical devices, servers can be deployed *virtually*, through software that *emulates* the functions of a physical device. Such “hardware virtualization” allows for more efficient deployment of computing resources by enabling a single physical server to host multiple virtual servers.

Virtualization is the key enabling technology behind many Cloud services. If a company needs powerful servers to launch a new social networking site, it can raise capital and invest in the hardware, software, physical plant and personnel needed to support a data center, with the attendant risk that it will be over-provisioned or under-provisioned as demand fluctuates. Alternatively, the startup can secure the computing resources it needs by using virtual servers hosted by a Cloud service provider like Amazon or Microsoft. Virtualization permits adding and paring back computing resources commensurate with demand and, being pay-as-you-go, requires little capital investment. Thus, a computing platform or infrastructure can be virtualized and leased, *i.e.,* offered as a service via the internet. Accordingly, Cloud Computing may be termed **PaaS** *(Platform as a Service)* or **IaaS** *(Infrastructure as a Service)*. Web-based applications are **SaaS** *(Software as a Service)*.

It's helpful for attorneys to be aware of the role of **virtual machines** (VMs) because the ease and speed with which VMs are deployed and retired as well as their isolation within the operating system can pose unique risks and challenges in e-discovery, especially with respect to implementing a proper legal hold and when identifying and collecting potentially responsive ESI.

Server Applications

Computers dedicated to server roles typically run operating systems optimized for server tasks and applications specially designed to run in a server environment. In turn, servers often support dedicated tasks such as serving web pages (*Web Server*), retaining and delivering files from shared storage allocations (*File Server*), organizing voluminous data (*Database Server*), facilitating the use of shared printers (*Print Server*), running programs (*Application Server*) or handling messages (*Mail Server*). These various server applications may run physically, virtually or as a mix of the two.

Network Shares

Eventually, all electronic storage devices fail. Even the RAID storage arrays previously discussed do not forestall failure, but instead afford a measure of redundancy to allow for replacement of failed drives before data loss. Redundancy is the sole means by which data can be reliably protected against loss; consequently, companies routinely back up data stored on server NAS and SAN storage devices to backup media like magnetic tape or online (*i.e.,* Cloud) storage services. However, individual users often fail to back up data stored on local drives. Accordingly, enterprises allocate a "share" of network-accessible storage to individual users and "map" the allocation to the user's machine, allowing use of the share as if it were a local hard drive. When the user stores data to the mapped drive, that data is backed up along with the contents of the file server. Although **network shares** are not local to the user's computer, they are typically addressed using drive letters (*e.g.,* M: or T:) as if they were local hard drives. Local network shares have their counterparts in the Cloud, including **Cloud Storage and File-Sharing Services** like Dropbox, Box, Google Drive, Apple iCloud and Microsoft OneDrive.

Practice Tips for Computers, Hard Drives and Servers

Your first hurdle when dealing with computers, hard drives and servers in e-discovery is to identify potentially responsive sources of ESI and take appropriate steps to inventory their relevant contents, note the form and associated metadata of the potentially responsive ESI, then preserve it against spoliation. Get a handle on data volumes, file types, metadata, replication and distribution as early in the litigation process as possible as these determine the overall cost to preserve, collect, process and host ESI in discovery.

Take stock of physical computing and storage devices. For each machine, device or repository holding potentially responsive ESI, you may wish to collect the following information (as applicable):

- Manufacturer and model
- Serial number and/or service or asset tag
- Operating system
- Custodian
- Location
- Type of storage (don't miss removable media, like SD and SIM cards)
- Aggregate storage capacity (in MB, GB or TB)
- Encryption status
- Credentials (user IDs and passwords), if encrypted
- Prospects for upgrade or disposal
- If you'll preserve ESI by drive imaging, it's helpful to identify device interfaces.

For servers, further information might include:

- Purpose(s) of the server (*e.g.,* web server, file server, print server, etc.)
- Names and contact information of server administrator(s)
- Time in service and data migration history
- Whether hardware virtualization is used
- RAID implementation(s)
- Users and privileges
- Logging and log retention practices
- Backup procedures and backup media rotation and retention
- Whether the server is "mission critical" and cannot be taken offline or can be downed.

When preserving the contents of a desktop or laptop computer, it's typically unnecessary to sequester any component of the machine other than its hard drive(s) since the ROM BIOS holds little information beyond the rare forensic artifact. Before returning a chassis to service with a new hard drive, be sure to document the custodian, manufacturer, model and serial number/service tag of the redeployed chassis, retaining this information with the sequestered hard drive.

The ability to fully explore the contents of servers for potentially responsive information hinges upon the privileges extended to the user. Be sure that the person tasked to identify data for preservation or collection holds administrator-level privileges.

Above all, remember that computers, hard drives and servers are constantly changing while in service. Simply rebooting a machine alters system metadata values for large numbers of files. Accordingly, you should consider the need for evidentiary integrity before exploring the contents of a device, at least until appropriate steps are taken to guard against unwitting alteration. Note also that connecting an evidence drive to a new machine effects changes to the evidence unless suitable write blocking tools or techniques are employed.

Hashing Data

Because all digital data are numbers, the arithmetic around parity values helps guard against data loss. More advanced math called "hashing" makes it possible to authenticate, deduplicate and cull digital data. Hashing is the use of mathematical algorithms to calculate a unique sequence of letters and numbers to serve as a reliable digital "fingerprint" for electronic data. These sequences are called "message digests" or, more commonly, "hash values."[14] Hashing is an invaluable tool in both computer forensics and electronic discovery and deployed by courts with growing frequency.[15]

Using a hash algorithm, any amount of data—from a tiny file to the contents of entire hard drives and beyond—can be expressed as an alphanumeric sequence of fixed length. The most common forms of hashing are MD5 and SHA-1. MD-5 is a 128-bit (16 byte) value typically expressed as 32 hexadecimal (Base16) characters.

A hash value is just a big, big, BIG number calculated on the contents of the file. A 128-bit number can be as large as 2^{128} – if you start doing the 2 x 2 x 2 x 2, etc. on that, you'll see how fast the values mount.

To say 128 bits or 2^{128} is a "big, big, BIG number" doesn't begin to convey its unfathomable, astronomic scale. In decimal terms, it's about 340 *billion billion billion billion* (a/k/a 340 undecillion). *That's 4 quadrillion times the number of stars in the observable universe!*

A SHA-1 hash value is an even larger 160-bit (20 byte) value typically expressed as 40 hex characters. So, a SHA-1 value is a WAY bigger number—*4.3 billion times bigger*.

The MD5 hash value of the plain text of Lincoln's Gettysburg Address is E7753A4E97B962B36F0B2A7C0D0DB8E8. ***Anyone, anywhere performing the same hash calculation on the same data will get the same unique value*** in a fraction of a second. But change "Four score" to "Five score" and the hash becomes 8A5EF7E9186DCD9CF618343ECF7BD00A. However subtle the alteration—an omitted period or extra space—the hash value changes markedly. The chance of an altered electronic document having the same MD5 hash—a "hash collision" in cryptographic parlance—is one in 340 *trillion, trillion, trillion*. Though supercomputers

[14] Please don't say "hash marks," unless you are speaking of insignia denoting military rank or the yard markers on a football field. The one-way cryptographic calculations used to digitally fingerprint blocks of data are "hash values," "hashes" or "message digests."

[15] In 2017, Federal Rule of Evidence 902 was amended to support self-authentication of digital evidence when supported by a process of digital identification like hashing. Fed. R. Evid. 902(14).

have fabricated collisions, it's still a level of reliability far exceeding that of fingerprint and DNA evidence.

Hashing sounds like rocket science—and it's a miraculous achievement—but it's very much a routine operation, and the programs used to generate digital fingerprints are freely available and easy to use. Hashing lies invisibly at the heart of everyone's computer and Internet activities[16] and supports processes vitally important to electronic evidence, including identification, filtering, Bates numbering, authentication, de-duplication and blockchain authentication.

Identification

Knowing a file's hash value enables you to find its identical counterpart within a large volume of data without examining the contents of each file. The government uses this capability to ferret out child pornography, but you might use it to track down company secrets that flew the coop when an employee joined the competition.

Filtering and De-NISTing

A common e-discovery process is to cull data collected from computers that couldn't be evidence because it isn't a custodian's work product. It's done by matching hash values of collected data files to hash values on the National Software Reference Library's (NSRL's) freely published list of hash values corresponding to common retail software and operating systems. The NSRL is part of the National Institute for Standards and Technology (NIST), so this process is commonly called **"de-NISTing**" a data set. For more information on the NSRL, visit http://www.nsrl.nist.gov/.

Bates Numbering

Hashing's ability to uniquely identify e-documents makes it a candidate to supplement, though not supplant, traditional **Bates numbering**[17] in electronic production in discovery. Though hash values don't fulfill the sequencing function of Bates numbers, they're excellent unique identifiers and enjoy an advantage over Bates numbers because they

Mechanical Bates Stamp

[16] For example, many web services store the hash value of your password, but not the password itself. This enables them to authenticate a user by comparing the hash of the password entered to the hash value on file; however, the password cannot be reversed engineered from the hash value. A remarkable feature of hash values is that they are one-way calculations meaning that although the hash value identifies just one sequence of data, it reveals nothing *about* the data, much as a fingerprint uniquely identifies an individual but reveals nothing about their appearance or personality —it's computationally infeasible to derive the source data from the hash of the source data.

[17] Bates numbering has historically been employed as an organizational method to label and identify legal documents, especially those produced in discovery. "Bates" is capitalized because the name derives from the Bates Manufacturing Company, which patented and sold auto-incrementing, consecutive-numbering stamping devices. Bates stamping served the dual functions of sequencing and uniquely identifying documents.

eliminate the possibility that the same number might be applied to different documents. An electronic document's hash value derives from its contents, so will never conflict with that of another document unless the two documents are identical. Similarly, because two identical documents from different custodians (sources) will hash identically, the documents' hash values won't serve to distinguish between the two despite their different origins.

Authentication

Forensic examiners extensively use hashing to establish that a forensically sound duplicate of a hard drive faithfully reflects every byte of the source evidence drive and to prove that their activities haven't altered the original evidence. As e-discovery gravitates to production of native file formats instead of static page images, concern about intentional or inadvertent alteration requires lawyers to have a fast, reliable method to authenticate electronic documents. Hashing neatly fills the bill. In practice, a producing party calculates and records the hash values of all items produced in native format. Were there suspicion, alteration is apparent merely by hashing the file.

De-duplication

In e-discovery, manually reviewing vast volumes of identical data is burdensome and poses a significant risk of conflicting relevance and privilege assessments. Hashing serves to flag identical documents, permitting a single, consistent assessment of an item that might otherwise have cropped up hundreds of times and been differently characterized. This is **hash de-duplication**, and it drastically cuts the cost of reviewing data for responsiveness and privilege. But because even the slightest difference triggers different hash values, insignificant variations between files (e.g., different Internet paths taken by otherwise identical e-mail messages) may frustrate hash de-duplication. An alternative is to hash relevant *segments* of e-documents to assess their relative identicality, a practice sometimes called "**near de-duplication**."

In practice, each file processed, and each constituent item extracted, is hashed and their hash values compared to the hash values of items previously processed and extracted to determine if the file or item has been seen before. Thereafter, files and items with matching hashes are suppressed as duplicates, and instances of each duplicate and associated metadata are noted in a deduplication or occurrence log.

Takeaways on Hashing

The most important things to know about hashing:

1. Electronically stored information of any type or size can be hashed;

2. The algorithms used to hash data are not proprietary, and thus cost nothing to use;

3. No matter the size of the file that's hashed, its hash value is *always* a fixed length;

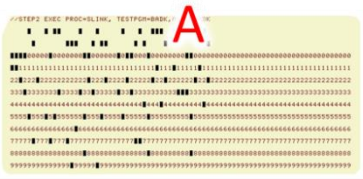
4. The two most common hash algorithms are called MD5 and SHA-1.

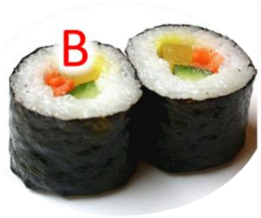
5. In a random population of hashed data, no one can reverse engineer a file's hash value to reveal anything about the file

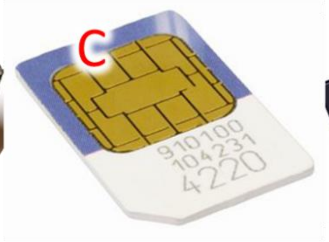
6. The chance of two different files accidentally having matching MD5 hash values (a so-called *hash collision*) is one in 340 *trillion trillion trillion (i.e*., 340 undecillion). So, it is highly improbable that two files with matching hash values are not identical

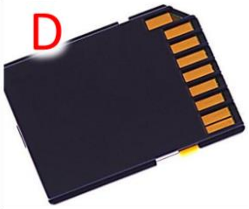
NOTE TO STUDENTS: We are seeking to lay a solid foundation in terms of your grasp of the fundamentals of information technology and the jargon used in e-discovery. Looking back over the preceding material, please list any topics and terms you don't understand and share your list with the instructor in order that we might go over those topics in class. Don't be shy!

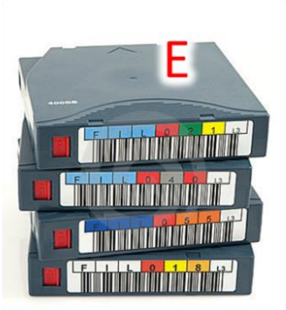
❁ Exercise 1: Identifying Digital Storage Media

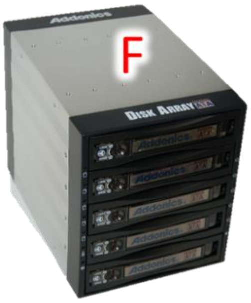


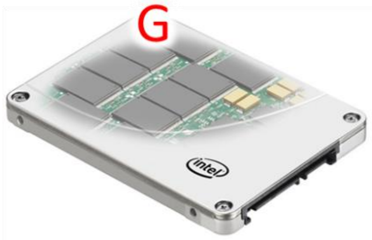


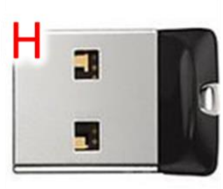


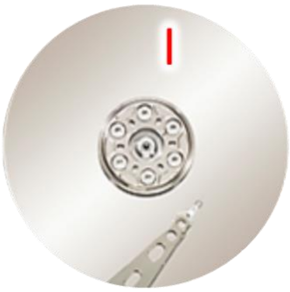


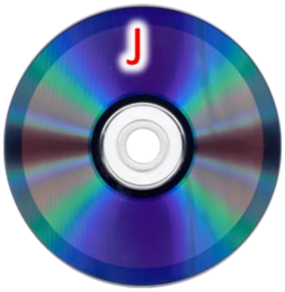


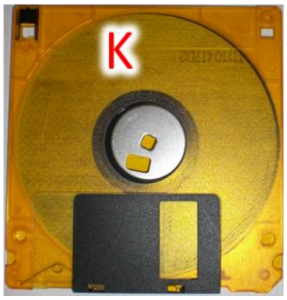


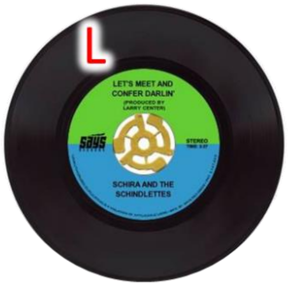












Insert the letter(s) of the media described in the adjacent blank:

1. Backup Tape _________
2. USB Thumb drive _________
3. SD media card _________
4. SIM Card _________
5. RAID array _________

1. Identify three items that record data *electromagnetically*: _____ _____ _____

2. Identify four *solid state* digital storage devices: _____ _____ _____ _____

3. Identify the three items with *the most meager* digital information storage capacity: _____ _____ _____

Introduction to Metadata

In the old joke, a balloonist descends through the fog to get directions. “Where am I?” she calls out to a man on the ground, who answers, “You’re in a yellow hot air balloon about sixty-seven feet above the ground.” The frustrated balloonist replies, “Thanks for nothing, Counselor.” Taken aback, the man on the ground asks, “How did you know I’m a lawyer?” “Simple,” says the balloonist, “your answer was 100% accurate and totally useless.”

If you ask a tech-savvy lawyer, “What’s metadata?” there’s a good chance you’ll hear, “Metadata is data about data.” Another answer that’s 100% accurate and totally useless.

It’s time to get past defining metadata as *data about data.*

It’s time to move past “data about data” and embrace more useful ways to describe metadata—ways that enable counsel to rationally assess relevance and burden attendant to metadata. Metadata may be the most misunderstood topic in electronic discovery. Requesting parties demand discovery of “the metadata” without specifying what metadata is sought, and producing parties fail to preserve or produce metadata of genuine value and relevance.

It’s Information *and* Evidence

Metadata is information that helps us use and make sense of other information. More particularly, metadata is information stored electronically that describes the characteristics, origins, usage, structure, alteration and validity of other electronic information. Many instances of metadata in many forms occur in many locations within and without digital files. Some are supplied by the user, but most metadata are generated by systems and software. Some is crucial evidence, and some is merely digital clutter. Appreciating the difference--knowing what metadata exists and understanding its evidentiary significance—is a skill essential to electronic evidence and discovery.

Metadata is Evidence!

If evidence is anything that tends to prove or refute an assertion as fact, then clearly metadata is evidence. Metadata sheds light on the origins, context, authenticity, reliability and distribution of electronic evidence, as well as provides clues to human behavior. It’s the electronic equivalent of DNA, ballistics and fingerprint evidence, with a comparable power to exonerate and incriminate.

In ***Williams v. Sprint/United Mgmt. Co.***, 230 F.R.D. 640 (D. Kan. 2005), the federal court ruled:

[W]hen a party is ordered to produce electronic documents as they are maintained in the ordinary course of business, the producing party should produce the electronic documents with their metadata intact, unless that party timely objects to production of metadata, the parties agree that the metadata should not be produced, or the producing party requests a protective order.

Within the realm of metadata lies discoverable evidence that litigants are obliged to preserve and produce. There's as much or more metadata extant as there is information and, like information, you don't deal with every bit of it. You *choose* wisely.

Often, files have *more* metadata than content.

A lawyer's ability to advise a client about how to find, preserve and produce metadata, or to object to its production and discuss or forge agreements about metadata, hinges upon how well he or she understands metadata.

It's Just Ones and Zeroes

Understanding metadata and its importance in e-discovery begins with awareness that electronic data is, fundamentally, just numbers. Though you've heard that before, you may not have considered the implications of information being expressed so severely. There are no *words*. There are no spaces or punctuation. *There is no delineation of any kind*. *Solely binary numbers.*

How, then, do computers convert this unbroken sequence notated as ones and zeroes into information that makes sense to human beings? There must be some *key*, some *coherent structure* imposed to divine their meaning. But where does it come from? We can't derive it *from* the data if we can't first make sense *of* the data.

It's Encoded

Consider that written English conveys all information using fifty-two upper- and lowercase letters of the alphabet, ten numerical digits (0-9), some punctuation marks and a few formatting conventions, like spaces, lines, pages, etc. You can think of these collectively as a seventy- or eighty-character "code." Alternatively, the same information could be communicated or stored in Morse code, where a three-signal code composed of dot, dash and pause serves as the entire "alphabet."

We've all seen movies where a tapping sound is heard and someone says, "Listen! It's Morse code!" Suddenly, the tapping is an encoded *message* because someone has furnished metadata ("It's Morse code!") *about* the data (tap, tap, pause, tap). Likewise, all those ones and zeroes on a computer only make sense when other ones and zeroes—*metadata*—reveal a framework for parsing and interpreting the data.

All those ones and zeroes on a computer only make sense when *other* ones and zeroes—*metadata*—reveal a framework for parsing and interpreting the data.

So, we *need* data *about* the data. We need information that tells us the data's encoding scheme. We need to know when information of one sort concludes and different information begins. We need the name, date, context, purpose and origin of information to support its utility and integrity. We need its ***metadata***.

The Metadata Continuum

Sometimes metadata is elemental, like the contents of a computer's master file table detailing where the sequences of one and zeroes for files begin and end. This metadata is invisible to a user without special tools called hex editors capable of peering through the walls of the Windows interface into the utilitarian plumbing of the operating system. Without file location metadata, each time a user sought to access a file or program, the operating system would have to peruse the entire drive to find it. It'd be like looking for someone by knocking on every door in town!

At other times, metadata supports enhanced functionality not essential to the operation of the system. The metadata that tracks a file's name or the dates a file was created or last modified may only occasionally be probative of a claim or defense in a case, but that information *always* makes it easier to locate, sort and segregate files.

Metadata is often instrumental to the intelligibility of information, helping us make sense of it. "Sunny and 70 degrees" aren't a very useful forecast without metadata indicating *when* and *where* it's the weather. Similarly, understanding information on a website or within a database, a collaborative environment like Microsoft's SharePoint or a social network like Facebook depends on metadata that defines its location, origin, timing and structure. It's even common for computerized information to comprise *more* metadata than data, in the same way that making sense of the two data points "sunny" and "70 degrees" requires *three* metadata points: location, date and time of day.

There's No Such Thing as "The Metadata"

As we move up the evolutionary ladder for metadata, some is recorded in case it's needed to support a specialized task for the operating system or an application. Standard **System Metadata** fields like "Camera Model" or "Copyright" may seem an utter backwater to a lawyer concerned with spreadsheets and word-processed documents; but, if the issue is the authenticity of a photograph or the origins of pirated music, these fields can make or break a case. ***It's all about relevance and utility.***

The point is, there's really no such thing as "the metadata" for a file or document. Instead, there's a continuum of **System** and **Application Metadata** that enlightens many aspects of ESI. The metadata that matters depends upon the issues presented in the case and the task to be accomplished; consequently, the metadata preserved for litigation should reasonably reflect the issues to be reasonably anticipated, and it must also address the file management and integrity needs attendant to identification, culling, processing, review and presentation of electronic evidence.

File Systems and Relative Addressing

Most of those ones and zeroes on q hard drive are files that, like library books, are written, read, revised and referenced. Computers use file systems to keep track of files just as libraries once used card catalogues and the Dewey Decimal system to track books.

Imagine you own a thousand books without covers that you stored on one very long shelf. You also own a robot named Robby that can't read, but Robby can count books very accurately. How would you instruct Robby to get a particular book?

If you know the order in which the books are stored, you might say, "Robby, bring me the 412th book." If it was a 24-volume set of encyclopedias, you might add: "…and the next 23 books." The books don't "know" where they're shelved. Each book's location is metadata *about* the book, but it's not stored within the book. The system tracks that metadata. *System. Metadata.*

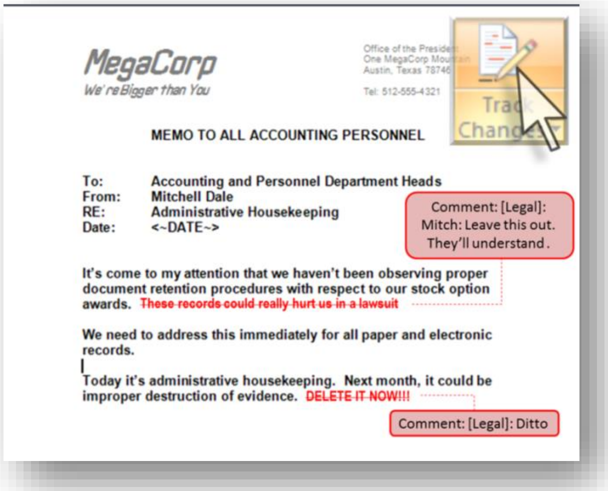
Locating something by specifying that it's so many units from a particular point is called *relative addressing* or *offset addressing*. The number of units the destination is set off from the specified point is called the *offset*. Computers use offset values to indicate the locations of files on storage devices as well as to locate information inside files.

Computers use various units to store and track information, so offsets aren't always expressed in the same units. As previously explained, a "bit" stores a one or zero, eight bits is a "byte," (sufficient to hold a letter in the Latin alphabet), 512 bytes is often a *sector or block* (see **Appendix A**) and (typically) eight contiguous sectors or blocks is a *cluster*. The cluster is the most common unit of logical storage, and modern computers tend to store files in as many of these 4,096-byte clusters, or "data baskets," as needed. Offset values are couched in bytes when specifying the location of information within files and as sectors when specifying the location of files on storage media.

Application Metadata

To the extent lawyers are familiar with metadata, it's likely just the type called **application metadata** with the fearsome potential to inadvertently reveal confidential or privileged information embedded within electronic documents. Computer programs or "applications" store data in files "native" to them, meaning that the data is structured and encoded to uniquely support the application. As these applications added features--like a word processor's ability to redline changes in or collaborate on a document--the files used to store documents now had to retain those tracked changes and collaborative comments.

Microsoft Word was once notorious for its potential to store information unseen by users, and a cottage industry grew up offering utilities to strip embedded information, like comments and tracked changes, from Word documents. Because of its potential to embarrass lawyers or compromise privilege, metadata acquired an unsavory reputation. But metadata is much more than embedded *application* metadata affording those who know how to find it the ability to dredge up a document's non-obvious content.



MegaCorp
We're Bigger than You

Office of the President
One MegaCorp Mountain
Austin, Texas 78746

Tel: 512-555-4321

MEMO TO ALL ACCOUNTING PERSONNEL

To: Accounting and Personnel Department Heads
From: Mitchell Dale
RE: Administrative Housekeeping
Date: <~DATE~>

It's come to my attention that we haven't been observing proper document retention procedures with respect to our stock option awards. ~~These records could really hurt us in a lawsuit~~

We need to address this immediately for all paper and electronic records.

Today it's administrative housekeeping. Next month, it could be improper destruction of evidence. ~~DELETE IT NOW!!!~~

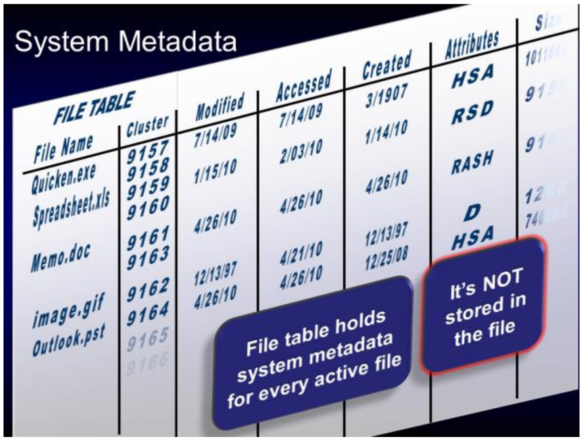
Comment: [Legal]: Mitch: Leave this out. They'll understand.

Comment: [Legal]: Ditto

By definition, **application metadata is embedded in the file it describes and moves with the file when copied.** But not all metadata is embedded for the same reason that cards in a library card catalog aren't stored between the pages of the books: *You need to know where information resides to reach it*.

System Metadata

Unlike books, computer files aren't neatly bound tomes with names embossed on spines and covers. Typically, files don't internally reflect the name they've been given or other information about their location, history or ownership. The information about the file *not* embedded within the file it describes but stored apart from the file is its **system metadata**. The computer's file management system uses *system* metadata to track file locations and store demographics about each file's name, size, creation, modification and usage.



System Metadata

FILE TABLE

File Name	Cluster	Modified	Accessed	Created	Attributes	Size
Quicken.exe	9157 9158	7/14/09	7/14/09	3/1907	HSA	1011
Spreadsheet.xls	9159 9160	1/15/10	2/03/10	1/14/10	RSD	91
Memo.doc	9161 9163	4/26/10	4/26/10	4/26/10	RASH	91
image.gif	9162	12/13/97	4/21/10	12/13/97	D	12
Outlook.pst	9164	4/26/10	4/26/10	12/25/08	HSA	74
	9165 9166					

File table holds system metadata for every active file

It's NOT stored in the file

System metadata is crucial to electronic discovery because so much of our ability to identify, find, sort, cull and authenticate information depends on its system metadata. For example, system metadata helps identify the custodians of files, what the file is named, when files were created or modified and the folders in which they are stored. System metadata stores much of the *who, what, when, where* and *how* of electronic evidence.

Every computer employs one or more databases to keep track of system metadata. In computers running the Windows operating system, the principal "card catalog" holding system metadata is called the Master File Table or "MFT." In the predecessor DOS operating system, it was called the

File Allocation Table or “FAT.” The more sophisticated and secure the operating system, the greater the richness and complexity of the system metadata in its file table.

Windows Shell Items

In the Windows world, Microsoft calls any single piece of content, such as a file, folder, message or contact, a “Shell item.” Any individual piece of metadata associated with a Shell item is called a “property” of the item. Windows tracks hundreds of distinct metadata properties of Shell items in 34 property categories**.** To see a list of Shell item properties on a Windows system, right click on the column names in any folder view and select “More….” Examining a handful of these properties in just four categories reveals metadata of great potential evidentiary value existing within and without files, messages and photos:

Category	Properties	
Document	ClientID	LastAuthor
	Contributor	RevisionNumber
	DateCreated	Template
	DatePrinted	TotalEditingTime
	DateSaved	Version
	DocumentID	
Message	AttachmentContents	FromAddress
	AttachmentNames	FromName
	BccAddress	HasAttachments
	BccName	IsFwdOrReply
	CcAddress	SenderAddress
	CcName	SenderName
	ConversationID	Store
	ConversationIndex	ToAddress
	DateReceived	ToDoFlags
	DateSent	ToDoTitle
	Flags	ToName
Photo	CameraManufacturer	CameraSerialNumber
	CameraModel	DateTaken
System	ApplicationName	ItemAuthors
	Author	ItemDate
	Comment	ItemFolderNameDisplay
	Company	ItemFolderPathDisplay
	ComputerName	ItemName
	ContainedItems	OriginalFileName

ContentType
DateAccessed
DateAcquired
DateArchived
DateCompleted
DateCreated
DateImported
DateModified
DueDate
EndDate
FileAttributes
FileCount
FileDescription
FileExtension
FileName
IsShared
OwnerSID
Project
Sensitivity
SensitivityText
SharedWith
Size
Status
Subject
Title
FileOwner
FlagStatus
FullText
IsAttachment
IsDeleted
IsEncrypted

Much More Metadata

The hundreds of Windows Shell item properties are by no means an exhaustive list of metadata. Software applications deploy their own complements of metadata geared to supporting features unique to each application. E-mail software, word processing applications and spreadsheet, database, web browser and presentation software collectively employ hundreds of additional fields of metadata.

For example, digital photographs can carry dozens of embedded fields of metadata called **EXIF data** detailing information about the date and time the photo was taken, the camera, settings, exposure, lighting, even precise geolocation data. Photos taken with cell phones having GPS capabilities contain detailed information about where the photo was taken *to a precision of about ten meters*.

Photos taken with cell phones having GPS capabilities contain detailed EXIF information about where the photo was taken.

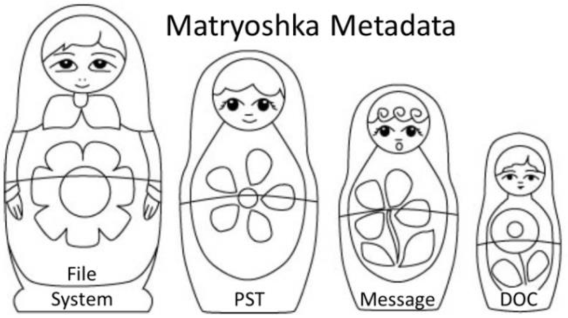
The popular Microsoft Outlook e-mail client application provides for more than 180 standard application metadata fields which users may select to customize their view.

But even this broad swath of metadata is only *part* of the probative information about information recorded by computers. Within the Master File Table and index records used by Windows to track all files, still more attributes are encoded in hexadecimal notation. In fact, an ironic aspect of Windows is that the record used to track information about a file may be larger than the file itself!

Stored within the hives of the System Registry—the “Big Brother” database that tracks attributes covering almost any aspect of the system—are thousands upon thousands of attribute values called “registry keys.” Other records and logs track network activity and journal virtually every action.

Matryoshka Metadata

Matryoshka are carved, cylindrical Russian dolls that nest inside one another. It’s helpful to think of metadata the same way. If the evidence of interest is a Word document attached to an e-mail, the Word document has its usual complement of application metadata that moves with the file; but, as it nests within an e-mail message, its “system” metadata is only that which is contained within the transporting message—likely just the file’s name and filetype. The transporting message, in turn, carries its own metadata concerning transit, addressing, structure, encoding and the like. The message is managed by Outlook, which maintains a rich complement of metadata about the message and about its own configuration. As configured, Outlook may store all messages and application metadata in a container file called Outlook.PST. This container file exists within a file system of a computer that stores system metadata about the container file, such as where the file is stored, under whose user account, when it was last modified, its size, name, associated application and so on.



Within this Matryoshka maelstrom of metadata, some information is readily accessible and comprehensible while other data is so Byzantine and cryptic as to cause even highly skilled computer forensic examiners to scratch their heads.

Forms of Metadata

Now that *your* head is spinning from all the types, purposes and sources of metadata, let’s pile on another consideration: the *form* of the metadata. Metadata aren’t presented the same way from field to field or application to application. For example, some of the standard metadata fields for Outlook e-mail are simply bit flags signifying “true” or “false” for, *e.g.,* “Attachment,” “Do Not Auto Archive,” “Read” or “Receipt Requested.” Some fields reference different *units*, *e.g.*, “Size” references bytes, where “Retrieval Time” references minutes. Several fields even use the *same* value to mean *different* things, e.g., a value of “1” signifies “Completed” for “Flag Status,” but denotes “Normal for “Importance,” “Personal” for “Sensitivity” and “Delivered” for “Tracking Status.”

The form of metadata is a key consideration when deciding how to preserve and produce the information. Not everyone would appreciate a response like, “for this message, item type 0x0029 with value type 0x000b was set to 0x00," when the question posed was whether the sender sought a read receipt. Because some metadata items are simply bit flags or numeric values and make sense only as they trigger an action or indication in the native application, preserving metadata can entail more than just telling opposing counsel, "we will grab it and give it to you." Context is essential.

It's not that locating and interpreting any particular item is hard, but counsel needs to know whether the firm, client or service provider has the tools and employs a methodology that makes it easy. That's why it's crucial to know what metadata is routinely collected and amenable to production before making commitments to opposing counsel or the court. E-discovery service providers should be able to readily identify the system and application metadata values they routinely collect and process for production. Virtually any metadata value can be readily collected and processed—it's just data; but a few items require specialized tools, custom programming or tweaking of customary workflows.

Relevance and Utility

How much of this metadata is relevant and discoverable? Would I be any kind of lawyer if I didn't answer, "It depends?" In truth, it *does* depend upon what issues the data bears upon, its utility and the cost and burden of preservation and review.

Metadata is unlike almost any other evidence in that its utility may flow from its probative value (its relevance as evidence), its utility its ability to support searching, sorting and interpretation of ESI) or both. If the origin, use, distribution, destruction or integrity of electronic evidence is at issue, the relevant "digital DNA" of metadata is essential, probative evidence that needs to be preserved and produced. Likewise, if metadata materially facilitates the searching sorting and management of electronic evidence, it should be preserved and produced for its utility.[18] Put simply, *metadata is an indispensable feature of ESI and should be considered for production in every case. Too, much*

[18] This important duality of metadata is a point sometimes lost by those who read the rules of procedure too literally and ignore the comments to same. Federal Rules of Civil Procedure Rule 26(b) states that, "Parties may obtain discovery regarding any nonprivileged matter that is ***relevant to any party's claim or defense*** and proportional to the needs of the case..." (emphasis added). The Comments to Rules revisions made in 2015 note, "[a] portion of present Rule 26(b)(1) is omitted from the proposed revision. After allowing discovery of any matter relevant to any party's claim or defense, the present rule adds: "including the existence, description, nature, custody, condition, and location of any documents or other tangible things and the identity and location of persons who know of any discoverable matter." Discovery of such matters is so deeply entrenched in practice that it is no longer necessary to clutter the long text of Rule 26 with these examples. The discovery identified in these examples should still be permitted under the revised rule when relevant and proportional to the needs of the case. ***Framing intelligent requests for electronically stored information, for example, may require detailed information about another party's information systems and other information resources***" (emphasis added). Though the Committee could have been clearer in its wording and have helpfully used the term "metadata," the plain import is that relevance "to a party's claims or defenses" is not the sole criterion to be used when determining the scope of discovery as it bears on metadata. Metadata is discoverable for its utility as well as its relevance.

of what is dismissed as "mere metadata" is truly substantive content, such as embedded comments between collaborators in documents, speaker notes in presentations and formulas in spreadsheets.

Does this then mean that every computer system and data device in every case must be forensically imaged and analyzed by experts? Absolutely not! *Once we understand what metadata exists and what it signifies, a continuum of reasonableness will inform our actions*. A police officer making a traffic stop routinely collects relevant "dog tag" data, *e.g.*, driver's name, address, vehicle license number, driver's license number and date, time and location of offense. We wouldn't expect the traffic cop to collect a DNA sample or fingerprint from the driver. But, make it a murder case and the calculus changes.

The crucial factors are burden and cost balanced against utility and relevance. The goal should be a level playing field between the parties in terms of their ability to see and use relevant electronic evidence, including its metadata.

So where do we draw the line? Begin by recognizing that the advent of electronic evidence hasn't changed the fundamental dynamics of discovery: *Litigants are entitled to discover relevant, non-privileged information, and relevance depends on the issues before the court.* Relevance assessments aren't static but change as new evidence emerges and new issues arise. Metadata irrelevant at the start of a case may become decisive when, *e.g.*, allegations of data tampering or spoliation emerge. Parties must periodically re-assess the adequacy of preservation and production of metadata and act to meet changed circumstances.

Periodically re-assess the adequacy of preservation and production of metadata, and act to meet changed circumstances.

Metadata Musts

There are readily accessible, frequently valuable metadata that, like the dog tag information collected by a traffic cop, we should expect to routinely preserve and produce. Examples of essential system metadata fields for any file produced are:

- **Custodian**
- **Source Device**
- **Originating Path** (file path of the file as it resided in its original environment)
- **Filename** (including extension)
- **Last Modified Date**
- **Last Modified Time.**

Any party producing or receiving ESI should be able to state something akin to, "This spreadsheet named *Cash Forecast.xls* came from the Documents folder on Sarah Smith's Dell laptop and was last modified on January 16, 2020 at 2:07 PM CST."

Another metadata "must" that informs time and date information is the UTC time zone offset applicable to each time value (unless all times have been normalized; that is, processed to a common time zone). UTC stands for both for *Temps Universel Coordonné* and Coordinated Universal Time. It's a fraction of a second off the better-known Greenwich Mean Time (GMT) and identical to Zulu time in military and aviation circles. Why UTC instead of TUC or CUT? It's a diplomatic compromise, for neither French nor English speakers were willing to concede the acronym. Because time values may be expressed with reference to local time zones and variable daylight savings time rules, you need to know the UTC offset for each item.

Application metadata is, by definition, embedded within native files; so, native production of ESI obviates the need to selectively preserve or produce application metadata. It's in the native file. But when ESI is converted to other forms, the parties must assess what metadata will be lost or corrupted by conversion and identify, preserve and extract relevant or utile application metadata fields for production in ancillary files called **load files**.

For e-mail messages, this is a straightforward process notwithstanding the dozens of metadata values that may be introduced by e-mail client and server applications. The metadata "musts" for e-mail messages are, as available:

- **Custodian** – Owner of the mail container file or account collected
- **To** – Addressee(s) of the message
- **From** – The e-mail address of the person sending the message
- **CC** – Person(s) copied on the message
- **BCC** – Person(s) blind copied on the message
- **Subject** – Subject line of the message
- **Date Sent (or Received**)– Date the message was sent (or received)
- **Time Sent (or Received)** – Time the message was sent (or received)
- **Attachments** – Name(s) or other unique identifier(s) of attachments/families
- **Mail Folder Path** – Path of the message to its folder in the originating mail account; and,
- **Message ID** –Unique message identifier.[19]

E-mail messages that traverse the Internet contain so-called header data detailing the routing and other information about message transit and delivery. Whether header data should be preserved and produced depends upon the reasonable anticipation that questions concerning authenticity, receipt or timing of messages will arise. The better question might be, "since header data is an

[19] In fact, few of these items are truly "metadata" in that they are integral parts of the message (*i.e.,* user-contributed content); however, message header fields like To, From, CC, BCC and Subject are so universally labeled "metadata," it's easier to accept the confusion than fight it.

integral part of every message, why should any party be permitted to discard this part of the evidence absent cause?"

The metadata essentials must further include metadata values generated by the discovery and production process itself, such as Bates numbers, attachment rangers, hash values, production paths, deduplicate identification, family relationships and the like.

When ESI other than e-mail is converted to non-native forms, it can be enormously difficult to preserve, produce and present relevant or necessary application metadata in ways that don't limit its utility or intelligibility. For example, tracked changes and commentary in Microsoft Office documents may be incomprehensible without seeing them in context, i.e., superimposed on the document. By the same token, furnishing a printout or image of the document with tracked changes and comments revealed can be confusing and deprives a recipient of the ability to see the document as the user ultimately saw it. As well, it often corrupts the extraction of searchable text using optical character recognition. If native forms will not be produced, the most equitable approach may be to produce the document twice: once with tracked changes and comments hidden and once with them revealed.[20]

For certain ESI, there is simply no viable alternative to native production with metadata intact. The classic example is a spreadsheet file. The loss of functionality and the confusion engendered by rows and columns that break and splay across multiple pages mandates native production. A like loss of functionality occurs with sound files (e.g., voice mail), video, animated presentations (i.e., PowerPoint) and databases, web content, SharePoint, social networking sites and collaborative environments where the structure and interrelationship of the information--reflected in its metadata—defines its utility and intelligibility. Forms of production are thoroughly addressed elsewhere but suffice to say that native production greatest strength may derive from its ability to make optimum use of metadata.

The Path to Production of Metadata

The balance of this section discusses steps typically taken in shepherding a metadata production effort, including:

- Gauge spoliation risks before you begin
- Identify potential forms of metadata
- Assess relevance and burden
- Consider authentication and admissibility
- Evaluate need and methods for preservation

[20] But the viability of this "solution" must be weighed against the greatly increased cost to load and host alternate versions of documents considering that vendors typically charge for services by the gigabyte. Two sets of static images substantially inflate the cost of discovery for the parties receiving such a double-whammy production.

- Collect metadata
- Plan for privilege and production review
- Resolve production issues

Gauge spoliation risks before you begin

German scientist Werner Heisenberg thrilled physicists and philosophy majors alike when he posited that the very act of observing alters the reality observed. Heisenberg's Uncertainty Principal speaks to the world of subatomic particles, but it aptly describes a daunting challenge to lawyers dealing with metadata: When you open any document in Office applications without first employing specialized hardware or software, metadata often changes, and prior metadata values may be lost. Altered metadata implicates not only claims of spoliation, but also severely hampers the ability to filter data chronologically. How, then, can a lawyer evaluate documents for production without reading them?

Begin by gauging the risk. Not every case is a crime scene, and few cases implicate issues of computer forensics. Those that do demand extra care be taken to preserve a broad range of metadata evidence. Further, it may be no more difficult or costly to preserve data using forensically sound methods than those that imperil metadata.

For the ordinary case, a working knowledge of the most obvious risks and simple precautions are sufficient to protect the metadata most likely to be needed.

Windows systems typically track at least three date values for files, called "MAC dates" for Last Modified, Last Accessed and Created. Of these, the Last Accessed date is the most fragile, yet least helpful. Historically, last accessed dates could be altered by previewing files and running virus scans. Now, ***last accessed dates are only infrequently updated in Windows*** (after Vista and Win7/8/10).

Similarly unhelpful in e-discovery is the Created date. The created date is often presumed to be the authoring date of a document, but it more accurately reflects the date the file was "created" *within the file system of a particular storage medium*. So, when you copy a file to new media, you've "created" it on the new media as of the date of copying, and the created date changes accordingly. Conversely, when you use an old file as a template to create a new document, the creation date of the template stays with the new document. ***Created dates may or may not coincide with authorship***; so, it's a mistake to assume they do.

The date value of greatest utility and stability is the Last Modified date. The last modified date of a file is not changed by copying, previewing or virus scans. It changes only when a file is opened and saved; however, it is not necessary that the user-facing content of a document be altered for the last accessed date to change. Other changes—including subtle, automatic changes to application metadata--may trigger an update to the last modified date when the file is re-saved by a user.

Apart from corruption, ***application metadata does not change unless a file is opened***. So, the easiest way to preserve a file's application metadata is to keep a pristine, unused copy of the file and access only working copies. By always having a path back to a pristine copy, inadvertent loss or corruption of metadata is harmless error. Calculating and preserving hash values for the pristine copies is a surefire way to demonstrate that *application* metadata hasn't changed.

An approach favored by computer forensic professionals is to employ write blocking hardware or software to intercept all changes to the evidence media. Modern e-discovery review tools are designed not to change metadata values when reviewing files.

Finally, containerized copies[21] can be transferred to read only media (e.g., CD-R or DVD-R), permitting examination without metadata corruption.

Identify potential forms of metadata

To preserve metadata and assess its relevance, you must know it exists. So, for each principal file type subject to discovery, assemble a list of associated metadata of potential evidentiary or functional significance. You'll likely need to work with an expert the first time or two, but once you have a good list, it will serve you in future matters. You'll want to know not only what the metadata fields contain, but also their location and significance.

For unfamiliar or proprietary applications and environments, enlist help identifying metadata from the client's IT personnel. Seek your opponent's input, too. Your job is simpler when an adversary conversant in metadata expressly identifies fields of interest. The parties may not always agree, but at least you'll know what's in dispute.

Assess relevance and burden

Are you going to preserve and produce dozens and dozens of metadata values for every document and e-mail in the case? Perhaps not, although you may find it easier to preserve all than selectively collect and cull just those values you deem relevant. Modern e-discovery tools extract a broad range of metadata fields during processing, so the burden to produce another field is nominal *so long as a specific request for the data is communicated before production*. Claims of "undue burden" tend to be overblown when it's just one more column of information exported to a load file along with file names, Bates numbers and other everyday metadata values.

[21] A containerized copy would typically be a compressed .Zip file. The reason to use containers rather than simply copy the data to optical media is that optical media employs a different file system than other storage media and cannot replicate the system metadata values of files stored on, e.g., an NTFS hard drive. The zip format better replicates a broader range of system metadata values.

Much metadata is like the weather reports from distant cities published in the daily newspaper. Though only occasionally indispensable, we want that information available when we need it.[22]

Relevance is always subjective and as fluid as the issues in the case. Case in point: two seemingly innocuous metadata fields common to Adobe Portable Document Format (PDF) files are "PDF Producer" and "PDF Version." These are listed as "Document Properties" under the "File" menu in any copy of Adobe Acrobat. Because various programs can link to Acrobat to create PDF files, the PDF Producer field stores information concerning the source application, while the PDF Version field tracks what release of Acrobat software was used to create the PDF document. These metadata values may appear esoteric and irrelevant but consider how matters change if the dispute turns on a five-year-old PDF contract claimed to have been recently forged. If the metadata reveals the PDF was created using a scanner introduced to market last year and a six-month-old release of Acrobat, that metadata supports a claim of recent fabrication. In turn, if the metadata reflects use of a very old scanner and an early release of Acrobat, the evidence bolsters the claim that the document was scanned years ago. Neither is conclusive on the issue, but both are relevant evidence needing to be preserved and produced.

Assessing relevance is another area where communication with an opponent is desirable. Often, an opponent will put relevance concerns to rest by responding, "I don't need that." For every opponent who demands "all the metadata," there are plenty who neither know nor care about more than the bare metadata "musts."

Consider Authentication and Admissibility

Absent indicia of authenticity like signatures, handwriting and physical watermarks, how do we establish that electronic evidence is genuine or attributable to one person versus another? Computers and accounts may be shared, hacked or passwords lost or stolen. Software enables convincing alteration of documents absent the telltale signs that expose paper forgeries. Once, we used dates and postmarks to establish temporal relevance, but now documents may acquire new dates each time they're opened, inserted by a word processor macro as a "convenience" to the user.

If the origins and authenticity of evidence may be in issue, preservation of original date and system user metadata is essential. When identifying metadata to preserve or request, consider, *inter alia,* system and network logs and journaling, evidence of other simultaneous user activity and version control data. For more on this, review the material on digital forensics, *supra.*

[22] Of course, we are more likely go online for weather information; but even then, we want the information available when we need it.

In framing a preservation strategy, balance the burden of preservation against the likelihood of a future need for the metadata, but remember, if you act to preserve metadata for documents supporting your case, it's hard to defend a failure to preserve metadata for items bolstering the opposition's case. Failing to preserve metadata could deprive you of the ability to challenge the relevance or authenticity of material you produce.

Chain of Custody

An important role of metadata is establishing a sound chain of custody for ESI. Through every stage of e-discovery--collection, processing, review, and production—metadata should facilitate a clear, verifiable path back to the source ESI, device and custodian.

"Chain of custody" describes the processes used to track and document the acquisition, storage and handling of evidence to be able to demonstrate that the integrity of the evidence has not been compromised. From movies and television, we're familiar with the signed and sealed evidence bags, police property rooms, sign in/out logs and other steps law enforcement agencies use to safeguard physical evidence.   But what are the corollary steps required for digital evidence?

As a rule, counsel should be able to reliably trace any item of digital evidence back to its origin. So, there must be a means to identify the device, network address, container file and custodian of the source of the data. When electronic evidence is collected, or media imaged for preservation, collections and images should be digitally fingerprinted upon acquisition and hash values recorded.

Though digital evidence is unique in that its ability to be duplicated and authenticated without compromising any iteration deemed to be "original,"[23] it remains the practice to restrict access to the data and log any actions with a potential to change the evidence or its hash values.

As we delve into forensic imaging and ESI processing, consider the features of each that support chain of custody and authentication.

Evaluate Need and Methods for Preservation

Not every metadatum is important in every case, so what factors should drive preservation? The case law, rulings of the presiding judge and regulatory obligations are paramount concerns, along with obvious issues of authenticity and relevance. Another aspect to consider is the stability of metadata. As discussed, essential metadata fields, like Last Modified Date, change when a file is used

If you fail to preserve metadata at the earliest opportunity, you may never be able to replicate what was lost.

[23] In the world of digital forensics, the notions of "original" or "best" evidence no longer mean much in that one hash validated copy of ESI is indistinguishable from any other.

and saved. If you don't preserve dynamic data, you lose it. Where a preservation duty has attached, by, *e.g.*, issuance of a preservation order or operation of law, the loss of essential metadata may, at best, require costly remedial measures be undertaken or, at worst, could constitute spoliation subject to sanctions.

How, then, do you avoid spoliation occasioned by review and collection? What methods will preserve the integrity and intelligibility of metadata? Poorly executed collection efforts can corrupt metadata. For example, when a custodian or reviewer opens files in native applications, copies responsive files to new media, prints documents or forwards e-mail, metadata is altered or lost. Consequently, metadata preservation must be addressed before a preservation protocol is implemented and good practices made a part of preservation directives, so-called "legal hold notices." Be certain to document what was done and why. Courts expect a modicum of transparency concerning data preservation, so consider communicating contemplated protocols to opposing counsel in sufficient time to allow adversaries to object, seek court intervention or propose an alternate protocol.

Collect Metadata

Because metadata is stored both within and without files, simply duplicating a file without capturing its system metadata is insufficient. Not all metadata preservation efforts demand complex and costly solutions; you can tailor the method to the case in a proportional way. As feasible, record and preserve system metadata values before use or collection. This can be achieved using software that archives the basic system metadata values to a table, spreadsheet or CSV (comma separated value) file. Then, if an examination results in a corruption of metadata, the original values can be ascertained. Even just archiving files ("zipping" them) may be a sufficient method to preserve associated metadata. Optimally, you'll use tools purpose-built for e-discovery and forensic imaging and/or engage vendors specializing in electronic discovery.

Whatever the method chosen, safeguard the association between the data and metadata. For example, if data is the audio component of a voice mail message, recordings may be of little use unless correlated with the metadata detailing the date and time of the call and the identity of the voice mailbox user. Similarly, email attachments must tie back to transmittals. These efforts termed, "**preserving family relationships**."

When copying file metadata, know the limitations of the environment and medium in which you're working. I learned this lesson the hard way many years ago while experimenting with recordable CDs to hold evidence files and metadata. Each time I tried to store a file and its MAC dates (modified/accessed/created) on a CD, I found that the three *different* MAC dates derived from the hard drive would always emerge as three matching MAC dates when read from the CD. I was corrupting the data I sought to preserve!

I learned that optical media like CD-Rs aren't formatted in the same manner as magnetic media like hard drives. Whereas the operating system formats a hard drive to store three distinct dates, CD-R media stores just one. In a sense, a CD file system has no "place" to store all three dates, so it discards two. When the CD's contents are copied back to magnetic media, the operating system re-populates the "slots" for the three dates with the single date found on the optical media. Thus, using a CD in this manner served to both corrupt and misrepresent the metadata. Similarly, different operating systems and versions of applications maintain different metadata and not all systems support metadata of the same length; so, don't be caught out as I was, *test your processes for alteration, truncation or loss of metadata*.

Plan for Privilege and Production Review

The idea of reviewing metadata for privilege may seem odd unless you consider that application metadata potentially reveals deleted content and commentary. The industry (sub)standard has long been to simply suppress the metadata content of evidence, functionally deleting it from production. This occurred without any entitlement springing from privilege. Producing parties didn't want to review metadata so simply, *incredibly,* purged it from production for their own convenience. *That won't fly anymore. Metadata must be assessed as any other potentially responsive ESI and produced when tied to a responsive and non-privileged information item.*

When the time comes to review metadata for production and privilege, the risks of spoliation faced in collection may re-appear during review. Ponder:

- How will you efficiently access (i.e., "see") metadata?
- Will the metadata exist in a form you can interpret?
- Will your examination alter the metadata?
- How will you tag metadata for production?
- How can you redact privileged or confidential metadata?

Fortunately, modern **e-discovery review platforms** (the software tools lawyers use to search, sort, read and tag electronic evidence) are designed to address these concerns; however, access and alteration remain a peril for lawyers mistakenly trying to use native applications as review tools.

Application Metadata and Review

Many lawyers deal with metadata by pretending it doesn't exist. They employ review methods that don't display application metadata like comments and tracked changes in Microsoft Office files review only what prints instead of all the information embedded in the document. Rather than tailor their workflows to the evidence, they suppress application metadata for fear they'll unwittingly produce privileged or confidential content (or simply from ignorance). They defend their actions claiming that the burden to review application metadata for privileged or confidential content is greater than the evidentiary value of that content. To ensure that requesting parties cannot access all that metadata the producing counsel ignored, producing parties instead strip away all metadata, either by printing the documents to paper or hiring a vendor to convert the ESI to static images (*i.e.,* **TIFF images**). Doing so successfully removes the metadata but impairs the utility, integrity and searchability of the evidence.

Sometimes, counsel producing evidence as static TIFF images undertake to reintroduce some of the stripped metadata and searchable text in ancillary productions called **load files.** The production of document images and load files is a high-cost, low utility, error-prone approach to e-discovery; but its biggest drawback is that it's unable to do justice to native files and metadata. When produced as images, spreadsheets become useless and incomprehensible. Multimedia files disappear. Interactive, animated and structured information breaks. In general, the richer the information in the evidence, the less likely it is to survive production as static TIFF images.

Despite these shortcomings, lawyers cling to cumbersome TIFF productions, driving up e-discovery costs. This is troubling enough, but raises a disturbing question: *Why does any lawyer assume he or she is free to unilaterally suppress--without review or proffer of a privilege log—integral parts of discoverable evidence?* Stripping away or ignoring metadata that's an integral part of evidence seems little different from erasing handwritten notes in medical records because you'd rather not decipher the doctor's handwriting!

In ***Williams v. Sprint/United Mgmt Co.,*** 230 F.R.D. 640 (D. Kan. 2005), concerns about privileged metadata prompted the defendant to strip out metadata from the native-format spreadsheet files it produced in discovery. The court responded by ordering production of all metadata as maintained in the ordinary course of business, save only privileged and expressly protected metadata.

The court was right to recognize that privileged information need not be produced, wisely distinguishing between surgical redaction and blanket excision. One is redaction following examination of content and a reasoned judgment that some matters are privileged. The other excises data haphazardly springing from a speculative concern that the data pared away ***might*** contain privileged information. The baby goes out with the bathwater. Moreover, blanket redaction based on privilege concerns doesn't relieve a party of the obligation to log and disclose such redaction. The defendant in *Williams* not only failed to examine or log items redacted, but Sprint left it to the plaintiff to discover that something was missing.

The upshot is that requesting parties are entitled to the metadata benefits available to the producing party. Producing parties may not vandalize or hobble electronic evidence for production without

The requesting party is entitled to the metadata benefits that are available to the producing party.

adhering to the same rules attendant to redaction of privileged and confidential information in paper documents. We address these issues depth later in these materials.

Resolve Production Issues

Metadata may be produced as a database or housed in a **delimited load file,[24]** in an image format, hosted within an online database or even furnished as paper printouts. However, metadata presents unique production challenges. One is that metadata may be unintelligible outside its native environment absent processing and labeling. How can you tell if an encoded value describes the date of creation, modification or last access without both decoding the value *and* preserving its significance with labels?

Another issue is that metadata isn't always textual. It may consist of a bit flag in an index entry—a one or zero—wholly without meaning unless you know what it signifies. A third challenge to producing metadata lies in finding ways to preserve the relationship between metadata and the data it describes and, when obliged to do so, to present both the data and metadata so as to be electronically searchable.

When files are separated from their metadata, we lose much of the ability to sort, manage and authenticate them. Returning to the voice mail example, unless the sound component of the message (e.g., the WAV file) is paired with its metadata, a reviewer must listen to the message in real time, hoping to identify the voice and deduce the date of the call from the message. It's a Herculean task without metadata, but a task made much simpler if the producing party, *e.g.,* drops the WAV file into an Adobe PDF file as an embedded sound file, then inserts the metadata in the image layer. Now, a reviewer can both listen to the message and search and sort by the metadata.

Sometimes, simply producing a table or load file detailing originating metadata values will suffice. Other times, only native production will suffice to supply relevant metadata in a useful and complete way. Determining the method of metadata production best suited to the case demands planning, guidance from experts and cooperation with the other side.

Beyond Data about Data

The world's inexorable embrace of digital technology serves to escalate the evidentiary and functional value of metadata in e-discovery. Today, virtually all information is born electronically, bound to and defined by its metadata as we are bound to and defined by our DNA. The proliferation

[24] Load files are used to produce searchable text and metadata. *Delimited load files* are composed of delimited text, i.e., values following a predetermined sequence and separated by characters like commas (so-called CSV files), tabs or quotation marks serving as "delimiters" (i.e., separators). We will explore the use and structure of load files later in the semester.

and growing importance of metadata dictates that we move beyond unhelpful definitions like "data about data," toward a fuller appreciation of metadata's many forms and uses.

Key Takeaways re: Metadata

Crucial Distinction: System versus Application Metadata:

File tables hold **system metadata** about the file (*e.g.*, name, locations on disk, MAC dates): it's **CONTEXT** residing **outside** the file

Files hold **application metadata** (*e.g.,* geolocation data in photos, comments in docs): it's **CONTENT embedded in** the file.

System Metadata Examples: File names, file sizes, Modified, Accessed and Created (MAC) dates, file locations (path), custodian.

Application Metadata Examples: Comments, tracked changes, editing times, last printed dates.

System Metadata values must be collected and produced in delimited text files called "Load Files." Application Metadata is embedded in native files, but when files are not produced in native formats, Application Metadata must likewise be extracted and produced in load files.

Parties seeking metadata in discovery should specify the fields of metadata sought.

Parties should preserve the "family relationships" between data and metadata.

Appendix A: Just Ones and Zeros

SECTOR
512 bytes

To give you a sense of what one sector sized snippet of ASCII texts looks like, the binary data above comprises a single hard drive sector storing a binary encoding of the text below (excerpted from *David Copperfield* by Charles Dickens):

I was born with a caul, which was advertised for sale, in the newspapers, at the low price of fifteen guineas. Whether sea-going people were short of money about that time, or were short of faith and preferred cork jackets, I don't know; all I know is, that there was but one solitary bidding, and that was from an attorney connected with the bill-broking business, who offered two pounds in cash, and the balance in sherry, but declined to be guaranteed from drowning on any higher bargain. Consequently the advertisement was withdrawn at a dead loss--for as to sherry, my poor dear mother's own sherry was in the market then--and ten years afterwards, the caul was put up in a raffle down in our part of the country, to fifty members at half-a-crown a head, the winner to spend five shillings. I was present myself, and I remember to have felt quite uncomfortable and confused, at a part of myself being disposed of in that way. The caul was won, I recollect, by an old lady with a hand-basket, who, very reluctantly, pr *[end of sector]*

A 3-terabyte hard drive contains some *5.8 billion* 512-byte sectors.

⚛ Exercise 2: Metadata and Hashing

GOALS: The goals of this exercise are for the student to:

1. Use cryptographic hashing to match duplicates and flag altered files.

2. Assess which actions alter file metadata and hash values

You may wish to review the information about hashing found at pp. 55-57.

Step 1: Download the file http://www.craigball.com/JJJS.zip and extract the three files it contains:
JJJS_DRAFT1.xlsx
JJJS_DRAFT2.xlsx
JJJS_FINAL.xlsx

DO NOT OPEN THESE FILES IN EXCEL or you risk altering the evidence!

Step 2: Collecting System Metadata *Without opening the files*, examine the properties of each file and record the values in the table below.

FILE NAME	LAST MODIFIED DATE AND TIME	FILE SIZE IN BYTES
JJJS_DRAFT1.xlsx		
JJJS_DRAFT2.xlsx		
JJJS_FINAL.xlsx		

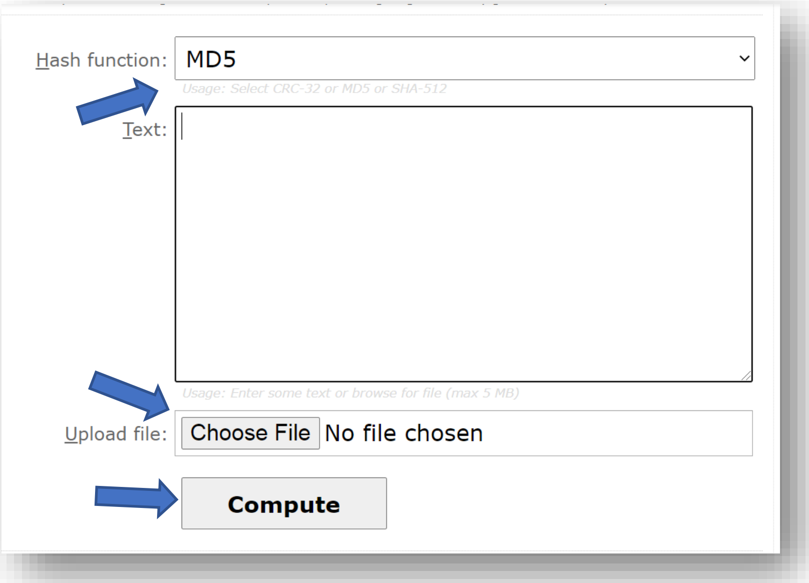
Step 3: Apart from having different file names, are they otherwise "identical" in terms of their system metadata values (date and time values and file size)? (yes or no): ________________

Step 4: Find an online or local (standalone) MD5 hashing (AKA "checksum") tool of your choice and acquaint yourself with its interface. There are dozens of free, online hash calculators that can be found by searching Google for "online MD5 hash calculator." Examples:

http://hash.urih.com/
https://defuse.ca/checksums.htm
http://www.fileformat.info/tool/md5sum.htm
http://hash.online-convert.com/md5-generator
https://www.pelock.com/products/hash-calculator
http://onlinemd5.com

If you use an online hash calculator, be sure to use one that will allow you to browse your machine for a file to hash (all the listed sites do). Should you elect to use a hash calculator that you install as a local application, know that there is no need to purchase software for this purpose as there are many freeware options out there.

Below right is the interface for the online hash calculator found at https://hash.urih.com. Note that all that's required is to ensure that MD5 has been selected as the hash function, then click **Choose File**, navigate to and select the file you wish to hash, then click **Compute** and the 32-digit MD5 hash value of the file you selected should appear.



Hash function: MD5

Usage: Select CRC-32 or MD5 or SHA-512

Text:

Usage: Enter some text or browse for file (max 5 MB)

Upload file: Choose File No file chosen

The online MD5 checksum calculators are the easiest method to use in this Exercise and support both Mac and Windows systems; nonetheless, if you're a Mac user and prefer to calculate MD5 hash values offline, you can follow these steps:

1. Open Terminal.
2. Type md5 and hit the SPACE button.
3. Drag the file you have downloaded into the Terminal Window. ...
4. Hit ENTER. You will see the MD5 Checksum

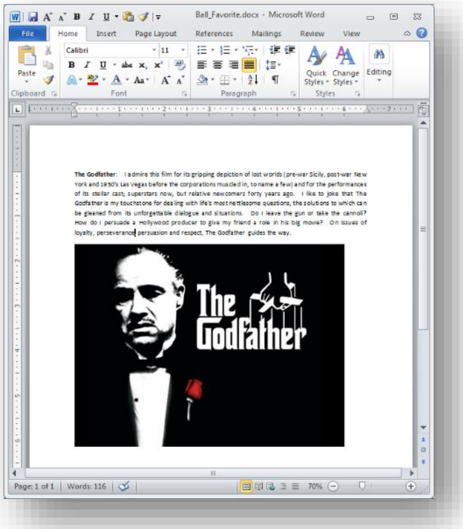
Step 5: Computing Hash Values Hash the named files and record the MD5 hash values for each in the table below.

FILE NAME	MD5 HASH VALUE (first eight characters will suffice here)
JJJS_DRAFT1.xlsx	
JJJS_DRAFT2.xlsx	
JJJS_FINAL.xlsx	

Step 6: Based on their hash values, are any of the three files truly identical in terms of their content? (which ones, if any?): _______________

Step 7: Create an Evidence File

Using the word processing application of your choice, please create a document that identifies by title one of your favorite books or films, followed by one sentence saying why you like it. What you choose to write is of no consequence; you're merely creating a unique file to serve as evidence for the exercise. Feel free to embellish as you wish, e.g., adding a still image from the film or of the cover of the book (as from Amazon.com) to paste into the document; but you can stick with plain text if you prefer. Five minutes on this part, max!



The Godfather: I admire this film for its gripping depiction of lost worlds (pre-war Sicily, post-war New York and 1950's Las Vegas before the corporations muscled in, to name a few) and for the performances of its stellar cast; superstars now, but relative newcomers forty years ago. I like to joke that The Godfather is my touchstone for dealing with life's most nettlesome questions, the solutions to which can be gleaned from its unforgettable dialogue and situations. Do I leave the gun or take the cannoli? How do I persuade a Hollywood producer to give my friend a role in his big movie? On issues of loyalty, perseverance persuasion and respect, The Godfather guides the way.

Name the document with your surname, an underscore and then "Favorite" (i.e., YOURNAME_Favorite.docx). **Save** and **close** the document and your word processor. **This is your original "evidence file" for purposes of this exercise**.

Step 8: Gather Baseline Metadata Values for the Evidence

To begin, establish the "true" or "baseline" system metadata for your original evidence file.

In Windows OS: Using Windows Explorer, determine the following metadata values for your original evidence file:

Filename: ____

Created Date and Time: ______________________________

Modified Date and Time: _____________________________

File size and size on disk: ________________________ _____

OR

In Mac OS: Use Get Info to determine the following metadata values for your original evidence file:

Filename: ____

Created Date and Time: _________________________________

Modified Date and Time: ________________________________

File size and size on disk: ________________________________

Record these values above.

Step 8: Establish Baseline Hash Values

Now, you need to establish the "baseline" hash values for your original evidence file.

Using the local or online hashing tool of your choice, determine the MD-5 hash value for your original evidence file.

Here is the hash value for my original evidence file. Your hash value will be different.

MD5 HASH	25e8b842e0c9383d37107b9ec0758039

Record the value you get for *your* file here for ready reference.

MD5 HASH	

Step 9: Identify Actions that Do and Don't Alter Metadata and Hash Values

Instructions: After completing each task below, determine the metadata and hash values for the resulting file and record them in the spaces provided (first eight characters of the hash will suffice):

a. **E-mail** a copy of your original evidence file to yourself as an attachment. ***Be sure to send the file to yourself using an alternate e-mail address than the transmitting account***.[25] When received, save the e-mailed attachment (*just* the attachment, not the whole message) from your e-mail client to disk (don't overwrite your original)[26] and record its metadata and hash values below:

Created Date and Time: ___________________________________

Modified Date and Time: __________________________________

File size and size on disk: _________________________________

MD5 HASH	

[25] The reason you must use an alternate address is because some e-mail systems won't transmit an attachment across the Internet when the sender and addressee are the same.

[26] In Windows, when you save files of the same name to the same folder, the operating system adds an incrementing number to the name; e.g., YOURNAME_Favorite(1).doc. In Mac, the OS may add the word "copy" to the name. For this exercise, don't be concerned if the file name changes in this way.

b. **Copy** (not Move) your original evidence file to another storage device than your computer (such as a thumb drive, external hard drive or your online storage within Canvas). Determine the metadata and hash values of the copy:

c.

Created Date and Time: ___________________________________

Modified Date and Time: __________________________________

File size and size on disk: _________________________________

MD5 HASH	

d. **Rename** your **<u>original</u>** evidence file using the file system,[27] *but make no other change to the file.* Rename it to something like "YOURNAME_Favorite_2.docx." Determine the metadata and hash values of the <u>renamed</u> document:

Created Date and Time: ___________________________________

Modified Date and Time: __________________________________

File size and size on disk: _________________________________

MD5 HASH	

e. **Edit** your original evidence file *using the application you used to create it* and add a single space somewhere in the document. Save the modified file by a different name (e.g., YOURNAME_Favorite_3.docx). Determine the metadata and hash values of the edited document:

Created Date and Time: ___________________________________

Modified Date and Time: __________________________________

File size and size on disk: _________________________________

MD5 HASH	

[27] To rename a file using the file system, DO NOT open the file in the application you used to create it. *Doing so will likely alter the hash value.* Instead, in Windows OS, right click on the file and select "rename." In MacOS, change the file's name in the "Name and Extension" field of the Get Info screen.

What does this exercise seek to demonstrate? (read but no need to respond)

1. It supplies hands on experience in using a hash algorithm to digitally fingerprint files for reliable comparison, identification and deduplication. Imagine being tasked to compare two printed 100-page documents to determine if they are identical in every respect. You'd need to compare them word-for word and line-by-line (or use a tool that makes the comparison). Note how use of hashing permits ready comparison of just 32-character values no matter the length or complexity of the source document.

2. It underscores the essential difference between system metadata and application metadata. Applications embed application metadata inside the file, so application metadata are part of the calculation when the file is hashed. Accordingly, **a change in <u>application</u> metadata alters the hash value of the file**, impacting the ability to deNIST and deduplicate the file. By contrast, system metadata values are stored outside the file and so are not hashed along with the file. Accordingly, **system metadata can be changed without altering the hash value of the file**. If you can change the name of a file without changing the file's hash value, what does that signify with respect to the location of a file's name record? What might it facilitate in terms of Bates numbering native production formats?

3. The interplay between system and application metadata confuses many students, a circumstance exacerbated by the tendency of software to display a mix of application and system metadata values on a file's properties screen. Some properties MUST move with the file to be of use (e.g., Last Printed Dates in Word documents). Others must *not* move with the file because they would be misleading in other environments (*e.g.*, a physical storage address on disk).

4. Metadata is dynamic and may be lost or changed inadvertently when files are copied or transmitted. How we handle files in e-discovery (our "chain of custody") impacts our ability to authenticate those files as evidence and may prompt charges of spoliation.

⚛ Exercise 3: Metadata: System and Application Metadata

GOALS: The goals of this exercise are for the student to:

1. Distinguish between system metadata and application metadata; and
2. Explore the range and volume of metadata in and for everyday ESI.

OUTLINE: Students will examine various file types to distinguish between metadata stored within files (application metadata) and metadata stored outside the file (system metadata).

Background

Computers may track dozens or hundreds of metadata values for each file, but the quantity and integrity of metadata values retained for any file hinge on factors peculiar to the file's type and history. Moreover, though metadata may be stored both within and without a file, every active file will have *some* complement of *system* metadata that's not contained within the file. Certain image file formats contain metadata tags called **EXIF data** (for Exchangeable Image File Format) that hold a wealth of data.

Step 1: Download the Zip file at www.craigball.com/filetypes.zip and extract its contents to a folder on your desktop or any other convenient location on your computer.[28] The extracted contents will comprise eleven folders (named BMP, DOC, DOCX, DWG, GIF, JPG, PDF, TXT, WAV, XLSX and XLS), each containing samples of file types commonly processed in e-discovery. Open the folder called JPG. You should see 12 files. Find the file called **MardiGras.jpg** and open it.

Step 2: View File Properties

On a Windows PC: Right click on the file and select "Properties." Open the "Details" tab.

On a Mac: In Preview, go to the "Tools" menu and select "Show Inspector." A box will open displaying the file's General Info properties. Note the four tabs at the top of this box. Click on the More Info tab (an "*i*" in a black circle), and note that four different tabs appear called General, Exif, IPTC and TIFF. Click on each of the four to see the range of data available.

Online Tool: If you prefer to use an online tool, Steps 2 and 3 of this exercise can be completed using the metadata features of, *e.g.*, http://fotoforensics.com or https://www.metadata2go.com

[28] I'm assuming all students have previously extracted and saved the contents of a Zip-compressed container. If you haven't, no worries. All Mac and Windows operating systems have the capability to view and extract the contents of Zip files or you may prefer to download and install a free Zip utility like 7-Zip for Windows at https://www.7-zip.org or the Unarchiver for Mac at https://theunarchiver.com/

Step 3: Collect Metadata

Determine the following for the photo called MardiGras.jpg:

1. Camera maker: ____________________

2. Camera model: ____________________

3. Date taken: _______________________

4. Flash mode: _______________________

5. Has the image been PhotoShopped? ________________

Step 4: Different Roles, Different Metadata

Locate the file "TwitterArticle.doc" in the DOC folder. In Windows, right click on it and select "Properties." In Mac, use Get Info.

Determine the following for the document named "TwitterArticle.doc" using the metadata displayed in the Properties box (Windows) or the Get Info box (Mac): NOTE: Some of the following metadata may not be accessible using a Mac OS.

1. Author: ____________________________________

2. Company: __________________________________

3. Date Created: _______________________________

4. Last Printed: ________________________________

5. Title: ______________________________________

6. Total editing time: ___________________________

7. Which, if any, of these values are **system metadata**? ______________________________

8. Can you alter any of the metadata values from the Properties/Get Info window? ________

❁ Exercise 4: Metadata: Geolocation in EXIF

GOALS: The goals of this exercise are for the student to:

1. Grasp the fundamentals of global positioning
2. Explore the range and volume of metadata in and for everyday ESI; and
3. Identify and use geolocation information in EXIF data to track stolen funds.

OUTLINE: Students will examine various image files to extract embedded EXIF metadata and assess its value as evidence, then use that information and online research to follow the ill-gotten gains.

Background

Some cameras and all mobile phones sold in the United States are GPS-enabled. For phones, the latter capability is legally mandated to support 911 emergency services. In fact, modern cell phones facilitate geolocation in at least[29] three distinct ways, by:

1. Communicating with cell towers
2. Using internal GPS capabilities; and
3. Interacting with WiFi hotspots.

Consider how many issues in civil and criminal litigation might be resolved by the availability of detailed and reliable geolocation evidence? But right now, how reliable *is* such data?

Global Positioning System (GPS)

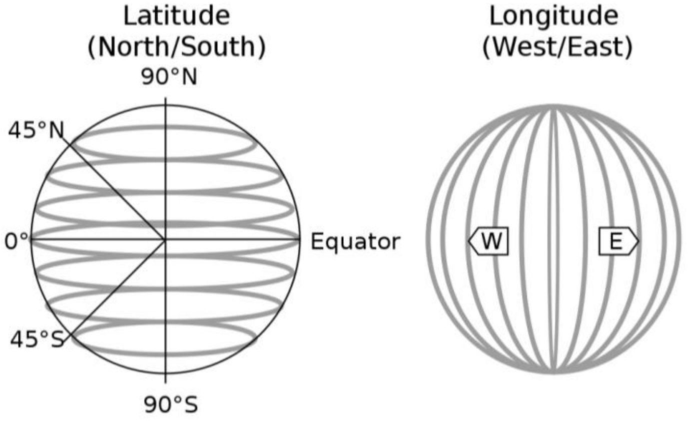
Few who use GPS have more than a vague idea that GPS-enabled devices establish their location by talking to a network of satellites. In fact, your phone doesn't talk but listens for signals from a minimum of four orbiting transmitters and atomic clocks, part of a constellation of about 31 U.S. GPS satellites covering the Earth twice a day. Each satellite continuously sends a digital signal communicating the precise time the signal was dispatched and GPS receivers pinpoint their location by computing the time difference between when the signal was sent and when it was received. Because the signal's speed is constant (the speed of light) and the position of each satellite is known, the time difference can be converted to distance and, in turn, the GPS receiver plots the distance from these four known points to their single point of terrestrial convergence, reflecting the receiver's location and elevation. Phones use **Assisted GPS** (A-GPS) to speed getting a fix on location. Rather than wait to receive an almanac of satellite positions from on high—a process that

[29] Phones also possess Bluetooth capabilities (*e.g.*, Apple's iBeacon), though the relatively short range of the common Class 2 Bluetooth radio limits its capacity for geolocation.

can take minutes—Assisted GPS gives phones a headstart by speedily downloading that almanac information via Wi-Fi or cellular data.

Geographic Coordinate System: Latitude and Longitude

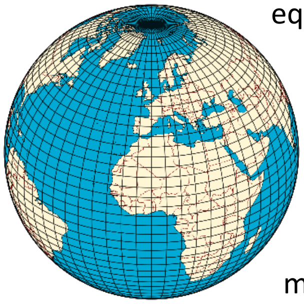
Global Positioning will tell you where you are, but to be truly useful, geolocation data must translate to a Geographic Coordinate System that ties where you are in relation to where everything else is, too. GPS systems employ the **World Geodetic System (WGS)** and its latest reference coordinate iteration, called **WGS 84**.



Latitude varies from 0° at the equator to 90° North and South at the poles

Longitude varies from 0° at Greenwich to 180° East and West

The WGS sections the planet vertically into **meridians of longitude**, drawn from pole-to-pole and divides the eastern and western hemispheres at the **Prime Meridian** at the Royal Observatory in Greenwich, England. Each hemisphere contains 180°. The globe further divides into horizontal **parallels of latitude**, concentric circles starting at the Equator and proceeding 90° north and south to the poles. Latitude is the angular difference between a line drawn from the center of the earth to the point sought to be expressed equator and a line drawn from the center of the Earth to the Equator, measured sexagesimally (in sixtieths) as degrees, minutes and seconds (or as an equivalent value of degrees and decimal minutes). Together, the parallels of latitude and meridians of longitude form a cage like grid called a **graticule**.



REMEMBER: The divisions from pole-to-pole are **meridians of longitude;** the concentric circumferential divisions are **parallels of latitude**. It helps to note that the Prime **Meridian** is a *north-south* demarcation and that parallel lines never touch, so **parallels** of latitude are belts *around* the Earth like the zero-degree meridian called" the Equator."

The Mission:

You've been hired to recover assets stolen in a massive Ponzi scheme. The culprit is a globetrotting gastronome named Bernie who's stashed stolen cash and ill-gotten diamonds all over the world. Each time Bernie opened a new account or safety deposit box, he stopped nearby for a celebratory meal *or drink and texted an iPhone snapshot to his wife in Palm Beach. Your client obtained copies of eleven of these cellphone photos through discovery, all in their native .jpg format. Your first task is to figure out where in the world the cash might be and when Bernie visited.*

Step 1: EXIF Geolocation Data

Download the Zip file at www.craigball.com/exif.zip and extract its contents to a folder on your desktop or any other convenient location. Locate ten numbered files called **yum1.jpg** through **yum10.jpg** and two others named **key.jpg** and **huh.jpg**.

Find the file called **yum1.jpg** and explore its properties:

In Windows: Right click on the file, select Properties>Details. Note the **Date Taken** (under "Origin"). This is the date and time the photo was taken (perhaps adjusted to your machine's local time zone and DST setting). Also, note the GPS coordinates for Latitude, Longitude and Altitude.

In MacOS: Open the file in Preview, go to the "Tools" menu and select "Show Inspector." A box will open displaying the file's General Info properties. Note the four tabs at the top of this box. Click on the More Info tab (an "*i*" in a black circle), then click on the GPS tab. Note the **Date Stamp** and **Time Stamp.** These are the date and time the photo was taken. Also, note the GPS coordinates for Altitude, Latitude and Longitude. Your Mac may even display a handy world map!

Step 2: Record the GPS coordinates and Date Taken

For the file **yum1.jpg**, locate the GPS Latitude and Longitude values embedded within the photograph's complement of EXIF data and the date the photo was taken.

You *should* see the following:

In Windows: Latitude: *33; 35; 47.9999999* **Longitude:** *7; 40; 25.2000000* **Date :** *7/2/2011 4:24PM*

In MacOS: Latitude: *33° 35' 49.2 N* **Longitude:** *7° 40' 26.4 W* **Date :** *7/2/2011 8:24PM*

The Windows time may vary.

Step 3: Where's Waldo, I mean Bernie?

Now let's determine exactly where this photo was taken. In Google, carefully enter the latitude and longitude values embedded within the photo as described below:

If the values you found were formatted as: Latitude AA; BB; CC.CCCCCC and Longitude XX; YY; ZZ.ZZZZZ, enter them in the Google search box in the following format:

AA BB**'** CC.CCCCCC**, -**XX YY**'** ZZ.ZZZZZ ***(note that minus sign!)***

So, for yum1.jpg: the Google search is**: 33 35' 47.9999999, -7 40' 25.200000** (Windows) or, using the Mac data: **33° 35’ 49.2 N, 7° 40’ 26.4 W.** Either way, all roads lead to Rome. I mean, Casablanca.

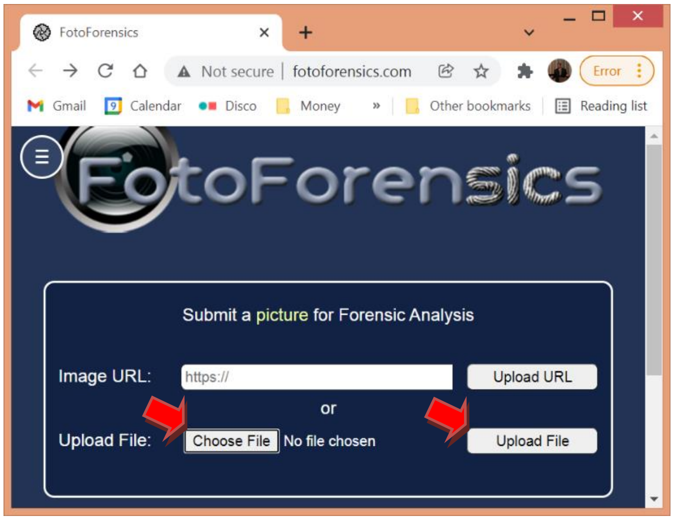
That’s right, if the map you retrieved points to a gin joint on Boulevard de la Corniche in Casablanca, this looks like the beginning of a beautiful friendship. If not, check your formatting and try again.

Be sure you include the apostrophe after the second longitude and latitude values, the comma separating the values and (for the Windows values only) the minus sign preceding the longitude value. By way of example, the coordinates for The University of Texas School of Law in a photo might appear as **Latitude 30° 17' 18.696" N, Longitude 97° 43' 49.846" W**. In Google, they must be entered as: **30 17' 18.696, -97 43' 49.846**.

Step 4: A Quicker Way

Since we must get through many of these, let’s find a quicker way. If you’re using a Mac, look for the “Locate” button at the bottom of the GPS menu where you found the coordinates. Clicking on this button for each will launch a map in your browser (if you have an Internet connection).

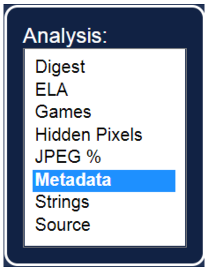
Windows users may be able to double click each photo and launch the image in the Windows photo viewer by default. If so, look for the three dots in the upper right corner of the app and click them, then select “File info” from the drop-down menu. The date, location and other information about the photo should appear in the File info panel on the left side of the screen. If that doesn’t work, you can find many online EXIF viewers by searching for same on Google, or you can use Jeffrey’s EXIF Viewer at http://exif.regex.info or FotoForensics at http://fotoforensics.com/ In FotoForensics, upload each file by clicking Choose File, select each photo listed in the following table from the location where you stored the photos on your machine, then select Upload File. Once you see the uploaded file, select “Metadata” from the Analysis menu.



Submit a picture for Forensic Analysis

Image URL: https://

Upload File: Choose File No file chosen



The page that appears will supply extensive EXIF data. Scrolling down, you’ll see a map image showing the approximate GPS location. **Determine where and when Bernie took the pictures, then add that information to the following table for each.**

Photo	**Location Taken**	**Date taken**
Yum1.jpg	Casablanca, Morocco	July 2, 2011
yum2.jpg*		
yum3.jpg		
yum4.jpg		
yum5.jpg		
yum6.jpg		
yum7.jpg		
yum8.jpg		
yum9.jpg		
yum10.jpg		
Key.jpg		

**Note: Yum2.jpg can be a bit tricky if you are manually searching the coordinates. Be sure to put a minus sign in front of the latitude for this one, but don't use a minus sign in front of the longitude, i.e. -35.745047, 150.213486. That is, be sure you are using south latitudes.*

Step 5: Follow the Money, Part 1

Sizable sums went abroad late in July or early in August during both 2015 and 2018. You've learned that Bernie boasted of splitting the ill-gotten gains: putting half into foreign banks and half in a safety deposit box in the form of loose diamonds purchased in the Netherlands. ***Key.jpg*** *shows Bernie holding a safety deposit box key outside a branch of Deutsche Bank. Using the EXIF data from the photos and online mapping resources, locate the most likely branch of Deutsche Bank to hold the cash and diamonds that went abroad late in July or early in August 2015 and 2018*

City	**Deutsche Bank Branch Street Address**

Step 6: Follow the Money, Part 2

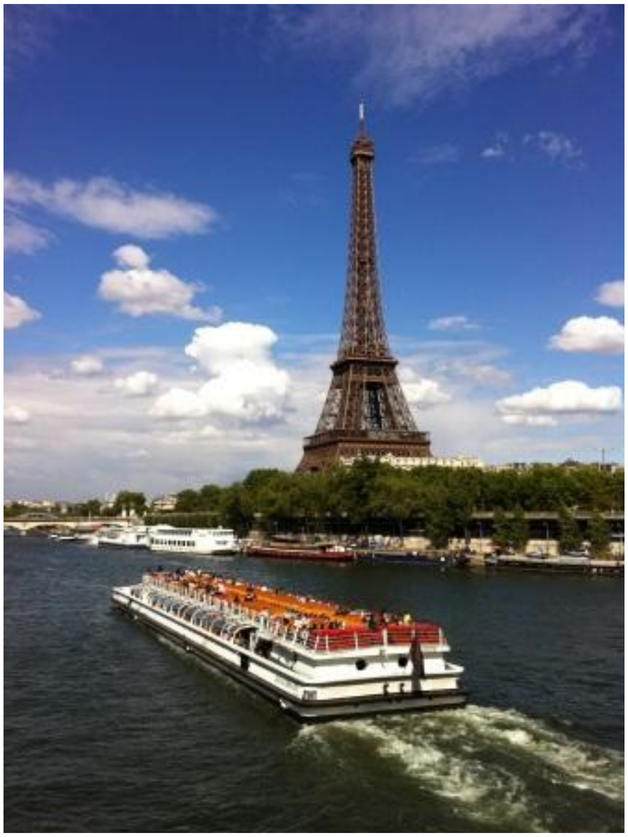
You suspect that Bernie carried funds and diamonds to three other branches of Deutsche Bank, once during late July 2016, a second time when he visited a fancy German spa town in August of 2018 and finally shortly after Christmas in 2019. Identify the cities and the three branches of Deutsche Bank he most likely visited in each instance. To answer this question, you must apply the date and geolocation data information gleaned from the photos to identify the correct cities then apply online resources to identify the (then) nearest Deutsche Bank outpost for each location.

Date	City	Deutsche Bank Branch Address
July 2016		
August 2018		
December 2019		

Step 7: How Trustworthy is EXIF data?

Let's do one more.

From the location where you stored the photos, open the one called **huh.jpg**. Ah, the Eiffel Tower; the Bateau-Mouche; one can almost hear Edith Piaf singing, *non*? Can this be anywhere but the City of Lights and Love? **Check the GPS data and map it.**



According to its EXIF geolocation data, where and when was this photo taken? How?

____

____

____

____

____

Introduction to Digital Forensics

Digital Forensics melds the fast-changing complexity and variety of digital devices and data with human behavior and motivation. The computer forensic examiner finds the human drama manifested as digital needles in staggeringly large data haystacks or in just a byte or two denoting actions like reading, deleting, tagging or altering a file or record. It's exacting work, and competence in the discipline demands examiners move at the breakneck pace of technology with the plodding precision of law.

VITAL VOCABULARY
Encoded Data
Unallocated Clusters
Slack Space
Formatting
Partitioning
Forensic Examiner
Examination Protocol
Windows Registry
File Carving
Binary Signature

What is Computer Forensics?

A computer's operating system or **OS** (e.g., Windows, Mac or Linux) and its installed software or **applications** generate and store more information than users know. Only some of this unseen information is **active data** accessible to users, though sometimes requiring skilled interpretation to be of value in illuminating human behavior. Examples include the data *about* data or **metadata** tracked by the operating system and applications. For example, Windows Explorer supports the display of geolocation data embedded within photos, but few users customize their folder views to see it.

Other active data reside in obscure locations or as encoded formats neither readily accessible nor comprehensible to end users but enlightening when interpreted and correlated by forensic experts. Log files, system files and information recorded in non-text formats are examples of **encoded data** that may reveal information about user behavior. To illustrate, in a data theft investigation, crucial evidence may reside within the Windows system Registry as time-stamped entries logging when USB storage devices were connected.

Finally, there are vast regions of hard drives and thumb drives that hold **forensic artifacts** in regions and formats operating systems don't access. These digital boneyards, called **unallocated clusters** and **slack space**, contain much of what a user, application or OS discards over the life of a machine. Accessing and making sense of this disjointed detritus demands specialized tools, techniques and skills.

Computer forensics is the expert acquisition, interpretation and presentation of the data within these three categories (**Active**, **Encoded** and **Forensic** data), along with its juxtaposition against other available information (*e.g.*, credit card transactions, keycard access data, phone records, surveillance video, fitness monitoring, social networking, voice mail, e-mail, documents and text messaging).

In litigation, computer forensics isn't limited to personal computers and servers, but may extend to all manner of devices harboring electronically stored information. Certainly, phones, tablets, external hard drives, thumb drives and memory cards are routinely examined, and increasingly, relevant information resides on IoT devices, Cloud repositories and even automobile navigation systems and air bag deployment modules. The scope of computer forensics—like the scope of a crime scene investigation—expands to mirror the available evidence and issues before the court.

How Does Computer Forensics Differ from Electronic Discovery?

Electronic discovery addresses the ESI accessible to litigants; computer forensics addresses the ESI accessible to forensic experts. However, the lines blur because e-discovery often requires litigants to grapple with forms of ESI deemed not reasonably accessible due to burden or cost, and computer forensic analysis often turns on information readily accessible to litigants, such as file modification dates.

The principal differentiators are **expertise** (computer forensics requires a unique skill set), **issues** (most cases can be resolved without resorting to computer forensics, though some will hinge on matters that can only be resolved by forensic analysis) and **proportionality** (computer forensics injects contentious issues of expense, delay and intrusion). Additionally, electronic discovery tends to address evidence as discrete information items (documents, messages, databases), while computer forensics takes a more systemic or holistic view of ESI, studying information items as they relate to one another and in terms of what they reveal about what a user did or tried to do. Lastly, electronic discovery deals almost exclusively with existing ESI; computer forensics frequently focuses on what's gone, how and why it's gone and how it might be restored.

When to Turn to Computer Forensics

Most cases require no forensic-level computer examination, so parties and courts should closely probe whether a request for access to an opponent's machines is grounded on a genuine need or is simply a fishing expedition. When the question is close, courts can balance need and burden by using a neutral examiner and a protective protocol, as well as by assessing the cost of the examination against the party seeking same until the evidence supports reallocation of that cost.

Certain disputes demand forensic analysis of relevant systems and media, and in these cases, the parties and/or the court should act swiftly to support appropriate efforts to preserve relevant evidence. For example, claims of data theft may emerge when a key employee leaves to join or become a competitor, prompting a need to forensically examine the departing employee's current and former business machines, portable storage devices and home machines. Such examinations inquire into the fact and method of data theft and the extent to which the stolen data has been used, shared or disseminated.

Cases involving credible allegations of destruction, alteration or forgery of ESI also justify forensic analysis, as do matters alleging system intrusion or misuse, such as instances of employment discrimination or sexual harassment involving the use of electronic communications. Electronic devices figure prominently in most crimes and domestic relations matters, too. It's the rare fraud or dalliance that doesn't leave behind a trail of electronic breadcrumbs in e-mail, messaging, online financial histories and mobile devices.

What Can Computer Forensics Do?

Though the extent and reliability of information gleaned from a forensic examination varies, here are some examples of the information an analysis may uncover:

1. Manner and extent of a user's theft of proprietary data
2. Timing and extent of file deletion or antiforensic (*e.g.,* data wiping) activity
3. Whether and when a thumb drive or external hard drive was connected to a machine
4. Forgery or alteration of documents
5. Recoverable deleted ESI, file structures and associated metadata
6. Internet usage, online activity, Cloud storage access and e-commerce transactions
7. Breach, intrusion and unauthorized access to servers and networks
8. Social networking
9. Clock and calendar tampering
10. Photo manipulation; and
11. Minute-by-minute system usage.

What Can't Computer Forensics Do?

Notwithstanding urban legend and dramatic license, there are limits on what a computer forensic examination can accomplish. To illustrate, an examiner generally cannot:

1. Recover information that has been completely overwritten—even once—by new data
2. Conclusively identify the hands on the keyboard if one person logs in as another
3. Conduct a thorough forensic examination without access to the evidence media or a forensically sound image of same
4. Recover data from a drive that has suffered severe physical damage
5. Guarantee that a drive won't fail during the acquisition process; or
6. Sit down at a suspect's computer, at a crime scene and in seconds, hack in and find the smoking gun.

NTFS

By way of review, the NTFS file system underlies Windows NT, 2000, XP, Vista and Windows 7-11. NTFS uses a powerful and complex database called the **Master File Table** or **MFT** to manage file storage. One unique aspect of NTFS is that, if a file is small enough in size (less than about 1,500 bytes), NTFS stores the file in the MFT to increase performance. Rather than moving the read/write heads to the beginning of the disk to read the MFT entry, and then elsewhere to read the actual file, the heads simply read both at the same time. This can account for a considerable increase in

speed when reading lots of small files.[30] It also means that forensic examiners need to carefully analyze the contents of the Master File Table for revealing information. Lists of account numbers, passwords, e-mails and smoking gun memos tend to be small files.

To illustrate this critical difference, if NTFS were an old card catalog at the library, it would have all books small enough to fit tucked right into the card drawer.

Formatting and Partitioning

There is a fair amount of confusion concerning formatting and partitioning of hard drives. Some of this confusion grows out of the way certain things were done in "the old days" of computing, i.e., thirty years ago. Take something called "low level formatting." Once upon a time, a computer user adding a new hard drive had to low-level format, partition, and then high-level format the drive. Low level formatting was the initial "carving out" of the tracks and sectors on a pristine drive. Back when hard drives were small, their data density modest and their platter geometries simple, low-level formatting by a user was possible. Today, low level formatting is done at the factory and no user ever low-level formats a modern drive. You couldn't do it if you tried; yet, you still hear veteran PC users talk about it.

Your new hard drive comes with its low-level formatting set in stone. You need only be concerned about the disk's partitioning into **volumes,** which users customarily see as drive letters (e.g., C:, E:, F: and so on) and its high level formatting, which defines the logical structures on the partition and the location of any necessary operating system files. For most users, their computer comes with their hard drive partitioned as a single volume (universally called C:) and already high level formatted. Some users will find (or will cause) their hard drive to be partitioned into multiple volumes, each appearing to the user as if it were an independent disk drive.

Cluster Size and Slack Space

By way of further review, a computer's hard drive records data in bits, bytes and sectors, all physical units of storage.

A common paper filing system uses labeled manila folders assembled into a master file for a case, client or matter. A computer's file system stores information on the hard drive in batches of sectors called clusters. Clusters are the computer's manila folders and, like their real-world counterparts,

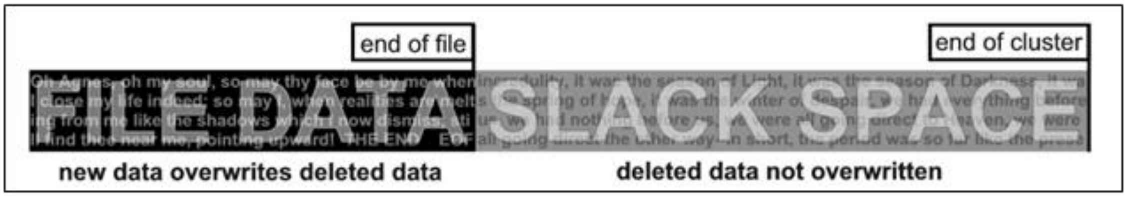
[30] I like to think of this as the difference between moving atoms versus electrons. Electric currents propagate through electrons at about 90% the speed of light or 270,000 km/s. Matter—"stuff" —is made up of atoms, and moving stuff takes time. Compared to the near lightspeed of electrical signals in a circuit, the movement of physical objects like the platters or read-write heads of a hard drive is glacially slow. If you consider that the outside edge of a hard drive's platter travels at about one mile per second (faster than the eye can see), you can begin to appreciate the glacial difference by noting that electrical signals travel *600 million* times faster.

collectively form files. These files are the same ones that you create when you type a document or build a spreadsheet.

Cluster size is set by the file system when it is installed on the hard drive and ranges from 1 sector to 128 sectors (64KB). Until the advent of multi-terabyte hard drives, Windows clusters have long been eight sectors in size (8 x 512 bytes = 4,096 bytes aka 4 kB). Remember that a cluster (also called an **allocation unit** or **block**) is *the smallest unit of data storage in a file system*. You might be wondering, "what about bits, bytes and sectors, aren't they smaller?" Certainly, but in setting cluster size, the file system strikes a balance between storage efficiency and operating efficiency. Again, the smaller the cluster, the more efficient the use of hard drive space; the larger the cluster, the easier it is to catalog and retrieve data.

Suppose you used 500-page notebooks to store documents. If you have just 10 pages to store, you must dedicate an entire notebook to the task. Once in use, you can add another 490 pages, until the notebook won't hold another sheet. For the 501st page and beyond, you must use a second notebook. The difference between the maximum capacity of the notebook and its contents is its "slack" space. Smaller capacity notebooks would mean less slack, but you'd have to keep track of many more notebooks.

In the physical realm, where the slack in the notebook holds empty air, slack space is merely inefficient. But on an electromagnetic hard drive,[31] where magnetic data isn't erased until it's overwritten by new data, the slack space is far from empty. When Windows stores a file, it fills as many clusters as needed. Because a cluster is the smallest unit of storage, the amount of space a file occupies on a disk is "rounded up" to an integer multiple of the cluster size. If the file being stored requires a single byte more than the clusters allocated to it can hold, another cluster will be allocated, and the single byte will occupy an entire cluster on the disc. The file can then grow without requiring further space allocation until it reaches the maximum size of the final cluster, at which point the file system will allocate another full cluster for its use. For example, if a file system employs 4-kilobyte clusters, a file that is 96 kilobytes in size will fit perfectly into 24 clusters, but if that file were 97 kilobytes, then it would occupy 25 clusters, with 3 kilobytes idle. Except in the rare instance of a perfect fit, a portion of the final storage cluster will always be left unfilled with new data. This "wasted" space between the end of the file and the end of the last cluster is slack space (also variously called "file slack" or "drive slack") and it can significantly impact available storage (See figure below).


end of file
end of cluster
FILE DATA
SLACK SPACE
new data overwrites deleted data
deleted data not overwritten

[31] The explanation supplied here applies only to conventional electromagnetic or "spinning" hard drives. Solid state drives employ a radically different storage mechanism that tends not to retain deleted data in unallocated clusters due to data maintenance routines termed "wear leveling" and TRIM.

When Windows deletes a file, it simply earmarks clusters as available for re-use. When deleted clusters are recycled, they retain their contents until and unless the entire cluster is overwritten by new data. If later written data occupies less space than the deleted data, some of the deleted data remains, as illustrated in the previous figure. It's as if in our notebook analogy, when you reused notebooks, you could only remove an old page when you replaced it with a new one.

Though it might seem that slack space should be insignificant—after all, it's just the leftover space at the end of a file—the reality is that slack space adds up. If file sizes were truly random then, on average, one half of a cluster would be slack space for every file stored. But most files are pretty small--if you don't believe it, look at your web browser's temporary Internet storage space. The greater the instance of small files, the greater the volume of slack space on your drive. It wasn't unusual for 10-25% of a drive to be lost to slack. Over time, as a computer is used and files deleted, clusters containing deleted data are re-used and file slack increasingly includes fragments of deleted files.

A simple experiment you can do to better understand clusters and slack space is to open Windows Notepad (usually in the Programs>Accessories directory). Type the word "hello" and save the file to your desktop as "hello.txt." Now, find the file you've just created, right click on it and select "properties." Your file should have a size of just 5 bytes (for the five letters in hello"), but the size it occupies on disk will be much larger, ranging from as little as 4,096 bytes to as much as 32,768 bytes.[32] Now, open the file and change "hello" to "hello there," then save the file. Now, when you look at the file's properties, it has more than doubled in size to 11 bytes (the space between the words requires a byte too), but the storage space occupied on disk is unchanged because you haven't gone beyond the size of a single cluster

Forensic Implications of Slack Space

In "Jurassic Park," scientists cloned genetic material harvested from petrified mosquitoes to bring back the dinosaurs. Like insects in amber, computers trap deleted data and computer forensics resurrects it. Though a computer rich with data trapped in file slack can yield a mother lode of revealing information, mining this digital gold entails tedious digging, specialized tools and lots of good fortune and patience.

Operating systems are blind to all information in slack space. Searching is accomplished using a forensically-sound copy of the drive and specialized examination software. File slack is, by its very nature, fragmented, and the information identifying file type of slack contents is the first data overwritten.

The search for plain text information is typically the most fruitful avenue in file slack examination. Experienced computer forensic examiners are skilled in formulating search strategies likely to turn

[32] If you see that the size on disk is zero, then Windows is correctly reporting that the small file is being stored within the Master File Table. You may not see Windows move the data out of the MFT until you reach about 550 characters in the file.

up revealing data, but the process is greatly aided if the examiner has a sense of what he or she is seeking before the search begins. Are there names, key words or parts of words likely to be found within a smoking gun document? If the issue is trade secrets, are there search terms uniquely associated with the proprietary data? If the focus is child pornography, is there image data or Web site address information uniquely associated with prohibited content?

Because most lawyers and litigants are unaware of its existence, file slack and its potential for disgorging revealing information is usually overlooked by those seeking and responding to discovery. In fairness, a request for production demanding "the contents of your computer's slack space" is absurd. In practice, the hard drive must be examined by a computer forensics expert employed by one of the parties, a neutral expert agreed upon by all parties or a special master selected by the court.

Bear in mind that while the computer is running, computer data is constantly being overwritten by new data, creating a potential for spoliation when forensic artifacts are recognized as important to the case. The most prudent course is to secure, either by agreement or court order, forensically sound duplicates (*i.e.,* **forensic images**) of potentially relevant hard drives. Such specially created copies preserve both the live data and the information trapped in the slack space and other locales. Most importantly, they preserve the status-quo and afford litigants the ability to address issues of discoverability, confidentiality and privilege without fear that delay will result in destruction of data. There's more on this topic (and a forensic imaging exercise) to follow.

Balancing Need, Privilege and Privacy

A computer forensic examiner sees it all. The Internet has so eroded barriers between business and personal communications that workplace computers are routinely peppered with personal, privileged and confidential communications, even intimate and sexual content, and personal devices routinely hold business data. Further, a hard drive is more like one's office than a file drawer. It may hold data about the full range of a user's daily activity, including private or confidential information and teem with trade secrets, customer data, email flirtations, salary schedules, Internet searches for pornography and escort services, bank account numbers, online shopping, medical records and passwords.

So how does the justice system afford access to discoverable information without inviting abuse or exploitation of the rest? With so much at stake, parties and the courts must approach forensic examination cautiously. Access should hinge on demonstrated need and a showing of relevance, balanced against burden, cost or harm. Direct access to storage media should be afforded an opponent only when, *e.g.*, it's been demonstrated that an opponent is untrustworthy, incapable of preserving and producing responsive information or that the party seeking access has some proprietary right with respect to the drive or its contents. Showing that a party lost or destroyed ESI is a common basis for access, as are situations like sexual harassment or data theft where the computer was instrumental to the alleged misconduct.

In Texas, for example, the process attendant to seeking forensic examination is described by the Texas Supreme Court in *In re: Weekley Homes, L.P.*, 295 S.W.3d 309 (Tex. 2009), a dispute concerning a litigant's right to directly access an opponent's storage media. The plaintiff wanted to run 21 search terms against the hard drives of four of defendant's employees to find deleted e-mails. The standards that emerged serve as a sensible guide to those seeking to compel an opponent to recover and produce deleted email, to wit:

1. Parties seeking production of deleted emails should specifically request them and specify a form of production

2. Responding parties must produce reasonably available information in the format sought. They must object if the information is not reasonably available or if they oppose the requested format

3. Parties should try to resolve disputes without court intervention; but if they can't work it out, either side may seek a hearing at which the responding party bears the burden to prove that the information sought is not reasonably available because of undue burden or cost

4. If the trial court determines the requested information is not reasonably available, the court may still order production if the requesting party demonstrates that it's feasible to recover deleted, relevant materials and the benefits of production outweigh the burden, *i.e.,* the responding party's production is inadequate absent recovery

5. Direct access to another party's storage devices is discouraged; but if ordered, only a qualified expert should be afforded such access, subject to a reasonable search and production protocol protecting sensitive information and minimizing undue intrusion; and

6. The requesting party pays the reasonable expenses of any extraordinary steps required to retrieve and produce the information.

In *Weekley Homes,* the Supreme Court further articulated a new duty:

> Early in the litigation, parties must share relevant information concerning electronic systems and storage methodologies to foster agreements regarding protocols and equip courts with the information needed to craft suitable discovery orders.

That's a familiar—though poorly realized—obligation in federal practice, but one largely absent from state court practice nationwide.

Weekley Homes shed light on how to access data on an adversary's drives, but offered scant guidance about what's needed to demonstrate whether it's feasible to recover deleted e-mail or what serves as a proper protocol to protect privilege and privacy. A thoughtful protocol balances what lawyers want against what forensic experts can deliver and what deserves protection from disclosure.

The parties may agree that one side's computer forensics expert will operate under a protocol barring disclosure of privileged and confidential information. Increasingly, federal courts appoint neutral forensic examiners to serve as Rule 53 Special Masters for the purpose of performing the forensic examination *in camera*. To address privilege concerns, the information developed by the neutral may first be tendered to counsel for the party proffering the devices, who then generates a privilege log and produces non-privileged, responsive data.

Whether an expert or court-appointed neutral conducts the examination, the order granting forensic examination of ESI should provide for the handling of confidential and privileged data and narrow the scope of examination by targeting specific objectives. The examiner needs clear direction in terms of relevant keywords and documents, as well as pertinent events, topics, persons and time intervals. A common mistake is for parties to agree upon a search protocol or secure an agreed order without first consulting an expert to determine feasibility, complexity or cost. Generally, use of a qualified neutral examiner is more cost-effective and ensures that the court-ordered search protocol is respected.

Who Performs Computer Forensics?

Historically, experienced examiners tended to emerge from the ranks of law enforcement, but this is changing as computer forensics training courses and college degree plans have emerged. Though the ranks of those offering computer forensics services are growing, there is spotty assessment or regulation of the profession. Only a handful of respected certifications exist to test the training, experience and integrity of forensic examiners. Some states require computer forensic examiners to obtain private investigator licenses, but don't demand that applicants possess or demonstrate expertise in computer forensics.

Computer experts without formal forensic training or experience may also offer their services as experts; but just as few doctors are qualified as coroners, few computer experts are qualified to undertake a competent digital forensics analysis. Programming skill has little practical correlation to skill in computer forensics.

Drafting Digital Examination Protocols

A computer or smart phone under forensic examination is like a vast metropolis of neighborhoods, streets, buildings, furnishings and stuff--*loads* of stuff. It's customary for a single machine to yield over a million discrete information items, some items holding thousands of data points. Searching so vast a virtual metropolis requires a clear description of what's sought and a sound plan to find it.

In the context of electronic discovery and digital forensics, an examination protocol is an order of a court or an agreement between parties that governs the scope and procedures attendant to testing and inspection of a source of electronic evidence. Parties and courts use examination protocols to guard against compromise of sensitive or privileged data and ensure that specified procedures are employed in the acquisition, analysis, and reporting of electronically stored information.

A well-conceived examination protocol serves to protect the legitimate interests of all parties, curtails needless delay and expense and forestalls fishing expeditions. Protocols may afford a forensic examiner broad leeway to adapt procedures and follow the evidence, or a protocol may tightly constrain an examiner's discretion to defend against waiver of privilege or disclosure of irrelevant, prejudicial material. A good protocol helps an examiner know where to start his or her analysis, how to proceed and, crucially, when the job is done.

As a litigator for 40 years and a computer forensic examiner for more than 25 years, I've examined countless devices and sources for courts and litigants. In that time, I've never encountered a forensic examination protocol of universal application. "Standard" procedures change over time, adapted to new forms of digital evidence and new hurdles--like full-disk encryption, solid-state storage and explosive growth in storage capacities and data richness. Without a protocol, a forensics examiner could spend months seeking to meet an equivocal examination mandate. The flip side is that poor protocols condemn examiners to undertake pointless tasks and overlook key evidence.

Drafting a sensible forensic examination protocol demands a working knowledge of the tools and techniques of forensic analysis so counsel doesn't try to misapply e-discovery methodologies to forensic tasks. Forensic examiners deal in artifacts, patterns and configurations. The data we see is structured and encoded much differently than what a computer user sees. The significance and reliability of an artifact depends on its context. Dates and times must be validated against machine settings, operating system functions, time zones and corroborating events.

Much in digital forensics entails more than meets the eye; consequently, simply running searches for words and phrases "e-discovery-style" is far less availing than it might be in a collection of documents.

If you can conceive of taking the deposition of a computer or smart phone, crafting a forensic examination protocol is like writing out the deposition questions in advance. Like a deposition, there are basic inquiries that can be scripted but no definitive template for follow-up questions. A good examiner--of people or computers--follows the evidence yet hews to relevant lines of inquiry and respects boundaries. A key difference is good advocates fit the evidence to their clients' narrative where good forensic examiners let the evidence tell its own story.

Common Elements

Though each is unique, examination protocols share common elements. They should, *inter alia*:

- Identify the examiner (or the selection process) and the devices and media under scrutiny
- set the scope of the exam, temporally and topically
- Ensure integrity of the evidence
- Detail the procedures and analyses to be completed
- Set deadlines and reporting responsibilities
- Require cooperation; and,
- Allocate cost.

Good protocols typically set out the goals of the exam and articulate the rights sought to be protected. As needed, a protocol should address the who, what, when and where of access to devices or media and the conditions under which acquisition and examination will occur. A proper chain of custody may be addressed, as well as who may be present when data is acquired or processed.

Identify the Examiner

If a neutral will perform the exam, ideally the parties will agree upon a qualified person. When they cannot, the Court may seek recommendations from other judges or the protocol can require each side to submit proposed candidates, including their *curriculum vitae* and a list of other matters in which the examiner candidates have served as court-appointed neutrals. The Court then looks at the training, experience, professional certifications and other customary indicia of expertise in selecting an appointee. The protocol should make clear whether the examiner is working for a party or serving as a neutral.

> **Exemplar language:** *The parties have until [DATE] to agree upon a computer forensic examiner ("Examiner") who will inspect and analyze the electronic devices and media pursuant to this Protocol. If the parties fail to agree on an Examiner, they shall submit two names each to the Court with a summary of the proposed Examiners' qualifications and experience, not to exceed one page each, and each Examiner's fee structure. The Court will select an Examiner from among the candidates submitted. The Examiner will serve as an officer of the court, agree to submit to the jurisdiction of this Court and be bound by the terms of this Protocol.*

Identify the Devices and Media

A forensic examination protocol should clearly define what devices and media must be tendered for acquisition and analysis. Designations may be as specific as "*Dell Inspiron laptop computer Service Tag XYZ123*" or as broad as "*all computers, cell phones and electronic data storage devices (thumb drives, external hard drives and the like) in the care custody or control of John Doe*."

Forensic examinations routinely turn up evidence pointing to the existence of other potentially relevant devices and storage media. This triggers mistrust and charges of concealment or spoliation. Accordingly, the parties should discuss the potential for other devices to turn up and draft the examination protocol to address whether such items fall within the scope of the examination.

Set the Scope of Examination

As noted, there is no more a "standard" protocol applicable to every forensic examination than there is a "standard" set of deposition questions applicable to every matter or witness. In either circumstance, a skilled examiner tailors the inquiry to the case, follows the evidence as it develops and remains flexible enough to adapt to unanticipated discoveries. Consequently, it is desirable for a court-ordered protocol to afford the examiner some discretion to adapt to the evidence and apply their expertise.

In framing a forensic examination order, it's helpful to set out the goals to be achieved and the risks to be averted. To illustrate, a protocol might state: "*The computer forensic examiner should, as feasible, recover and produce from Smith's computer, phone and storage media tendered for examination all e-mail communications between John Smith and Jane Doe, but without revealing Smith's personal confidential information or the contents of privileged attorney-client communications to any person other than Smith's counsel.*"

The court issued a clear, succinct order in **Bro-Tech Corp. v. Thermax, Inc., 2008 WL 724627 (E.D. Pa. Mar. 17, 2008)**. Though it assumed some existing familiarity with the evidence (*e.g.*, referencing certain "Purolite documents"), an examiner should have no trouble understanding what was expected:

(1) Within three (3) days of the date of this Order, Defendants' counsel shall produce to Plaintiffs' computer forensic expert forensically sound copies of the images of all electronic data storage devices in Michigan and India of which Huron Consulting Group ("Huron") made copies in May and June 2007. These forensically sound copies are to be marked "CONFIDENTIAL--DESIGNATED COUNSEL ONLY";

(2) Review of these forensically sound copies shall be limited to:
(a) MD5 hash value searches for Purolite documents identified as such in this litigation;
(b) File name searches for the Purolite documents; and
(c) Searches for documents containing any term identified by Stephen C. Wolfe in his November 28, 2007 expert report;

(3) All documents identified in these searches by Plaintiffs' computer forensic expert will be provided to Defendants' counsel in electronic format, who will review these documents for privilege;

(4) Within seven (7) days of receiving these documents from Plaintiffs' computer forensic expert, Defendants' counsel will provide all such documents which are not privileged, and a privilege log for any withheld or redacted documents, to Plaintiffs' counsel. Plaintiffs' counsel shall not have access to any other documents on these images;

(5) Each party shall bear its own costs;

Of course, this order keeps a tight rein on the scope of examination by restricting the effort to hash value, filename and keyword searches. Such limitations are appropriate where the parties are seeking a small population of well-known documents but would severely hamper a less-targeted effort. It bears mention that the *Bro-Tech* protocol was barely a forensic examination as it focused exclusively on active data, not forensic artifacts. As such, it's a poor template for a deeper inquiry.

Set the Temporal Scope

Parties routinely seek to impose time constraints on a forensic examination in terms of what data the examiner should search. While limiting an examiner to review of information in a relevant interval may seem wise, it's often infeasible to assign dates to artifacts and a strict temporal limitation may serve to frustrate the ends of the exam. No forensics tool can limit a search of unallocated clusters and forensic artifacts to a date range. There are few temporal guideposts for forensic artifacts because date information is usually absent or may be unreliable. Even for active data, there won't always be metadata in the master file table to support a reliable time limitation.

For example, log files contain information pertaining to dates other than the dates of the log files themselves. Excluding log files based on their file dates serves to prevent scrutiny of temporally relevant log entries. Moreover, file metadata misleads those who don't fully understand its significance. A file's creation date often bears no relation to the date the file's contents were authored. A file's last modified date may relate to events outside a relevant interval although the contents of the file are precisely what the parties seek. An examiner can limit the date range only for items that have reliable temporal metadata, but not otherwise.

So, be wary of language like, "*All searches are restricted to the time period from November 1, 2020 through January 18, 2021*." Interval limitations on search often don't fly, and you won't know what you're missing.

A preferable approach in a protocol might be to specify that the examiner should not *produce* information to counsel if the examiner determines that the information falls outside of the relevant interval specified in the protocol. The distinction is that, while an examiner may not be able to limit a search to an interval, an examiner can often glean enough information about items found to make a reasonable assessment of their temporal relevance.

Exemplar Language: *The parties intend that the scope of the examination be, as feasible, limited to the Relevant Interval: [Date 1] through [Date 2]. Except as otherwise specified herein, Examiner should make reasonable efforts to exclude from production the information that the Examiner determines falls outside of the Relevant Interval.*

Assess Evidence Integrity

Lawyers seeking forensic exams often have cause to suspect fraud or spoliation. So, it should come as no surprise that evidence tendered for forensic examination may be swapped, fabricated, sterilized, reformatted, reimaged or otherwise corrupted. Why then, do so many lawyers framing examination protocols fail to explicitly require that the integrity of the evidence be assessed?

A threshold step in any forensic examination should include consideration of whether the evidence supplied is what it purports to be and whether it appears that its contents have been wiped or manipulated to subvert the exam.

Exemplar Language: *The Examiner shall assess the integrity of the evidence by, e.g., checking Registry keys to investigate the possibility of drive swapping or fraudulent reimaging and looking at logs to evaluate BIOS clock manipulation. The Examiner may take other reasonable steps to determine if the data supplied is consistent with its stated origins, including but not limited to:*

a. Looking at the dates of key system folders to assess temporal consistency with the device, operating system and events;
b. Looking for instances of applications employed to alter file metadata or erase/alter system cache and history data; and,
c. Noting the presence and nature of any recently installed applications and/or antiforensic "privacy" tools.

The Examiner shall promptly report any irregularities concerning the integrity of the evidence to counsel for the parties.

Provide for Cooperation

Hardened device security has made it difficult for computer forensic examiners to bypass passwords and encryption. Today, it's common that users must supply their access credentials to facilitate examination.

Exemplar Language: *The Parties shall cooperate with the Examiner insofar as promptly supplying non-privileged information and passwords and credentials required to access and decrypt data on the Image and accurately interpret same. No passwords or credentials obtained from the image or furnished by the parties will be used by the Examiner to access data other than found on the Image.*

Plot the Process

The most daunting feature of a forensic examination protocol is detailing the procedures and analyses to be completed. It's important to lay out these steps because forensic examiners use different tools, call things by different names and don't all possess the same grasp of forensic artifacts and their significance. The best way to ensure that the work gets done is to describe what's required in language the examiner will clearly understand and as steps the examiner has the tools and expertise to complete.

You can't do that by blindly borrowing language from a protocol in a different case. Instead, counsel must bone up on common forensic artifacts or, better yet, consult a forensic examiner to lay out what needs to be done, describing the steps in enough detail that any examiner using one of the leading forensic tools will know what to do and where to look.

The single biggest mistake lawyers make in drafting forensic examination protocols is requiring examiners to do things they can't do. Forensic examiners can't always tell what files a user copied or what files were deleted. We can't always tell who logged in using another's password. Despite what we see on television, computers don't track everything, and they don't simply log all events, not even in the so-called event logs!

Forensics is a powerful tool; but, it's not magic. Most forensic artifacts on which examiners rely exist only by way of happy accidents.

How Windows Deletes a File

Windows can be downright obstinate in its retention of data you don't want to hang around. Neither emptying the Recycle Bin nor performing a quick format of a disk will obliterate all its secrets. How is that deleting a file doesn't, well, *delete* it? The answer lies in how Windows stores and catalogs files. Remember that the Windows files system deposits files at various locations on a disc drive and then keeps track of where it has tucked those files away in its Master File Table--the table of contents for the massive tome of data on the drive.

The MFT keeps tabs on what parts of the hard drive contain files and what parts are available for storing new data. When a user deletes a file, Windows doesn't scurry around the hard drive vacuuming up ones and zeroes. Instead, it adds an entry to the master file table noting "this file has been deleted" and, by so doing, makes the disk space containing the deleted data ("**unallocated clusters**") available for storage of new data. But permitting a file drawer to be used for new stuff and clearing out the old stuff are different things. The old stuff—the deleted data—stays on an electromagnetic hard drive until new data overwrites it.

If we return to our library card catalog analogy, pulling a card from the card catalog doesn't remove the book from the shelf, although when consulting the card catalog, you wouldn't know it's there.

Deleting a computer file only removes its “card” (the file table record). The book (the file’s content) hangs around until the librarian needs the shelf space for new titles.

“True, I can’t take it with me, but I can take the access codes to it.”

© Cartoonbank.com

Let’s assume there is a text file called secrets.txt on your computer and it contains the account numbers and access codes to your Panamanian numbered account. Sadly, the bloom has gone off the rose that was your marriage, and you decide that perhaps it would be best to get this information out of the house. So, you copy it to a thumb drive and delete the original. Now, you’re aware that though the file no longer appears in its folder, it’s still accessible in the Recycle Bin. Consequently, you open the Recycle Bin and execute the “Empty Recycle Bin” command, thinking you can now rest easy. In fact, the file is not gone. All that’s happened is that Windows has flipped a bit in the Master File Table to signal that the space once occupied by the file is now available for reuse. The file and the passwords and account numbers it holds is still on the drive and, until the physical space the data occupy is overwritten by new data, a forensic examiner can read the contents of the old file or undelete it. Even if the file is partially overwritten, some of its contents may be recoverable if the new file is smaller in size than the file it replaces. This is true for your text files, financial files, images, Internet pages you’ve visited and so on.

Happy Accidents: LNK Files, Prefetch, Windows Registry and Other Revealing Artifacts

You can roughly divide the evidence in a computer forensic examination between evidence generated or collected by a user (*e.g.,* an Excel spreadsheet or downloaded photo) and evidence created by the system which serves to supply the context required to authenticate and weigh user-generated evidence. User-generated or -collected evidence tends to speak for itself without need of expert interpretation. In contrast, artifacts created by the system require expert interpretation, in part because such artifacts exist to serve purposes having nothing to do with logging a user’s behavior for use as evidence in court. Most forensic artifacts arise because of a software developer’s effort to supply a better user experience and improve system performance. Their probative value in court is a happy accident.

For example, on Microsoft Windows systems, a forensic examiner may look to machine-generated artifacts called LNK files, prefetch records and Registry keys to determine what files and applications a user accessed and what storage devices a user attached to the system.

LNK files (pronounced "link" and named for their file extension) serve as pointers or "shortcuts" to other files. They are like shortcuts users create to conveniently launch files and applications; but, these LNK files aren't user-created. Instead, the computer's file system generates them to facilitate access to recently used files and stores them in the user's RECENT folder. Each LNK file contains information about its target file that endures even when the target file is deleted, including times, size, location and an identifier for the target file's storage medium. Microsoft didn't intend that Windows retain information about deleted files in orphaned shortcuts; but, there's the happy accident–or maybe not so happy for the persons caught because their computer was trying to better serve them.

Similarly, Windows seeks to improve system performance by tracking the recency and frequency with which applications are run. If the system knows what applications are most likely to be run, it can "fetch" the programming code those applications need in advance and pre-load them into memory, speeding the execution of the program. Thus, records of the last 128 programs run are stored in series of so-called "prefetch" files. Because the metadata values for these prefetch files coincide with use of the associated program, by another happy accident, forensic examiners may attest to, *e.g.,* the time and date a file wiping application was used to destroy evidence of data theft.

Two more examples of how much forensically significant evidence derives from happy accidents are the USBSTOR and DeviceClasses records found in the **Windows System Registry** hives. The Windows Registry is the central database that stores configuration information for the system and installed applications—it's essentially everything the operating system needs to "remember" to set itself up and manage hardware and software. The Windows Registry is huge and complex. Each time a user boots a Windows machine, the registry is assembled from a group of files called "hives." Most hives are stored on the boot drive as discrete files and one—the Hardware hive—is created anew each time the machine inventories the hardware it sees on boot.

The Registry can provide information of forensic value, including the identity of the computer's registered user, usage history data, program installation information, hardware information, file associations, serial numbers and some password data. The Registry is also one area where you can access a list of recent websites visited and documents created, often even if the user has taken steps to delete those footprints. A key benefit of the Registry in forensics is that it tracks the attachment of USB storage media like thumb drives and external hard drives, making it easier to track and prove data theft.

When a user connects an external mass storage device like a portable hard drive or flash drive to a USB port, the system must load the proper device drivers to enable the system and device to communicate. To eliminate the need to manually configure drivers, devices have evolved to support so-called Plug and Play capabilities. Thus, when a user connects a USB storage device to a Windows system, Windows interrogates the device, determines what driver to use and—importantly—

records information about the device and driver pairing within a series of keys stored in the ENUM/USBSTOR and the DeviceClasses "keys" of the System Registry hive. In this process, Windows tends to store the date and time of both the earliest and latest attachments of the USB storage device.

Windows is not recording the attachment of flash drives and external hard drives to enable forensic examiners to determine when employees attached storage devices to steal data. The programmer's goal was to speed selection of the right drivers the next time the USB devices were attached; but the happy accident is that the data retained for a non-forensic purpose carries enormous probative value when properly interpreted and validated by a qualified examiner.

Shellbags

If you've ever wondered why, when you change the size and shape of a Windows Explorer folder your preferences are retained the next time you use that folder, the answer lies in Windows retention of folder configuration data in "keys" (entries) within the system Registry called **Shellbags**.

When a forensic examiner locates a shellbag key for a folder, the examiner can reasonably conclude that the folder has been opened--a significant observation if the folder contains, say, child pornography or other data the user was not permitted to access. Shellbags are also a trove of other data respecting the folder, relevant dates and even files that formerly resided within the folder but have been moved or deleted.

Swap and Hibernation Files

Just like you and me, Windows needs to write things down as it works to keep from exceeding its memory capacity. Windows extends its memory capacity (RAM) by swapping data to and from a file on disk called a **"swap file**." When a multitasking system such as Windows has too much information to hold in memory at once, some of it is stored in the swap file until needed. If you've ever wondered why Windows seems to always be accessing the hard drive, chances are it's reading or writing information to its swap file. Windows uses the term **"page file"** (because the blocks of memory swapped around are called *pages),* but it's essentially the same thing: a giant digital "scratch pad."

The swap file contains data from the system memory; consequently, it can contain information that the typical user never anticipates would reside on the hard drive. Moreover, we are talking about a considerable volume of information. How much varies from system-to-system, but it runs to *billions* of bytes. For example, the page file on the windows machine used to write this article is currently 16 megabytes. The swap file is a hefty 2.5 gigabytes, holding a swath of whatever kind of information exists (or used to exist) on my computer, running the gamut from word processing files, e-mail, Internet web pages, database entries, you name it. It also includes passwords and decryption keys. If the user used it, parts of it are floating around the swap file.

Because the memory swapping is (by default) managed dynamically in Windows, the swap file tends to disappear each time the system is rebooted, its contents relegated to unallocated space and recoverable in the same manner as other deleted files.

Another system file of a similar nature is the Windows hibernation file (Hiberfile.sys), 13.3 GB on my machine. It records the system state when the computer hibernates to promote a faster wake-from-sleep. Accordingly, it stores to disk all data from running applications at the time the machine went into hibernation mode.

The Windows swap and hibernation files are forensic treasure troves, but they are no picnic to examine. Although filtering software exists to help in locating so-called **named entities**, e.g., passwords, phone numbers, credit card numbers and fragments of English language text, it's a labor-intensive effort like so much of computer forensics in this day of multi-terabyte hard drives.

Windows NTFS Log File

The NTFS file system increases system reliability by maintaining a log of system activity. The log is designed to allow the system to undo prior actions if they have caused the system to become unstable. The log file is a means to reconstruct aspects of computer usage. The log file is customarily named $LogFile, but it is not viewable in Windows Explorer, so don't become frustrated looking for it.

TMP Files

Every time you run Microsoft Word, Excel, PowerPoint, etc., these programs create temporary files. The goal of temp files is often to save your work in the event of a system failure and then disappear when they are no longer needed. Temp files do a respectable job saving your work but, much to the good fortune of the forensic investigator, they do a lousy job of disappearing. Computers orphan temp files when a program locks up, power fails or due to other atypical shutdowns. When the application restarts, it creates new temp file, but rarely does away with its orphaned predecessor. It just hangs around. Even when the application deletes the temp file, the contents of the file tend to remain in unallocated space until overwritten.

As an experiment (for Windows users), search your hard drive for all files with the .TMP extension. You can usually do this with the search query "*.TMP." You may have to adjust your system settings to allow viewing of system and hidden files. When you get the list, forget any with a current date and look for .TMP files from prior days.[33] Open those in Notepad or WordPad and you may be shocked to see how much of your work hangs around without your knowledge. Word processing applications are by no means the only types which keep (and orphan) temp files.

[33] I found more than 2,600 old .TMP files on my machine on May 1, 2019.

Files with the .BAK, .WBK or .ASD extensions usually represent timed backups of work in progress maintained to protect a user in the event of a system crash or program lock up. Applications like word processing software create .BAK and .ASD files at periodic intervals. While these files are supposed to be deleted by the system, they often linger on.

Volume Shadow Copies

Microsoft has been gradually integrating a feature called Volume Snapshot Service (a/k/a Volume Shadow Copy Service) into Windows since version XP; but until Windows 7, you couldn't truly say the implementation was so refined and entrenched as to permit the recovery of almost anything from a remarkable cache of data called **Volume Shadow Copies**.

Volume shadow copies are largely unknown to the e-discovery community. Though a boon to forensics, volume shadow copies may prove a headache in e-discovery because their contents represent reasonably accessible ESI from the user's standpoint.

Much of what e-discovery professionals believe about file deletion, wiping and even encryption goes out the window when a system runs any version of Windows with Volume Snapshot Service enabled (and it's enabled by default). Volume Shadow Copies keep virtually everything, and Windows keeps up to 64 volume shadow copies, made at daily or weekly intervals. ***These aren't just system restore points: volume shadow copies hold user work product, too.*** The frequency of shadow copy creation varies based upon multiple factors, including whether the machine is running on A/C power, CPU demand, user activity, volume of data needing to be replicated and changes to system files. So, 64 "weekly" shadow volumes could represent anywhere from two weeks to two years of indelible data, or far less.

How indelible? Consider this: most applications that seek to permanently delete data at the file level do it by deleting the file then overwriting its storage clusters. As you've learned, these are called "unallocated clusters," because they are no longer allocated to storage of a file within the Windows file system and are available for reuse. But, the Volume Shadow Copy Service (VSS) monitors *both* the contents of unallocated clusters and any subsequent efforts to overwrite them. Before unallocated clusters are overwritten, VSS swoops in and rescues the contents of those clusters like Spiderman saving Mary Jane.

These rescued clusters (a/k/a "blocks") are stored in the next created volume shadow copy on a space available basis. Thus, each volume shadow copy holds only the *changes* made between shadow volume creation; that is, it records only *differences* in the volumes on a block basis in much the same way that incremental backup tapes record only changes between backups, not entire volumes. When a user accesses a previous version of a deleted or altered file, the operating systems instantly assembles all the differential blocks needed to turn back the clock. It's all just three clicks away:

1. Right click on file or folder for context menu;
2. Left click to choose "Restore Previous Versions;"
3. Left click to choose the date of the volume.[34]

It's an amazing performance…and a daunting one for those seeking to make data disappear.

From the standpoint of e-discovery, responsive data that's just three mouse clicks away is likely to be deemed fair game for identification, preservation and production. Previous versions of files in shadow volumes are as easy to access as any other file. There's no substantial burden or collection cost for *the user* to access such data, item-by-item. But, as easy as it is, few of the standard e-discovery tools and protocols have been configured to identify and search the previous versions in volume shadow copies. It's just not a part of vendor workflows; but eventually, someone will see the naked emperor and ask why we ignore this data in discovery.

These are examples, and we must recognize that artifacts are different for different operating systems (Windows versus MacOS) and even for different releases of the same operating system. Artifacts are radically different on phones versus computers. It's complicated, and it changes, literally, every day..

If you will be using a neutral examiner, draft the protocol to provide for the parties to confer with the examiner to establish the scope of work. Too often, examiners are saddled with unwieldy protocols poorly tailored to answering the parties' questions because the protocol was drafted without professional guidance.

What you should *not* expect to occur is your expert gaining direct access to your opponent's digital media. The more-likely result is a protocol laying out the steps to be followed by your *opponent's* expert or by a court-appointed neutral examiner.

Establish Who Pays

Though the forensic preservation of a desktop or laptop machine tends to cost no more than a short deposition, the cost of a forensic examination can vary widely depending upon the nature and complexity of the media under examination and the issues. Forensic examiners usually charge by the hour with rates ranging from approximately $200-$600 per hour according to experience, training, reputation and locale. Costs of extensive or poorly targeted examinations can quickly run into five and even six figures. Nothing has a greater influence on the cost than the scope of the examination. Focused examinations communicated via clearly expressed protocols tend to keep costs down. Searches should be carefully evaluated to determine if they are over- or under inclusive. The examiner's progress should be followed closely, and the protocol modified as

[34] This GUI access capability was removed in Windows 8 but restored in Windows 10 for users who enable the "File History" featyres.

needed. It's prudent to have the examiner report on progress and describe work yet to be done when either hourly or cost benchmarks are reached.

In all events, the examination protocol should make clear how, when and by whom the Examiner is compensated for professional time and reimbursed for expenses.

Exemplar Language: *Charges for Examiner's professional time and time in transit shall be timely paid by Plaintiffs at the Examiner's customary rates, along with reasonable and customary expenses according to the terms of the rate sheet submitted before appointment. In the event Examiner's charges equal or exceed $_______, the Examiner shall report progress to the parties and project further charges expected to be incurred to completion.*

Address Onsite Acquisition and Supervision

A party whose systems are being acquired and examined may demand to be present throughout the process. This may be feasible while the contents of a computer are being *acquired* (duplicated); otherwise, it's an unwieldy, unnecessary and profligate practice. Computer forensic examinations are commonly punctuated by the need to allow data to be processed or searched. Such efforts consume hours, even days, of "machine time," but not examiner time. Examiners sleep, eat and turn to other cases and projects until the process completes. However, if an examiner must be supervised during machine time operations, the examiner cannot jeopardize another client's expectation of confidentiality by turning to other matters. Thus, the "meter" runs all the time, without any commensurate benefit to either side except as may flow from the unwarranted inflation of discovery costs.

Demanding that forensically sound acquisition occur on a client's premises versus in an examiner's lab can hugely inflate cost. On-site acquisition may be unavoidable for mission-critical systems like servers; but otherwise, I push back against demands to work on a party's premises versus in my own lab. In the lab, I can turn to other tasks and stop billing. Onsite acquisition and analysis run up the bill unnecessarily and require I be furnished a workspace that's suitable and secure, perhaps for days or longer. The pandemic has prompted adoption of remote collection techniques as never before..

Recovering Deleted Data

Although the goals of forensic examination vary depending on the circumstances justifying the analysis, a common aim is recovery of deleted data. One court ordered, "if the files…have been deleted or altered using a drive-wiping utility, [forensic examiner] will also recover all deleted files and file fragments." *Schreiber v. Schreiber*, 2010 WL 2735672 (N.Y. Sup. Ct. June 25, 2010). That's not such a good idea.

The Perils of "Undelete Everything"

Examination protocols shouldn't direct the examiner to, in effect, "undelete all deleted material and produce it." That *sounds* clear, but it creates unrealistic expectations and invites excessive cost. Here's why:

A computer manages its hard drive in much the same way that a librarian manages a library. The files are the "books" and their location is tracked by an index. But there are two key differentiators between libraries and computer file systems. Computers employ no Dewey decimal system, so electronic "books" can be on any shelf. Further, electronic "books" may be split into chapters, and those chapters stored in multiple locations across the drive. This is called "**fragmentation**." Historically, libraries tracked books by noting their locations on index card in a card catalog. Computers similarly employ directories (called "**file tables**") to track files and fragmented segments of files.

When a user hits "Delete" in a Windows environment, nothing happens to the actual file targeted for deletion. Instead, a change is made to the master file table that keeps track of the file's location. Thus, akin to tearing up a card in the card catalogue, the file, like its literary counterpart, is still on the "shelf," but now—without a locator in the file table—the deleted file is a needle in a haystack, buried amidst millions of other unallocated clusters.

To recover the deleted file, a computer forensic examiner employs three principal techniques:

1. File Carving by Binary Signature

Because most files begin with a unique digital signature identifying the file type, examiners run software that scans each of the millions of unallocated clusters for file signatures, hoping to find matches. If a matching file signature is found and the original size of the deleted file can be ascertained, the software copies or "carves" out the deleted file. If the size of the deleted file is unknown, the examiner designates how much data to carve out. The carved data is then assigned a new name and the process continues.

Unfortunately, deleted files may be stored in pieces, as discussed above, so simply carving out contiguous blocks of fragmented data grabs intervening data having no connection to the deleted file and fails to collect segments for which the directory pointers have been lost. Likewise, when the size of the deleted file isn't known, the size designated for carving may prove too small or large, leaving portions of the original file behind or grabbing unrelated data. Incomplete files and those commingled with unrelated data are generally corrupt and non-functional. Their evidentiary value is also compromised.

File signature carving is frustrated when the first few bytes of a deleted file are overwritten by new data. Much of the deleted file may survive, but the data indicating what type of file it was, and thus enabling its recovery, is gone.

File signature carving requires that each unallocated cluster be searched for each of the file types sought to be recovered. When a court directs that an examiner "recover all deleted files," that's an exercise that could take excessive effort, followed by countless hours spent examining corrupted files. Instead, the protocol should, as feasible, specify the *file* types of interest based upon how the machine was used and the facts and issues in the case.

Notably, file carving of deleted information from unallocated clusters is fast becoming untenable by the emergence of solid state and encrypted media. Storage optimization techniques used by solid state drives serve to routinely overwrite once-recoverable data.

2. File Carving by Remnant Directory Data

In some file systems, residual file directory information revealing the location of deleted files may be strewn across the drive. Forensic software scans the unallocated clusters in search of these lost directories and uses this data to restore deleted files. Here again, reuse of clusters can corrupt the recovered data. A directive to "undelete everything" gives no guidance to the examiner respecting how to handle files where the metadata is known but the contents are suspect.

3. Search by Keyword

Where it's known that a deleted file contained certain words or phrases, the remnant data may be found using keyword searching of the unallocated clusters and slack space. Keyword search is a laborious and notoriously inaccurate way to find deleted files, but its use may be warranted when other techniques fail. When keywords are too short or not unique, false positives ("**noise hits**") are a problem. Examiners must painstakingly look at each hit to assess relevance and then manually carve out responsive data. This process can take days or weeks for a single machine.

Better Practice than "Undelete" is "Try to Find"

The better practice is to eschew broad directives to "undelete everything" in favor of targeted directives to use reasonable means to identify specified types of deleted files. To illustrate, a court might order, "*Examiner should seek to recover any deleted Word, Excel, PowerPoint and PDF files, as well as to locate potentially relevant deleted files or file fragments in any format containing the terms, 'explosion,' 'ignition' or 'hazard.'*"

Reporting and Deadlines

In the context of digital forensics, "reporting" means many things. As a lawyer-examiner, I create narrative reports setting forth in plain language what I'm seeing in the evidence and what my training and experience suggest it signifies. But, most forensic examiners regard reporting as a machine-generated process. It's common for a forensic "report" to consist of dozens or hundreds of pages of mostly unintelligible gibberish spit out by software. So, it's smart to deal with that in the protocol. If the parties need specific questions answered in a narrative fashion, say so. If the analysis must be completed by a time certain, set deadlines for preliminary and final reporting and establish whether meeting those deadlines is feasible for the examiner (recognizing that the examiner has seen no evidence and probably has more questions than answers).

Forensic Acquisition versus Preservation

Parties and courts are wise to distinguish and apply different standards to requests for forensically sound *acquisition* versus those seeking forensic *examination*. Forensically sound acquisition of implicated media guards against spoliation engendered by continued usage of computers and by intentional deletion. It also preserves the ability to later conduct a forensic examination, if warranted.

Forensic *examination* and analysis of an opponent's ESI is both intrusive and costly, necessitating proof of egregious abuses before allowing one side to directly access the contents of the other side's computers and storage devices (something I caution courts against ordering). By contrast, forensically duplicating and preserving the *status quo* of electronic evidence costs little and can generally be accomplished without significant inconvenience or intrusion upon privileged or confidential material. Accordingly, courts should freely order forensic preservation upon a showing of good cause.

During the conduct of a forensically sound acquisition:

1. Nothing on the evidence media is altered by the acquisition;
2. Everything on the evidence media is faithfully acquired; and,
3. The processes employed are authenticated to confirm success.

These standards cannot be met in every situation—notably, in the logical acquisition of a live server or physical acquisition of a phone or tablet device—but parties deviating from a "change nothing" standard should disclose and justify that deviation.

Exemplar Acquisition Protocol

An exemplar protocol for acquisition follows, adapted from the court's order in *Xpel Techs. Corp. v. Am. Filter Film Distribs*., 2008 WL 744837 (W.D. Tex. Mar. 17, 2008):

> The motion is GRANTED and expedited forensic imaging shall take place as follows:
> A. Computer forensic acquisition will be performed by ____________ (the "Examiner").
> B. Examiner's costs shall be borne by the Plaintiff.
> C. Examiner must agree in writing to be bound by the terms of this Order prior to the commencement of the work.

D. Within two days of this Order or at such other time agreed to by the parties, Defendants shall make the specified computer(s) and other electronic storage devices available to Examiner to enable Examiner to make forensically-sound images of those devices, as follows:

i. Images of the computer(s) and any other electronic storage devices in Defendants' possession, custody, or control shall be made using hardware and software tools that create a forensically sound, bit-for-bit, mirror image of the original hard drives (*e.g.,* EnCase, FTK Imager, X-Ways Forensics or Linux dd). A bitstream mirror image copy of the media item(s) will be captured and will include all file slack and unallocated space.

ii. Examiner should document the make, model, serial or service tag numbers, peripherals, dates of manufacture and condition of the systems and media acquired.

iii. All images and copies of images shall be authenticated by cryptographic hash value comparison to the original media.

iv. The forensic images shall be copied and retained by Examiner in strictest confidence until such time the court or both parties request the destruction of the forensic image files.

v. Without altering any data, Examiner should, as feasible, determine and document any deviations of the systems' clock and calendar settings.

E. Examiner will use best efforts to avoid unnecessarily disrupting the normal activities or business operations of the Defendants while inspecting, copying, and imaging the computers and storage devices.

F. The Defendants and their officers, employees and agents shall refrain from deleting, relocating, defragmenting, overwriting data on the subject computers or otherwise engaging in any form of activity calculated to impair or defeat forensic acquisition or examination

Pulling It Together in an Exemplar Protocol

The following exemplar examination protocol was accepted by the Court in a case where the parties sought to determine what a user was doing on a laptop a laptop machine on a single day. As the machine was in Illinois, it was determined that the forensic image would be acquired by a local examiner and the image shipped.

Examination Protocol for Windows Laptop

I. **GOALS:** The purpose of Protocol is to guide, Craig Ball, Texas attorney and Certified Computer Forensic Examiner ("Examiner") in identifying and interpreting active and latent artifacts tending to shed light on the nature, extent and timing of usage, if any, of a Windows laptop machine ("Machine") during specified relevant intervals, as well as in assessing the integrity of the Machine and its contents for data loss, destruction and alteration during and following the relevant interval (*Date 1 through Date 2*).

II. **EVIDENCE:** This protocol assumes that Examiner will receive a forensically-sound, hash-authenticated bitstream image ("Image") of the Machine's data storage device(s) along with customary chain-of-custody information and baseline data establishing the accuracy or deviation of the Machine's system clock at the time of Image acquisition. Unless otherwise agreed by the parties and the Examiner, only a duly-certified Computer Forensic Examiner shall image the Machine and authenticate the chain-of-custody and baseline data.

III. **DUPLICATION:** The Examiner will make hash-authenticated working and archival copies of the Image. The Image supplied will not otherwise be used for analysis but will be secured until return or disposal.

IV. **COOPERATION AND CREDENTIALS:** The Parties shall cooperate with the Examiner insofar as promptly supplying non-privileged information and passwords and credentials required to access and decrypt data on the Image and accurately interpret same. No passwords or credentials obtained from the image or furnished by the parties will be used by the Examiner to access data other than found on the Image.

V. **AUTHORIZATION AND SCOPE:** The Examiner may:

1. Load an authenticated working copy of the Image into an analysis platform or platforms and examine the file structures for anomalies.
2. Assess the integrity of the evidence by, *e.g.*, checking Registry keys to investigate the possibility of drive swapping or fraudulent reimaging and looking at logs to evaluate BIOS clock manipulation. The Examiner may take other reasonable steps to determine if the data supplied is consistent with its stated origins.
3. Look at the various creation dates of key system folders to assess temporal consistency with the machine, OS install and events.
4. Look for instances of applications employed to alter file metadata or erase/alter usage cache and history data.
5. Note recently installed applications and any antiforensic "privacy" tools.
6. Refine the volume snapshot to, *e.g.,* identify relevant, deleted folders, applications and files, orphaned file records, Host Protected Areas, hidden partitions, inter-partition data and encrypted volumes.
7. Further refine the volume snapshot to unpack compound files (e.g., compressed and container files), compare binary file signatures with file extensions, identify possible encrypted files using entropy testing, hash all files, extract application metadata and process contents of Volume Shadow Copies.
8. Carve the unallocated clusters for file artifacts using binary signature analysis, seeking deleted files and deleted cache content, temp files, fragments and system artifacts.
9. Locate and extract Registry hives for analysis.

10. Look at the LNK files, index files, TEMP directories, cookies, Registry MRUs, shellbags, jump lists, thumbnails, shadow copies and, as relevant, system and event logs and Windows prefetch area, to assess usage of applications, files and network accesses.
11. Generate and export complete file listings with associated file size, file path, hash and temporal metadata values (other metadata values as relevant and material).
12. If indicated, run keyword searches against the contents of all clusters (including unallocated clusters and file slack) seeking relevant data, then review same.
13. Sort the data chronologically for the relevant Modified, Accessed and Created (MAC) dates to assess the nature of activity within the relevant interval.
14. As feasible, generate a network activity report against, *inter alia*, index.dat and comparable network activity artifacts to determine, *inter alia,* if there has been web surfing web search, e-mail, texting, download or upload activity or research conducted at pertinent times concerning, *e.g.,* how to destroy or alter electronic evidence, conceal system and network usage and the like.
15. Filter for e-mail messaging formats (*e.g.,* PST, OST, NSF, DBX, MSG, EML, etc.), and extract messaging for processing in preferred application. Check OLK folders (Outlook attachment temp storage).
16. Examine container files for relevant email in the relevant interval(s). If web mail, look at cache data. If not found, carve UAC to reconstruct same.
17. Identify mobile device (*e.g.*, iTunes, Android) and Cloud (*e.g.,* DropBox) synch sources.
18. Gather the probative results of the efforts detailed above, assess whether anything else is likely to shed light on the documents and, if not, share conclusions as to what transpired.
19. Make recommendations for further lines of inquiry or sources of data, if any.

VI. **COST:** Charges for Examiner's professional time and time in transit shall be timely paid by Plaintiffs at the Examiner's customary rates, along with reasonable and customary expenses according to the terms of the Examiner's Engagement Agreement.

Hashing

The order above empowers the examiner to "hash all files." As you've seen, hashing is the use of mathematical algorithms to calculate a unique sequence of letters and numbers to serve as a "fingerprint" for digital data. These fingerprint sequences are called "message digests" or, more commonly, "hash values." It's an invaluable tool in both computer forensics and electronic discovery, and one deployed by courts with growing frequency.

The ability to "fingerprint" data enables forensic examiners to prove that their drive images are faithful to the source. Further, It allows the examiner to search for files without the necessity of examining their content. If the hash values of two files are identical, the files are identical. This file matching ability allows hashing to be used to de-duplicate collections of electronic files before review, saving money and minimizing the potential for inconsistent decisions about privilege and responsiveness for identical files.

A court may order the use of hash analysis to:

1. Demonstrate that data was properly preserved by recording matching hash values for the original and its duplicate
2. Search data for files with hash values matching hash values of expropriated data alleged to be confidential proprietary
3. Exclude from processing and production files with hash values matching known irrelevant files, like the Windows operating system files or generic parts of common software in a process termed "de-NISTing;" or,
4. Employ hash values instead of Bates numbers to identify ESI produced in native formats.

Hashing is often a pivotal tool employed to conclusively identify known contraband images in prosecutions for child pornography.

Although hashing is an invaluable and versatile technology, it has a few shortcomings. Because the tiniest change in a file will alter that file's hash value, hashing is of little value in finding contraband data once it's been modified. Changing a file's name won't alter its hash value (because the name is generally not a part of the file), but even minimally changing its contents will render the file unrecognizable by its former hash value. Another limitation to hashing is that, while a changed hash value proves a file has been altered, it doesn't reveal how, when or where within a file change occurred.

What's Missing?

Privilege and Confidentiality Concerns: The preceding protocol involved a matter where privileged and confidential material and communications weren't a concern; but protocols more typically need to provide for non-waiver of privilege and for counsel's review of the examiner's reporting before it's seen by opposing counsel so that objections can be asserted to disclosure of privileged or protected content. A protocol should also address *ex parte* communications with the Examiner.

Exemplar Language: *To the extent the Examiner has direct or indirect access to information protected by the attorney-client privilege, such access will not result in a waiver of the attorney-client privilege. Unless counsel for all parties are included, there shall be no communications between any party or party's counsel aside from purely ministerial communications necessary to complete the tasks set out in this Protocol.*

All data and analyses governed by this Protocol are deemed protected material. Possession of such material is limited to the Examiner the attorneys of record in the captioned cause and their experts. Counsel and their experts may not share or review the protected material in any manner with any other person, including their respective clients.

Any data or reporting resulting from the Examination will be produced by the Examiner to the attorney for the device/media owner for review. No data will be provided to opposing counsel until it has been reviewed and released by the attorney for the device/media owner. A listing of the data that was forwarded to counsel for the device/media owner will be included with the data. This index will include the file name, the date last modified and the file size, as feasible. Data not in the form of a file will be identified on the listing in a reasonably clear and practical manner. The attorney for the device/media owner will identify on the listing any items that will not be produced and the basis for withholding such items. Items not withheld shall be produced to the party or parties requesting the data along with the listing showing the items withheld and the basis for withholding such items.

Parties may object to withholding of any data, and counsel for the parties shall cooperate on procedures to resolve disputes about withheld data. If the parties cannot resolve a dispute as to the production of withheld data, then any party may move for protection or for an order to compel production.

Forms of Production: The protocol also doesn't address the challenge of delivering forensic artifacts to counsel in usable formats. Lawyers and courts are conditioned to expect "documents" and are rarely prepared for data. A crucial forensic artifact may be no more than a few bytes of encoded information bobbing in a sea of unallocated clusters. A handful of these might be converted to a document-like format for review; but what if there are hundreds of thousands of such instances to examine (as commonly occurs when running keyword searches against unallocated clusters)? Lawyers can't expect that the fruits of a forensic examination can be loaded into an e-discovery review platform and treated like documents. Too, lawyers can't expect to load native files into native software applications without altering the evidence. Native applications modify native files.

Skilled forensic examiners are experienced in working with lawyers to facilitate review of forensic artifacts in practical, scalable ways. Since it's not always practical or possible to provide for a form of production in advance of a forensic examination, a protocol should afford the examiner some leeway to supply deliverables in forms suited to assist the parties in their review (and the Court in any *in camera* review).

Ethical Boundaries

Unless the Court expressly permits, or the parties agree, a forensic examiner should never use the devices tendered for examination or information derived in the exam to access information beyond that stored on the physical devices and media when tendered for examination. Most examiners know this and will act ethically; however, a thorough protocol should make that restraint clear, so none need worry that an overeager examiner will abuse a booted clone device or a user's log in credentials.

Exemplar Language: *Examiner shall not use the devices and storage media tendered for examination, or any information or credentials derived from same, to access any electronic information not present on the devices and storage media when tendered for examination. This prohibition includes but is not limited to accessing private online or Cloud accounts, e-mail accounts or servers, private social media sites and banking and credit card accounts and transactions.*

Other Points to Ponder

A protocol may need to address topics such as disposition of evidence after analysis, data retention and destruction duties (including financial responsibility for same), amenability to discovery (deposition and subpoena), applicability of protective orders.

It's useful to empower the examiner to make recommendations for further lines of inquiry or sources of data. Certainly, the parties and the Court must be sensitive to suggestions that smack of make-work; but, a skilled, ethical examiner will often have the best ideas where to go to find other relevant electronic evidence.

Conclusion

Crafting a forensic examination protocol demands more than finding a good form to filch. It requires a clear sense of about what you seek to accomplish through an examination and the ability to express those goals with enough technical specificity to guide a diligent examiner to the artifacts that will answer your questions. There's often a tension between one side's wish to rein the examiner in and the others' to turn the examiner loose. A good protocol balances the two and affords the examiner just enough discretion to follow the electronic evidence and let it tell its tale.

Frequently Asked Questions

How do I preserve the status quo without requiring a party to stop using its systems?

The ongoing use of a computer system erodes the effectiveness of a computer forensic examination and serves as an ongoing opportunity to delete or alter evidence. Where credible allegations suggest the need for forensic examination may arise, the best course is to immediately secure a forensically sound image of the machine or device, acquired by a qualified technician and authenticated by hashing. Alternatively, the party in control of the machine may agree to replace the hard drive and sequester the original evidence drive so that it will not be altered or damaged.

A party wants to have its technicians make "Ghost" images of the drives. Are those forensically sound images?

No, only tools and software especially suited to the task collect every cluster on a drive without altering the evidence. Other software (like Norton Ghost) that IT staff use to duplicate data for installation on new hardware may be called "imaging tools," but they aren't *forensically sound* imaging tools. The failure to employ write protection will effect changes to the evidence and only forensically sound imaging tools collect data in all regions of storage media important to a thorough forensic examination. Even the right software and hardware in unskilled hands is not a guarantee of a forensically sound acquisition.

The use of other imaging methods may be entirely sufficient to meet preservation duties when issues requiring computer forensics aren't at stake.

Do servers need to be preserved by forensically sound imaging, too?

Though forensic examiners may differ on this issue, generally, forensically sound imaging of servers is unwarranted because the way servers operate makes them poor candidates for examination of their unallocated clusters. This is an important distinction because the consequences of shutting down a server to facilitate forensic acquisition may have severe business interruption consequences for a party. For preservation in e-discovery, live acquisition of the server's active data areas is usually sufficient and typically doesn't require that the server be downed.

What devices and media should be considered for examination?

Though computer forensics is generally associated with servers, desktops and laptops, these are rarely the only candidates for examination. When they may hold potentially relevant ESI, forensic acquisition and/or examination could encompass external hard drives, thumb drives, tablet devices, smart phones, web mail accounts, Cloud storage areas, media cards, entertainment devices with storage capabilities (e.g., iPods and gaming consoles), digital cameras, optical media, legacy media (*e.g.,* floppy and ZIP disks), automobile air bag modules and incident data recorders ("black boxes"), GPS units, backup tape and any of a host of other digital storage devices and Internet of Things sources. Moreover, machines used at home, legacy machines sitting in closets or storage rooms and machines used by "proxies" like secretaries, assistants and family members must be considered as candidates for examination.

How intrusive is a computer forensic examination?

A computer forensic examination entails that the devices and media under scrutiny be acquired in a forensically sound manner. Remote acquisition may be feasible, though more often the process requires a user to surrender his or her computer(s) for several hours, but rarely longer than overnight. If a user poses no interim risk of wiping the drive or deleting files, acquisition can generally be scheduled so as not to unduly disrupt a user's activities.

A properly conducted acquisition makes no changes to the user's data on the machine, so it can be expected to function exactly as before upon its return. No software, spyware, viruses or any other applications or malware are installed.

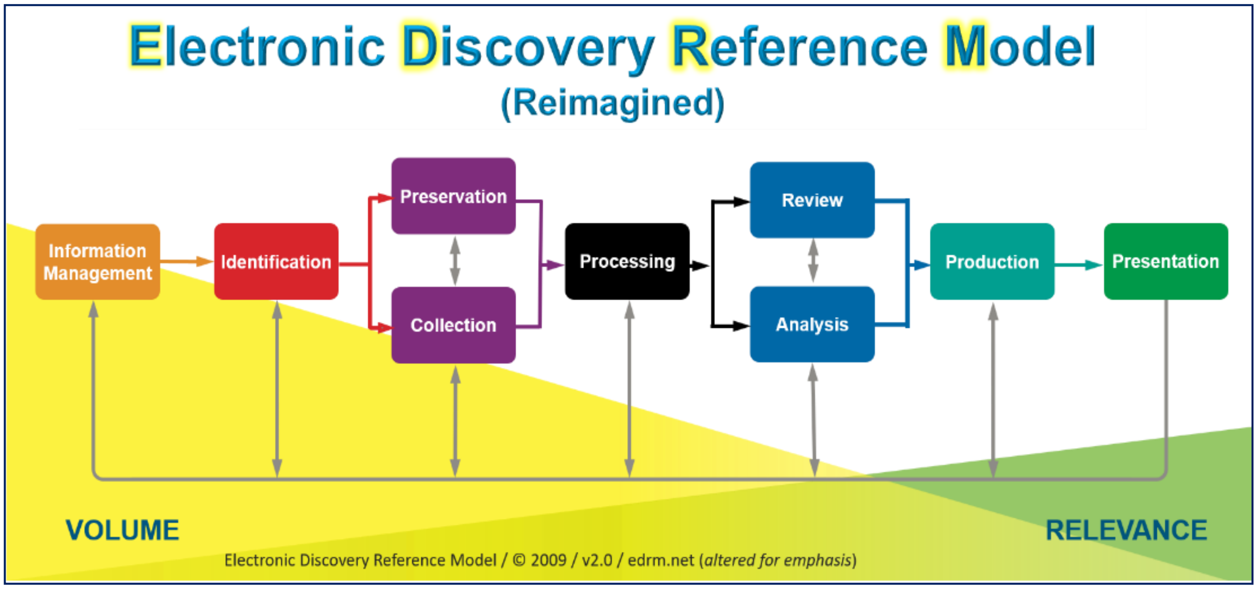
The intrusion attendant to forensic examination flows from the fact that such examination lays bare any and all current or prior usage of the machine, including for personal, confidential and privileged communications, sexual misadventure, financial and medical recordkeeping, storage of proprietary business data and other sensitive matters. Though it may be possible to avoid intruding on such data within the orderly realm of active data, once deleted, these relevant and irrelevant data cannot easily be segregated or avoided. Accordingly, it's important for the court to either impose strict limits on the use and disclosure of such information by the examiner or require that the examination be conducted by a neutral examiner obliged to protect the legitimate discovery and privacy concerns of both sides.

What does it cost?

Though the forensic preservation of a desktop or laptop machine tends to cost no more than a short deposition, the cost of a forensic examination can vary widely depending upon the nature and complexity of the media under examination and the issues. Forensic examiners usually charge by the hour with rates ranging from approximately $200-$600 per hour according to experience, training, reputation and locale. Costs of extensive or poorly targeted examinations can quickly run into five and even six figures. Nothing has a greater influence on the cost than the scope of the examination. Focused examinations communicated via clearly expressed protocols tend to keep costs down. Keyword searches should be carefully evaluated to determine if they are over- or underinclusive. The examiner's progress should be followed closely, and the protocol modified as needed. It's prudent to have the examiner report on progress and describe work yet to be done when either hourly or cost benchmarks are reached.

Processing in E-Discovery

Processing is the "black box" between preservation/collection and review/analysis. Though the iconic Electronic Discovery Reference Model (EDRM) positions Processing, Review and Analysis as parallel paths to Production, processing is an essential prerequisite—"the only road"—to Review, Analysis and Production.[35] Any way you approach e-discovery at scale, you must process ESI before you can review or analyze it. If we recast the EDRM to reflect processing's centrality, the famed schematic would look like this:



There are hundreds—perhaps thousands—of articles delving into the stages of e-discovery that flank processing in the EDRM. These are the stages where lawyers have had a job to do. But lawyers tend to cede processing decisions to technicians. When it comes to processing, lawyer competency and ken is practically non-existent, little more than "stuff goes in, stuff comes out."

Why process ESI in e-discovery? Isn't it "review ready?"

We process information in e-discovery to catalog and index contents for search and review. Unlike Google, e-discovery is the search for *all* responsive information in a collection, not just one information item deemed responsive. Though all electronically stored information is inherently electronically searchable, computers don't structure or search all ESI in the same way; so, we must process ESI to ***normalize*** it to achieve uniformity for indexing and search.

Thus, "processing" in e-discovery could be called "normalized access," in the sense of extracting content, decoding it and managing and presenting content in consistent ways for access and review.

[35] That's not a flaw. The EDRM is a conceptual view, not a workflow.

It encompasses the steps required to extract text and metadata from information items and build a searchable index. The closest analogy is the creation of a Google-like capability with respect to a discrete collection of documents, data and metadata.

ESI processing tools perform five common functions.[36] They must:

1) Decompress, unpack and fully explore, *i.e.,* **recurse** ingested items.
2) **Identify** and apply templates (filters) to **encoded** data to **parse** (interpret) contents and **extract text, embedded objects**, and **metadata**.
3) **Track** and **hash** items processed, **enumerating** and **unitizing** all items and tracking failures.
4) **Normalize** and **tokenize** text and data and create an **index** and **database** of extracted information.
5) **Cull** data by file type, date, lexical content, hash value and other criteria.

Files

If we polled lawyers asking what to call the electronic items preserved, collected and processed in discovery, most would answer, "documents." Others might opt for "data" or reflect on the initialization "ESI" and say, "information." None are wrong answers, but the ideal response would be the rarest: "*files*." Electronic documents are *files*. Electronically stored information resides in *files*. *Everything* we deal with *digitally* in electronic discovery comes from or goes to physical or logical data storage units called "data files" or just "files." Moreover, all programs run against data files are themselves files comprising instructions for tasks. These are "executable files" or simply "executables."

So, what is it we process in the processing stage of e-discovery? The answer is, "**we process *files***." Let's look at these all-important files and explore what's in them and how are they work.

A Bit About and a Byte Out of Files

A colleague once defended her ignorance of the technical fundamentals of electronically stored information by analogizing that "she didn't need to know how planes stay aloft to fly on one." She had a point, but only for passengers. If you aspire to be a pilot or a rocket scientist—if you want to be at the controls or design the plane—you must understand the fundamentals of flight. If you aspire to understand processing of ESI in e-discovery and manage e-discovery, you must understand the fundamentals of electronically stored information, including such topics as:

- *What's* stored electronically?

[36] While a processing tool may do considerably more than the listed functions, a tool that does less is unlikely to meet litigants' needs in e-discovery.

- *How* is it stored?
- *What forms* does it take?

The next few pages are a crash course in ESI storage, particularly the basics of encoding and recording textual information. If you can tough it out, you'll be undaunted by discussions of "Hex Magic Numbers" and "Unicode Normalization" yet to come.

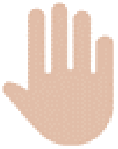
Digital Encoding

All digital evidence is encoded, and how it's encoded bears upon how it's collected and processed, whether it can be searched and in what reasonably usable forms it can be produced. Understanding that electronically stored information is numerically encoded data helps us see the interplay and interchangeability between different forms of digital evidence. Saying "*it's all ones and zeroes*" means nothing if you don't grasp *how* those ones and zeros underpin the evidence.

Electronic evidence is just data, and data are just numbers; so, understanding the numbers helps us better understand electronic evidence.

Decimal and Binary: Base 10 and Base Two

Understanding encoding requires we hearken back to those hazy days when we learned to tally and count by numbers. Long ago, we understood quantities (numeric *values*) without knowing the **numerals** we would later use to symbolize quantities. When we were three or four, "five" wasn't yet Arabic **5**, Roman **V** or even a symbolic tally like 𝍸.
More likely, five was this:



If you're from the Americas, Europe or Down Under, I'll wager you were taught to count using the **decimal system**, a **positional notation** system with a **base** of 10. Base 10 is so deeply ingrained in our psyches that it's hard to conceive of numeric values being written any other way. Decimal just *feels* like one, "true" way to count, but it's not. Writing numbers using an alternate base or "**radix**" is just as genuine, and it's advantageous when information is stored or transmitted digitally.

Think about it. Human beings count by tens because we evolved with ten digits on our hands. Were that not so, tasteless old jokes like this one would make no sense: "*Did you hear about the Aggie who was arrested for indecent exposure? He had to count to eleven.*"

Had our species evolved with eight fingers or twelve, we would have come to rely upon an octal or duodecimal counting system, and we would regard those systems as the "true" positional notation system for numeric values. Ten only feels natural because we built everything around ten. Again, if we'd evolved with eight or ten fingers, it really wouldn't matter because you can express any number—and consequently any data—in any number system. So, it happens that computers use the base two or binary system, and computer programmers are partial to base sixteen or hexadecimal. Data is just numbers, and it's all just counting.

When we were children starting to count, we had to *learn* the decimal system. We had to *think* about what numbers *meant*. When our first-grade selves tackled big numbers like 9,465, we were overtly aware that each digit represented a decimal multiple. The nine was in the thousands place, the four in the hundreds, the six in the tens place and the five in the ones. We might even have parsed 9,465 as: *(9 x 1000) + (4 x 100) + (6 x 10) + (5 x 1)*.

But soon, it became second nature to us. We'd unconsciously process 9,465 as nine thousand four hundred sixty-five. As we matured, we learned about powers of ten and now saw 9,465 as: *(9 x 10^3) + (4 x 10^2) + (6 x 10^1) + (5 x 10^0).* This was exponential or "base ten" notation. We flushed it from our adolescent brains as fast as life (and the SAT) allowed.

Computers don't have fingers; instead, computers count using a slew of electronic switches that can be "on" or "off." Having just two states (on/off) makes it natural to count using Base 2, a binary counting system. By convention, computer scientists *notate* the status of the switches using the numerals one and zero. So, we tend to say that computers store information as ones and zeroes. Yet, they don't.

Computer storage devices like IBM cards, hard drives, tape, thumb drives and optical media store information as physical phenomena that can be reliably distinguished in either of two distinct states, *e.g.,* punched holes, changes in magnetic polar orientation, minute electric potentials or deflection of laser beams. We *symbolize* these two states as one or zero, but you could represent the status of binary data by, say, turning a light on or off. Early computing systems did just that, hence all those flashing lights.

You can express any numeric value in any base without changing its value, just as it doesn't change the numeric value of "five" to express it as Arabic "5" or Roman "V" or just by holding up five fingers.

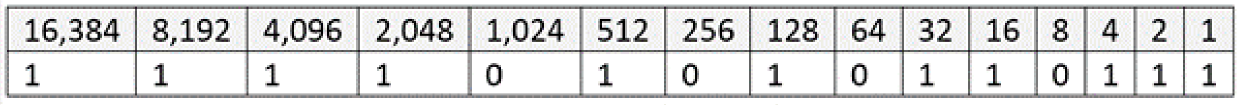
In positional notation systems, the order of numerals determines their contribution to the value of the number; that is, their contribution is the value of the digit multiplied by a factor determined by the position of the digit and the base.

The base/radix describes the number of unique digits, starting from zero, that a positional numeral system uses to represent numbers. So, there are just two digits in base 2 (binary), ten in base 10 (decimal) and sixteen in base 16 (hexadecimal). E-mail attachments are encoded using a whopping *64 digits* in base 64.

We speak the decimal number 31,415 as "thirty-one thousand, four hundred and fifteen," but were we faithfully adhering to its base 10 structure, we might say, "three ten thousands, one thousand, four hundreds, one ten and five ones. The "base" ten means that there are ten characters used in the notation (0-9) and the value of each position is ten times the value of the position to its right.

10,000	1,000	100	10	1
3	1	4	1	5

The same *decimal* number 31,415 can be written as a *binary* number this way: **111101010110111**



16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1
1	1	1	1	0	1	0	1	0	1	1	0	1	1	1

In base 2, two characters are used in the notation (0 and 1) and each position is twice the value of the position to its right. If you multiply each digit times its position value and add the products, you'll get a total equal in value to the decimal number 31,415.

A value written as five characters in base 10 requires 15 characters in base 2. That seems inefficient until you recall that computers count using on-off switches and thrive on binary numbers.

The decimal value 31,415 can be written as a base 16 or hexadecimal number this way: **7AB7**

4,096	256	16	1
7	A	B	7

In base 16, sixteen characters are used in the notation (0-9 and A-F) and each position is sixteen times the value of the position to its right. If you multiply each digit times its position value and add the products, you'll get a total equal in value to the decimal number 31,415. But how do you multiply letters like A, B, C, D, E and F? You do it by knowing the letters are used to denote values greater than 9, so A=10, B=11, C=12, D=13, E=14 and F=15. Zero through nine plus the six values represented as letters comprise the sixteen characters needed to express numeric values in hexadecimal.

Once more, if you multiply each digit/character times its position value and add the products, you'll get a total equal in value to the decimal number 31,415:

	7 x 4,096	=	28,672
The A means 10 so:	10 x 256	=	2,560
The B means 11 so:	11 x 16	=	176
Finally, we add:	7 x 1	=	7

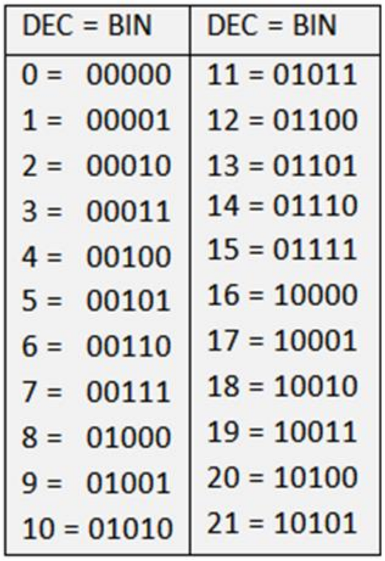
			31,415

Bits

To recap, computers use **binary digits** in place of decimal digits. The word **bit** is even a shortening of the words "**B**inary dig**IT**." Unlike the decimal system, where we represent any number as a combination of *ten* possible digits (0-9), the binary system uses only *two* possible values: zero or one. This is not as limiting as one might expect when you consider that a digital circuit—essentially an unfathomably complex array of switches—hasn't got any fingers to count on but is very good and very fast at being "on" or "off."

In the binary system, each binary digit—each "bit"—holds the value of a power of two. Therefore, a binary number is composed of only zeroes and ones, like this: 10101. How do you figure out what the value of the binary number 10101 is? You do it in the same way we did it above for 9,465, but you use a base of 2 instead of a base of 10. Hence: (1 x 2^4) + (0 x 2^3) + (1 x 2^2) + (0 x 2^1) + (1 x 2^0) = 16 + 0 + 4 + 0 + 1 = **21**.

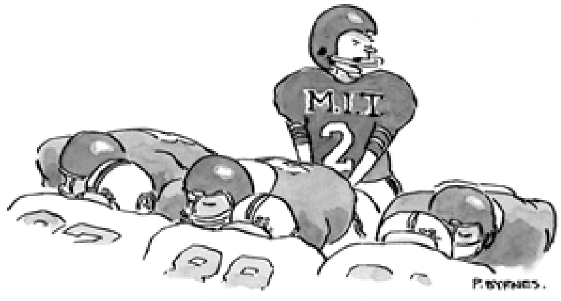
Moving from right to left, each bit you encounter represents the value of increasing powers of 2, standing in for zero, two, four, eight, sixteen, thirty-two, sixty-four and so on. That makes counting in binary easy. Starting at zero and going through 21, decimal and binary equivalents look like the table at right.



DEC = BIN	DEC = BIN
0 = 00000	11 = 01011
1 = 00001	12 = 01100
2 = 00010	13 = 01101
3 = 00011	14 = 01110
4 = 00100	15 = 01111
5 = 00101	16 = 10000
6 = 00110	17 = 10001
7 = 00111	18 = 10010
8 = 01000	19 = 10011
9 = 01001	20 = 10100
10 = 01010	21 = 10101

Bytes

Computers also work with binary data in eight-character sequences called **bytes**. A byte is a string (sequence) of eight bits. The biggest number that can be stored as one byte of information is 11111111, equal to 255 in the decimal system. The smallest number is zero or 00000000. Thus, you can store 256 different numbers as one byte of information (0-255). So, what do you do if you need to store a number larger than 256? Simple! You use a *second* byte. This affords you all the combinations that can be achieved with 16 bits, being the product of all of the variations of the first byte and all of the second byte (256 x 256 or 65,536). So, using bytes to express values, we express any number greater than 256 using at least two bytes (called a “word” in geek speak), and any number above 65,536 requires three bytes or more.



“01101001, 00111011, 00011010, but, but!”

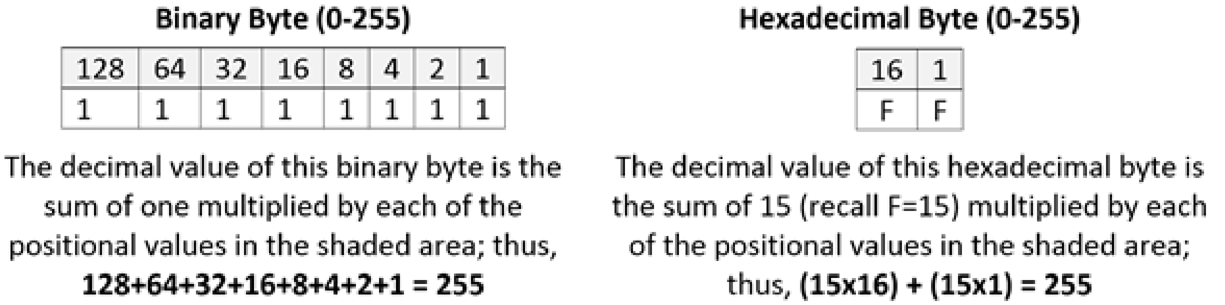
Why are eight-bit sequences the fundamental building blocks of computing? It just happened that way. In these times of cheap memory, expansive storage and lightning-fast processors, it’s easy to forget how scarce and costly such resources were at the dawn of the computing era. Seven bits (with a leading bit reserved) was the smallest block of data that would suffice to represent the minimum complement of alphabetic characters, decimal digits, punctuation and control instructions needed by the pioneers in computer engineering. It was, in a sense, all the data early processors could bite off at a time, perhaps explaining the name “byte” (coined in 1956 by IBM scientist Dr. Werner Buchholz).

Werner Buchholz

Hexadecimal

Once more, a binary sequence of eight ones and zeros (“bits”) can be arranged in 256 unique ways. Long sequences of ones and zeroes are hard for humans to follow, so happily, two hexadecimal characters can also be arranged in 256 unique ways, meaning that just two base-16 characters can replace the eight characters of a binary byte (*i.e.*, a binary value of 11111111 can be written in hex as FF). Using hexadecimal characters allows programmers to write data in just 25% of the space required to write the same data in binary, and it's easier for humans to follow.

Let's take a quick look at why this is so. A single binary byte can range from 0 to 255 (being 00000000 to 11111111). ***Computers count from zero***, so that range spans 25**6** unique values. The following table demonstrates why the largest value of an eight-character binary byte (11111111) equals the largest value of just two hexadecimal characters (FF):



Binary Byte (0-255)

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

The decimal value of this binary byte is the sum of one multiplied by each of the positional values in the shaded area; thus, **128+64+32+16+8+4+2+1 = 255**

Hexadecimal Byte (0-255)

16	1
F	F

The decimal value of this hexadecimal byte is the sum of 15 (recall F=15) multiplied by each of the positional values in the shaded area; thus, **(15x16) + (15x1) = 255**

Hexadecimal values are everywhere in computing. Litigation professionals encounter hexadecimal values as MD5 hash values and may run into them as IP addresses, Globally Unique Identifiers (GUIDs) and even color references.

The Magic Decoder Ring called ASCII

Back in 1935, American kids who listened to the Little Orphan Annie radio show and drank lots of Ovaltine could join the Radio Orphan Annie Secret Society and obtain a Magic Decoder Ring, a device with rotating disks that allowed them to read and write numerically encoded messages.[37]

Similarly, computers encode words as numbers. Binary data stand in for the upper- and lower-case English alphabet, as well as punctuation marks, special characters and machine instructions (like carriage return and line feed).

Encoding Text

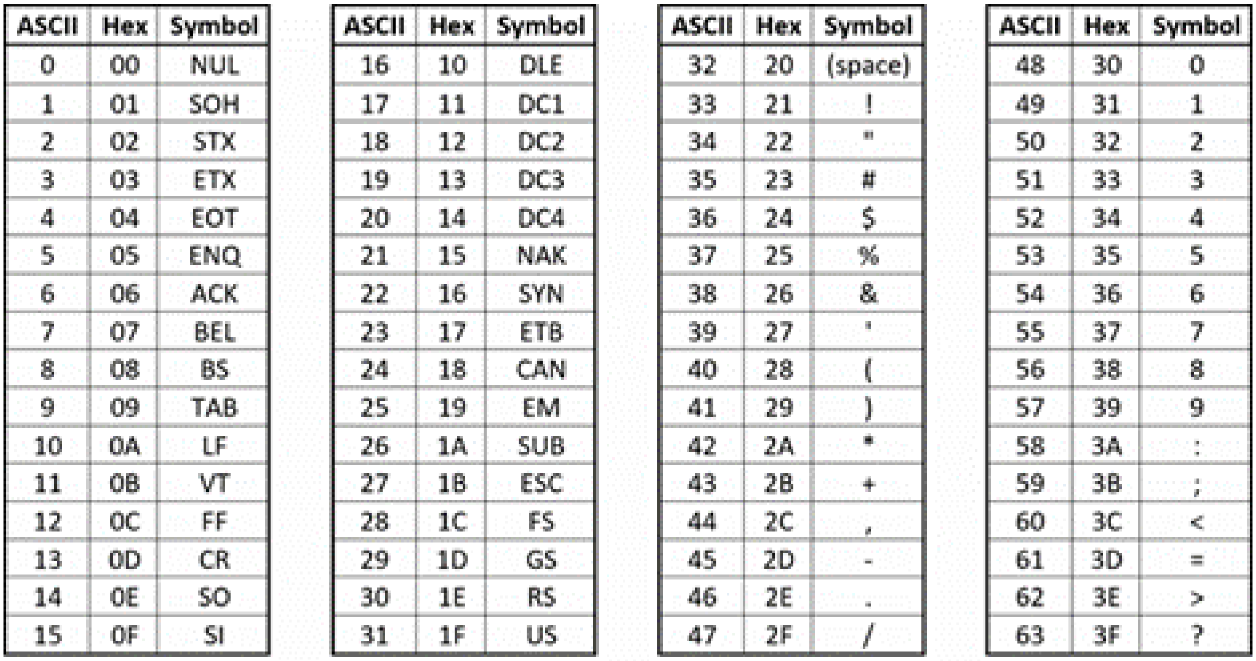
So far, I've described ways to encode the same numeric value in different bases. Now, let's shift gears to describe how computers use those numeric values to signify intelligible alphanumeric information like the letters of an alphabet, punctuation marks and emoji. Again, data are just numbers, and those numbers signify something *in the context of the*

[37] A similar toy, a secret decoder *pin*, was depicted in the movie, *A Christmas Story*.

application using that data, just as gesturing with two fingers may signify the number two, a peace sign, the V for Victory or a request that a blackjack dealer split a pair. What numbers mean depends upon the **encoding scheme** applied to the values in the application; that is, the encoding scheme supplies the essential context needed to make the data intelligible. If the number is used to describe an RGB color, then the hex value 7F00FF means violet. Why? Because each of the three values that make up the number (7F 00 FF) denote how much of the colors red, green and blue to mix to create the desired RGB color. In other contexts, the same hex value could mean the decimal number 8,323,327, the binary string 11111110000000011111111 or the characters 缀ÿ.

ASCII

When the context is text, there are a host of standard ways, called **Character Encodings** or **Code Pages**, in which the numbers denote letters, punctuation and symbols. Now nearly sixty years old, the **American Standard Code for Information Interchange** (ASCII, "ask-key") is the basis for most modern character encoding schemes (though both Morse code and Baudot code are older). Born in an era of teletypes and 7-bit bytes, ASCII's original 128 codes included 33 non-printable codes for controlling machines (*e.g.*, carriage return, ring bell) and 95 printable characters. By limiting the ASCII character set to just 128 characters (0-127), we can express any character in just seven bits (2^7 or 128) and so occupy only one byte in the computer's storage and memory. The ASCII character set follows:

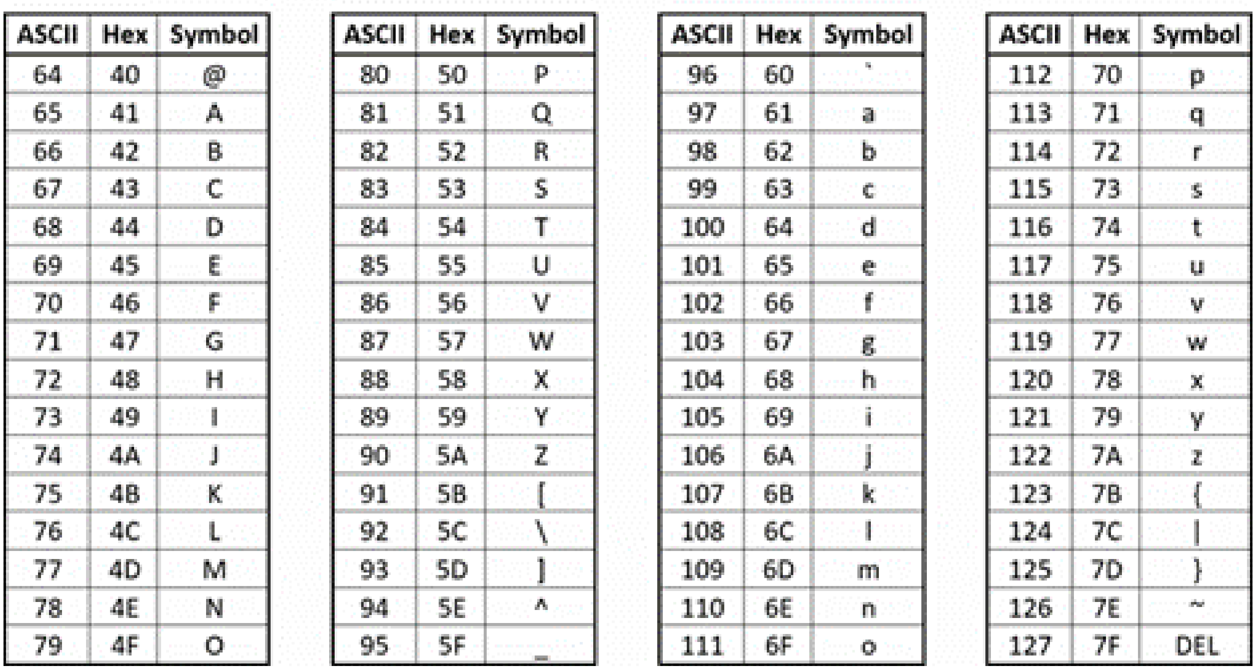


ASCII	Hex	Symbol
0	00	NUL
1	01	SOH
2	02	STX
3	03	ETX
4	04	EOT
5	05	ENQ
6	06	ACK
7	07	BEL
8	08	BS
9	09	TAB
10	0A	LF
11	0B	VT
12	0C	FF
13	0D	CR
14	0E	SO
15	0F	SI

ASCII	Hex	Symbol
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US

ASCII	Hex	Symbol
32	20	(space)
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/

ASCII	Hex	Symbol
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?



ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	DEL

Here's the same ASCII character set expressed in binary values:

Binary	Decimal	Character	Binary	Decimal	Character	Binary	Decimal	Character
00000000	000	NUL	00101011	043	+	01010110	086	V
00000001	001	SOH	00101100	044	,	01010111	087	W
00000010	002	STX	00101101	045	-	01011000	088	X
00000011	003	ETX	00101110	046	.	01011001	089	Y
00000100	004	EOT	00101111	047	/	01011010	090	Z
00000101	005	ENQ	00110000	048	0	01011011	091	[
00000110	006	ACK	00110001	049	1	01011100	092	\
00000111	007	BEL	00110010	050	2	01011101	093	]
00001000	008	BS	00110011	051	3	01011110	094	^
00001001	009	HT	00110100	052	4	01011111	095	_
00001010	010	LF	00110101	053	5	01100000	096	`
00001011	011	VT	00110110	054	6	01100001	097	a
00001100	012	FF	00110111	055	7	01100010	098	b

Binary	Decimal	Character
00001101	013	CR
00001110	014	SO
00001111	015	SI
00010000	016	DLE
00010001	017	DC1
00010010	018	DC2
00010011	019	DC3
00010100	020	DC4
00010101	021	NAK
00010110	022	SYN
00010111	023	ETB
00011000	024	CAN
00011001	025	EM
00011010	026	SUB
00011011	027	ESC
00011100	028	FS
00011101	029	GS
00011110	030	RS
00011111	031	US
00100000	032	SP
00100001	033	!
00100010	034	"
00100011	035	#
00100100	036	$
00100101	037	%
00111000	056	8
00111001	057	9
00111010	058	:
00111011	059	;
00111100	060	<
00111101	061	=
00111110	062	>
00111111	063	?
01000000	064	@
01000001	065	A
01000010	066	B
01000011	067	C
01000100	068	D
01000101	069	E
01000110	070	F
01000111	071	G
01001000	072	H
01001001	073	I
01001010	074	J
01001011	075	K
01001100	076	L
01001101	077	M
01001110	078	N
01001111	079	O
01010000	080	P
01100011	099	c
01100100	100	d
01100101	101	e
01100110	102	f
01100111	103	g
01101000	104	h
01101001	105	i
01101010	106	j
01101011	107	k
01101100	108	l
01101101	109	m
01101110	110	n
01101111	111	o
01110000	112	p
01110001	113	q
01110010	114	r
01110011	115	s
01110100	116	t
01110101	117	u
01110110	118	v
01110111	119	w
01111000	120	x
01111001	121	y
01111010	122	z
01111011	123	{

Binary	Decimal	Character	Binary	Decimal	Character	Binary	Decimal	Character
00100110	038	&	01010001	081	Q	01111100	124	|
00100111	039	'	01010010	082	R	01111101	125	}
00101000	040	(	01010011	083	S	01111110	126	~
00101001	041	)	01010100	084	T	01111111	127	DEL
00101010	042	*	01010101	085	U	**Note: 0-127 is 128 values**		

So, "E-Discovery" would be written in a binary ASCII sequence as:

0100010100101101**01000100**01101001**01110011**01100011**01101111**01110110**01100101**01110010**01111001**

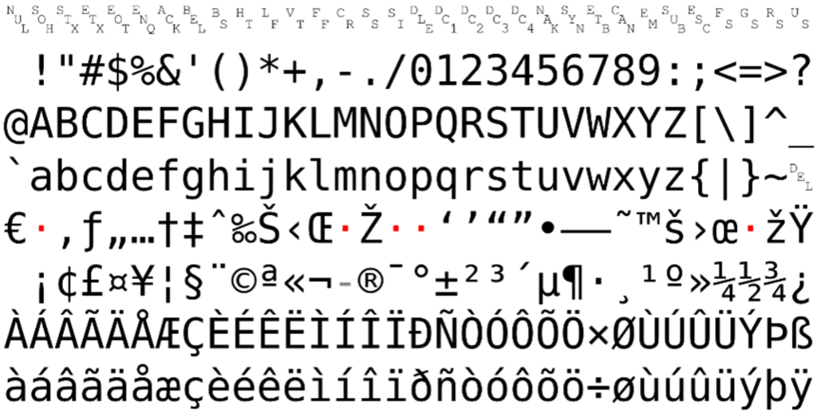
It would be tough to remember your own name written in this manner! *Hi, I'm Craig, but my computer calls me* ***01000011****01110010****01100001****01101001****01100111.***

Windows-1252

Note that each leading bit (i.e., the first character) of each byte in the binary table above is a zero. It wasn't used to convey any encoding information; that is, they are really all *7-bit* bytes. Later, when the byte standardized from seven to eight bits, 128 additional characters could be added to the character set by simply changing the leading bit to a one, prompting the development of *extended* character encodings that include, *e.g.*, accented characters used in foreign languages and line drawing characters.

In the mid-1980s, international standards began to emerge for character encoding, ultimately resulting in various code sets issued by the International Standards Organization (ISO). These retained the first 128 American ASCII values and assigned the upper (extended) 128-byte values to characters suited to various languages (*e.g.,* Cyrillic, Greek, Arabic and Hebrew). ISO called these various character sets ISO-8859-*n,* where the *"n"* distinguished the sets for different languages. ISO-8859-1 was the set suited to Latin-derived alphabets (like English) and so the nickname for the most familiar code page to U.S. computer users became "**Latin 1**."

Arguably the most used single-byte character set in the world is the **Windows-1252 code page,** the characters of which are set out in the following table (red dots signify unassigned values).



```
NUL SOH STX ETX EOT ENQ ACK BEL BS HT LF VT FF CR SO SI DLE DC1 DC2 DC3 DC4 NAK SYN ETB CAN EM SUB ESC FS GS RS US
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~DEL
€·‚ƒ„…†‡ˆ‰Š‹Œ·Ž··‘’“”•–—˜™š›œ·žŸ
 ¡¢£¤¥¦§¨©ª«¬-®¯°±²³´µ¶·¸¹º»¼½¾¿
ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞß
àáâãäåæçèéêëìíîïðñòóôõö÷øùúûüýþÿ
```

Note that the first 128 control codes and characters (from NUL to DEL) match the ASCII encodings and the 128 characters that follow are the extended set. Each character and control code has a corresponding fixed byte value, *i.e.*, an upper-case B is hex 40 and the section sign, §, is hex A7. To see the entire code page character set and the corresponding hexadecimal encodings on Wikipedia, click here. Again, ASCII and the Windows-1252 code page are *single byte* encodings so they are limited to a maximum of 256 characters.

Unfortunately, these extra characters weren't assigned in the same way by all computer systems. The emergence of different sets of characters mapped to the same high byte values prompted a need to identify these various **character encodings** or, as Microsoft calls them in Windows, these "**code pages**." If an application used the wrong code page, some information displayed as gibberish. This is such a familiar phenomenon that it has its own name, **mojibake** (from the Japanese for "character changing"). If you've encountered a bunch of Asian language characters in an e-mail or document you know was written in English, you might have glimpsed mojibake.

Note that we are speaking here of textual information, not typography; so, don't confuse character encodings with fonts. The former tells you whether the character is an A or b, not whether to display the character in Arial or Baskerville.

Unicode

ASCII dawned in the pre-Internet world of 1963—before the world was flat, when the West dominated commerce and personal computing was the stuff of science fiction. The Windows-1252 code page works reasonably well so long as you're writing in English and most European languages; but sporting only 256 characters, it won't suffice if you're writing in, say, Greek, Cyrillic, Arabic or Hebrew, and it's wholly unsuited to Asian languages like Chinese, Japanese and Korean.

Though programmers developed various *ad hoc* approaches to foreign language encodings, an increasingly interconnected world needed universal, systematic encoding mechanisms. These methods would use ***more than one byte*** to represent each character, and the most widely adopted such system is **Unicode**. In its latest incarnation (version 14.0, effective 9/14/21), Unicode standardizes the encoding of 159 written character sets called “scripts” comprising 144,697 characters, plus multiple symbol sets and emoji characters.

The Unicode Consortium crafted Unicode to co-exist with the longstanding ASCII and ANSI character sets by emulating the ASCII character set in corresponding byte values within the more extensible Unicode counterpart, UTF-8. UTF-8 can represent all 128 ASCII characters using a single byte and all other Unicode characters using two, three or four bytes. Because of its backward compatibility and multilingual adaptability, UTF-8 has become the most popular text encoding standard, especially on the Internet and within e-mail systems.

Mind the Gap!

Now, as we talk about all these bytes and encoding standards as a precursor to further discussion of hexadecimal notation, it will be helpful to revisit how this all fits together. A byte is eight ones or zeroes, which means a byte can represent 256 different decimal numbers from 0-255. So, two bytes can represent a much bigger range of decimal values (256 x 256 or 65,536). Character encodings (aka “code pages”) like Latin 1 and UTF-8 are ways to map textual, graphical or machine instructions to numeric values expressed as bytes, enabling machines to store and communicate information in human languages. As we move forward, keep in mind that *hex, like binary and decimal, is just another way to write numbers*. Hex is not a code page, although the numeric values it represents may correspond to values *within* code pages. [38]

Hex

Long sequences of ones and zeroes are very confusing for people, so **hexadecimal notation** emerged as more accessible shorthand for binary sequences. Considering the prior discussion of base 10 (decimal) and base 2 (binary) notation, it might be enough to say that hexadecimal is base 16. In hexadecimal notation (**hex** for short), each digit can be any value from zero to fifteen. Accordingly, we can replace four binary digits with just a single hexadecimal digit, and more to the point, we can express a byte as just two hex characters.

The decimal system supplies only 10 symbols (0-9) to represent numbers. Hexadecimal notation demands 16 symbols, leaving us without enough single character *numeric* values to stand in for all the values in each column. So, how do we cram 16 values into each column? The solution was to substitute the letters A through F for the numbers 10 through 15. So, we can represent 10110101 (the decimal number 181) as "B5" in hexadecimal notation. Using hex, we can notate values from 0-255 as 00 to FF (using either lower- or upper-case letters; it doesn’t matter).

[38] Don’t be put off by the math. The biggest impediment to getting through the encoding basics is the voice in your head screaming, “I SHOULDN’T HAVE TO KNOW ANY OF THIS!!” Ignore that voice. It’s wrong.

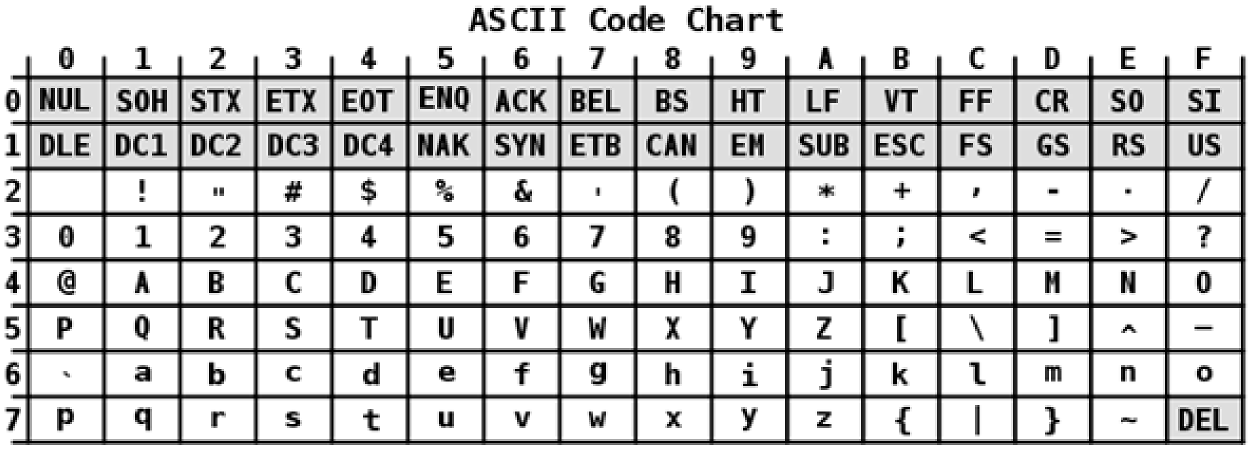
It’s hard to tell if a number is decimal or hexadecimal just by looking at it: if you see "37", does that equate to 37 ("37" in decimal) or a decimal 55 ("37" in hexadecimal)? To get around this problem, two common notations are used to indicate hexadecimal numbers. The first is the suffix of a lower-case "h." The second is the prefix of "0x." So "37 in hexadecimal," "37h" and "0x37" all mean the same thing.

The ASCII Code Chart below can be used to express ASCII characters in hex. The capital letter “G” is encoded as the hex value of 47 (i.e., row 4, column 7), so “**E-Discovery**” in hex encodes as:

0x45 2D 44 69 73 63 6F 76 65 72 79

That’s easier than:

0100010100101101**01000100**01101001**01110011**01100011**01101111**01110110**01100101**01110010**01111001?**



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
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	N	O
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	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

NOTE TO STUDENTS:

For most students of this course—law students particularly—the last twelve pages on digital encoding constitute the most challenging material we cover. If your eyes glazed over on first reading, I understand. But it’s foundational stuff, so give it a second chance when you’re fresh and be sure to ask questions about the material until it’s clear to you. You’ll be glad you did when you start taking the quizzes.

⚛ Exercise 5: Encoding: Decimal, Binary, Hexadecimal and Base64

All digital evidence is encoded, and its encoding bears upon how it's collected, whether it can be searched and in what reasonably usable forms it can be produced. Understanding that electronically stored information is, in essence, numerically encoded data helps students to see the interplay and interchangeability between different forms of digital evidence. Simply saying, "*it's all ones and zeroes*" means nothing if you know nothing of how those ones and zeros underpin evidence you must authenticate or undercut.

GOALS: The goals of this exercise are for the student to:

1. Understand the correspondence between binary data and hexadecimal; and
2. Understand the correspondence between data in hex and encoded text and content.

OUTLINE: We will examine evidence data in Text and Hex modes, noting the correspondence between text and its hexadecimal equivalents. We will then examine the role of Base64 as an encoding scheme for e-mail attachments.

Exercise 5A: Notate ASCII as Hex

Please write your surname in ASCII/hex:

____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____

Exercise 5B: Viewing data in Hex

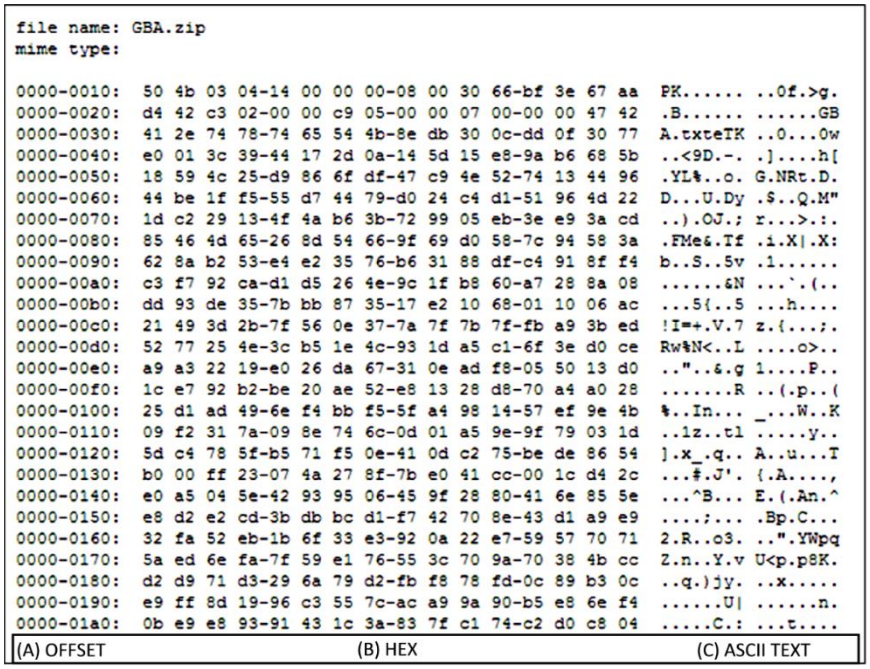
In this exercise, you will use online data viewer tools to examine common file types in hexadecimal. Remember that hexadecimal is just a method to notate numeric values. Such values can be expressed in any notation, e.g., base 2 (binary) or base 10 (decimal) or any other base. *It's all just numbers that are written differently but mean the same thing*. Still, be mindful of the distinction between the notation employed to *record* the information (the "numbers") and the encoding scheme used to *express* the information (e.g., ASCII, ANSI, Unicode, etc.). The notation is akin to an alphabet (e.g., Roman, Cyrillic, etc.) and the encoding scheme is like the language (e.g., English, French, Russian, etc.).

In the preceding exercise, the encoding scheme was ASCII and the notation was hexadecimal. Put another way, ASCII supplied the translation table, and hex served to record the locations within that table.

Step 1: View the File Structure of a Compressed Container (Compound) File

Download the compressed archive file called GBA.zip from http://www.craigball.com/gba.zip. Save the file to your desktop or any other convenient location on your computer.

Using your web browser or the hex viewer of your choice,[39] go to the Online HexDump Utility at http://www.fileformat.info/tool/hexdump.htm and click "choose File." Using the selection box that will appear, navigate to the file **gba.zip** you just saved and select it. Click "Open." Now click the blue "Dump" button on the Online HexDump Utility page. You should see this:



```
file name: GBA.zip
mime type:

0000-0010:  50 4b 03 04-14 00 00 00-08 00 30 66-bf 3e 67 aa  PK...... ..0f.>g.
0000-0020:  d4 42 c3 02-00 00 c9 05-00 00 07 00-00 00 47 42  .B...... ......GB
0000-0030:  41 2e 74 78-74 65 54 4b-8e db 30 0c-dd 0f 30 77  A.txteTK ..0...0w
0000-0040:  e0 01 3c 39-44 17 2d 0a-14 5d 15 e8-9a b6 68 5b  ..<9D.-. .]....h[
0000-0050:  18 59 4c 25-d9 86 6f df-47 c9 4e 52-74 13 44 96  .YL%..o. G.NRt.D.
0000-0060:  44 be 1f f5-55 d7 44 79-d0 24 c4 d1-51 96 4d 22  D...U.Dy .$..Q.M"
0000-0070:  1d c2 29 13-4f 4a b6 3b-72 99 05 eb-3e e9 3a cd  ..).OJ.; r...>.:.
0000-0080:  85 46 4d 65-26 8d 54 66-9f 69 d0 58-7c 94 58 3a  .FMe&.Tf .i.X|.X:
0000-0090:  62 8a b2 53-e4 e2 35 76-b6 31 88 df-c4 91 8f f4  b..S..5v .1......
0000-00a0:  c3 f7 92 ca-d1 d5 26 4e-9c 1f b8 60-a7 28 8a 08  ......&N ...`.(..
0000-00b0:  dd 93 de 35-7b bb 87 35-17 e2 10 68-01 10 06 ac  ...5{..5 ...h....
0000-00c0:  21 49 3d 2b-7f 56 0e 37-7a 7f 7b 7f-fb a9 3b ed  !I=+.V.7 z.{...;.
0000-00d0:  52 77 25 4e-3c b5 1e 4c-93 1d a5 c1-6f 3e d0 ce  Rw%N<..L ....o>..
0000-00e0:  a9 a3 22 19-e0 26 da 67-31 0e ad f8-05 50 13 d0  .."..&.g 1....P..
0000-00f0:  1c e7 92 b2-be 20 ae 52-e8 13 28 d8-70 a4 a0 28  .......R ..(.p..(
0000-0100:  25 d1 ad 49-6e f4 bb f5-5f a4 98 14-57 ef 9e 4b  %..In... _...W..K
0000-0110:  09 f2 31 7a-09 8e 74 6c-0d 01 a5 9e-9f 79 03 1d  ..1z..tl .....y..
0000-0120:  5d c4 78 5f-b5 71 f5 0e-41 0d c2 75-be de 86 54  ].x_.q.. A..u...T
0000-0130:  b0 00 ff 23-07 4a 27 8f-7b e0 41 cc-00 1c d4 2c  ...#.J'. {.A....,
0000-0140:  e0 a5 04 5e-42 93 95 06-45 9f 28 80-41 6e 85 5e  ...^B... E.(.An.^
0000-0150:  e8 d2 e2 cd-3b db bc d1-f7 42 70 8e-43 d1 a9 e9  ....;... .Bp.C...
0000-0160:  32 fa 52 eb-1b 6f 33 e3-92 0a 22 e7-59 57 70 71  2.R..o3. ..".YWpq
0000-0170:  5a ed 6e fa-7f 59 e1 76-55 3c 70 9a-70 38 4b cc  Z.n..Y.v U<p.p8K.
0000-0180:  d2 d9 71 d3-29 6a 79 d2-fb f8 78 fd-0c 89 b3 0c  ..q.)jy. ..x.....
0000-0190:  e9 ff 8d 19-96 c3 55 7c-ac a9 9a 90-b5 e8 6e f4  ......U| ......n.
0000-01a0:  0b e9 e8 93-91 43 1c 3a-83 7f c1 74-c2 d0 c8 04  .....C.: ...t....
```

The three columns of information represent, from left to right, (A) the byte offset (location) of the hex value within the file, expressed in hexadecimal notation, (B) the contents of the file in hexadecimal notation and (C) the hexadecimal content expressed as ASCII text (to the extent the hex values *have* a corresponding ASCII value).

Note that the first two ASCII characters in the file are PK and the first two hex values are 50 4b.

If you check the ASCII Code Chart, you'll see that everything matches up: 50 4B = PK. That PK at the start of the file serves an important purpose. It's the file's **binary header signature**. In computing, a file's header refers to data occurring at or near the start of the file that serves to identify the type of data contained in the file and may also furnish information about the file's length, structure or other characteristics. [Don't confuse *file* headers with *mail* headers, which carry information about, *e.g.*, sender, addressee(s), routing, subject, etc. for e-mail messages.] That PK means that the file data that follows is encoded and compressed with Zip compression. In other words, as a file header," PK" signals to the operating system that the data will only make sense if it is interpreted as Zip-compressed content.

[39] a capable alternate online hex editor can be found at https://hexed.it/?hl=en

Why PK? Because the unfortunate fellow who came up with the Zip compression algorithm was named Phil Katz! Phil insured his place in history by using his initials as the binary header signature for Zip files. So long as it's not already used to identify another file type, a binary header signature can be almost anything, and the person or entity that originates the file structure/type gets to choose it. More on that to come.

Step 2: Unpack the Archive

Open the zip file and extract (unzip) its contents to a convenient location on your machine.

The zip file should hold the seven files listed below:

	Name	File Type
1.	**GBA.doc**	Word Document
2.	**GBA.docx**	Word Document
3.	**GBA.htm**	Web Page
4.	**GBA.pdf**	Adobe PDF
5.	**GBA.rtf**	Rich Text Format
6.	**GBA.txt**	Text
7.	**GBA.eml**	E-mail

Remember where you stored these extracted files.

Step 3: Exploring the Contents of the Archive

Six of these files hold precisely the same famous text, but each in their own unique ***encoded*** way. The seventh, an e-mail, also holds the text, but encoded as *both* an image and attachment.

One-by-one, load each file except GBA.eml into the Online HexDump Utility, http://www.fileformat.info/tool/hexdump.htm, (or the hex viewer of your choice) and explore each file's hex and ASCII content. Now, please answer the following questions about the files:

Exercise 5C: Encoding Anomalies

1. Who is the famous author of the text? ____

2. As you go through each file, can you identify any **date or time values** (e.g., application metadata values like Creation Date, Last Modified Date, Date Last Printed or the like)?

    ____

3. Which, if any, of these files do <u>not</u> show the famous text as human-readable text anywhere in the ASCII Text column? ____

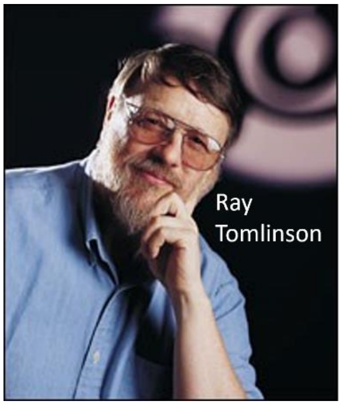
4. What are the first four **hex** values seen in the file GBA.doc? _________________________

5. Do you note anything peculiar about the file GBA.txt in terms of file signatures?

____

Base64

Internet e-mail was born in 1971, when a researcher named Ray Tomlinson sent a message to himself using the "@" sign to distinguish the addressee from the machine. Tomlinson didn't remember the message transmitted in that historic first e-mail but speculated that it was probably something like "qwertyuiop." So, not exactly, "*Mr. Watson, come here. I need you*," but then, Tomlinson didn't *know* he was changing the world. He was just killing time.



Also, back when the nascent Internet consisted of just four university research computers, UCLA student Stephen Crocker originated the practice of circulating proposed technical standards (or "protocols" in geek speak) as publications called "Requests for Comments" or RFCs. They went via U.S. postal mail because there was no such thing as e-mail. Ever after, proposed standards establishing the format of e-mail were promulgated as numbered RFCs. So, when you hear an e-discovery vendor mention "RFC5322 content," fear not, it just means plain ol' e-mail.

An e-mail is as simple as a postcard. Like the back-left side of a postcard, an e-mail has an area called the message body reserved for the user's text message. Like a postcard's back right side, we devote another area called the message header to information needed to get the card where it's supposed to go and to transmittal data akin to a postmark.

We can liken the picture or drawing on the front of our postcard to an e-mail's attachment. Unlike a postcard, we must convert e-mail attachments to letters and numbers for transmission, enabling an e-mail to carry any type of electronic data — audio, documents, software, video —not just pretty pictures.

The key point is that *everything in any e-mail is plain text*, no matter what's attached.

And by plain text, I mean the plainest English text, 7-bit ASCII, lacking even the diacritical characters required for accented words in French or Spanish or any formatting capability. No **bold**. No <u>underline</u>. No *italics*. It is text so simple that you can store a letter as a single byte of data.

The dogged adherence to plain English text stems in part from the universal use of the Simple Mail Transfer Protocol or SMTP to transmit e-mail. SMTP only supports 7-bit ASCII characters, so sticking with SMTP maintained compatibility with older, simpler systems. Because it's just text, it's compatible with any e-mail system invented in the last 50 years. Think about that the next time you come across a floppy disk or CD and wonder how you're going to read it.

How do you encode a world of complex digital content into plain text without losing anything?

The answer is an encoding scheme called Base64, which substitutes 64 printable ASCII characters (A–Z, a–z, 0–9, + and /) for any binary data or for foreign characters, like Cyrillic or Chinese, that can be represented by the Latin alphabet.

Base64 is brilliant and amazingly simple. Since all digital data is stored as bits, and six bits can be arranged in 64 separate ways, you need just 64 alphanumeric characters to stand in for any six bits of data. The 26 lower case letters, 26 upper case letters and the numbers 0-9 give you 62 stand-ins. Throw in a couple of punctuation marks—say the forward slash and plus sign—and you have all the printable ASCII characters you need to represent <u>any</u> binary content in six-bit chunks. Though the encoded data takes up roughly a third more space than its binary source, now any mail system can hand off any attachment. Once again, *it's all just numbers.*

Exercise 5D: Exploring Base64

In this exercise, we will open the e-mail GBA.eml in a plain text viewer and locate its Base64-encoded content. If you are using a Windows machine, you can use Notepad as your text viewer; else, you can use the free application at **http://www.rapidtables.com/tools/notepad.htm.**

Step 1: Open the File in the Text Viewer

Returning to the seven files you extracted from the GBA.zip archive, find the file named GBA.eml and, using the commands, File>Open, open the file in your preferred plain text viewer. Once visible, scroll down until you see this block of data:

```
--089e0118431ed478e705164be95e--
--089e0118431ed4790705164be960
Content-Type: image/gif; name="GBA.gif"
Content-Disposition: attachment;
filename="GBA.gif"
Content-Transfer-Encoding: base64
X-Attachment-Id: f_i9suf1i90
```

This snippet tells the recipient system that the attachment data is encoded in base64 and that it should be interpreted as a GIF image file named GBA when decoded. The first two lines are boundaries signaling where sections of the message begin and end.

Now, look at the gibberish text that follows, all the way until the end of the message. What you are seeing is a .gif image file—a drawing--that's been numerically encoded to be able to traverse the network as an e-mail attachment. Note that the entirety of the data preceding the end boundary of the message: **==--089e0118431ed4790705164be960—** are composed of 64 characters: the 26 lower case letters, 26 upper case letters, the numbers 0-9 and the forward slash and plus sign.[40]

The Base64-encoded content you see should begin:

```
R0lGODlhGQNOAXAAACH5BAEAAMcALAAAAAAZA04BhwAAAKy3wtDOzlJ7ocnHx8/Nza+np8nGxs7K
yj1qk4mFhcHAwM7MzLSysjszM46KisTBwcbDw6ekpDhhh4WCgri2ts3Ly7y6uo6MjDIyMqCensjG
xp6cnC4uLjhhhjNafBUSEoaEhMrIyK+srF5eXnt5eba0tMzKyq6rq1BOTjNZezVcf0E+PsbExKin
p1RUVGNgYLCtrcC9vZKQkDVbfjtmjYN/f8vJybe1tVdXV7q4uGRfXzpliz9tlrSwsMvIyNDNzc7L
y8PDw83MzMzLy0Bwm0BAQM/MzNDMzMvLy8jIyGlaWkFxnJCQkNHNzdLMzN+/v9HOzq6urgoKCtTJ
yc/Pz8XCwru7uzs6OjAvL2tpaQ4ODjk4OHd2drq1tW9fXxISEgcHBykoKF1bWwQEBBUVFTQzM6qh
oScmJiAfH83KyiUkJB0dHU1NTTo5OYSCgtHMzA8PD8vKyrOzs8O/v8fExAgICFdUVElISG5ubpCP
jz08PAkJCU5NTRYWFlJRUREQEL+/v4yKioF/fxoaGjExMRAQECEhIaCfn2FfXwICAoaFhaupqXt6
elhWVh8fH3JxcVtaWp6dnaCgoD89PQMDA8C8vFhXV4SDg3Z0dA0NDScnJ0dHRykpKR4eHiUlJUNC
Qjg2Njc2Ni8vLzg3NxkZGW9ubmlpaYiGhkZFRSckJGxra1xbW0FAQCUiIhEREQ8MDAYGBp+ennNz
c3JwcEVDQwEBAaimpo+OjjIxMRsbGyQjIz8/P19dXTw6Ojw8PJ2ammZlZW9tbTc3N2hoaBQUFJGP
jwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

…and continue for another 333 lines of base64-encoded information.

Takeaway: The different ways in which data is encoded—and the different ways in which collection, processing and search tools identify the multifarious encoding schemes—lie at the heart of significant challenges and costly errors in e-discovery. It's easy to dismiss these fundamentals of information technology as too removed from litigation to be worth the effort to explore them; but, understanding encoding, binary signatures and the role they play in collection, indexing and search will help you realize the capabilities and limits of the tools you, your clients, vendors and opponents use.

To that end, we're just getting started.

[40] Turning all that binary data into alphanumeric data has a dark side in e-discovery reminiscent of those apocryphal monkeys at typewriters who will, in the fullness of infinite time, type Shakespeare's sonnets. Trillions of seemingly random alphabetic characters necessarily form words by happenstance--including keywords searched in discovery. Very short keywords occur in Base64 with alarming frequency. If the tools employed in e-discovery treat encoded base64 attachments as text, or if your search tool doesn't decode base64 content before searching it, false or "noise" hits may prove a significant problem.

⚛ Exercise 6: Encoding: Running the Bases

All ESI is encoded, so success in electronic discovery often depends upon the ability to extract intelligible text from encoded forms to facilitate search and review. Each of the following questions for you to answer is encoded in a common digital format. The first question is in binary (base2), the second is in hex (base16) and the third is in base64. Decode each to find the three questions you must answer in this exercise. Again, decode each to find the question, then answer the three questions.

GOALS: The goals of this exercise are for the student to:

1. Decode each of three questions encoded in various ways: and

2. Answer the questions posed once decoded.

You are encouraged to use free online tools[41] and solicit help as needed, with the proviso that you must be prepared to demonstrate your solutions to the problems. The tasks should take no more than about 15-30 minutes. The exercise has **five** parts (Questions 1-5). Answer and submit responses to all five.

IMPORTANT: Please don't try to decode these by hand (an electronic version of the text can be found at http://www.craigball.com/runningbases2021.txt). And remember: Google and Wikipedia *are* your friends!

Question 1: Easy

```
01010100 01101000 01100101 00100000 01010011 01110101 01101101 01100101 01110010 01101001 01100001
01101110 00100000 01100001 01100010 01100001 01100011 01110101 01110011 00100000 01101001 01110011
00100000 01110100 01101000 01100101 00100000 01100110 01101001 01110010 01110011 01110100 00100000
01101011 01101110 01101111 01110111 01101110 00100000 01110000 01101000 01111001 01110011 01101001
01100011 01100001 01101100 00100000 01101001 01101110 01110011 01110100 01110010 01110101 01101101
```

[41] Examples:
Binary decoders:
http://www.roubaixinteractive.com/PlayGround/Binary_Conversion/Binary_To_Text.asp
https://paulschou.com/tools/xlate/
http://nickciske.com/tools/binary.php
Hex decoders:
http://www.convertstring.com/EncodeDecode/HexDecode
http://www.unit-conversion.info/texttools/hexadecimal/
http://bin-hex-converter.online-domain-tools.com/
Base64 decoders:
http://codebeautify.org/base64-to-image-converter
https://onlineimagetools.com/convert-base64-to-image
http://www.freeformatter.com/base64-encoder.html

```
01100101 01101110 01110100 00100000 01100110 01101111 01110010 00100000 01110100 01101000 01100101
00100000 01110000 01110101 01110010 01110000 01101111 01110011 01100101 00100000 01101111 01100110
00100000 01100011 01100001 01110010 01110010 01111001 01101001 01101110 01100111 00100000 01101111
01110101 01110100 00100000 01100011 01101111 01101101 01110000 01110101 01110100 01100001 01110100
01101001 01101111 01101110 00101110 00100000 00100000 01001001 01110100 00100000 01100010 01101111
01110010 01100101 00100000 01101100 01101001 01110100 01110100 01101100 01100101 00100000 01110010
01100101 01110011 01100101 01101101 01100010 01101100 01100001 01101110 01100011 01100101 00100000
01110100 01101111 00100000 01110100 01101000 01100101 00100000 01101101 01101111 01100100 01100101
01110010 01101110 00100000 01000011 01101000 01101001 01101110 01100101 01110011 01100101 00100000
01100001 01100010 01100001 01100011 01110101 01110011 00101100 00100000 01110100 01100001 01101011
01101001 01101110 01100111 00100000 01110100 01101000 01100101 00100000 01100110 01101111 01110010
01101101 00100000 01101111 01100110 00100000 01100001 00100000 01100110 01101100 01100001 01110100
00100000 01110011 01110101 01110010 01100110 01100001 01100011 01100101 00101100 00100000 01110011
01110101 01100011 01101000 00100000 01100001 01110011 00100000 01100001 00100000 01110011 01110100
01101111 01101110 01100101 00100000 01110100 01100001 01100010 01101100 01100101 01110100 00100000
01101001 01101110 01100011 01101001 01110011 01100101 01100100 00100000 01110111 01101001 01110100
01101000 00100000 01110000 01100001 01110010 01100001 01101100 01101100 01100101 01101100 00100000
01101100 01101001 01101110 01100101 01110011 00101100 00100000 01100001 01101110 01100100 00100000
01110000 01100101 01100010 01100010 01101100 01100101 01110011 00100000 01110101 01110011 01100101
01100100 00100000 01100001 01110011 00100000 01100011 01101111 01110101 01101110 01110100 01100101
01110010 01110011 00100000 01110100 01101111 00100000 01110100 01110010 01100001 01100011 01101011
00100000 01110001 01110101 01100001 01101110 01110100 01101001 01110100 01101001 01100101 01110011
00101110 00100000 00100000 01010100 01101000 01100101 00100000 01010011 01110101 01101101 01100101
01110010 01101001 01100001 01101110 01110011 00100000 01101101 01100001 01100100 01100101 00100000
01110011 01101001 01100111 01101110 01101001 01100110 01101001 01100011 01100001 01101110 01110100
00100000 01100011 01101111 01101110 01110100 01110010 01101001 01100010 01110101 01110100 01101001
01101111 01101110 01110011 00100000 01110100 01101111 00100000 01101101 01100001 01110100 01101000
01100101 01101101 01100001 01110100 01101001 01100011 01110011 00100000 01100001 01101110 01100100
00100000 01100001 01110010 01100101 00100000 01100011 01110010 01100101 01100100 01101001 01110100
01100101 01100100 00100000 01110111 01101001 01110100 01101000 00100000 01100010 01100101 01101001
01101110 01100111 00100000 01110100 01101000 01100101 00100000 01100110 01101001 01110010 01110011
01110100 00100000 01110100 01101111 00100000 01110101 01110011 01100101 00100000 01110011 01111001
01101101 01100010 01101111 01101100 01110011 00100000 01110100 01101111 00100000 01110010 01100101
01110000 01110010 01100101 01110011 01100101 01101110 01110100 00100000 01100111 01110010 01101111
01110101 01110000 01110011 00100000 01101111 01100110 00100000 01101111 01100010 01101010 01100101
01100011 01110100 01110011 00100000 01110100 01101111 00100000 01100011 01101111 01101101 01101101
01110101 01101110 01101001 01100011 01100001 01110100 01100101 00100000 01101100 01100001 01110010
01100111 01100101 00100000 01101110 01110101 01101101 01100010 01100101 01110010 01110011 00101110
00100000 00100000 01010100 01101000 01100101 01111001 00100000 01100001 01100100 01101111 01110000
01110100 01100101 01100100 00100000 01100001 00100000 01100011 01101111 01110101 01101110 01110100
01101001 01101110 01100111 00100000 01110011 01111001 01110011 01110100 01100101 01101101 00100000
01100010 01100001 01110011 01100101 01100100 00100000 01101111 01101110 00100000 01110100 01101000
01100101 00100000 01101110 01110101 01101101 01100010 01100101 01110010 00100000 00110110 00110000
00101100 00100000 01110111 01101000 01101001 01100011 01101000 00100000 01101000 01100001 01110110
01100101 00100000 01110101 01110011 00100000 01110100 01101000 01100101 00100000 01110011 01101001
01111000 01110100 01111001 00101101 01101101 01101001 01101110 01110101 01110100 01100101 00100000
01101000 01101111 01110101 01110010 00101100 00100000 01110011 01101001 01111000 01110100 01111001
00101101 01110011 01100101 01100011 01101111 01101110 01100100 00100000 01101101 01101001 01101110
01110101 01110100 01100101 00100000 01100001 01101110 01100100 00100000 01110100 01101000 01100101
00100000 01101110 01101111 01110100 01100001 01110100 01101001 01101111 01101110 00100000 01110011
01111001 01110011 01110100 01100101 01101101 00100000 01110101 01110011 01100101 01100100 00100000
01110100 01101111 00100000 01110000 01101100 01101111 01110100 00100000 01110000 01101111 01101001
01101110 01110100 01110011 00100000 01101111 01101110 00100000 01110100 01101000 01100101 00100000
01100111 01101100 01101111 01100010 01100101 00101100 00100000 01100001 01101100 01110011 01101111
00100000 01110101 01110011 01100101 01100100 00100000 01110100 01101111 00100000 01100100 01100101
01101110 01101111 01110100 01100101 00100000 01101100 01100001 01110100 01101001 01110100 01110101
01100100 01100101 00100000 01100001 01101110 01100100 00100000 01101100 01101111 01101110 01100111
01101001 01110100 01110101 01100100 01100101 00100000 01110110 01100001 01101100 01110101 01100101
01110011 00100000 01100101 01101101 01100010 01100101 01100100 01100100 01100101 01100100 00100000
01101001 01101110 00100000 01110100 01101000 01100101 00100000 01000101 01011000 01001001 01000110
00100000 01100100 01100001 01110100 01100001 00100000 01101111 01100110 00100000 01110011 01101101
01100001 01110010 01110100 01110000 01101000 01101111 01101110 01100101 00100000 01110000 01101000
01101111 01110100 01101111 01100111 01110010 01100001 01110000 01101000 01110011 00101110 00100000
00100000 01010100 01101000 01100001 01110100 00100000 01100111 01100101 01101111 01101100 01101111
01100011 01100001 01110100 01101001 01101111 01101110 00100000 01100100 01100001 01110100 01100001
00100000 01101101 01100001 01101011 01100101 01110011 00100000 01000111 01010010 01000101 01000001
01010100 00100000 01100101 01110110 01101001 01100100 01100101 01101110 01100011 01100101 00101110
00100000 00100000 01010100 01101000 01100001 01101110 01101011 00100000 01111001 01101111 01110101
00101100 00100000 01010011 01110101 01101101 01100101 01110010 01101001 01100001 01101110 01110011
00100001 00001101 00001010 00001101 00001010 01010001 01110101 01100101 01110011 01110100 01101001
```

```
01101111 01101110 00111010 00100000 01010111 01101000 01100001 01110100 00100000 01101001 01110011
00100000 01110100 01101000 01100101 00100000 01100010 01100001 01110011 01100101 00101101 00110110
00110000 00100000 01110011 01111001 01110011 01110100 01100101 01101101 00100000 01101111 01110010
01101001 01100111 01101001 01101110 01100001 01110100 01100101 01100100 00100000 01100010 01111001
00100000 01110100 01101000 01100101 00100000 01010011 01110101 01101101 01100101 01110010 01101001
01100001 01101110 01110011 00100000 01100011 01100001 01101100 01101100 01100101 01100100 00111111
```

ANSWER:____

Question 2: Harder

```
54 68 65 20 66 61 6d 65 64 20 45 6e 67 6c 69 73 68 20 70 6f 65 74 20 61 6e 64 20 6e 6f 74 6f 72 69
6f 75 73 20 70 6c 61 79 62 6f 79 2c 20 4c 6f 72 64 20 42 79 72 6f 6e 2c 20 68 61 64 20 61 20 64 61
75 67 68 74 65 72 20 68 65 20 61 62 61 6e 64 6f 6e 65 64 20 61 66 74 65 72 20 66 69 76 65 20 77 65
65 6b 73 2e 20 20 53 68 65 20 77 61 73 20 72 69 67 6f 72 6f 75 73 6c 79 20 65 64 75 63 61 74 65 64
20 69 6e 20 6d 61 74 68 20 61 6e 64 20 73 63 69 65 6e 63 65 2c 20 65 78 74 72 61 6f 72 64 69 6e 61
72 79 20 66 6f 72 20 61 20 77 6f 6d 61 6e 20 69 6e 20 74 68 65 20 65 61 72 6c 79 20 31 39 74 68 20
63 65 6e 74 75 72 79 20 62 75 74 20 69 6e 74 65 6e 64 65 64 20 74 6f 20 64 69 73 74 61 6e 63 65 20
68 65 72 20 66 72 6f 6d 20 74 68 65 20 6c 69 62 65 72 74 69 6e 65 20 69 6e 63 6c 69 6e 61 74 69 6f
6e 73 20 6f 66 20 68 65 72 20 70 61 70 61 2e 20 20 53 68 65 20 6d 65 74 20 6d 61 74 68 65 6d 61 74
69 63 69 61 6e 20 61 6e 64 20 69 6e 76 65 6e 74 6f 72 20 43 68 61 72 6c 65 73 20 42 61 62 62 61 67
65 20 61 74 20 61 20 64 69 6e 6e 65 72 20 70 61 72 74 79 20 69 6e 20 31 38 33 33 20 77 68 65 72 65
20 42 61 62 62 61 67 65 20 73 68 6f 77 65 64 20 68 65 72 20 61 20 70 72 6f 74 6f 74 79 70 65 20 6f
66 20 68 69 73 20 64 69 66 66 65 72 65 6e 63 65 20 65 6e 67 69 6e 65 2c 20 74 68 65 20 66 69 72 73
74 20 61 75 74 6f 6d 61 74 69 63 20 63 61 6c 63 75 6c 61 74 6f 72 2e 20 20 49 6e 20 31 38 34 33 2c
20 73 68 65 20 70 75 62 6c 69 73 68 65 64 20 61 6e 20 61 72 74 69 63 6c 65 20 66 6f 72 20 53 63 69
65 6e 74 69 66 69 63 20 4d 65 6d 6f 69 72 73 20 69 6e 20 77 68 69 63 68 20 73 68 65 20 69 6e 63 6c
75 64 65 64 20 64 65 74 61 69 6c 65 64 20 69 6e 73 74 72 75 63 74 69 6f 6e 73 20 66 6f 72 20 73 65
74 74 69 6e 67 20 74 68 65 20 42 61 62 62 61 67 65 20 64 69 66 66 65 72 65 6e 63 65 20 65 6e 67 69
6e 65 20 74 6f 20 63 6f 6d 70 75 74 65 20 42 65 72 6e 6f 75 6c 6c 69 20 6e 75 6d 62 65 72 73 2e 20
20 48 65 72 20 61 6c 67 6f 72 69 74 68 6d 20 61 6e 64 20 69 6e 73 74 72 75 63 74 69 6f 6e 20 73 65
74 20 61 72 65 20 67 65 6e 65 72 61 6c 6c 79 20 72 65 67 61 72 64 65 64 20 61 73 20 6f 6e 65 20 6f
66 20 74 68 65 20 66 69 72 73 74 20 70 75 62 6c 69 73 68 65 64 20 63 6f 6d 70 75 74 65 72 20 70 72
6f 67 72 61 6d 73 2e 0a 51 75 65 73 74 69 6f 6e 3a 20 57 68 6f 20 77 61 73 20 74 68 69 73 20 72 65
6d 61 72 6b 61 62 6c 65 20 65 61 72 6c 79 20 70 72 6f 67 72 61 6d 6d 65 72 3f 0a
```

ANSWER:____

Question 3: Hardest

```
Content-Type: image/gif; name="unicode.gif"
Content-Disposition: attachment; filename="unicode.gif"
Content-Transfer-Encoding: base64

R0lGODlhgAG7AHAAACH5BAEAAP8ALAAAAACAAbsAh////+/v74yUlPfv94SEjGtrY+bm5lpjY729
vc7W1vf/98XOzmtrc8W9xffv5t7e3nuEhJyclFpSWpScpaWlpUpCSsW9tUpSSntze2tze97v5kJC
OlpaSrWttbWtpQgZEObm1vf3/zoxMc7e3q2trQAAALW1vZytpYyEhN7Fve/v3sXOvQicpTpCSjoQ
SnuEa3u9a6U6UnM6GaUQUnMQGZzexa1a5t6c5kpapd5C5pxjQq0Z5q1atd6ctd5Cta0ZtXveOnuc
OmtjGRBaWnveEHucEHtzc0pazkreEEoZhEoZzkqcEKWllGs6St7e1t7m5iEhGUJaGdbWxWve5ine
5imc5hlaMRlapRla7xneMRkZpRkZ7xmcMRnvYxmtY0re5gje5gic5hnOYxmMY6WUlObF3rW9rXtC
hHsQhKU6GXvva2sQUq29a6UQGUpaWt7Wva2tta3eOjopGa2cOq3eEK2cEJyUzjoxQjEIEEpa7Orv
Y0reMUoZpUoZ7OqcMUqtYxAISkrOYOqMY2taWqWMa1Jrc5xrhK1ChKOQhM7O3nOElK3va4yEc973
Ut73Gd73jDExMUpSOpyU7+8ZjO8ZOu8ZY+8ZEM4ZjM4ZOs4ZY84ZEJyMjHtrYxkpMd73va3mnHtr
5u9r5u8Q5q1jEHsp5ntrte9rte8QtXsptYzmnHtK5s5r5s4Q5oxjEHsI5ntKtc5rtc4QtXsIta2t
nK3m7+/OjO9ajO9aOu/OOu+UjO+UOu/OY+9aY+9aEO/OEO+UY++UEBlrEBlrhBlrzhnvEBkphBkp
zhmtEFKtpUpSWozm787OjM5ajM5aOs7OOs6UjM6UOs7OY85aY85aEM7OEM6UY86UEBlKEBlKhBlK
zhnOEBkIhBkIzhmMEFKMpZS15qW1vbW91mut3mvvpSnvpSmtpUqt3krvpQjvpWuM3mvOpSnOpSmM
pUqM3krOpQjOpcXF1ntzhFprYxApWqVjY5TFnO/m5nOtpebe3kpKezpjWq2Mpeb3/3uMhP/v/wAA
AAj/AAEIHEiwoMGDCBMqXMiwocOHECNKnEixosWLGDNq3Mixo8ePIEOKHEmypMmTKFOqXMmypcuX
MGPKnEmzps2bOHPq3MmzJ0QFFYFWnAUgAL+HARIm9cmUoAIHJI5WRPCA5IMFSzkqUIEg60qhAPgp
ILEQKFiCCxYMdOD1oZQNkO6ckAglAIEIDhEcQBgAA9mKCxAsfPD3YARIkCS0lUjYpooKDiwWEHww
AN6MCDDg+/hAggaKD+BcVIAAEhQoIlZ0Yujg8J0PAgZuIjBwwSB+JKo+tIBhaYAEC8oAxbcgQdG0
SVUscLJaBT8VV9UqiP55YICqVzcTfKAA39UH2QGA/1+g20naolelMGAQIEDazfhGWPi8qcUDsQVn
VUgq1h5y8cCpkAhW0/33QFJO2LWfeOQVJQVW3zmFT1IHKrCAFGJJkYh21kUHgANPoDfdgwJihU8L
aalwHIH4JJKAA+QtddYgETQYQIgHIiBCImepMEKDAJinW1pVQXceApx8dmFbKuBolIYIArndCmWs
IIGB5wXghBRPuIfVcSvgQ0AyARRY1W8QSqkQARMAoIAcT3zpnhQyOsVBIm4CQAAGwH2ogQOQkFBd
```

QyscIABZZkBRABQLqCDBBZyMwEkFFRSgAAYVDAKFBn6RUYEEkCBg2yAlnOBPnghI8FgBUVk3GQbJ
gP9awKcacCKBBHdc+GkFJJjx6SYiQEHCArYqNqsEFWjAARQERFbQIGOKFEABHFTAgAa3ipBMARsQ
4MSttHICDgBQOFEBFAKsUICthEFxZQUYODbQpXNVgEAEEkBQBgnrHpCVAmZOUoAIuVXATwMcdEZp
pZBE4IDAlSqwyboYaDBIC3dJgIFaBh3wqX17AsDJCS/gkQGeAl16xyCQpCVBAQU8YMEFyIajH7IR
rNAbATBj4BQBtNnrQQUELOBBARhIUJVZAdw6WQvIFgBABJwE+gAkFVDA78t6VsABCedGgMAgST8w
i7srbIBBbAxhsIlAUGQAic8TF5DBYmQUkFUEGxT/AAkcFkiAABT27iYBAxAoQIYEADAiAL4CeXAA
Pw5IwDN3UPCDAThkZADALBmQEXSdZjDwgAhdpVzU5oxAILIiACRDLG2OD0JWqiR4NnWbDyBAwqac
tAnJA1YixIFoAuV9lBy2gQOoexU8wIFxIiywMQAtaGABAyJLAMcBGcwySFEbkHBWAIyMeyUjMStQ
QQYkdEKZm88kM/XGjNv2QDIjKJA5CRhwAiT48QAaVUAAE2DUIMygp0GMwCkpk8BcOPE4121OegVx
wNpEJoBBFAAcf+OAYCJQABICYAH24oRqKECADTxQIA7YRGwGgYATKA0AHGAA2OaSMg/YL0gtuM4G
/34DDk5k4AEVeKAUEECACtjmgQrwgOdgljsImDAAkADHWRLygjY5IHNSKMACOhGB+nBsXgVoEwDv
JTUEOHF8V4KIzpZCAu4x4i6e+1w8HHApDHAiRJBAnxnIEBsyFMJXGdAbQcxgqQRgIB4yUgAnwsGI
EyiAAHNJhhPEBYAXCKAC4QCAlUgAgaREIDazMQMkRrA5AGwAHqmaF0E44brk5XED4NiY+7QkgQS8
oCoVMAMGBHOHB5ihEACwGwLA4YQTxEMgC4CAIt1ELwBcIJQRyEAEBkEABCAgREI55v2KtwClFUAK
CoAEAIRpLrZgAALQ8qYTrqeAE4ysIApQwGSmZv9F14OMiVJJGSPaRAB4xsN3/JDDA0mgzdg84A46
M8Md4GCG1AmFADNEwFgKYMNuguOFQGECMoPkLyQugBERIAADEnAbADAgAiewV8wEUkcASCAeZgDH
AyJQSylAgBP/QsgGAaDOBUzmDiTwZlsCV5Vw1CilAEiAE/fCgTM2JHAnoAoJ9uI41SBgAghowSkv
kIADEIARmdscVGdRgA5AgqNSAQoCCjBPRhxgUCpg3QT8MVAAVCABk8odOPBFgkHOIzNA2USyOICB
RD2hlatcASQSgQD25OmEkDjUCWYhgcdVQAVifIwCVvDZufGNU4b6QABUudNBgGMWiYiAz5xwl2T/
zYsREogAowiQuwi8AAMrEMC/LCC1bKoAEicYxAWwdSA5AACACYACBAjgmc7eq1Zq2UTuCOAB4c7r
YiRYJdh+B4cHlG9QkuRVy2x4r74cwAO5jR/VMqAzDVQgAnBo1UA8dQI8cG67jGAAAmahGzdJARIe
iAB8CWivZPCLAd/6DME2sQGLQYACCPDAHTTASwGPYBPPlIIAxmTUAh8EA3gJgHPDuL9unqBOl6ql
bALV2RWMj2gmXogKZLuxBIjGmwDIDAm0BLSqRBMOmwgAAkawAguckAQRcHBuAXCqIMEhALMoGkEc
QBULqEVUz+UUBCBgvrG8YC4JSJ0KDvoAAWwi/wJKfmAE8MGPTZxgp215ANAEswKg2UMBOp0aP/Ax
CydkYLoPVAFvN6EAf/BWySgWpWAcMAtGhJIgAUBgBDTwgEqGJQLx8EBAASAF0ViAgeHgrQcCUN6p
BckMATgBCRhhHCcIYGOrrQpt8TLUgZiBBNwtCrBhOxYB6AYojMgAAcyggFj7hY8TSB8AyLA24RrT
AWIiwAoKEoAJRCACD+AwnOvpl6M464TxyI35LKNkRpAADhqYxamkgFG86JkAGlCO+UgpAH+sgDKi
W4A97vwhhJiBY7HRwJAV3U1uk0AKBVkBlCNAQLIs4AVWVcipRq0QsGzRIAFO1bhkORFOqg6CHf9P
eT6peZCPE2tQly1JlfH58ZQT5Cgrb7OJa47ytWCyIYTkOD5JTnSE+EMocQ2KVspCEp5fRMk10kij
Tg4TGC1GIk4vekSEsnKVOCHHCHmAE3iedZZ43CFlh0naMcL1oVO95w0R+kTOsvaFyN0kbUeIWWLe
FI7c/e1DT3vdT7L3nQy+IWT3SdYPjxLGdz3vIkn82xfvkJn/RCN0P3niay75s6fsLHE9VXcyDsEA
rODqBhGKlvAElJk/5bnz+/xBku4mpcLd6UcveFhYzncIelz0vTeLUFTQgIR0ne+Fn5frUz8Qjm+x
8J53kwooQKdhwX3oarK55kfrFahEABzL8Uf/QBv9cbOIvspXkdG5r/8QD1Kd67mXZTvdhHOtz4sB
igrAg/ix/I8fLebxJxD4UAAgFSSgUkyXNXN3IXsEMV4x53JSUxC9dXx5EoADEYCrdUIFwCFUpnmi
RwZt0npuYgH2oHt85yxS4C/zUn6Wd4EwFHORIYKX5Sg+AwAWsAF3gBflJALRw10gwAAx+HZ+5FSF
c4HRNxCnpDpjET2iVAFy4DMDiDVLA0MG4ytQ0CZKtgk6c2zIJxAtSBBbeBAx+IUO8S2Z8yHX0R4T
IhDQkRQBoALtsTT44BzYRmfNNiFJIXpDAwAjcG8q0oZFgW1JsRX2YEjoMYjQQTnu4UADgQ8V/zAX
BwIA3jGI3tFXAbCGn8cJBHAUk4iGmLhOjBMAGrCGnPACRwGIb9glb3gg1wEUaWiDyTJoAriGcegV
6IOFE6IAYYQ8eFgUKmAPlGMPAZACycCKReEdaKgC2nEdGpAUvbiC+KAB5vaGR5F7C6AonxEJ8FYG
j2EGz6EC8CMFP6R60Ug5vNI0cyGKSxMA9qAB/tAkRiGJisY2QOEBHFCMANAA0KFQjPBMaUQQHuAz
C4APvpIUX+VXVsUWE6KIbhIAbJGG12GDkdIevhgiRdEeqfh3B1FJ/3gYFSACciMBpDEIEuAvjJAY
pAUAm3AHLUMC1bIJjlIBb0UQB2YG/uCIUP9gOiyjN0kjhebSAu8TAJOyH78jAX0GCRuQRAORAOo0
ELOwAVCjPUj5iAVEkgW2AHLQCbPgK4nxBIcRRwJRPMgiAgegAq+RVRUAlHqGLCQQK3JwAKACZ5iy
SpzwAbZTACpAAi2wAYrhQVjzQm7CJm4kAcJCBlDQCQmQOxoDAAdwB0ozCJ3QWFADCbRBABwACeGw
AJCgKUM2KR+wUoMAl1Y1NImhaFATe2TACBggGpxgOs9VAIOCARMgBRJQfw4wMA2DAB9AmVDQAr4j
ARxAAI5yB0VTALdCPJQiAvTYkBKAMkUBCU9gBgeQKs4pMhMgFG4UjZxwFJxwGUCRNDJJGBj/wA8e
YDp8qSjJIAcIsALJ4ISlhC8V0CaEkzE31HuBJwEkoB8AgAGeEwkTMC1WAgELUD4MADvmMgK2ZYML
QF0B4ERvMmpDA5uEVBTeVAGJ8ALPhAEnsEERwB4LsAJyAA+Q4wR3IB5MGDkVMC9QECLBE0eDsKEF
sAIXwDYCMUz4QEM2dQKMMFIDETgKcAGCIQJ1hhfV0zhWJDWOmBbJYgH2swJSpVMpGkZPoFDJlFyw
I5spE0MTEDgagEJ98UByIGsXQFqiYUKidAf8QFoWQiwQYAEGI0Ukup+ZkQFzlUeS6EQA8KLJ5hQB
UFUY4DPTwiuyFVec8HCMg4RSg0SghSec/7Ac8amb0jMurTlX8RA81lkQBQRFDSSJkHlXBBEJl6ZP
DnUZPDVLnuM4AblOR5REbkSehUCb/cMocuBUxQgFD8QJvSFH1gIBmyIAM4QV4qIx37agZIFEbzEQ
KFY2B/BAq1QQ/NBZWSZK3YkinoZitrNODGA9J9AyoGOD/qICjOgmTLkWq9FJn2Qcj2OcUGZV/8SI
ayNDESeScRQJI4BRACAHSTEB8UJx4PoEUlAI+leMGEAG5YNBZXAA9gAFApEBExCjl0KjKjkBwqR/
QdSo/uNtmxZHdyQQtElqEkBvE9OmUiM5VyMABOhB33ZpHzJELpVSEGshzOIxL2QGyWBXWf8hm5OR
UBsEruW0bYMQDlZCABSwCZKCJx7kbTY2LhEAOzO4rAJhBoqhJ/7OTFUWCYKhAB3aFoQwngPhaaBm
Bp5TOt9ier1ROtbzGaECBScAU4AiEBpgsqTXcvsZQLMDNDiEANhFSwnQAQ+goUD0BPc1LFDQAJbJ
XFg0KB5ABlZyAhQgONS1ALnit2sTAdNZAfGALwe2AJ0DAMcFbJCAVxqzAuCWg3BwJYPwAiSwAScg
ABmwAHAAmCLzYkizAtWzCUw7EKmiABIwAg7Qg7RBNqliBgtYQHEyCNDjK1J1ZyIQDtLZoALgRs0j
STR6ixO7AnegT/HgPhMQGE2jFkODABT/VZso5CtGFQ/lBAAeMAhOIAfJQBskFBhnFACDcFBOxCZO
oYliVwHg0E0EcABI1KbDMjId26P6S12xoxaDcHqDIAALwLeDYBzvSxUYMBkbALF5WT6rlVkjUCsE
WACwIxTXSgYE8yXICrEbxLoL8DUvMwIcwA9gu1qDcGDfFj3B1ADhoGI05VaFsRCUxjFtBmQRMAIB
QAJCfGuzsFqhpGgnFC8ECTRDNgEhgm8DoWga6g92sWkoRQD2YAbbVlEAYMQnoAKzQQBx8heZ8Wbg
5LZ7Ily01UoB8AIEQAbh4A+y5V0DYX16FmpBFnvigRfgdklbLBgB0A9+scdFAWf48J9i/4y1d7QA
/pBlEncdEDBMU8Mjy7RI4eAEZCEmqgoeqkkWxuYmspUbeIEPcAZAAgAHzXRCWYVUBUAAlhEvXqFn
L+AB67Rt87IJ1ZFUs6ZleRwBKgAODZBpTgFABFAV4PY5apEAe9IVBHbIa9MeBSXHaMEz8UAYr2y+
OxUvn7Fy1MYPbLUnJnYBVkUC2+Y7ALChMGUPDqPJ0/Gfp4QBeKIz2qYAtcREdtwQrjdz/ReP5ScU
QeiCTnFuzUZlQCEWoucbzXducYV0Xsg0awEW+GF/HifReQJ5Cah670cQ4ict87J+Ex0WFEhNr4dp
zfd28Vh/DSkQlENNYGMbriMW5Ud7l/9F0K7IbS+odw7N0tTEH+ezgifoAPgRGe0B0bJklh9ydF7h
BDNFcmCRFVVG0FT2L3XiJuu3doWn0gzodjWXe2TIdPbXgcmH0ZOXeZyndyjXdlX2fA9t0Ayxd9C3
ghbYhc4KeGXBRLD2gOz3EE53hBtt15OHeJAnVzBHkwtA0yaocWTNeIEH1ErogW5iect3OReokYjt
dpRdgZlH0Y/de/EngmdneWeXeZ9ddFsEfKUNFOsn2ZTtLI2meW59eZeVd86y1iKd1vaZdEIx17O9
0cJ3e1Um2lttn17ofFUmd/jxeH+N0Vgt1q5XeOenhApw2WXNclLBR4CtfNm91cAHg0b/KIPPN2pd
Z4HC3dt7PXsah9bJp30uuHK/nXK27XvSLXfTzdfG197x/dq4Ld2ydNbsTX9lAd5eSNxvnaXywtb7
nXqvDct5QgaAiQ+5sdeuHdho7d0LsMOYTXRBzHxlsX6YTXZc5gG+o7kQJxDmUeHEXXMOQAYcg+By
WxYeUHWGsQL3dN4UsdmXJXFOjb4L4OFODdYdl3eb3Ro7JxFOAGfkghAO4C2Id3JyUGD4ORBMXZ3s
p9xE19d3fKhuMkj3PRAbEHuCJy3cQ+AMQW2UWU6FYQEk8NVkzeFFcSUWcAITnuFGZxCRpjqnssBU
ftpcbeOzYD5k2LgIQUGy3cdgF9bXBpdPx6y7fP8cEa0xN8e1AJtQFaYMZx4ACa6ZMhZAAE5GW0es
OJwrGJ87Ft87ECkQPKE8xJPu1iuguUO2CZPWAZugFhZwL4IhBR1ABuZzQrAuEAggOnbqiHiK12L9
FN/mALYjNh/iAQSwGbU+ZJs+aUwk6XgQxjtVyr6DAGIyZDp9NQIRPBHAQIHxXAws5bcuGBbAjd9m

kLCODxbQAXCmM/p0B9FydFCRZA7wb7P+IUzkZOfxoehLALDejycg42GJBxDQjCm1GWZgAf/ZiBFA
BkkhBdrlAOwMZ/deFcOihU3ziKTG6SPIY5g2tH3xOILhD2Yw67pIAhJPW7YcZC4mB/HwJ8ceADb/
s2rfVoIFp2cSP23o7CYeIPPq4m1qwWXHvHsNoR/99gFIcwAKEA8TszEVUEpOmVveckCDwB5FhZcb
5ikC4F8DAQItYCiBpJr4ogFCsQ8NExcvIAHMCzT7UQDOzjxkQEaBIgUV0AH4KVknIAfc44qDUIz4
EgHB+XlkUzSDkIO5YgbxwAhkcjEUdRcbUAaDHwFnk5nQwp8CVABSJTaFXRBILTINkyuiA7UI4GQC
eAcalKLJOGcdtDFfswCJ8IQHAEk16j2zBAFkEESDcEB3EABkkAwEC++xkVJz9TsLSkZUKhAPIAcC
kAhIO0RPOPvcHpj4Ukp34KtzxQACsOnapCoV/9C+TmQr4YD3uAO5EyAHdnopZpUZ2V8BgHMXkGAP
GJCDD1BCEgAOvdVYB+QoADGBQIEHFQqA21QgXCIADZ1UiFAAQwAoByw0BCBFhIAHEiDK4ZfQDCQV
GE2exLgCgwYFkAAsIKghHAEJCgowVNBw0ImGsyQAUNHigcsFK0U8OQggacM3BUYAqEBCggAEQjEy
IgAgggAAGE4oWDGLwwIGCAAwEhAhKxkIjA7AnMCJBAASDDAqiAAB6goACCo0VOAACkYJZrqScLCA
BKQAnMABUJAC3IETGeYCKAhg1iASJCokGKQBACcIDFHexSeiYQE4ZyeQIbBiEBzRGCsgGJSMRP+B
J/iKZnAi4UQA3PxW/GxIhhE/jMEj+F3AybAIJ2MBbCKA9jqBCJs0VNCAYQIACY8hQ9XgxCWADQsG
mcf4wAKGeBEwNERQIAAAfnLgwDkgkUgSUYCTcLZSgJBkYLIvArrsaggfDh5oSDsC9FrBgwqKcvCl
/CK67AGnVoiKjALyY8gnvnJSwETMKhiKOYwcGGQEBySABwAO8oMgnA1PAxKmB1p6qQASI2AkmQAG
WcCkQcbzR6oXhwJgBf3uGMEx8i4DwIkCmtzAMwIQgGO/htQCQAABFMCABEMYiKCCRFjrKq14OoyH
JgRIeIITriK6C81IHvMLowBAamjBriZgpAD/Ejpp7DGfZknmBE7GU0ADl2aJCpwFrHwCgAA8QyCA
/fxBCZ+/AJjNtRMYCcCJDJA784Cd7hCAnw3sO0CFBwioQLaXlOwJgpwaWoADAsBBQIMXwgEAkmQf
w0oAvQgQgIwD7HNNRy6jfUC9hjYcBNoZd+UkAwEywE8CCgGQg6oOFPgsgALM2MRBAW7j84WsIoAQ
qB+7+koA7hwVYQHuXhJBqgkK4Ckj3qSAhARwGkiEASlEPRIjnzATaj1DJSDwLajAGQSDPc0E8i4A
8JGg18FgQkCCgZIB4AB3MZIKAwIewuCArCQoIDcVINGAjAoYgQI+KQxiQIIABBiEgBNk7Cor/0YU
GQ2cdAlor4AKIIAxgk7AXmCBCgggQL6xK2AXIzKk3USCoLlUYBAGMFhAAgQKFIARBgiQw97HyLC5
AnB6juDo5wrIYExQu3pBgqIinrGhBwZjdS4M7Cwq6Luc+GDMEszaaJAvCQj6OH4sqEBGvwg4ls0C
uIsU3hEQrzztDRipIC1I1g7PwagAA9feAjjRb0eT/NlVAuCSuV2lA/geyGp/RHhABUfNCL4gCOJ5
QG1GWrivIQfqxmCFS7OOYBABpP28SoiS2QSBZPy0h+gyrmcEOJxwgDJoJR7CwcjTgiYAipzEAUU7
DpgWcAIJYKVlJznWAibgAGglJjFm8ECTFP9DoWMh4ASGSUAEzLAfFUzADAt4HT8UAAcSlGpGC4BD
BCikAhKcwAIsAgAInqIBCi1AAwEwAwnCoYIMQIAETYqfGf5WpQiQYD97QkCTDDWLJmHRZQE4AQl8
I5oEPEEDNXzMAkqiAARIUQVIPAG0FBMAKcziBA9IDHNWUMOXaPE0WlwAhZwwggd0r4csywkc3mhD
J3RmBKOaxSObpAI/stGK5+FhDRWwgCvKEAET0CIWNcCJMd3uAaFaQahKmJMH9JBCK2AZskhghgS8
pDMBaCUZAuAAEjwHABbYTyAVwMcFVNEeAEhiFk3Cwwg8QQpDcoITkFjDWL2rjXCgEAJmIcf/CWhA
BVVcwCb384ATIABrAEjhCvuCkkaWbz/AJM5zjnVBl9ETg/bE5zzpqc/T2MQskNkE1/qJT4IW1CT8
NGhCUYJQhRaUoQ3ZBPUs11CKVnSfFsXoBR+aUSBtlKAebagDsBZLjpb0PCZF6UU5qgBTkTSlL9Uo
TBOqT5DG9KQyxWjt7vnSmiKPnz09CapwulOUAjWnJT2nSY26ONMI9TRJRV5UDSpUhEL1pkNl6kGB
mNKHzhOhCnDqefxRO6WW1Ks8bZlHw1rP85T1oFKtqE5PCtSzwhUlVs3qTF22OZ/G5ykoQRVzGOrR
s1qAQEQFjBT+SdWrXrWuDnwASbcq1sfm/wRVvpHqLBCZE8HaVKovjKpO5cpW0eJnY+pzyJgA89WU
aJFFx3LASTT71mPJ8KOQyUnagDTWnOCDAN9C1QOm6NODxnaubyXucUXFScQ6AAH4QK5BjStVfm6C
hGPFCAIokNecjKpJO5XRTy2Lkfq1zAPQklJD+CFUoTqApilBwHSXKYAFOAEcZpIvQ/FhxbA6YQIk
ldZJY6vP2GItvzcdhIMIgKqcHBgw2G0Ig0/DgE2ghBNluYttTRLCrtwJMmsV1ROj1SSwvvc8eG2s
VjBQW8gkFXGPgTAypfbhecoXSDqcK4SdiqoA4PguLCLDkCRgrrC69IJSiEADXiZP51rAiv+kI0At
c+KBCGhACtBaQQQiEA4QNIAEK/olJFbSlwj4kTgr84BZFIDkJjFiOKUKABnmElwOSCAslpviN81p
qFvCeQFSYEAFMGeSB3ACARgQNJ/JwA/nRmA/ZpBCLyEBAVP5OgnxRYAUKADdSCygA/jZDxIx0yR8
XAQBjsbMcwKwAAtQIADzGdUHSEBCMlNICl4mcZab9AC+SKFJSC4zBiIAh39iJl5PCACVj5mCOXZJ
BF+yDwL+6QQCnPaXBglAsCZwWguUOSNNckI4u/0UXb8k0uFQQGRI8Dk2+hgoFCCBBiTklPNkuQxm
YMDiIhQBC/AjAA2AAydvCZQIdGABcsj/QJO6LUgnx/OvydZAIL2NKjPIQREpgNisIdPtUBW0AVVj
gAYkEBEy4KMTDJDIE+RAAC0CbNawic4EoHBqSBAgYQ0hUchpwgi/aQ4KnJBAzRNWhtvBaBOBM9bI
B9KQAHDAIggonQg6kO0T7ITpGGAAJ9yzAAf8FnjjQYn3wsGIC4BdKZGjss+aF5U4EaAxNDnBCuIR
AEignAH+6HS58BGglw3iARFoQY8FYAaJHCDbz3lIynC2HxJAAdXXgZME7EGAmkPiAeCLwBO6cBwA
wKYjcYIDI2quOM2JOQEYKMABNSC2CezQICmEwvzSJgGpmItEhQjABtYmgSdQcBMVgIMH/+4zC5bT
fgGHJgAHAoCBmm+gDBa4AOHiQSIRiyplBfhSBRigRfAFDgFitpm94pTpTqhMLeFPWTw8sIGEw4Ez
MGLEHU4Aon/igwGFrgAGJsowCARnaSLgBxCpACMDEp/ZOBEYAdYhkb4oAKh4l+uoEX4gg2zJijt4
gE24k1Z5Ga0DAChgDkbwMCfoBBU7C67gtRZYgPg7AGR6DxJoAfJyEBIYBADAFk4pCkrLiAFsiLFY
PgTwAAjAGqx5AL4xg/T5JYIQlUiYANgZgdt4CU4YIveBAAvIALpbNd7gnrwQvgZjBIH4gETgAAIp
AxcsCM0qCG+CBBlRgTu4C4gIh/d4H//O8IrOwwAz+Am1IIEqVBm2KUHbcIIH2ID9uIOCMLvRmAsQ
rJNNeIsEawgVwIDH6DTvqRmzsMN167zYSJixaiXL4QSe8AoBuI8I6AcS8aMVYMMAWBC5wAir64tB
4AczWMHyGbkFBIoU3D6RYZLGiIcmHDu9mMRjUQEJGAFHKZCIoZERUICJChZOUIQFuAM/sid/aYhl
6AQKKKfMsAAJ0JW/ggwEEIEJmIUpHLkCwAuuKYAOYSKzgIL9IIAQ/AsyeIEAHLwKmIVOGAGeO0YN
0TIuEbYHAQBCiIfF6CUtkoINkJENiRQzuBOWOgnvSQASwMFWpJAAEIHf+ooKeIoUyID/yBKBCJAA
CFiBKgTA4CgIWaQMngmeBLsDBaAJAbgDFQgPvzkAZNuAd3GAEQQMEYiHWYi3NmEVAeAEO9CMDNAf
rYgNiCBHDJgFPLGNQAJAHfEa+BiN1jCYs1AE4Omlv1IBDMPIxvCMLkqGWQhFY1mA6CMDDhAAy5lD
tLgQrVCEACCRrKgSNkzGyyGM1gAAr+GHmimIOJmAFVhBe+GE1lkVqOCkAjiAXvo7TDmOuMGHm3AM
NuEKoGCSuoQKOFieXqoNfPKAkROAMhA0MkgEEZSxvOGbhqCyc4yAgBSBZBAAe9ga8ugQ5bEXYTOZ
uTQgqySBHYGCROiHtJCA4BAAPkoO/8uhIKtMALYzDRVIGR3iuTgBBxI5gTIYhFgSkU+5gIjxnvoI
h8EkATjAO4YAgetJAEiomSnMgKOpEt7jHhrZAG6UAiiYPihgFO54gROwPALYhAywwyewB6RRH0gg
g9ogjVuCGFY5AbGgIDNQD06BgPVLBgLgBz85i7gBgE5rDGFLhkIbNEbQBykggBcYDaoQjmZSHwwI
De5hTnAYiAg4gOicCg7AACnYhM8ZzAi4AJjwxMEbuQp4gQYAGLlUgeDBgPTsRLnZls4oAH5YhgUx
T4iwAJwZIFA0vOBhoHRZAW0hgf/Sjm4jgBV8GWF8j7wBOxrBgOAAkxUYiC1NKIXkiv8FiAdGSIAA
aA18MAyFIUi24Qdfc1Gp+JkmMQNuLIoRUAG2gY9R6aMqmVMtY6AVIFQrUph4KDYVcKJLewm+UAkC
4EZqg4AhYRuNUwsFYCCTIA7RODqMWAABeIHIAsW5iDfIAIcXYCYBSAR8MKdZ8LdSuaS0NAlegqKn
uMAIGA4SeIFSgQPmWDSeEYCOm5r6YCOG8ABktA8SYA6FjIAUEg6JWABy8yMSOCZqY7m+eECgWJYy
aJIVANTP6biXwIBWCoBXHJLVvAziIwNOYptHwkMNQIACQtdhahM1YpsHdAKseArDChTuEC7MAE8z
0M9HwksLyBZwUIDTEwB8oLZSUQv/R1sAhsCH1XwXBVghMxgSBDCNlwBJBKAQK+qx+iBAnGq7F8Wq
mP0xmSXVA6VZgqKULx20m+XZnvVZrFIAJygDKXArs/rZlGqpo0Orc3UCEFPap4XaqJVaiiraqbXa
q8XarNXajqJal9qlekq3AHittMKg2pmsrd0nU7ktru2qlrUnt6UotbWosEXbknqmmE0MsbWhgZoF
e1gAJXueKhmOk2gbqIozaFwttf2qBTgmDIqttEGVTJrIzhChzhDZhiKOca0oJxA2FBsoyCCBkqCn
U7OrfXKAuYMMKWAI4kixi+LcwbUoJIPblVIBRrukgtJchcIADHBalGLOBSCDwLug/7bZk9PAt83c
wWJbJgzAS35agcm8qhtRXpOAHR3EPqLpigPgBBghAVIqAbxMKO/iqIgYkoSKrRX4AAttGRehKKvQ
CqmJgGRIt4YSACRsqHmiwnXFqUJlx9y9IFe0KD9hyapBJoDhBGTbBAIOnYnovNsJgPDgmx47R6ES
kfilEMIjiAIZCCiCS3MBAA+4HvmoQu5AoOvLCinQP3BwgKApgAoLgDWVkLM5DXsJBwtICNSFjBTE
AH5QT6RQG7MA0JchluX1SS27HcwYCIUQWRLIHpYKUlGJiAh4HQHYPjLwisuqAEgIBzKwj8srmkfC
DjmkN4zghHjgABlSC0ZcN9IAgP8TsAuxZOCG2A3e+LugUTiJYAz1GQSyGRJ8W4kV9orlq5oWWQEN
2IT9wIcNkJbGEBq6IACtywvOeIGbQKbdFQ0ZxD4zGcLC4A8AcICIiIfX0Zs2SwhX3d0uKYDyGCbW

4QQocBfNUqLOS4CiuJ3NgN5ZKAG+ERHaSyhGgAAeyg9wCF4EiIcCsFO1WQAP6ATqO47TO4Ea5aSI
SAQ7Yw4VOIAS+K4XKgABUAFZO2URSIQa9ZhMDclsewGpAI2D8JsFkKKVxL4FkBYr7hsEqJtwoq5B
yLIWwMwOwTlXdABImIDxIxGAxohtRglhQxGikYpiahNp+RKgQwAOCAcPuAlHQYD/A+gIem6BBiA8
EkBGyOAEwxsEPJOAHKqARDaYiqGJGQEPLYGEgOREv+EAOBDL6ECAdQGMNCMABniItBhAUnqUdTUD
KOAaKWAStgmAD9Ch3T2OMErlySwQqSEN3HgVDgCHglAEl5wAATgALzk14FAc/auNRgKf2pBBODCD
RAYHj6SItBiEBDC+UaKKDUgAoDs+Qcu94mEVKoMEU7wXOfG5LPKZqkBcevocCb4DEgjHlxjAiJAD
cKDoEmQ+DJADkNwAtyuAO8AAKIiYxxwb/MAAEUhPl8iP73CCCBAoBRA+9nOf76CQxDwINpkAfGgU
KAiPx9hmDsg/KAgHfNspB3gL/+ygQQ9rCAtY0pdgnXIBwTgRDVXhxmORA6JZ5vcgGK9wgAHkmwP9
HEYov43gBM3mbD6OF2HECGxhFcMAlA4cgRoZoABITo+JCrtBjwJZTZ6IAMuoD0hIZZyBjCJMhuAs
gCO6gwQ4YwW4A5YhA4T7pQ9ggDpTz2BsklkoBFYJYhrkCkhp48HhCuYcgeDApduAgqgJlr8Iydr4
lUHohMb15YYAH0+2nDvgh+tMkizSQeHoQqDgvWxbvvHA59VkFdw2DyJJFtFgRZDKCT/xDNm2DzZK
hohYksWxC5+xD3/zN0csAKyg8gjJvunkgAY4AUXAB9LWj05DbaaTAw0AmHKOBP90K49B+Bth2+YA
6ITw4Arh4AQpLonI5gd8EK0DODWuIAD1xY39cC/ABMzD5AlG2GFRga5xgTPyePMIIAMogISv0DpV
RAtuEVtQBADR2ApPDoBZEN6zYARWaQ3EYalO6IgHkAKMdoIX3EEGqF8O0IALKJWDqB8MgIBZiAcr
PiIzcQLnM4NkcIL7U4FOQNM1fJdgJJFwIJGWuklRKYyQiAcnQHEP3gS9SBhqX5hgDBdXrwDjjiy4
3IAHOSYFUHQDf5edFhULeEGSTEMRKRX12E2GCJY5hBkCAQ8DnOk8/PG8KTbG2DoASIbpbRkVIANb
7xvorMe7yQ/Z8wB24eKHYB3/hoaYmgHFZU+4h0iA29njIe2LHeYeLma6CpgAm0mAF8g9u2GAxuA6
q2EdmxuNCB1A/WGd8klsitZfB6hi+9aLjQu04dBPBIIcmgiVQTAMuijMnMXscHIzt9iECbjHEXjE
sziBvoEAvolLDBgBLWPXdDmWT+dAL8kA+VEBTtCAMqhCuG7EGQsABtAQuzmAAumETbiAM/ploanD
tqca4HiBJ1jDCcVjM+EWO8zQF7ivVfFImkiAtMA37Pp0xHGLEdgEnkCKcFkJJ5Dfoh8Ou4kZy4oI
AjhwZLmelbCbAtCLcn+AF4CDEHWXgUA9DVgWT76e6hiV65EWiByNv+HAHYyH/xQsvNmNj39i5zmF
DQTgCwBIhGgNAHugkEICii/LCQQwl1aCIcAog8c9oieKrEkqA5V7gvA3lM2LrCG5DXAoiTnFDD7P
IhiiJHXqEo/GuXBYNfbKCWmC/gcQ1GRaNTMAiDIAADhAQEIDgAAL+AFQgOHEwIELSDwAsCCAggeJ
CmBAAAXOgwAJAgB4gPABiQUDNcxS+aAigAQkSA58CcDJk4EBzKiMiTGBAgc98Y1wUFNDBQQISHIS
gKCizQAeEDCsaSZB0RH+FKhUAO7BwoYLEAJAkBDOUosRZargivFiTZgL4OSUkhOAFH4qEvhTmBCB
B4YqzLwkqYAfghVhG6JM0HaQRLiBKxqW0duhogIEZhg6IYvSL9cVJm2OLUkWgAoECBt4UBDxNeyB
/FwPpD1wFonYERm6tt1QN/DfuiMwCB7RAYbTxn0bry38NW3my1/vhA7cwax4EQjQnN78dVUA4Z9b
j+3wtIDJ39c7/86bPfz10sm3j28f//582NF9B+jOPnpEtjGXX0n+5deWbgAKWB+DASb4oHj6BTfg
cy+NJ5wCBF4434Ll0bdVQkbp5J+E/zUI4W8E3gedP7Wp6GBv7WWom4j0qehaVbOBeJ9tNaIo33cA
7gbbeC9OWKJ0toEoZEQtvhcfk0xaZ+R0tO14nIwOrgglbCK2SCGGVMomZm+utRikbl8CoKaNPC65
5W9fFkmfcz6eaF57DkiJJ5xx1namloHSOSOdzekJHJgD1SgmeYcWiqKYF2JYIpbrGUXSmXbSONCV
xl16J4sKtChiVY5SyeZ8NYrogKTxSUEBfGT0BKWZAKzQwFPGSQFTra+xaZ8TmHmIJ/9tDngwa3wd
hDPfmbQFMNkKHQ0bayLCaUpedDW2SuGLvq0Ah0KAOrdVABR0MBN1gjL0QATDSqeCAAFQhKhwZixF
AHRGqTAvvQEyd6UCFsARkRQnlIvsf7RZIMGPEXEyQXMxQudaABjEU4Cww0G0JaqwkRCBuNPqhkG7
MIoMACOb9BgcPsVxQsAIHqroYwG54RdRABHk1GGWCYYc2ywZ4JOcr7Y9IAEBGZC0bgQkOrdABYEe
eAC+BSDsZ0SbEIAPJLBhgFcBMbeZqodkZBARAgwcfXVzo3pAAAYHWERABBogoJIF4TiwnVkFgAMA
BRgIoMIDHmyCgROv4XO4WQTTHQH/P+2mJm92uTVQERl0I6ACuDnHtF3GDTlgBj4AzGQGCRjktoAT
ZlSLQAMqbLKJWQhovfcmCzjAuEMbv6ZCBKrbqnW7AcxCgAQCONdBzIRJEYEAm/SHgBNQMGDPAxgQ
YBYJZESwgPcErEBAvIoT8EIinEBAd0LB/73CJgKgRPJWZDAyARlyZKCBFFqHZAYB0DWQ5xHgf8FT
iQrgtoKckYwgqeNEPJ7lAHAEbzILiAAjxBcBEjSuIYyARAQXgAFGIMsCGzhBAQHgAZIFQAokIAB3
VAjDFqggAxfhG3RS1674qaBrESFDCU7QAQxoTyX42E5PWnQsANQtAEuxwPqkUBAB/3wMAgAwAxGt
hi8nbMdpwUHAIAjAiUI8gBOz0F4EGMaJzAUwbA9BgAQiUIB4rOAOAnhB1CKygOOF7TgqCABC8GAR
jG2gAAKoQAAY8L1kpLEACZBAAJ4gB4cIII2ios4g9CaBBXAiGZvYZAYJILeakWwCBdCABCTAwQqA
ZI9kqIAG4gGx9pSBDCSoQPZaQAJILAABFSDBBvAVkUFA5CEe6EQEKnACJ4hABXdgACq1JoEHVOAA
HIREMuMoAd+hTJGcTJ4E7rZBSASAAJBIiS3DaYY4RsAMFVBEGThxAjlSswAImBFyAigBfEigAIxA
JAEK8L2ARmAQE1hnQTOwgkE8YP8D2kMkBiAwiGSExDC0mUUFBleBCEBgmgSDQhoF4I8NqtIDd5Bj
AcP4gjuogBMj0B4JGIqzCJDBb9DDxx1e4xESrOACGdgEQAkwi0F0hxObcMAH7GABew5CAB2l5gEQ
QIZ4SCF5BZCACgbCCdV5ETjauyIDVgCFuOFyAwvIQAI2cABO6JIT4HgYAJZKggLwIwAVEJsCgFeA
CsADOFC4SQaouQAFuFV1jFDeXB9wgABQM60HwMAHrNNS3UlgBPIEwEO8J1irQaIABbgDAqIak4zy
wx7Bk8MCCOA7HKngBBjg5WEBUAAKMMKKgnPOIHKDAXCQ4WschZpswwGCTgykAHD/GETtGDaL4kQg
HrUJwAZgUgCIkI+BUHhAbE2HAWU2EAD2GIQGFtBZBvDSaq9ZQCEh+4ACROYOI4juQETwOgm8oF0R
UMQCMHaBESggKS4lgRUDtQJO8MMDyUCNcSOyMBUqDYCQEIAZzuaBbX7NCTTMAAI2MIgMQKJxQYGD
OfEnPx+KrmtSCJsCoKABEQwCsne5YgbIIAFrxuOMCdlmAap1AgjMogAqLEREkmmG+DDiBT+2AC4B
6RB9cAIAG0AAPhCSATjAFQE1Y0AAQIDLiBAxvyr5GTkTgLG7OsQMnHihFRMrAQ044Q5SOMBFHoBP
ndgzABTlxMAKMAHoVSwZX4NE/wMACYBwMmaE2lsbASBmmIEwAgOKXYF2AMAJ6Fkxu8Ut5glcy8R4
OCFq4exhQ/zGAZV0wMfHy5kVaSPfgQzib4xQxOECAAkNkO+K03yBAG4LAFQ+YaELeAJjc3xe5AJS
A5xQSSca2hOzAgAcB4iH8g4I3mk6oAKc7IS1PZTfB8CxJG6NyAoObAbOVoyKVgztGQPQZg10hAPg
AOSioaYCkp1aBOFxQNfWW5HrhjOS9z5AUhwKjqAFQAWtZoBKyACBmAKg1LxRwAh++SjYLOyQWOZI
AK+IB4ikcTvGhgPUBBDHBYzVs69pNAH2G5vZcMKzsORAIggbjhfAwQnB1B8A/P/Mgah5Nn4B0PNx
kMeJTrAbHA7pnvJm8QGzYMCQBFABA7jXdAHEY9KdSAAjwLGCCrxnEwLlZaIljRY7Ptg5Mn4BJAb+
NYU/UrZmkUAGOIHlTZbF1ARQAEcD9EkIIADPCqAbQeXALoiRYJu/REAy4KaBDT+gxR7HgGOOI/fm
VmywuExjAD+JvBUgngAVgAAnUSlnDli5AASIjIScMIh4IAUCTQ/PChi2wrTesp1WnDAqXyCBC8z7
pQKNoIJ/2YITkEGj0BFBBDwQD/7KgR+MOMB2ZuMwKChgEBXIyOebPu9qzUIR1PS8jxNCgH92cD2J
iMADGpAQEhisIe8eCAK+Z5H/iiRA/hbIAAls9hoSHEs5AeLHCSDACDjRT2hAAlTEAlCQj6HEAqiE
E03AE6xXtQRISiRCAKQAQizAAwTLX9AEHLSEReyMB7wf6lzEr2kNdIBDSgRAIiAgAqYETKANRWgA
PqiEEwygWZSGAuQfSaRFDZaEDeJVQyDACdjDr+FFzBhEOKjbBErVAsRMAshKScyCBuzgCTyAUJCI
V0yAnC1AW6SFVCEEAlCAFEgEGYCFCpggRlhA7AmA0i2abyyAwTjACZDAeASABeyaSqwAB73ECOhF
ZIzAFS4FXMQfwvDhUzwADsIGSjzASDSbeMRfRdQIt8VE4wRAHSYEXPwgu4BF/1W0FttsCaOMT6uI
YmyYgXMFB0r0yYqQgYFwCSsCSSzO4sQ1R/ZU0tc0DC2yIqPsoi6iyYoUnJH0ImwUnC/6YrfMIjEe
I4TwzDEqhPo9x430C6Uwo4lYY3NsC6jM4q8kycmYxzBqyTLejKB8YyzuCTXConfwiSyKDKFEzImM
IzZe46AA4340xJlshZBIjIu8hlEkIz7uo9Toh2+Iyoy8CUGKI6UUJDYSiDya4zwyI6M8ZEQCYzlK
ZJ5sB8VhmTh2WqXgTM9AxxkSBGFQnP4pijjSHZ48SVcNhEaKBz8eSUXOJE3WpE3eZE06wAO0SN4p
gDFy0kWICD6QRKcJ2kBgz/+NNU1JlI5u3IEDupVP2oNRzMVARJlOyFlV9pcDDuWaoIai6EtVCuUT
HMp2/FFt4IM9EETBBQCrcCVF4iRcxqVczqUyAsARaY8CxI8UEMDFNKAEQIDV+EPqII4TQALc5AYA
ic8CyEFlkUA8tJ9+ZNu8gQ/cGJlBMBEDcACWMUBuWMAIQcII4AMMzYJh5M4DEEAi1FACCQ4/WADd
3EUaERFQ9BYGaGCuwYEHvEA8MMJdvCVd/iZwBudvtogTdIBYjUD8IMAdoI6b3cE8QVJKcAABDJcZ
pBHUkMAs3AEqKQ8UkIA9YIpvbEBk/JxlbZIEUFBtWpsFwNEJLOYCNJUZdEL/IjQae7pkBqTAB5yA
GTAAPpCAO7OnBgCbS7YAJ6mOBKxAMiVARpWTtSUPp/imcEaohE7oMbrGXm4CL8UPANVVhxGYFFzf
dnCAAHyot3WUAMBNxfzNXBGApGglgsGD36AM8QHeQJjBIDDELyHSoD2ACNRXBqDedYrABGQSCJCM
U6YAR9wFIShPaAlUQlRAhFUEIdwDZikPhV4plmYpXDIClmXoQe1TYeaXurXAA8hBIkFAmz3LAZjB

BSwAPshZjJbBmyGABcQgq7UL1ChAAQzZQ2DQXDHEhFUEL22SrI1AU9nDCGRVhqTSGalYBByAClTA
VOCdlUaAPgAAEygSBvCD/4LmV8zED8rEwwRdi5aWqqmeqnOgmQRAwvlB2Kp22AJgGdQEgAQkAzY9
wB1UACRQFySIAL4QABSYwQFO1hMQ2ms4ga1CqWwNDCOcAAXgSwFAwjQJACRgEwDMAhRIQAs4wYcS
q6IwQgsAwAEwzAKIwLAuAIZWwATKGCRY28/1KvU4GhMpD/kAF6reK77mK3ykmy9q45AYRymu3M+c
Y9voq8EerJaWCWwQoRdBqMhIio9wCzkeY8AirMVe7EOCikNWIzvKpINOimzOi7gMbMcmioQQCkM6
LMauLMtiZHmorDXOyS/Sy8DCbMveLM7SyoNCJDymYzrabL+gY84OLdHy4ivE9gpDXuQ2cgrTTkvS
9uyjAG3RTi3V7qLUVi3WZq3Wbi3Xdq3Xfi3Yhq3Y/84kqcaHl6wM1Jril5AsQXyHj5Rtl2CjlzRK
fLSkhNaI3XIt3JbqyLqtOS7terAtbAjuLlbFz1RsbOwthSru1jLuljiufcRkRIKscRCuOlYkcyAu
wmru13KufPCDiJAIU3qjP3ZHf/jH3eAMQhAIPgTsJb1IQajEokSEExBAT/jk6XZH6o5I29JJRfhr
myhkbLBhecwCU3psL4UHh5RjdzytuyhtbGAX6t1JfvTHiNBE7sbtiWRv9l4vSXRv98ZG7lovg5iB
QCCJfXjAKeFFTtFuHn0kSt7EjDFEVVnPaXTXtX7NBPjHl/wcsvjIempve/gDBMBKbGxFBaSQAuBD
Bf9UgBy4l058G6PtrwC1BwODhOF2ZZMObcjurGyICAxFxL5UhAjcxZUYrm1kjpV85HioCr+ApIPM
bvz+q/T9ybcWwvGCiI6sXG8UhBxAhBQMAiRoZ8XIwa6GLPo9h1F8SQRU61N8QAPj1SZAcTwogAdU
q3M5RLsuywxbhAN/wB2c5I7saeJ6MJXsZO54ZC8lggPkBbeRgEA4OVSIiAKQwWNhBFx5DYaZBWOt
gAjEiwvNigM4AWMZhErYjWpOUh4ugBmMpyH/BQkwwCyJhX/ygxTIwfvp1AE8nOAsUZR5wEYpBOyk
yGC9sRS4hiGDBUikhgcghAstRWbkykSoRy81Fbj/JTAAsFJKNEQ4LJFOkcAI9BgczARXzEJkqEA4
EEbhoAtKfIUIQEBIGERFOIFSdMdEPEANIgC4WAQH6ckCeIDeFE5aSMRftpDWxRUHoYZmIEAiTIUG
Qx3QhcSgRUDs8UMHFAJN4Fs8jOUYLkZcEdVAeKQ/Qk9EIGBlrRPOxhWQJUQJ/gU4SAC4GMQiZuGk
NIcHMEAak+ljEQY5TdReKcDz3UEL+AcW6UUC225VEFYLYIAI7JEEHFOO+ar+2dV6VUABiIAGfJIA
LAAHOCcnEcAgNCAGDMIpJdNe6d+HcoIyUQAeeJ9wBEAeE4RH49JhRQAUZMDCZFxVmule7ZW82HTq
/1TAscGN6TEQBxxUGC3AQoncAljAHWCAHFip6TCmBkRCAQzCmpHByxxADIKyQFFABWSAMuFDo1WA
E2ArBlhA6pz1OXDELMhBIYSDXkPAYg1CCzydRETC9pgBJDAABOTQ4ahr7HEEPoTRdNWGBcjBAbj1
AdDOBTCApFrAWFUA3EFC5EXE/QTIBoADchSAaEkGY06EWCdFkLUA9NBgBdTWi8nYQzREQX2N3FWS
f1yIjQ4EBvSTBNSfWIvAV7TcNDXfe8YMo1xOGuNSGCUEJFTf30ABjGpA8T3JCunFg/1T4gzEC/xq
PESybFXEgv5GAAzCCABdq1nntSoPhp5AYA+G5/89ACTIGQNsDD9sgnPBEXSRiII272VtU6JlzxNE
qlI2hAKQO4MDgFO2GqOdjQbIwfpAwhMYhAR0hCoBAAGE8Xwrj64lBPoAgAgsi1txgEAZdOS4220U
wE4cmEFAgYtLxgkMQiEgNKulpk+RwQdwklzjRQVAzILF3giswASIaPDUaHcmA0AnBCMsyx3kRAOZ
jQWEa6NJ2mrJ+CTL2GwwggQo02tIQGQIOdxURQSE8YtbROyx6ECkoUc8QOQMwmIdgCIhTUX/RrdF
KkIElDONmhnISwROwhNAgACYjXMk41LlxAqEK3YxQJsRlkqApvWdJ2zkpl3CUi7PSnMHz37SakX/
ONr6ogaA4xkA5FZZhJNGBlQZrMAI9BYcjQA5ldMsOQS+rMABrNiieVAuRvCQTZdqSYHaWITIifAd
pKiOJ8CoDcQmtEsCtHQD2lW0fM25MOIGSEEDPEEDhXBA25CTmUTNjBxcDEQkRGPm2NoCbFIF9PLZ
rBcJbAJnyg2raSAxlfsLTC9B4MPLzJVFEBMn9NOEwM1AyFgCPKEI75YDXEB/xGjQWEA+yLjySPBv
NGuNrpmt2codvMeBvjoTMQKf+1i4RYQYxQYv6YSkXtYJhB+eeMDIaxlCyNGPShp61tGz+OqQ3ccg
MABg8wQMwRI5sRCAoBaiY4DYNIQTYEAykIBD/5heZXFZJ8wCJPChjyUD6L2MRyUEgE/Zru8ZGXC3
B7QA5ynTTjPCdLlXy126/q3AOTmodtpGD2V9QwBcnyfAJkwAPy0c9LRvQrBU1un4BcVR6ijPz6nO
FSL5BTB+BhTA1oHYA5iBc8a1CHtWfw3gxAfUx0wgAHzSBMwTBOBhMmQYyWlGcXTaXDGAglKdBnhW
AmAAA7xNxXCTCGFABpgBFIhRBkwYOGwUHE3ACzD4BDgFzrzAPbk6E/zSBqTEpRICvlyWothoIYyA
FEBCBnTAAqTAOV13d3jWBVUARv2NZIBTOz2ExFkb3WxXxWBQMowAQFioEEECCXAFAvgDsJChmf8D
CyUUiMBhAYZBEaCAY1SABJQnDjZU0BCggj0AChimTIlSQYATBFYs9EBAgwIKABBoAEBmhIiXFR4s
VOCBkaIIAFREEKCTYQoEmxAAGAEOgAUCACwICHrSgYcACIKaefBAQFQFEUgAWECABL8AESKYCcAP
7opEDBWsYLtwFr+F/hQ4SbvQb15wBEYAWLFArQB8JGgydIkzKIkAASA/WLwwKYGgCzaRWPA2QgCc
bBX4MxMBTmK/agnYI6Hia+JZBKKmvL1ASgMAD+AAIBGBjIYHVKviDr7WDD8QBBIoIENgQd6tCx2Q
WIpAwokJCyGbuYsANwAnBIpLBqug7Ukz1AH/4OuglrFdyZsI4H6AX8ByAqVTegCx7OBLSa+0HJiO
MScgMY+AEOx7QKnEAHgvKikQeI0hB+LL7SwB7nIpgnAeCKA/CjjEIB6cJHgNJZVgjFFGzgY5IYIM
TJtRxx157NFHGF/8UcghiSzSyCMXsgADJFdiOskZRTMyyCP54YSxCKR4csYHSLBMyy+n/FLMMcks
U6UA7jJTTbx4DHNNGR1YQMM36awTSDCJdNPOPJnU8yQh/VQzOCa1HPRNQ//ck082U1IIRg4TRZIl
odok1EyFEK1UxzDdDDTTO8ucMlNHixT1SJRI9fEBBEzLjlIAAvCAMUv/1MueSSUDUKVJIzhB/6hU
ZXQUOkg3uQklBJZ89TcvYWUknL9e9LNToVJDa8cyCkgJAUgIYGjOSH9MLUguc2QU2hmDtBJcDv1y
gFMZA/NM2V3xErbJTxHIzVwHJNjwVQwWSOuBIMMcVkfAcsVqVh7zBQDhH/3ygAHTpLgDu4W+YswB
fjA1GAAPMNApAIPfMm2uPxXgJCOMjnLRNH5Q2hjlRPnhUFwAWMaOhGxjZugBMl7DhwGzyhXKAYUG
MUOhYUO22oEXvoMVY1gLW6ECaGvEOFqZHVZgYw6HbYkrqf/MUYqouYbVZkoDeLFtvBSg6KRyHwbA
gQggMJpeDlfghCkOc4QZOqPxMi2Cbu22N//HojFGYNYcUTJZDgA05FAhCcAajOSOp8aOZsqjzVGF
TQKAI6iROXUAkga5+hZdnAoI6oEKMIsA4I+deAsDBBxAVl4PIOAERwEJSPOkTfhZS1cGK1iAkQz+
cxcyMoYlXcIn8ImAH5Bj68yMeI6aDoOho4ogEcwQIMEBfAhg5IBw7ibAg5SkiOeEBSApSAPbo9rE
DAzSEgD8FOA7tvMKAgSwCRVOgBEEuNpCFiCHAizgPBiwwFc2EQF8cOY/1MnPZUgAQAWE43CkS8FL
PEMu1nDiWVKAAAbkdSwMKLCCZvgNIzYRiXBgRoROAJ9JkFKBO4AjOzBkCngAaIECEAADjFn/CyMe
4AAClAUfAmgibGBSgRakhQQQOMpaPAOZwWhrAekTACMS8r/SQGJ7jEGeCv4DCceZrldXJA8DdPUV
LwZAQId7SreKKICREYAfJHiAFKbIuJRYIAMSoEoZbAc5HnmgADpxgkjuwLI7aGB3GMiAGSZghgog
oADxYFAHBnEAfFRgOBJ4kQpE8IBBgOhF+JAABALAiCOmAw4ewFwBBMCQApzAAh8IB8jMcIcVDCIe
UvhABB5gv1lIQJQZIAMu8QCBBXACDpW8ZQQgkQhOQMAMEmCM1yqgFVlCYFUTwEAlRYBHDiSiABkg
ASTgsACOQKFKFYCHKEkwiAcqAAR3uN8g/whghk7IMhnweJEFoHACDkCCBN2RQmiggA9dIqAC4CBD
J8xQSgQUAh9yiICABqGBOyDgBf1aCIPAcb+BzKJ5zJwFFPQ5CPwtgAOeYUoAOHEAAWKOEwxoyEpJ
4IEKeJF2XZLAUkqgQRKcgBEVUAFCZ9EA4e1Ppx39KAIsAIkFLGxuANwERZ/KHe1oQAQBwABV5KAB
95kBCunLgBTkMAEGhEwECxhE3hYiSp4SM6KD6AQJNhAOmnIHAiqARAAmOJDzmWshnEBASAEwiAwk
QJEzQgADnmAekXAiMVAAajlzw4i8XfIE/UJAPMwAidsUgEIkeQtHTIMSBgSHAN/BwASKGv9CRTDE
ApjrBBkksILfKoYBAlnIb7lZoQzgY5jyYCUGnFCIJ0zOAU2EhABsFxy+MCAtg6CKYNxHkhJxAhzm
BEAEJqACCnREAy/wFd4AkIAHnmQQiSClaWrECF8xZAX9ms57BXkCOMghEZsIpmpnsSKQzYIBGpCA
Biz8gA08YAIZEIovkzQIfvCDEe+8C3sZUADUUoAjGorAcDlwXtYBgBHBVEwGHqCATlTlBKUs6ELO
NotOIEDEC4EvAApwABJwIgKzWJICMLC7XT1IKQCYRQYikC27PfatALjDCAbxLPN+bwEH8EeBNRyA
DETtLKzEowWSAYD/4Mx23OHzHdyqT5b/FS1aCoBClPkZUl3tSCFSkMBdCKIBOgIACvUNhyNhxcSP
HUCpVp6gWBfwgNfAMgAqSAQktqIAjtDYV5w4wTtHkGnJyAEDAuBEC+S6JIkt4IEvAACeASCAbA1C
BCvgwAHg4IQCPOGxDzCnCEQzGuw8oLgLKABVxkmCA2gAKIoGBweiUiMCxM4jBKAKGbJlhhbghSLF
Da0IwsEI5KCkuO8lgAIcvIkCfHoE8cgbMGcRZwMyYHYBcIIEErCB/1CoKg8UyMiAOUEAbABZ/B7N
AxZQEOy8OLNHESVDnpekkDlADiPAHAYgoIEG8SsCCLhDAi58khoDoKgjSMD+VkQbEuBZ/9l2YwQc
8PNeBswiW24RgajTEtdRj9ID8fArPwrMIPRAjgwiWcAT2p1zU+fnvTj+eQFYtYBkgGMsMCLDAZTS
HX+MgJVCikBIJDACJ2SgtJy87AUqgAEQwL2jCpBABSqAEAxUgAObYIg/NuxAkTDkrHCAGstpIgEJ
6DQlg2iB3fulAbqDZdx0fsAdOFABo27iA3LtRAAi/pZOZD4copQABoBI4wqkfRaQwM/qL1yBEUD2
sndY/QlIoPnOY8BXKkgG3i+QqEEOQAEEaMEFMODWMbLI3SYSgCgLgFNw/p5LkPh9OFZQCBWIxAnJ
cIIIOnEHAThKAY6XwHlaEHe65A/TsP/XAAE24N4kQSECBa7ABfRFEs2HNMeNHsTdM8Cxnmvx2IgR
IKF5EAAKBCABkkHxJGRJnGAQLiB2GGBWpOgEHEznAuDuLmxyTuAOBqHztuUA6orM+gXO8AEKIKEF

xAsAQOAAVm8BViDOqM7ULgnvFmD7FC3dMq8iECclYgw8JCACLmADyIpH+GEsVMBu3ia05kIB8KFE
kOIBmEIFqDBHxoIpUEInrLBcWqJETuYyfmMstiYAuFB0RkAFYGZx/MIKR4JyTAaIWqVEVGBgsjAl
nkAK+ahtrjA+MCZCcqdt8MEeTMItMAYMU0IN7WYsYEZsNsRkIBEAnuAJ7CGXcMk0yCD/AINiY/xh
CR0A9KyKlNLkC0+nDxVgLN7GOE7HCZZwQx4gtFBxTtxQDBXAJAIgDedicYwjt8jwFE1CFWuGF4MC
ZFIiFyPxFoPCFqnQJPwhFj9RalqinN4CRxjCCnPMH3IrG6fQknZsYNwQKRQJdU5iJB4g9TQFVOBm
SDAFWBiiIqLsHV8gHuLxBWDoOlLCL1Ilwr7EdQQFXsylMGakVxoCsODl/9KJWs4xRuomVIrECRKA
VqgkUVRA8SSuSFRA4AgjRj4FUFQCZhCySRoRXNCRcijFCeDiBMjARiZgFiYgJdEit+glUvKRVADy
HmFkIVMDIQkmJnHlL3xSJ2llYDSS/1GEskemxCP9EVyK8iQeZlgAo1PepSMTZSchcikbhSRTwmCC
hB+9BVR6hl5cZGZE8kfq5kWQEiLB5EXycVo2Eh0HRVoy0m7G0lD08U/IL0Y0pCdjskgUgh3pBVGo
Ui/Hcl4cRixfhycpRU/aUiUQBldQgh8NJS/Rkk+mRUoQUx2H0kmE0msysy6VUjF3pC5NJR0TkyhN
kzIPMzPR5S4/cjDjkkhYs2CC5TRlBCDV8u+UxTMzJUwWUi7PUTQpK1xOoieDBFMYwh7MAB8QYINe
MyF3pAGO0DkfhQQ8hDZHsjGDMyE9MzsThZOwcleM0yqNMlGM01KoEoIGTTqDJH0uxv8195Ihsico
BkZeXuc8d3NMKlNIIMVR2LEuf+YtjohmfHM0MwUzZgU062UqE2Vw4iFqcnMo71Iti5MlwuZeuuZX
/IWytHIhUqDyXidVNpRN9BE4Y9IvlehlWOIuvZBNWHMTCJJGbLPP5oVDpGAQaktuztNcXgk+ZPRP
PAB57NI8kg83m+TmmBJVgsQJJmYl/FJHnCctsgcu9oMMFoI8gEYFSMAeTsAe4AhAguQtPKMrMMAD
/KGMMkhASEABFgAO4MLf2NQCEGD2NmETPpECCEAK8JEhZsEqEEC5GCMAyOCK3mITGMBX3mN+pCAC
NiF73oOsBiZ9CABON4ExOiPH0q3/AxQAWaICWQIoUDGgRBQ1KKxiflRCL8iAj9gUQKwiAvarHf/j
MojDAhRgAhjBeIQCTLMkhNJCCjogVBkhLeCUACgAkaLCAf5nFgAAQwhgEzIVgKLiMYqmjBQDfhQk
OQQGMjhhuFKCDD7AM7LniphyIbJHXsiCAEirS90rSIqIMWaCSstIUb8if3LncB4AH1BkAUIIEiLA
XRgCIzQxPx4yQPAuLSzkvcbiRzsBA/BBSarTR8xgEOCgOxDgAyhABO6AVReApirqTico7f7q6SSy
qEqOAAqClW6KVS+2r8CJIJ4PCiagAXbnY+1h25QqtBJl9SIAD/YKIU6PILYJjyAh6xzAjaMWgAxc
NgCgYK8GoWjwgQOG8AOUFgAGArl4SjSkAKc0YxNOoK82odU44AkQKsvwx3FUAh8wILYQwAPo75YK
igQyr2zlZwMM6QMEoAwyYHceKys5QSLSjZoGggL0VYs2QQLS7WIPcBNkr+mubZ+yDAKUCxyuZprY
hAyMigwYQbDsgTsetiUloPVsjCTDKhz+jRHIABJslivg4pYCQF854Q6MDQMYAQoIzm42IRm6ZBM4
YGMtoBPkjQEuSZDC9gAegAMogHAZgXDD0gPuwJDSiQA4wB5VgAMwYASQCwGM8OvwA6CcQA5E12P/
eOR4DyNm5ewoBCxnVKATsMXY9Glo5iQCKaR53gt8usXJACC5quxGrsxhbM12uE4CcMN1VSISRiMD
cjDtoGAhOGIDnqD4AgqAMIACTu0kduwBFO5FHADrAKBBnA0BkqFXCkC5MOB8hjCK0id3/+O/oACU
dKoAJhVGVoCjTuCzAOAE4oEMXkC0dkUKykkAnMBiXpAt5A872GghRCAxEOAADoziXoAEGOE7eqt+
aY0xMvFwAKC91i4AjJD3VCJ/mUARIo4CqqQA4KCBcqaLVeIiq6LCcOY7xgVZdudqHiADTiClMusu
3GYQcqOjKqQAIkwtKkDUniUjsiwRnClbQKYn/51AqJxgAxYisVTC1KAM6zAHAOSgBTrtwgIAoWa3
Ukh2MZygnGhMAEjs1IJpAVpAlowtDTnKLAqN8RbOAuQMAl6i6qQWNDzMdvLXrcAhUyNhAvy3jMJS
ahNBBVgIBAwYge+pRFSsACqQVWznJERALWzUN1WgAJ6lQZQLoMLhglSg6ahwfLaNo5iYASBgh6Ag
n57OHgSAdlJiAfqqAnCvcocJA/iBoOAHyO7gYRdw3B7gAiaVd4rRdBuZMWahEMggmKj4BXDuKOYs
GXiqfOwnbzzgAFYgAzSACVZvRcGNxiAgMJ5qb8MBLAgIZ4IpSGAJJ+iYmnxy+/AmA1bAYvzKDP+S
4QlI4pwwZhAGI3eFgyMUIVP/eBBWwM9kxTeQq1+crCYB4A0kAPSiuZa7kptS5gASgFU07NCMzTQ4
CSESYSP9qgAmiDtImsYmQOskIMaEjQEF5P/0ZSHWGXNyrwDMiQmW5EakdgUioAUEChzMwMNUAICy
rAJWYJqWiHH6CrJWAB+Swa0UDyHuunXBobgK4FPpmtHMI4hhBYMP2K84KSIigJNSydkkQMBOTwQM
DwoErvg4uyLgWpIWmQDkIJ7fK3YU720TZXgxQOAYhK0j4geFwoHwOPf+LxFs5yzypkFXborqtwGg
z5zoN9+2SQMcAAo8TCWk4AIOYAMggJQ6OOL/CoC0EOACBOpFYUUCDkAKMGCxJan84LqS/s9pR4AT
FM8IX0XckFiUDgBzPABHXhrJ7uA8UtucDkAApGADhjRaWkACKFjxhlQ3xG8GmVkDLuIEpu0CzOn/
PjUZmDNTnAAB4IeP4iNCTGMBLMCJUqpENONAq7GMaqKMnAAAMAxWZMcSN4OPYMbEyRb0yFYlcuwU
+RBm+tQJV4BEltC/qqPrFCBL/G1OHEAK//R0XvguVLxMc0ItVsAJtukEHgDUAiDIH0DK7XERw2E0
PNwK88uTX0UBymAx2uZxXjg6HeCFQ6uMHpLMqdDF9/A3dILJyVYFuDDHNE0B5BtGygABXi5j/0wj
AbAvxBHACXSCVE5xNDTghV8GZVQgzIMCH7BP0zJGCovxhT9jxGG8JeY8JxygAYS8DJAiMcZqM1Fi
wzutT8NxrNTCccoUVqTAH6Z6y1cAMzJAMJ/EAUQgWb6kyvBTUXzEgSKis5LS2FXTSMpgA7jSTBZz
SAYl2glz2YVELJidI9/TMrmzNkMTT77TSILZPbddI8WzTyQl3M09Naf93PGyUEpzJD3l2ovE2odT
THISSb4lMEe0NYVz3JtdOhdlV+DS4HX0ULrdRwIlT3194QlzSrx3QglFBczReyHzMnHFw8MVSR8U
OBEULN2k0xx+4JMyP8sz4lVzNCdzMlHC4v+DEzjxobOWUi8fPh230zAXUzwHBjDwwXv38t+BxDGj
6EcQoDSEHlVOQkOOvlz4VSgBwx9AILRmAkbacKspxWYsOOvzvTA5REnhCB9MZevFUul9EyryfWmq
4ylVIk4CAwNacTAJpig9/jZ98+49Xi7ZnjWT9So+9aWSz02eXul7PlLQwuPR4jWcYCsq9BSDwloW
lEMJqT3ZRz4dxQq1vjVfxAkwwB+kgHG8d21I9EX2gzm7i0IgRShfozH9clJGM+V4RCEgLlVIJfV1
PDrpxUh3hkirVNiB/kdoTS0Yp95pZQW+/DcGYejjo19WoJIe9CzbE+9b3t53BB+q44I4Ixz/cnRe
tjMRNjlnnsuGGPMkZiGYmExGIA5GtpxxPOBzz+UnkYIxFO5Cef8vwRMpGmBxXBnbGwVXAMIfgIEE
CXYKUDChQjMRNByQACkCAEYSCmjA8EAFgwCcDhSIQEaDBIgkABCQAEUCQimQoBDw0ElChQcBCsjB
gK/ChwMCASwYuUnByTucADwY1GmTA06QKiwAgOHOBgkPBHDEAKmAPykSREBCQKACAYQASHCoUCEA
vgKQMEwMt6JCCwQDC3ygGsGrxIGDShzwMFIOOABmZJYcGBfiggdoSTDOCmCWzAr4AoQtYA/BgQol
FSjAUAEziQJoI2QNEGBQ0wdOIHB8MNAJ/wYHYccCUCASEoHRFSIQeCpBhZQKkGaRIVETEid+jAQM
ZAnFdSE5Ep4CAAGpBAkMB1rM9PdiZAALBC5WcAsAX3lOWDHqHMT8jkUHgypAQUCxwgoMCD5AIlOB
URgkcltckJTEiQQMEITPIBJgIAUDsBVAwlhmbFDBBok8QBoB/JDRVQGoSXBTABRRRRAZG9yxgmzE
UUAYJINAoYFCNg5EBgYqQLGABdVV589RGjhBFQeKAOAbPlCMMNoCg1R15EAYSORBWh9pEE4CByBg
hlYDKVABAgtw4CQECkAyAgYQAKDBJgtGQOEBKkRQgRMWcQABh04i8EALUhSwAD8EYSCBAv+cQKBC
OClIMAIn4HgASY2IVcCPjxqU0QJsALzRgk8VhFNYAE51qSmkAUSAwQpQPBGABOBwdMImKr0AQagF
CEDGHWQBQIaIGBAQgQQaDHJAqwv8ys8mGaxQwBMVjDDQTxrIoYECAznwqwIBkNFCAA9wsoACd3B4
AmpTqiBmAeEwMgFBnDhHgFsvxHPtJvQyUiwGAsySwQMvKIJAAQ9AEuhzAlcggJ0IzBIPP4UB8JFv
HI5QwZorSACABHSROZqgCljwwAkFJLABTe5GkAAGJwwCjgpoAqofSylbJcKxBQDg1CYY8HOqhALc
PBAC+Ej2wAYrSHHHE4Mg4EQn/PR0Y0H/HuR5BwBSHPBAsBEcNQI+wg7yFBnlQeJTAYUtwO6g7ZKw
IAEQ4PMCUwggsKCgKkxVgAQrOIozCXxDfAIAFmDACQEAOJGMExkovYADGcDxEQIYnwSHA7f5FpmE
L2DQyQKcIODBggSVwQEATGDgGcuxBcisBoyNgAcDeT8xkAc3W5DMAgESCRsZGWAu+SadZCDBBPwW
xEloEkQQgQD8NA/AARZE8pTtmtnDAbSHq0QnCYICsPRAs0DAjwqA4vyACLRDNQE+EMi9iXNSSqQ6
3etHsOaUSEKwCSQZDCIAH3FIU0YZxBPIVBM4eIAAChAeaCKgrIAFgEyeAgAHwsGrAhDA/znW2k8F
5HQS7w0EEhWRABxm8SsBSIETkHuCP8y2mQyQKW0cAMcBzDAQTrSgAAXIAEFQmAwJKO4BCuhEAgwI
AEgQMGo/xIAGyLanB6zgAyNwVZWG9ZQkdUJwg4hLTL43EdfM4gBhDFYAkhEO3FULAA4Ik7dqMpgw
McI1D8CAiHR2ggOMYBN1sohT4IiVFiwgAFKYxR0kRRENcEIRBBiESDwHBxJgjCAke4JkEoGAC0jK
CZAYj4hWkKkNjCAAT/ieB3TFiGV1SgUXOAG5ZKVICIxRAU7Ylg8JQogM8OMBzBFAA9ckgXAUzkTx
WAEDnKW9FSQjAAsgwUxuc6gHZG1N5v+bwAo6cZQJ0IQTJ4jAMl0Vv0GtBwMBwMALvhcsqETgeRnI
0RttN4IDrM9qhXhdAkM3glnAh4hk6ATCcPYUCwRIAidQwAPkwAB4DCQAGyABPJ1AgkjxpVy8BEEF
qOU53aVxA8KMACGZ064NgCOVT3hAvFwnKSU9gASJK8AIHACFB7hqAVAAIxOD5lEy4gMDC4iXW3pl
x3ISiCE1OdxuNCgAwyFGAiRAgHNIYM0MNHIBKpCAUsuSAX0FQABPAdcCDoABCmigABjIAAIMZUcn
jgUDLxUAAgYxpXiARV+WQ9IdIOBIZp3EpAtwakHKmScCZIATexmIBAI4lrXwYxZkpUD/Xc3QCQgc
YEO3HA0GUGcaAsizJpcdwQrkNxANMOCyD5gFCfxBgsARYAUcuuxiVMKJ9dXxKlZ5DlhJQIK9kOAA

```
FAlAwMwagQ4sQINgm0XggkahB8iDcAS6zU8QEAEcksCXl33BAnoaANcwiAAqYCttWKvBq7CVER+J
R8oScDi3/KwBCihAMgiiAFSpKWW/IstwfwWb8ACgDB6Ckx2lSNaxnAAOAABXa0NyWQxg8Da/KkA8
VFCejAFAZIxIy02/BADUBOk2CFGBdfhRyxqpACGoAQDtHOAAZ4KFbASRpj9IjBBpakBQ+FjiAwYJ
AHuQ2FrMJDGOMzzKAGjLKAt9wAUE/xDRQVpnIASIR4mNQhMF4CPD81xoLQGQgCVqKxEl5seUATCC
JQPAA4Pw1kAI+ICXhvHJzITNkwkSgCxnOAA8g3GGBymUAvAMvlPu8V8H+WaaSOrGc07PAzTw4l1B
+c6KNihNSGwPIMPGAah5M6X9oQKBlFiaAODHYjLGnyPX1aDP4YcDGFBYNo7AZCpIgLXgTNWB7Ll8
ApEmndmE4xLrGscKWMCXrzUCFdSILLBRQBk0oOgLKztqUMuwbyIgZgwr+9XLrjYAwLKbhFiLS9bu
NhMX0K6bIqAD3lb2TlcwkGYv26bqZna5rW3TqEkhqRNI9l83gRBqv3vf1da3rJfdbv9+F9pGddU2
tf1dbmv5W+EF13bABW4jaj/8RvG2tsJvY3D4YnzfCLcRPzpOkInfaOIVx7CQL2xvhYDc2iIviL/V
vfKovfrVKYd4QqA2c5fbfNkN1/iNau5zbROk5zeNecKlvfOkb1zpTL+ROZmo76c3PehST4jliu1t
Sje86suOutB/rYARC/zNEi/I1gke9HczPAA4n3ra3b5zrkNd7XDvtgNMlnCMIOYwNyXiqQjkALl3
PeIYFhFBfmOjNyo7Ah4QeouTDfKtBn7nWSNgzuuOeZsb/fKZ77xCslbljxP933DewPoaQJebLgU1
BUi9sluebsffCAoFCVPigaZzhRT/INw2MqmkkL50ZxMe6V5XyOAs7/nkd/vpgvd28ZUPXwx8IC0I
aAxBbHee2ghOJoOswBM2QYIVwEEFB0geXSRTASdHZvokIF4yNpEeBixPATnKfm8I8qfenIo0qKpA
4JglAR0QAJtQAA3lDxdje/7zTxoDFVZFFhhQAi5lFhXAe0/AQ4ajP41ETp8RGhEABU7xAnDlDxVG
FalBGk/hFj9RAXBgLRxSHeXkYFhWH/UWAYMgIhXGHwWwGa43etDngz8IhAD3MYMQDi6TAIWhPRiQ
OBsAAQuAUQsQARshFgegAfjzAFAQDgKAEaJUAGtiFCIADw+AJyMQKQRAADnRKAcQ/xe/sUWC0gDX
1UkFwACqcgLOZD4koDti6FYVoAEWIAU6owBQ0FQSkADVgSrk8hzq0wAsExea8hHDdSwn4AEZIBsC
EAGcMAJv6CpFFAFPcALCYk4P8AGJwAhO1EmEJWcOACe08yt2QgIJsAIroCscoEtkoEd95X/OFIS7
yIu9eFP7IUUBAj6dEU4MOBOeUwHLJActQGBkkCcB0iUJ0EkCAFp86ADqAgAigABngQE8wgiB03oA
AAWvJgUEUABLggEEljRVVX0VAQnhcD5KdCwVUCid1F5nwwHJUAB3EG6iEhkZgBCqAwAYQhrgkI6c
IB8XYI4EZhScQES6MhGGkwB1Uv81XeIE45I9BeEE5zFBoYEwsnGOijQY+oIY/RIANeWLKamSu5gC
8rAAnMRMB/AUDsAIR6I6HPBWTVUsHHASGjA+AyM4IwMJMvJ7G+BrBYBuNcNNPMMPKgMxBLZFhjUW
cvAE6ZhE/MAVXwVoigQPgZg+C8AvKtA5fxIO1fEuPPNq6RMAo7FqFYBeOAMOe8ZNZqgAkXYr3nIU
Z3VI/hAv9EeJZINPd6ACgwAHAZBSVSUAg1AuCMEBVnEQPDUR5BQAZsAJGiCWRlFlK6mZmzl1MMgR
g2UbvBI4/AE+IxAvnLARlWmJuSUSPsEIceEbDDkRDAAWwPEAklOKczKSGFQBr4b/KmGRNYMhLGtx
KrxVHoinEmpCVq0ih24BKERCANYla4eSMoOFHoQxCNEJYeMnAZuTCFeDAU72AgKjEofDWxdUVYRB
AA/wXraYjp6xHQUADitwAPHwGxEQD4xwBwrACBhEfr9iBgTAKlZSErDHmQeKoHNnFCsgKGLyV7Rj
MrzkACuwaliZYcfGKm72AB4wF3nzPQHAWibjBC2IADVSGVJmLUs2oYuxSzHmD6TWawggZPjgGU+h
AGIiZE5wYwhxZbdpefiAZytgHa/mBHzCJiSWJWdWohlWhESkpDVibBlGRNaxACtAFiDaXD5aIBuC
G2ShAXwCoO+BdwlKpmWakhCWH1Sut3xmupLNx6ZvCqeep3hxSqd1aqd3iqd5qqd7yqf/feqnfwqo
gSqog0qohWqoyeemh6qoi8qoApeojQqpkSqpnbdwQfioFpd7mPpyGHdxpNcTJddsIOcxBAGqnRZ7
SPep1zKpq+pznkGolwpwSRdzBsqqtSpzFwar5SYQ+iYQJRd8g6dyENepvxp8ohpyThesuCp0uWqr
byerxGpzJdeDlDp8yCp7wBd0L/pu34Nwr6atlXqrfzV4B3dwpwqtzTqu1jJqSmcttIqru+py8LqL
r7aut6EAPaGt06qtBbGvKuemnfqtybp1D+eq7VquF3evBwc1Hzd06carF+er6AqE1BaxHGcjW9E4
82qtx1p0ChGq8BZ746EpUjet5oqt/wo6d0X6Y+4qsRiWoqARGtEmcMGipphaEBGVerJoG6mBERIA
kSh7rvvWIORkFodhLSchswThBDaYIrYHfDslMJi6CVBwXITHFfTCZE5BeCbCVAaDFqHRO9z4e2Zw
HgwCGslgG4yABx8ABQzwe8w6qZBSAsnAASXwAUbrGT2RsKM6qruaAHigVJxQAhJBrtr6ogogKIGn
cPKKJBEoKIxQAlCAXk5QAmkBCR8gKT1RrzjTSWinqvdqr4druAU7ECrgH+VUAl44EBVQAoNBrAvw
AR/gLiXAQcGnAiIwihhmOaBrr8MKgVGiqsHbH1FpQSXAkIH3cS+6kSVQAlFiAbE7Ev8bEQEl4E8l
gDHY8gHUm291SxwlsDOp9RMlcFUt2234sAGtKz4l0AK0w2ZkoQCUlm+WU2mEUQIZIGK8tFDzy0Z0
pnVf8mYFsQJ4sEVtZLeDQQapGwCXaw9omb/z6wBrOwIMy0Zz1r51xmOKBr+WY2qVAQAqgA/c2sFO
Cr+KlgIfQLyC2y6ooS0lBr8eU2kujBoFR2lQMbgNHG8eULkEwbqu20Y3Yy2bMBW0GzR40ALfgw9y
QCMBIAedkAhOIAKbULmC8gYfUDVOQMS7gsMSULHkeyMWoL5kcbsl0MTWmx5QMMAeaLf0shZ2KwIr
wLzM+y0fEDgWcL6RS2CzUL9yUAL/kEAgbhwgo8pjF1ACI6AAayvEnPABjXMHd8G8hlNV2Zu6KsC6
zFsUlMS600fIMzII2TtanLwSEsC8UJDCS3wqQkwQdVuEm1y5ZLEVZjwoJTAB/MAJUKDKWhwAIgAF
I7oJ2fsBdIHGlQsbVbXHeuwcS5u9X1EQXtwpA1G3xwsJy1QQUOy8VCx2nHQQBVYCgeMZe5y42YEA
J5C6CjcCfREP3srFynrAenYtrDsClPteIAAFVUMClZsXvTzPB2CQ3WW94cARrXvEHzABEeAfD0AG
sSsADFC/ggMFZGQjggsOXnyOyaAAS+xlImC9As3HHOJQiwwXsTsBzdXBcvABBCC4/1rcF72hx3dg
GiUQLsQ8AXrcNUSsAZALWjjD0tPLAHBwSwNhAQI8u+0iuCpt0S0jAp3wBDh8Bx3AAA1gBiVwB2aA
0FpcOmgcy33BAATAtmDkxcJYvK+SW3JwByQAB4hkyj39AS3QApwgKrTrgR9gB9fi1N9jBpC8vrXD
vMmwROesbPtQAvqguXW7ACAwt+mxxIYSuXcABSUQD16sGwjhxreEyOAwz1gLgQpUAguyAHOLEAtA
O5yHxxkwteDQBIk8fR2MuwH5AYORNXSiHQCgxzYlBXZ7B5f7AQrQF4Nx2xNRAmIy2CZBuyd5ENJc
EKiM1A1REFIABYAwu+skuIEDxf8C4AAbYNS5zcwl4J+L/ARIDBsQKAAasLYicAdwTBDPWzXVDQ6w
m9ixSzADIdzXBgUfcIHXZh+dkB2doQAlsAEeYwgpkR3OIzh+Aberai1mgAf7+Rwi/QSxjTFKMg8O
ULfx0E0RAC3OtMd9mNDYHA4H3C782boHTC8pYL0lR21SUODSTcMMgAftgg+4a2qbrCjZAQl40Nrv
/TOCgdSxMgECuMlPORco2LpePEnP/QBE/BmwXHssTX+se0hKu7azCw6GjUGQW29LbA86sWTnSyCS
/AH2YMKPC8sPQL3MQwZwAEYgXt423TgP0ABNsSEEQdM8/QHJ0GwBUAZimLtstMf/AzHXGOOEq8JG
jqHX3nY3qdtpgutD+KC+H5LnkOuFg8Qa4GO8ma3OggsHbtwtDoC7C+ABH+BDUhDnAWBRWNtzYGK3
GDMLhkwL+xUFgBBpOw7FECAS2ZxEJfAEk3cdTl0j0mTbqY3hug0ODkC9T0Dqg2TGA2jK1c0nRATK
rnsb51vM2QEtDg0A0uwAd4BNkEsvZOA5Ca3MBDwBBGwVSEwgzBR7gSjrqlICZOEAz5wQz83TRAxG
CYAQCBDFX5LovDLYA5O7E7kXAb6qCCDj7828x0TAHyDSVXNQkcu2iTC973OQVxi5FvgBBNYXHyDj
zuHhANAAJQAfTb3MuSe4Rk5T/3meHpcbAA4gAR+QRrEr8BSfxtFnt+9dzhyg8pDu43QBgYa8M/hw
xW+uw9bd8CasaP2h8EJcE8ar21MOCU/A82w78Sqgx+8NBQqFw2yb2O0yvQrP1Z6x6ZFrx9EsQuvH
vFTcV9MHRn2R2GiSHiIg43brURYt8PKDxwwd6MCqABXCeKCc7qciAE8AB/1YIVyVYWLjUQNRBgKA
byuwnkGzQc0lBdDmbGbgDw7gVlHzAPEDpQuUeikWy9e2CVP2hxPQAAIAG/5AAYiXzJtAAIXp+bBB
HrCxAC+gKZ+F+qZ7SOUI0oLDXdsCLPbmAKe/CdojOAIALY+fCHdfb85GAAvGD/8kAAH7EGnPQQAU
8ASQbxSqPwGSQm32ODz7kJkJUY4EoPp9fyokcHFVuBvf4wAnYIaq3wB/Lv4F08GoX/f9lhAD2C15
+q8Qp4ohDhAABA5UMNDgQYQJFS4UyI8hw4IIHT6UCEBBRIoZNW7k2NHjR5AhM05EiPFhQZIiD/pT
uXEEHkgkPJp8yNIiSJsMc1Lc2XJhT59BhQ4lWlQh0JsfaRpdeHHBA6YHIy5dGtXqVaxZtXKsujUh
Uo8pOyogibFr0pNRz2Zd69XtW4JSr64t2Dbu3bEGqeKtevYiWrwQIYINLPinz6lNBdKF29jxY8iR
JU+mXNnyZcyZNW/m3NnzZ9AUoUWPJl3a9GnUqVWvZt3a9WvYsWUHz6Zd23blgAA7
```

Hint: Be sure to decode *only* Base 64 content. *Don't cut or paste content before the R0lGOD… or after the final …bLgAA7*

Note also that you are converting this encoded data into a specified content-type (*i.e.*, a GIF image); so, *some online converters may require you to save the decoded base64 data to a file with a .gif extension before you can view the image*. If one online converter doesn't work for you, another likely will.

ANSWER:____

Question 4: Look at the base64 data in Question 3, above. Can you spot any combinations of letters within the data that form English words or names of three, four or even five letters? For example, I spotted "GOD," "PILE," "PUB," "NECK" and "MOM." List at least two others you found:

____

____

Think about the ways the occurrence of these happenstance words and names may complicate electronic searches for relevant documents?

Question 5: Find the Secret Word

Question 5: Download the file at http://www.craigball.com/Find_the_Secret_Word.zip Extract the two files in the Zip and use what you've learned in your slack space and encoding readings (along with your use of a hex viewer) to Find the Secret Word. Once you've found the secret word (by solving a <u>three-part</u> puzzle), enter the Secret Word below (and e-mail it before class with your other answers):

The secret word is: _________________________________

Encoding and Steganography

We record information every day using 26 symbols called "the alphabet," abetted by helpful signals called "punctuation." So, you could say that we write in hexavigesimal (Base26) encoding.

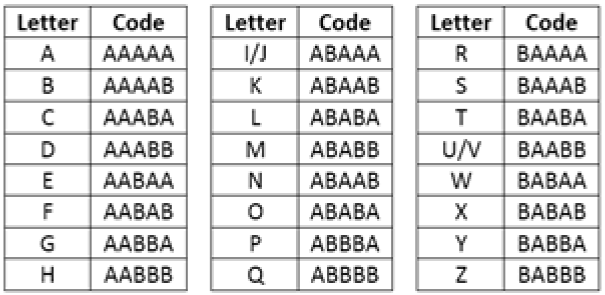
"Binary" or Base2 encoding is notating information using just two symbols. It's often said that "computer data is stored as ones and zeroes;" but that's a fiction. In fact, binary data is stored physically, electronically, magnetically or optically using mechanisms that permit the detection of two clearly distinguishable "states," whether manifested as faint voltage potentials (*e.g.,* thumb drives), polar magnetic reversals (*e.g.,* spinning hard drives) or pits on a reflective disc deflecting a laser beam (*e.g.,* DVDs). Ones and zeroes are simply a useful way to *notate* those states. You could use *any* two symbols as binary characters, *or even two discrete characteristics of the "same" symbol.*

I free you from the trope of ones and zeroes to plumb the evolution of binary communication and explore an obscure coding cul-de-sac called Steganography, from the Greek, meaning "covered writing." But first, we need an aside of Bacon.

I mean, of course, lawyer and statesman Sir Francis Bacon (1561-1626). Among his many accomplishments, Bacon conceived a bilateral cipher (a "code" in modern parlance) enabling the hiding of messages *omnia per omnia,* or "anything by anything."

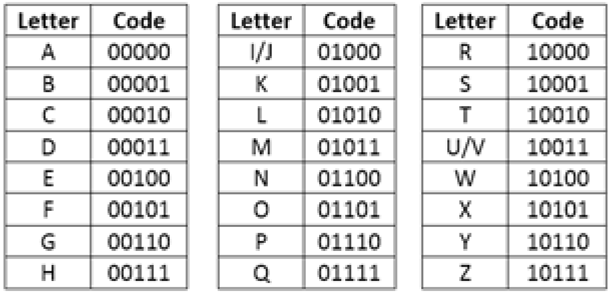
Bacon's cipher used the letters "A" and "B" to denote binary values; but if we use ones and zeros instead, we see the straight line from Bacon's clever cipher to modern ASCII and Unicode encoding.

As with modern computer encoding, we need multiple binary digits ("bits") to encode or "stand in for" the letters of the alphabet. Bacon chose the five-bit sets at right:



Letter	Code	Letter	Code	Letter	Code
A	AAAAA	I/J	ABAAA	R	BAAAA
B	AAAAB	K	ABAAB	S	BAAAB
C	AAABA	L	ABABA	T	BAABA
D	AAABB	M	ABABB	U/V	BAABB
E	AABAA	N	ABAAB	W	BABAA
F	AABAB	O	ABABA	X	BABAB
G	AABBA	P	ABBBA	Y	BABBA
H	AABBB	Q	ABBBB	Z	BABBB

If we substitute ones and zeroes (right), Bacon's cipher starts to look uncannily like contemporary binary encodings.

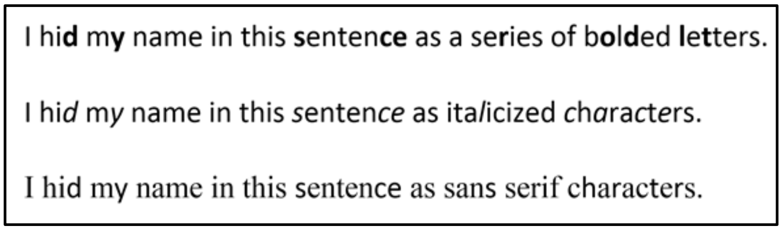


Letter	Code	Letter	Code	Letter	Code
A	00000	I/J	01000	R	10000
B	00001	K	01001	S	10001
C	00010	L	01010	T	10010
D	00011	M	01011	U/V	10011
E	00100	N	01100	W	10100
F	00101	O	01101	X	10101
G	00110	P	01110	Y	10110
H	00111	Q	01111	Z	10111

Why *five* bits and not three or four? The answer lies in binary math ("*Oh no! Not MATH!!*"). Wait, *wait*; it won't hurt. I promise!

If you have one binary digit (2^1), you have only two unique states (one or zero), so you can only encode two letters, say A and B. If you have two binary digits (2^2 or 2×2), you can encode four letters, say A, B, C and D. With three binary digits (2^3 or 2x2x2), you can encode eight letters. Finally, with four binary digits (2^4 or 2x2x2x2), you can encode just sixteen letters. *So, do you see the problem in trying to encode the letters of a 26-letter alphabet?* You must use at least *five* binary digits (2^5 or 32) unless you are content to forgo ten letters.

Sir Francis Bacon wasn't especially interested in encoding text as bits. His goal was to *hide* messages in any medium, permitting a clued-in reader to distinguish between differences lurking in plain sight. Those differences—whatever they might be—serve to denote the "A" or "B" in Bacon's steganographic technique. For example:

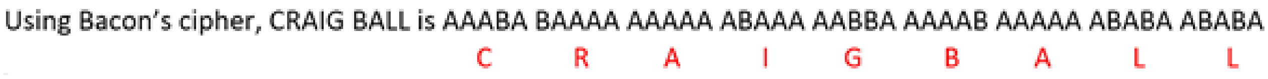


I hi**d** m**y** name in this **s**enten**ce** as a se**r**ies of b**o**l**d**ed **l**e**t**ters.

I hi*d* m*y* name in this *s*enten*ce* as ita*l*icized *c*h*a*ra*c*t*e*rs.

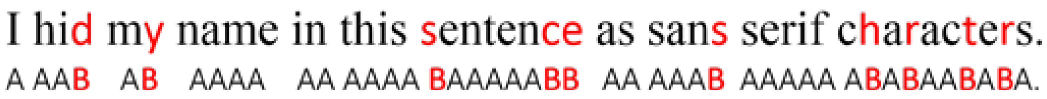
I hid my name in this sentence as sans serif characters.

That last one is quite subtle, right? Here's how it's done:



Using Bacon's cipher, CRAIG BALL is AAABA BAAAA AAAAA ABAAA AABBA AAAAB AAAAA ABABA ABABA

To conceal my name in each of the respective examples, every unbolded/unitalicized/serif character signifies an "A" in Bacon's cipher and every boldface/italicized/sans serif character signifies a "B" (ignore the spaces and punctuation). The bold and italic approaches look wonky and could arouse suspicion, but if the fonts are chosen carefully, the absence of serifs should go unnoticed. Take a closer look to see how it works:



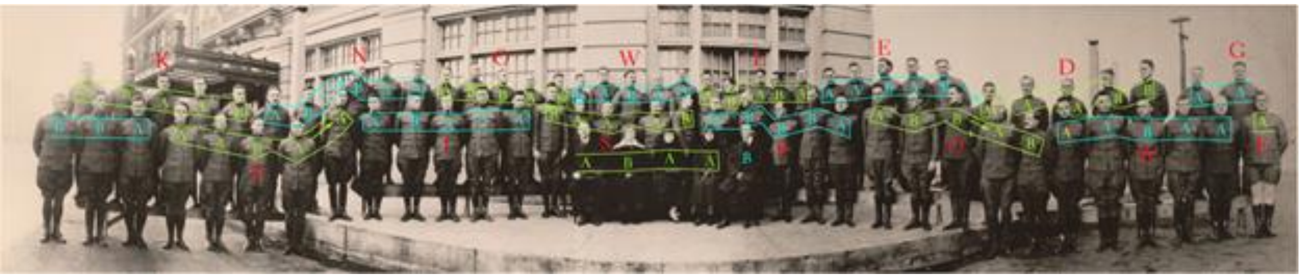
I hid my name in this sentence as sans serif characters.

A AAB AB AAAA AA AAAA BAAAAABB AA AAAB AAAAA ABABAABABA.

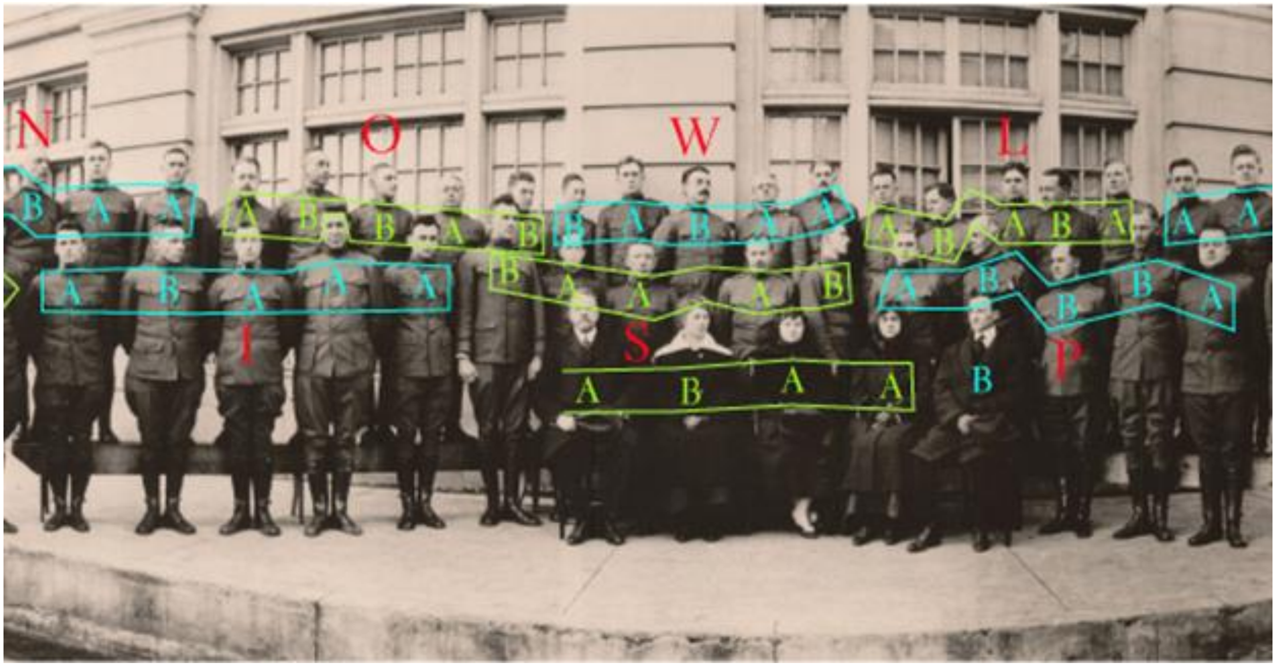
In my examples, I've used Bacon's cipher to hide text within text, but it can as easily hide messages in almost anything. My favorite example is the class photo of World War I cryptographers trained in Aurora, Illinois by famed cryptographers, William and Elizabeth Friedman.[42] Before they headed for France, the newly minted codebreakers lined up for the cameraman; but there's more going on here than meets the eye.

[42] For this material, I'm indebted to "How to Make Anything Signify Anything" by William H. Sherman in Cabinet no. 40 (Winter 2010-2011).

Taking to heart *omnia per omnia*, the Friedmans ingenuously encoded Sir Francis Bacon's maxim "knowledge is power" within the photograph using Bacon's cipher. The 71 soldiers and their instructors convey the cipher text by facing or looking away from the camera. Those facing denote an "A." Those looking away denote a "B." There weren't quite enough present to encode the entire maxim, so the decoded message actually reads, "KNOWLEDGE IS POWE." Here's the decoding:



A closer look:



Steganography is something most computer forensic examiners study but rarely use. Still, it's a fascinating discipline with a history reaching back to ancient Greece, where masters tattooed secret messages on servants' shaved scalps and hit "Send" once the hair grew back. Digital technology brought new and difficult-to-decipher steganographic techniques enabling images, sound and messages to hitch a hidden ride on a wide range of electronic media.

Processing Part II

Why Care about Encoding?

All this code page and Unicode stuff matters because of the vital role that electronic search plays in the identification and exploration of digital evidence. Modern evidence isn't sitting in a box, but must be teased out of databases, media and vast repositories. Little of it is organized. It's "**unstructured" data**. If we index an information item for search using the wrong code page, the information in the item won't be extracted, won't become part of the index and, accordingly, won't be found even when the correct search terms are run. Many older search tools and electronic records are not Unicode-compliant. Worse, because encrypted content may include a smattering of ASCII text, a search tool may conclude it's encountered a document encoded with Latin-1 and won't flag the item as having failed to index. In short, encoding issues carry real-world consequences, particularly when foreign language content is in the collection. If you aren't aware of the limits of the processing tools you, your opponents or service providers use, you can't negotiate successful protocols. What you don't know *can* hurt you.

The many ways in which data is encoded—and the diverse ways in which collection, processing and search tools identify the multifarious encoding schemes—lie at the heart of significant challenges and costly errors in e-discovery. It's easy to dismiss these fundamentals of information technology as too removed from litigation to be worth the effort to explore them; but, understanding encoding and the role it plays play in processing, indexing and search will help you realize the capabilities and limits of the tools you, your clients, vendors and opponents use.

Let's look at these issues and file formats through the lens of a programmer making decisions about how a file should function.

Hypothetical: TaggedyAnn

Ann, a programmer, must create a tool to generate nametags for attendees at an upcoming law school reunion. Ann wants her tool to support different label stock and multiple printers, fonts, font sizes and salutations. It will accept lists of pre-registrants as well as create name tags for those who register onsite. Ann will call her tool TaggedyAnn. TaggedyAnn has never existed, so no computer operating system "knows" what to do with TaggedyAnn data. Ann must design the necessary operations and associations.

Remembering our mantra, "ESI is just numbers," Ann must lay out those numbers in the data files so that the correct program—her TaggedyAnn program—will be executed against TaggedyAnn data files, and she must structure the data files so that the proper series of numbers will be pulled from the proper locations and interpreted in the intended way. Ann must develop the **file format**.

A file format establishes the way to encode and order data for storage within a file. Ideally, Ann will document her file format as a published specification, a blueprint for the file's construction. That way, anyone trying to develop applications to work with Ann's files will know what goes where and how to interpret the data. But Ann may never get around to writing a formal file specification or, if she believes her file format to be a trade secret, Ann may decide not to reveal its structure publicly. In that event, others seeking to read TaggedyAnn files must reverse engineer the files to deduce their structure and fashion a document filter that can accurately parse and view the data.

In terms of complexity, file format specifications run the gamut. Ann's format will likely be simple—perhaps just a few pages—as TaggedyAnn supports few features and functions. By contrast, the published file specifications for the retired binary forms of Microsoft Excel, PowerPoint and Word files run to 1,125, 649 and 578 pages, respectively. Microsoft's latest file format specification for the .PST file that stores Outlook e-mail occupies 193 pages. Imagine trying to reverse engineer such complexity without a published specification! For a peek into the Byzantine structure of a Word .DOCX file, see Illustration 2: Anatomy of a Word DOCX File.

File Type Identification

In the context of the operating system, Ann must link her program and its data files through various means of **file type identification**.

File type identification using binary file signatures and file extensions is an essential early step in e-discovery processing. Determining file types is a necessary precursor to applying the appropriate document filter to extract contents and metadata for populating a database and indexing files for use in an e-discovery review platform.

File Extensions

Because Ann is coding her tool from scratch and her data files need only be intelligible to her program, she can structure the files any way she wishes, but she must nevertheless supply a way that computer operating systems can pair her data files with only her executable program. Else, there's no telling what program might run. You can force Word to open a PDF file, but the result will be unintelligible.

Filename extensions or just "file extensions" are a means to identify the contents and purpose of a data file. Though her executable files must carry the file extension .EXE, any ancillary files Ann creates can employ almost any three-letter or -number file extension Ann desires *so long as her choice doesn't conflict with one of the thousands of file extensions already in use*. That means that Ann can't use .TGA because it's already associated with Targa graphics files, and she can't use .TAG because that signals a DataFlex data file. So, Ann settles on .TGN as the file extension for TaggedyAnn files. That way, when a user loads a data file with the .TGN extension, Windows can direct its contents to the TaggedyAnn application and not to another program that won't correctly parse the data.

Binary File Signatures

But Ann needs to do more. Not all operating systems employ file extensions, and because extensions can be absent or altered, file extensions aren't the most reliable way to denote the purpose or content of a file. Too, file names and extensions are *system metadata*, so they are not stored inside the file. Instead, computers store file extensions within the *file table* of the storage medium's file system. Accordingly, Ann must fashion a unique **binary header signature** that precedes the contents of each TaggedyAnn data file in order that the contents are identifiable as TaggedyAnn data and directed solely to the TaggedyAnn program for execution.

A binary file signature (also called a **magic number**) will typically occupy the first few bytes of a file's contents. It will always be hexadecimal values; however, the hex values chosen may correspond to an intelligible alphanumeric (ASCII) sequence. For example, PDF document files begin 0x 25 50 44 46 2D,[43] which is **%PDF-** in ASCII. Files holding Microsoft tape archive data begin 0x 54 41 50 45, translating to **TAPE** in ASCII. Sometimes, it's a convoluted correlation, *e.g.,* Adobe Shockwave Flash files have a file extension of SWF, but their file signature is 0x 46 57 53, corresponding to **FWS** in ASCII. In other instances, there is no intent to generate a meaningful ASCII word or phrase using the binary header signatures; they are simply numbers in hex. For example, the header signature for a JPG image is 0x FF D8 FF E0 which translates to the gibberish **ÿØÿà** in ASCII.

Binary file signatures often append characters that signal variants of the file type or versions of the associated software. For example, JPG images in the JPEG File Interchange Format (JFIF) use the binary signature 0x FF D8 FF E0 00 10 4A 46 49 46 or **ÿØÿà JFIF** in ASCII. A JPG image in the JPEG Exchangeable Image File Format (Exif, commonly used by digital cameras) will start with the binary signature 0x FF D8 FF E1 00 10 45 78 69 66 or **ÿØÿá Exif** in ASCII.

Ann peruses the lists of file signatures available online and initially thinks she will use 0x 54 41 47 21 as her binary header signature because no one else appears to be using that signature and it corresponds to **TAG!** in ASCII.

But, after reflecting on the volume of highly-compressible text that will comprise the program's data files, Ann instead decides to store the contents in a ZIP-compressed format, necessitating that the binary header signature for TaggedyAnn's data files be 0x 50 4B 03 04, **PK..** in ASCII. All files compressed with ZIP use the **PK..** header signature (again, because Phil Katz, the programmer who wrote the ZIP compression tool, chose to flag his file format with his initials).

[43] The leading "0x" signals that the data that follows is notated in hexadecimal.

How can Ann ensure that only TaggedyAnn opens these files and they are not misdirected to an unzipping (decompression) program? Simple. Ann will use the .TGN extension to prompt Windows to load the file in TaggedyAnn and TaggedyAnn will then read the **PK..** signature and unzip the contents to access the compressed contents.

But, what about when an e-discovery service provider processes the TaggedyAnn files with the mismatched file extensions and binary signatures? Won't that throw things off? It could. We'll return to that issue when we cover compound files and recursion; but first, an exercise to illustrate what you've read.

❁ Exercise 7: Encoding: File Extensions

GOALS: The goals of this exercise are for the student to:

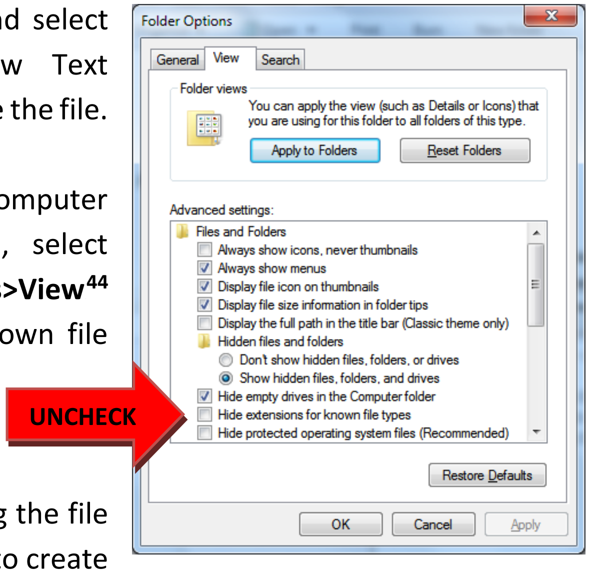
1. Through experimentation, understand the function of file extensions in Windows or Mac; and
2. Assess the utility and reliability of file extensions as a filtering tool in e-discovery.

OUTLINE: Section A: Students using Microsoft Windows operating systems will modify file extensions for a known file type to determine the impact on file appearance and application association. **Section B:** Mac users will investigate the treatment of file extensions and "Open With" option in the Info window. **Do Section 7(A) or Section 7(B), not both.**

7(A). File Extensions in Windows (Students with Windows machines do this exercise)

Step 1: Right click on an open area of your Desktop and select **New>Text Document**. Open the document "New Text Document.txt" that was created and type your name. Save the file.

NOTE: If you cannot see the file's extension of .txt, your computer may be configured to hide file extensions. If so, select **Start>Computer>Organize>Folder and Search Options>View**[44] and **<u>un</u>check** the folder option "hide extensions for known file types."



Folder Options

Folder views

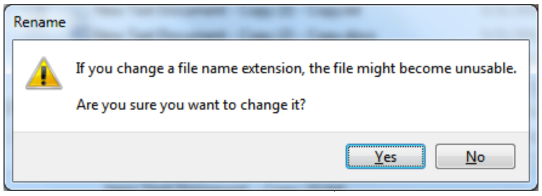
You can apply the view (such as Details or Icons) that you are using for this folder to all folders of this type.

Advanced settings:

Files and Folders
- □ Always show icons, never thumbnails
- ☑ Always show menus
- ☑ Display file icon on thumbnails
- ☑ Display file size information in folder tips
- □ Display the full path in the title bar (Classic theme only)
- Hidden files and folders
  - ○ Don't show hidden files, folders, or drives
  - ● Show hidden files, folders, and drives
- ☑ Hide empty drives in the Computer folder
- □ Hide extensions for known file types
- □ Hide protected operating system files (Recommended)

Step 2: Locate the New Text Document.txt file you just created and, ***while depressing the Ctrl key,*** click and drag the file to create a duplicate of the file. **Do this four more times** to create five identical copies of the file (Windows will change the filenames slightly to allow them to co-exist in the same folder).

Step 3: Right click on any copy of New Text Document.txt and select Rename. Change <u>only</u> the extension of the file from .txt to .doc. Hit the Enter key to accept the change. You should get a warning message asking if you are sure you want to change the extension. Select "Yes."



Rename

If you change a file name extension, the file might become unusable.

Are you sure you want to change it?

[44] In Windows 8, you can get there by typing 'folder options" at the Start screen to locate the Folder Options menu, then click the Folder Options menu and select the View tab and Advanced Options to continue to <u>un</u>check the option "hide extensions for known file types" option.

Step 4: Change the file extensions for each of the four other copies of the file to .zip, .xls, .ppt and .htm.

Did you notice any change to the icons used by Windows to reflect the file type?

Step 5: Take a screen shot of the folder showing the icons for all six files. You will submit this screen shot as your work on the exercise. This is all you need to submit for Exercise 7.

Step 6: Try to launch each of the five copies with their new extensions. **What application launches for each?**

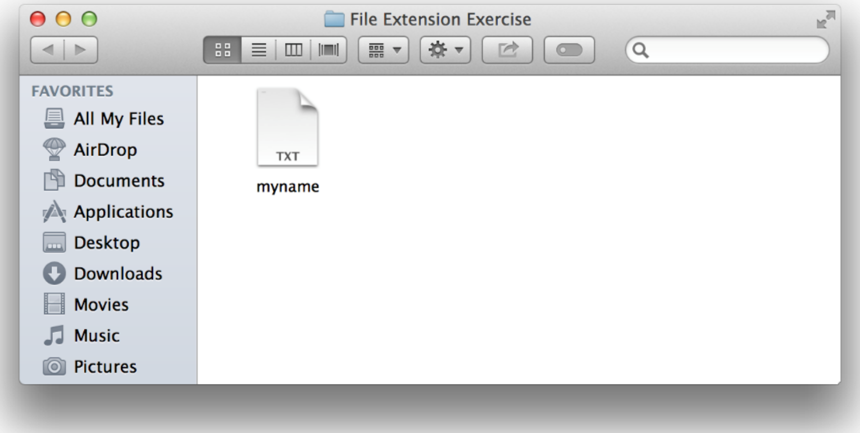
Step 7: Save one of the renamed files with <u>no</u> file extension and launch it. **What happens?**

Discussion questions:

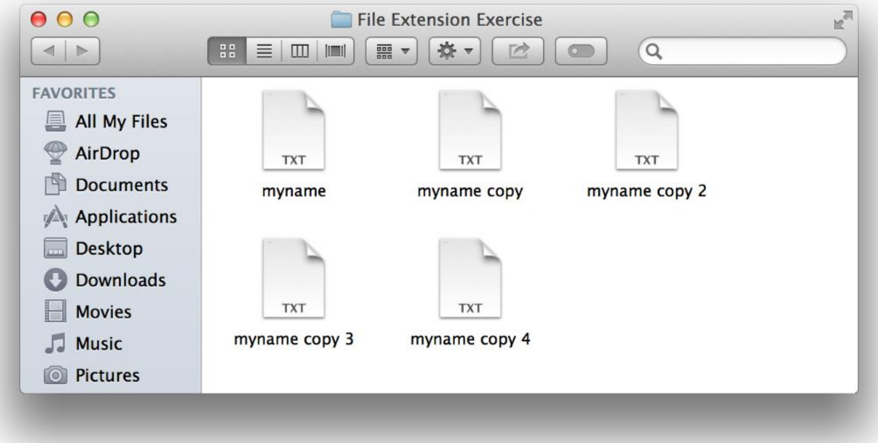
1. How might a file acquire a file extension other than the correct one or lack a file extension?
2. How does Windows determine what application to run for each file extension?
3. How might you determine what application should be used to open a file with an unfamiliar file extension? Check out FileExt.com.

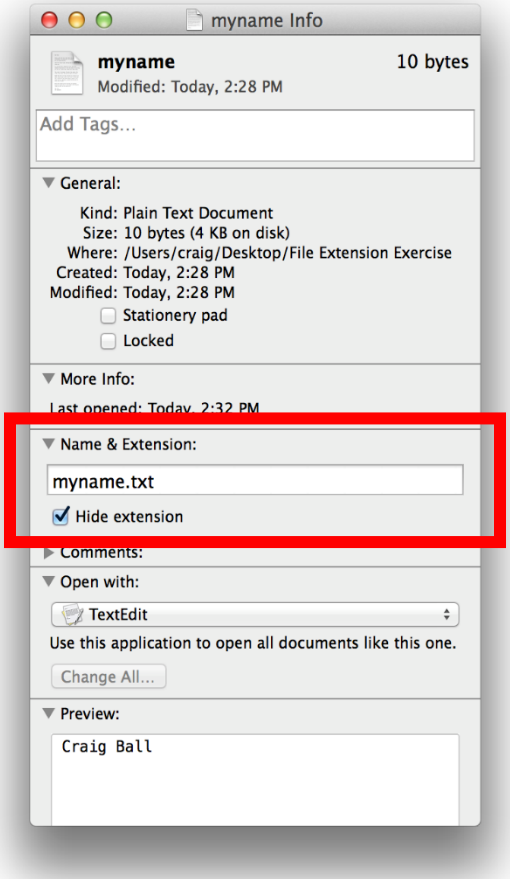
B. File Extensions in MacOS (Students with MacOS machines do this exercise)

Step 1: Create an empty folder called "File Extension Exercise" on your Desktop. Run the text editor called TextEdit (found in your Applications folder). Create a new blank document by either clicking File>New or Command-N. Type your name in this new blank document and save the file as "**myname.txt**" in your exercise folder. Be sure to save it in the new folder, like so:



Step 2: Locate the file **myname.txt** you just created, select it and type *Command-D* (⌘**-D**) four times to create four duplicates of the file. You can also right click and select Duplicate to achieve the same result, repeating The Apple Mac operating system hides file extensions by default; so, depending upon your system settings, you may or may not see file extensions for the five files. In the next steps, you will change settings, if necessary, to show file extensions. Your folder should now look like this:





myname Info

myname 10 bytes
Modified: Today, 2:28 PM

Add Tags...

General:

Kind: Plain Text Document
Size: 10 bytes (4 KB on disk)
Where: /Users/craig/Desktop/File Extension Exercise
Created: Today, 2:28 PM
Modified: Today, 2:28 PM
☐ Stationery pad
☐ Locked

More Info:

Last opened: Today, 2:32 PM

Name & Extension:

myname.txt
☑ Hide extension

Comments:

Open with:

TextEdit
Use this application to open all documents like this one.
Change All...

Preview:

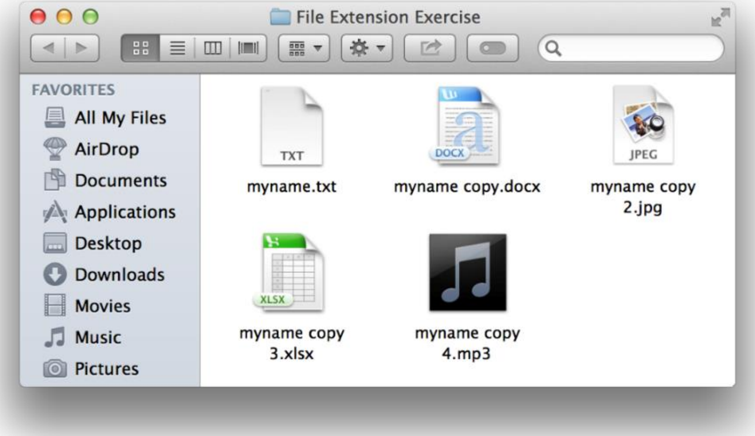
Craig Ball

Step 3: Right click on the file **myname.txt** and select **Get Info**. As you can see in the figure at right, "Craig Ball" is nine letters and a space, so the file size (top right corner) is 10 bytes, reflecting the ten ASCII characters comprising its contents.

Note in the **Name & Extensions** region that the option "Hide Extensions" is checked. This option can be turned on and off for individual files. If it's checked for your machine, uncheck the selection and note that the file icon now shows the name and extension, **myname.txt**.

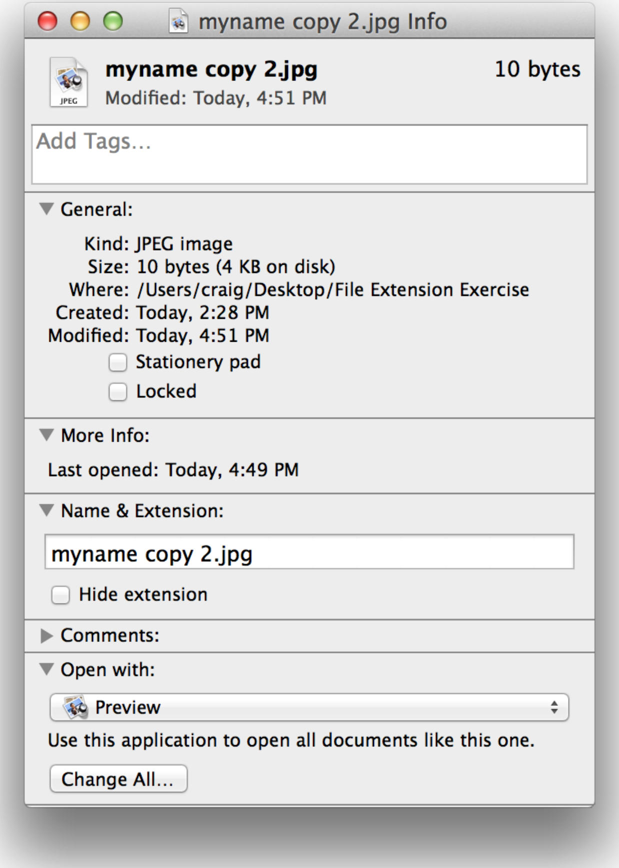
Step 4: Select the first duplicate file you created called **myname copy** and Get Info. In the **Name & Extensions** region, change the file's extension from .txt to .docx and uncheck "Hide extension." Hit the Enter key to accept the change. and when your system asks, "**Are you sure you want to change the extension from ".txt" to ".docx"?**" select "**Use .docx.**"

Step 5: Do the same thing for each of the three other copies of the file: uncheck "Hide extension" and change the file extensions of each to, respectively, .jpg, .xlsx and .mp3. Now, the contents of your exercise folder should look like mine at right.



Step 6: Take a screenshot of your exercise folder showing the icons for all five files.[45] You will submit the screenshot to show your work on the exercise. This is all you need to submit for Exercise 7.

Step 7: Try to launch each of the altered icons and note what application opens (or fails to open). If you have Microsoft Word installed, clicking on **myname copy.docx** will launch Word instead of TextEdit. It works because Word knows how to parse text. Launching the next three files with altered extensions will prompt your system to try to launch the associated application for JPG images, spreadsheets and music, but all will fail because the applications are incompatible with the contents of the file. Still, that didn't stop the operating system from changing the appearance of the icons to show the incompatible applications or from trying to launch them for incompatible files.


myname copy 2.jpg Info

myname copy 2.jpg 10 bytes
Modified: Today, 4:51 PM

Add Tags…

General:

Kind: JPEG image
Size: 10 bytes (4 KB on disk)
Where: /Users/craig/Desktop/File Extension Exercise
Created: Today, 2:28 PM
Modified: Today, 4:51 PM
☐ Stationery pad
☐ Locked

More Info:

Last opened: Today, 4:49 PM

Name & Extension:

myname copy 2.jpg
☐ Hide extension

Comments:

Open with:

Preview
Use this application to open all documents like this one.
Change All…

Step 8: Unlike Microsoft Windows, Mac OS doesn't slavishly rely on file extensions to pair files with the proper applications. Pull up the Get Info screen for named **myname copy.jpg** and locate the "Open with:" panel (it probably displays Preview or another photo viewer app). Select the pull-down menu and select "Other…" In the following menu, change "**Enable: Recommended Applications**" to "**Enable: All Applications**." Scroll down the list of applications, select "**TextEdit**" and click Add. The file still displays a JPG icon; but now, launching the file starts TextEdit, *not* Preview.

[45] On my Mac, the key combo **Shift-Command-4 plus space bar** generates a camera cursor that saves a screenshot of any folder or open window I select by clicking my mouse. A screenshot captured this way is stored on the desktop and features a white border around the window with a bit of a drop shadow.

Common File Extensions and Associated File Types

Extension	Associated File Type
Text Files	
.doc	Microsoft Word Document
.docx	Microsoft Word Open XML Document
.pages	Pages Document
.rtf	Rich Text Format File
.txt	Plain Text File
Mail Files	
.edb	Microsoft Exchange database
.eml	Outlook Express Mail Message
.mime	Multi-Purpose Internet Mail Message File
.msg	Outlook Mail Message
.nsf	IBM/Lotus Notes container file
.ost	Microsoft Outlook synchronization file
.pst	Microsoft Outlook container file
Data Files	
.csv	Comma Separated Values File
.dat	Data File
.key	Keynote Presentation
.pps	PowerPoint Slide Show
.ppt	PowerPoint Presentation
.pptx	PowerPoint Open XML Presentation
.vcf	vCard File
.xml	XML File
Audio Files	
.m3u	Media Playlist File
.mid	MIDI File
.mp3	MP3 Audio File
.mpa	MPEG-2 Audio File
.ra	Real Audio File
.wav	WAVE Audio File
.wma	Windows Media Audio File
Video Files	
.asf	Advanced Systems Format File
.asx	Microsoft ASF Redirector File
.avi	Audio Video Interleave File

.mov	Apple QuickTime Movie
.mp4	MPEG-4 Video File
.mpg	MPEG Video File
.rm	Real Media File
.swf	Shockwave Flash Movie
.vob	DVD Video Object File
.wmv	Windows Media Video File

Raster Image Files

.bmp	Bitmap Image File
.gif	Graphical Interchange Format File
.jpg	JPEG Image File
.png	Portable Network Graphic
.psd	Adobe Photoshop Document
.tif	Tagged Image File

Vector Image Files

.ai	Adobe Illustrator File
.drw	Drawing File
.eps	Encapsulated PostScript File
.ps	PostScript File
.svg	Scalable Vector Graphics File

Page Layout Files

.pdf	Portable Document Format File

Spreadsheet Files

.wks	Works Spreadsheet
.xls	Excel Spreadsheet
.xlsx	Microsoft Excel Open XML Spreadsheet

Database Files

.db	Database File
.dbf	Database File
.mdb	Microsoft Access Database
.pdb	Program Database
.sql	Structured Query Language Data

Executable (Program) Files

.app	Mac OS X Application
.bat	DOS Batch File
.cgi	Common Gateway Interface Script

.com	DOS Command File
.exe	Windows Executable File
.gadget	Windows Gadget
.jar	Java Archive File
.pif	Program Information File
.vb	VBScript File
.wsf	Windows Script File

CAD Files

.dwg	AutoCAD Drawing Database File
.dxf	Drawing Exchange Format File

GIS Files

.mxd	Map Exchange Document
.kml	Keyhole Markup Language File

Web Files

.asp	Active Server Page
.cer	Internet Security Certificate
.csr	Certificate Signing Request File
.css	Cascading Style Sheet
.htm	Hypertext Markup Language File
.html	Hypertext Markup Language File
.js	JavaScript File
.jsp	Java Server Page
.php	Hypertext Preprocessor File
.rss	Rich Site Summary
.xhtml	Extensible Hypertext Markup Language File

Font Files

.fnt	Windows Font File
.fon	Generic Font File
.otf	OpenType Font
.ttf	TrueType Font

System Files

.cab	Windows Cabinet File
.cpl	Windows Control Panel Item
.cur	Windows Cursor
.dll	Dynamic Link Library
.dmp	Windows Memory Dump
.drv	Device Driver

.lnk	File Shortcut
.sys	Windows System File

Settings and Configuration Files

.cfg	Configuration File
.ini	Windows Initialization File
.keychain	Mac OS X Keychain File
.prf	Outlook Profile File

Encoded Files

.bin	Macbinary Encoded File
.hqx	BinHex 4.0 Encoded File
.uue	Uuencoded File

Compressed Files

.7z	7-Zip Compressed File
.deb	Debian Software Package
.gz	Gnu Zipped Archive
.pkg	Mac OS X Installer Package
.rar	WinRAR Compressed Archive
.sit	Stuffit Archive
.sitx	Stuffit X Archive
.tar.gz	Tarball File
.zip	Zipped File
.zipx	Extended Zip File

Disk Image Files

.dmg	Mac OS X Disk Image
.iso	Disc Image File
.toast	Toast Disc Image
.vcd	Virtual CD

Developer Files

.c	C/C++ Source Code File
.class	Java Class File
.cpp	C++ Source Code File
.cs	Visual C# Source Code File
.dtd	Document Type Definition File
.fla	Adobe Flash Animation
.java	Java Source Code File
.m	Objective-C Implementation File
.pl	Perl Script

File Structure

Now, Ann must decide how she will structure the data within her files. Once more, Ann's files need only be intelligible to her application, so she is unconstrained in her architecture. This point becomes important as the differences in file structure are what make processing and indexing essential to electronic search. There are thousands of different file types, each structured in idiosyncratic ways. Processing normalizes their contents to a common searchable format.

Some programmers dump data into files in so consistent a way that programs retrieve particular data by offset addressing; that is, by beginning retrieval at a specified number of bytes from the start of the file (*offset* from the start) and retrieving a specified extent of data from that offset forward (*i.e.,* grabbing *X* bytes following the specified offset).

Offset addressing could make it hard for Ann to add new options and features, so she may prefer to implement a chunk- or directory-based structure. In the first approach, data is labeled within the file to indicate its beginning and ending, or it may be tagged ("marked up") for identification. The program accessing the data simply traverses the file seeking the tagged data it requires and grabs the data between tags. There are many ways to implement a chunk-based structure, and it's probably the most common file structure. A directory-based approach constructs a file as a small operating environment. The directory keeps track of what's in the file, what it's called and where it begins and ends. Examples of directory-based formats are ZIP archive files and Microsoft Office files after Office 2007. Ann elects to use a mix of both approaches. Using ZIP entails a directory and folder structure, and she will use tagged, chunked data within the compressed folder and file hierarchy.

Data Compression

Many common file formats and containers are compressed, necessitating that e-discovery processing tools be able to identify compressed files and apply the correct decompression algorithm to extract contents.

Compression is miraculous. It makes modern digital life possible. Without compression, we wouldn't have smart phones, digitized music, streaming video or digital photography. Without compression, the web would be a much different, much duller place.

Compression uses algorithms to reduce the space required to store and the bandwidth required to transmit electronic information. If the algorithm preserves all compressed data, it's termed "**lossless compression**." If the algorithm jettisons data deemed expendable, it's termed "**lossy compression**."

JPEG image compression is lossy compression, executing a tradeoff between image quality and file size. Not all the original photo's graphical information survives JPEG compression. Sharpness and color depth are sacrificed, and compression introduces distortion in the form of rough margins called "jaggies." We likewise see a loss of fidelity when audio or video data is compressed for storage or transmission (*e.g.*, as MPEG or MP3 files). The offsetting benefit is that the smaller file sizes facilitate streaming video and storing thousands of songs in your pocket.

In contrast, file compression algorithms like ZIP are lossless; decompressed files perfectly match their counterparts before compression. Let's explore how that's done.

One simple approach is **Run-Length Encoding**. It works especially well for images containing consecutive, identical data elements, like the ample white space of a fax transmission. Consider a black and white graphic where each pixel is either B or W; for example, the image of the uppercase letter "E," below left:

```
WWBBBBBWW
WWBBWWWWW
WWBBBBWWW
WWBBWWWWW
WWBBBBBWW
```

The image at left requires 45 characters but we can write it in 15 fewer characters by adding a number describing each sequence or "run,“ *i.e.,* 2 white pixels, 5 black, 2 white.

```
2W5B2W
2W2B5W
2W4B3W
2W2B5W
2W5B2W
```

We've compressed the data by a third. Refining our run-length compression, we substitute a symbol (|) for each 2W and now need just 23 characters to describe the graphic, like so:

```
|5B|
|2B5W
|4B3W
|2B5W
|5B|
```

We've compressed the data by almost half but added overhead: we must now supply a dictionary defining |=2W.

Going a step further, we swap in symbols for 5B (\), 2B (/) and 5W (~), like so:

```
|\|
|/~
|4B3W
|/~
|\|
```

It takes just 17 characters to serve as a compressed version of the original 45 characters by adding three more symbols to our growing dictionary.

As we apply this run-length encoding to more and more data, we see improved compression ratios because we can apply symbols already in use and don't need to keep adding new symbols to our dictionary for each swap.

ZIP employs a lossless compression algorithm called DEFLATE, which came into use in 1993. DEFLATE underpins both ZIP archives and the PNG image format; and thanks to its efficiency and being free to use without license fees, DEFLATE remains the most widely used compression algorithm in the world.

Tools for processing files in e-discovery must identify compressed files and apply the correct algorithm to unpack the contents. The decompression algorithm locates the tree and symbols library and retrieves and parses the directory structure and stored metadata for the contents.

Identification on Ingestion

Remember Programmer Ann and her struggle to select a TaggedyAnn file extension and signature? Now, those decisions play a vital role in how an e-discovery processing tool extracts text and metadata. If we hope to pull intelligible data from a file or container, we must first reliably ascertain the file's structure and encoding. For compressed files, we must apply the proper decompression algorithm. If it's an e-mail container format, we must apply the proper encoding schema to segregate messages and decode all manner of attachments. We must treat image files as images, sound files as sounds and so on. Misidentification of file types guarantees failure.

The e-discovery industry relies upon various open source and commercial file identifier tools. These tend to look first to the file's binary header for file identification and then to the file's extension and name. If the file type cannot be determined from the signature and metadata, the identification tool may either flag the file as unknown (an "exception") or pursue other identification methods as varied as byte frequency analysis (BFA) or the use of specialty filetype detection tools designed to suss out characteristics unique to certain file types, especially container files. Identifiers will typically report both the apparent file type (from metadata, *i.e.*, the file's name and extension) and the actual file type. Inconsistencies between these may prompt special handling or signal spoofing with malicious intent.

The table below sets out header signatures aka "magic numbers" for common file types:

File Type	Extension	Hex Signature	ASCII	Notes
ZIP Archive	ZIP	50 4B 03 04	PK..	
MS Office	DOCX XLSX PPTX	50 4B 03 04	PK..	Compressed XML files
Outlook mail	PST	21 42 44 4E 42	!BDN	
Outlook message	MSG	D0 CF 11 E0 A1 B1 1A E1		

File Type	Extension	Hex Signature	ASCII	Notes
Executable file	EXE	4D 5A	MZ	For Mark Zbikowski, who said, “when you're writing the linker, you get to make the rules.”
Adobe PDF	PDF	25 50 44 46	%PDF	
PNG Graphic	PNG	89 50 4E 47	.PNG	
VMWare Disk file	VMDK	4B 44 4D 56	KDMV	
WAV audio file	WAV	52 49 46 46	RIFF	
Plain Text file	TXT	none	none	Only binary files have signatures

Media (MIME) Type Detection

File extensions are central to Microsoft operating systems. Systems like Linux and Mac OS X rely less on file extensions to identify file types. Instead, they employ a file identification mechanism called **Media (MIME) Type Detection**. MIME, which stands for **M**ultipurpose **I**nternet **M**ail **E**xtensions, is a seminal Internet standard that enables the grafting of text enhancements, foreign language character sets (Unicode) and multimedia content (*e.g.,* photos, video, sounds and machine code) onto plain text e-mails. **Virtually all e-mail travels in MIME format.**

The ability to transmit multiple file types via e-mail created a need to identify the content type transmitted. The **I**nternet **A**ssigned **N**umbers **A**uthority (IANA) oversees global Internet addressing and defines the hierarchy of media type designation. These hierarchical designations for e-mail attachments conforming to MIME encoding came to be known as **MIME types**. Though the use of MIME types started with e-mail, operating systems, tools and web browsers now employ MIME types to identify media, prompting the IANA to change the official name from MIME Types to **Media Types**.

Media types serve two important roles in e-discovery processing. First, they facilitate the identification of content based upon a media type declaration found in, *e.g.,* e-mail attachments and Internet publications. Second, media types serve as standard classifiers for files after identification of content. Classification of files within a media type taxonomy simplifies culling and filtering data in ways useful to e-discovery. While it’s enough to specify “document,” “spreadsheet” or “picture” in a Request for Production, e-discovery tools require a more granular breakdown of content. Tools must be able to distinguish a Word binary .DOC format from a Word XML .DOCX format, a Windows PowerPoint file from a Mac Keynote file and a GIF from a TIFF.

Media Type Tree Structure

Media types follow a path-like tree structure under one of the following standard types: **application, audio, image, text** and **video** (collectively called **discrete** media types) and **message** and **multipart** (called **composite** media types). These top-level media types are further defined by subtype and, optionally, by a suffix and parameter(s), written in lowercase.

Examples of file type declarations for common file formats:
Note: File types prefixed by x- are not IANA. Those prefixed by vnd. are vendor-specific formats.

Application

Word .DOC:	application/msword
Word .DOCX:	application/vnd.openxmlformats-officedocument.wordprocessingml.document
Adobe PDF:	application/pdf (.pdf)
PowerPoint .PPT:	application/vnd.ms-powerpoint
PowerPoint .PPTX:	application/vnd.openxmlformats-officedocument. presentationml. presentation
Slack file:	application/x-atomist-slack-file+json
.TAR archive:	application/x-tar
Excel .XLS:	application/vnd.ms-excel
Excel .XLSX:	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
ZIP archive .ZIP:	application/zip

Audio

.MID:	audio/x-midi
.MP3:	audio/mpeg
.MP4:	audio/mp4
.WAV:	audio/x-wav

Image

.BMP:	image/bmp
.GIF:	image/gif
.JPG:	image/jpeg
.PNG:	image/png
.TIF:	image/tiff

Text (Typically accompanied by a **charset** parameter identifying the character encoding)

.CSS:	text/css
.CSV:	text/csv
.HTML:	text/html
.ICS:	text/calendar

.RTF:	text/richtext
.TXT	text/plain

Video

.AVI	video/x-msvideo
.MOV:	video/quicktime
.MP4:	video/mp4
.MPG:	video/mpeg

When All Else Fails: Octet Streams and Text Stripping

When a processing tool cannot identify a file, it may flag the file as an exception and discontinue processing contents; but the greater likelihood is that the tool will treat the unidentifiable file as an **octet stream** and harvest or "strip out" whatever text or metadata it *can* identify within the stream. An octet stream is simply an arbitrary sequence or "stream" of data presumed to be binary data stored as eight-bit bytes or "octets." So, an octet stream is anything the processor fails to recognize as a known file type.

In e-discovery, the risk of treating a file as an octet stream and stripping identifiable text is that whatever plain text is stripped and indexed doesn't fairly mirror relevant content. However, because *some* text was stripped, the file may not be flagged as an exception requiring special handling; instead, the processing tool records the file as successfully processed notwithstanding the missing content.

❁ Exercise 8: Encoding: Binary Signatures

GOALS: The goals of this exercise are for the student to:

1. Identify and parse binary file signatures and hex counterparts; and
2. Better appreciate the role data encoding plays in computing and e-discovery.

OUTLINE: Students will examine the binary content of multiple files of various types to determine consistent binary and hex file signatures suitable for filtering, processing and carving in e-discovery and computer forensics.

Binary and Hex File Signatures

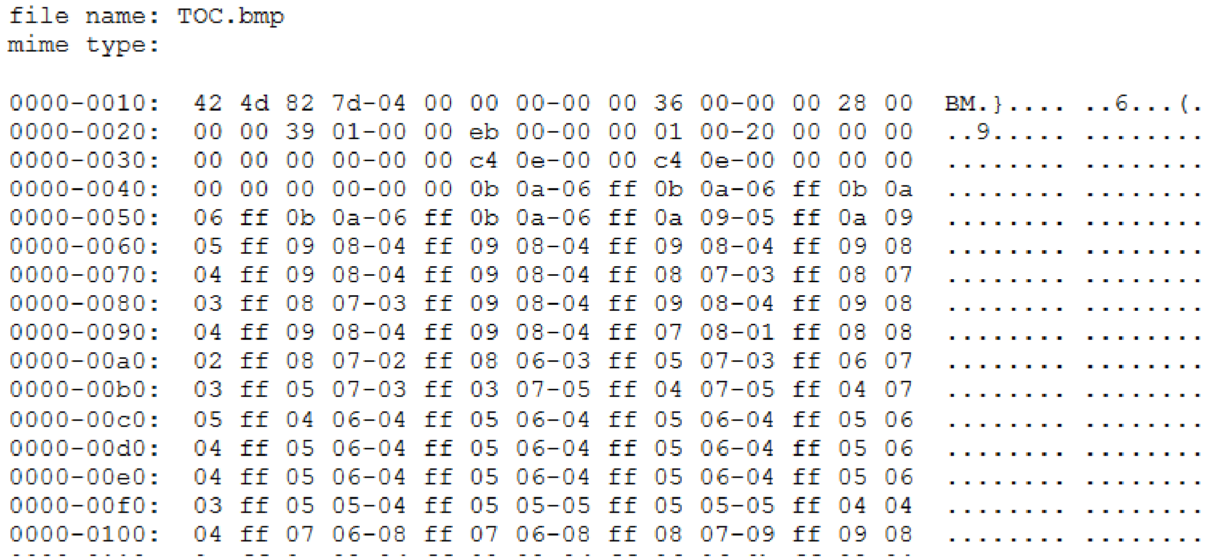
As we saw earlier, a file's header is data at or near the start of the file that serves to identify the type of data contained in the file (as well as information about the file's length, structure or other characteristics). File headers play a crucial role in the recovery of deleted data and the identification of hidden files. Computer forensic examiners often recover deleted files by scanning the recycled areas of hard drives called "unallocated clusters" for file signatures in a process called "data carving."

Step 1: Download the Zip file at www.craigball.com/filetypes.zip and extract its contents to your desktop or any other convenient location on your computer.

Step 2: The extracted contents will comprise nine folders (named BMP, DOC, DWG, GIF, JPG, PDF, TXT, WAV and XLS), each containing samples of file types commonly processed in e-discovery.

Step 3: Identify file header signatures for common file types

Using your web browser, go to the Online HexDump Utility at http://www.fileformat.info/tool/hexdump.htm and click "choose File." Using the selection box that will appear, navigate to the folder just extracted called BMP (you should see seven files) and select the file called TOC.bmp. Click "Open." Now click the blue "Dump" button on the Online HexDump Utility page. You should see this:



```
file name: TOC.bmp
mime type:

0000-0010:  42 4d 82 7d-04 00 00 00-00 00 36 00-00 00 28 00  BM.}.... ..6...(.
0000-0020:  00 00 39 01-00 00 eb 00-00 00 01 00-20 00 00 00  ..9..... ........
0000-0030:  00 00 00 00-00 00 c4 0e-00 00 c4 0e-00 00 00 00  ........ ........
0000-0040:  00 00 00 00-00 00 0b 0a-06 ff 0b 0a-06 ff 0b 0a  ........ ........
0000-0050:  06 ff 0b 0a-06 ff 0b 0a-06 ff 0a 09-05 ff 0a 09  ........ ........
0000-0060:  05 ff 09 08-04 ff 09 08-04 ff 09 08-04 ff 09 08  ........ ........
0000-0070:  04 ff 09 08-04 ff 09 08-04 ff 08 07-03 ff 08 07  ........ ........
0000-0080:  03 ff 08 07-03 ff 09 08-04 ff 09 08-04 ff 09 08  ........ ........
0000-0090:  04 ff 09 08-04 ff 09 08-04 ff 07 08-01 ff 08 08  ........ ........
0000-00a0:  02 ff 08 07-02 ff 08 06-03 ff 05 07-03 ff 06 07  ........ ........
0000-00b0:  03 ff 05 07-03 ff 03 07-05 ff 04 07-05 ff 04 07  ........ ........
0000-00c0:  05 ff 04 06-04 ff 05 06-04 ff 05 06-04 ff 05 06  ........ ........
0000-00d0:  04 ff 05 06-04 ff 05 06-04 ff 05 06-04 ff 05 06  ........ ........
0000-00e0:  04 ff 05 06-04 ff 05 06-04 ff 05 06-04 ff 05 06  ........ ........
0000-00f0:  03 ff 05 05-04 ff 05 05-05 ff 05 05-05 ff 04 04  ........ ........
0000-0100:  04 ff 07 06-08 ff 07 06-08 ff 08 07-09 ff 09 08  ........ ........
```

Note the first few bytes of the file. Load and peruse each of the remaining six bitmap files in BMP folder and identify text within the first few bytes of each that is **common to all** of the files in the BMP folder. Do you see that the first two characters of all of the BMP files are BM (hex 42 4D)? BM is the binary signature header identifying the content of each file as a bitmap image. Even if you changed the files' extensions to something other than BMP, that header signature gives away its content.

Now, use hexDump to view each of the six files in the folder called DWG (DWG is an extension typically denoting AutoCAD drawing files). Look at each of the six DWG files. **Can you identify a common binary header?** Note that all of the files begin "AC10" but the next two values vary from 15 to 18 to 24.

Header variation may indicate file formats that have changed over time. In the case of these DWG files, the headers AC1015, AC1018 and AC1024 reference AutoCAD files created using different releases of the AutoCAD program. AC1015 indicates that the drawing was made using version 15 of AutoCAD, sold in the year 2000. Version 18 was released in 2004 and version 24 in 2010.

Step 4: Identify Binary Signatures for Common File Types

Because file headers can vary, like the DWG files above, it's important to identify signatures that are common to all the files of a particular file type.

Examine the files in the DOC, GIF, PDF, TXT, WAV and XLS folders to determine the common binary signature you'd use to identify each of those file types. Now, record those signatures as hexadecimal values. ***Remember: you want a file signature to be as long as possible to assure it's***

precise, but you must not include characters that are not common to all files of that file type lest you fail to find those variations. **Show your answers below:**

File Type	Binary Signature	Hex Signature
DOC		
GIF		
PDF		
WAV		
XLS		
TXT		

Discussion questions (NOT ASSIGNED):

1. Do all files have unique binary signatures?
2. How do you distinguish between the various MS Office files?
3. Do file signatures always start with the first byte of a file?
4. Can a file's binary signature be changed?
5. Do files have footers (signatures at the end of files)?
6. How are file signatures used by e-discovery service providers and forensic examiners?
7. Can you find a leetspeak message (Google it) in the *hex* headers of Microsoft Office files?

❁ Exercise 9: Encoding: Unicode

GOALS: The goals of this exercise are for the student to:

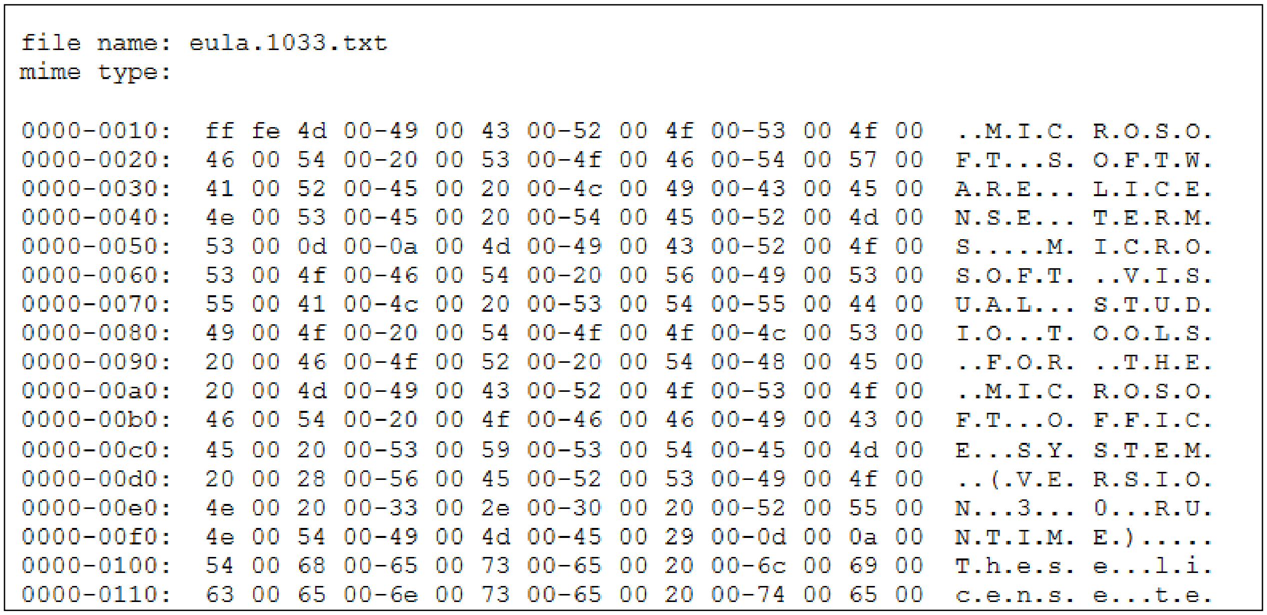
1. Gain further familiarity with the concept of encoding character sets and code pages;
2. Understand the significance of single byte and multibyte encoding schemes (e.g., Unicode); and
3. Appreciate the role that encoding schemae play in EDD processing and search.

OUTLINE: Students will examine files of like content in different foreign languages and character sets and, correspondingly, encoded using different multibyte code pages. You might want to re-read the brief discussion of Unicode at pp. 103-104.

Step 1: Use the files you extracted from www.craigball.com/filetypes.zip in Exercises 3 and 8 (in folders BMP, DOC, DWG, GIF, JPG, PDF, TXT, WAV and XLS).

Step 2: Identify file header signatures for common file types

Using your web browser, go to the Online HexDump Utility at http://www.fileformat.info/tool/hexdump.htm and click "choose File." Using the selection box that will appear, navigate to the folder called TXT. You should see 24 files. Select the file called eula.1033.txt. Click "Open." Now click the blue "Dump" button on the Online HexDump Utility page. You should see this:



```
file name: eula.1033.txt
mime type:

0000-0010:  ff fe 4d 00-49 00 43 00-52 00 4f 00-53 00 4f 00  ..M.I.C. R.O.S.O.
0000-0020:  46 00 54 00-20 00 53 00-4f 00 46 00-54 00 57 00  F.T...S. O.F.T.W.
0000-0030:  41 00 52 00-45 00 20 00-4c 00 49 00-43 00 45 00  A.R.E... L.I.C.E.
0000-0040:  4e 00 53 00-45 00 20 00-54 00 45 00-52 00 4d 00  N.S.E... T.E.R.M.
0000-0050:  53 00 0d 00-0a 00 4d 00-49 00 43 00-52 00 4f 00  S.....M. I.C.R.O.
0000-0060:  53 00 4f 00-46 00 54 00-20 00 56 00-49 00 53 00  S.O.F.T. ..V.I.S.
0000-0070:  55 00 41 00-4c 00 20 00-53 00 54 00-55 00 44 00  U.A.L... S.T.U.D.
0000-0080:  49 00 4f 00-20 00 54 00-4f 00 4f 00-4c 00 53 00  I.O...T. O.O.L.S.
0000-0090:  20 00 46 00-4f 00 52 00-20 00 54 00-48 00 45 00  ..F.O.R. ..T.H.E.
0000-00a0:  20 00 4d 00-49 00 43 00-52 00 4f 00-53 00 4f 00  ..M.I.C. R.O.S.O.
0000-00b0:  46 00 54 00-20 00 4f 00-46 00 46 00-49 00 43 00  F.T...O. F.F.I.C.
0000-00c0:  45 00 20 00-53 00 59 00-53 00 54 00-45 00 4d 00  E...S.Y. S.T.E.M.
0000-00d0:  20 00 28 00-56 00 45 00-52 00 53 00-49 00 4f 00  ..(.V.E. R.S.I.O.
0000-00e0:  4e 00 20 00-33 00 2e 00-30 00 20 00-52 00 55 00  N...3... 0...R.U.
0000-00f0:  4e 00 54 00-49 00 4d 00-45 00 29 00-0d 00 0a 00  N.T.I.M. E.).....
0000-0100:  54 00 68 00-65 00 73 00-65 00 20 00-6c 00 69 00  T.h.e.s. e...l.i.
0000-0110:  63 00 65 00-6e 00 73 00-65 00 20 00-74 00 65 00  c.e.n.s. e...t.e.
```

Note the "dots" that appear between the letters in the document. This is how Unicode text appears when viewed using a tool that treats it like ASCII. Looking at the same content in hex, you can see the second byte used to encode each letter is hex 00. Because the second byte isn't needed for the Latin alphabet, it’s ‘zeroed out’ and appears as a dot separating each letter when treated as ASCII.

Step 3: Open in Default Text Viewer

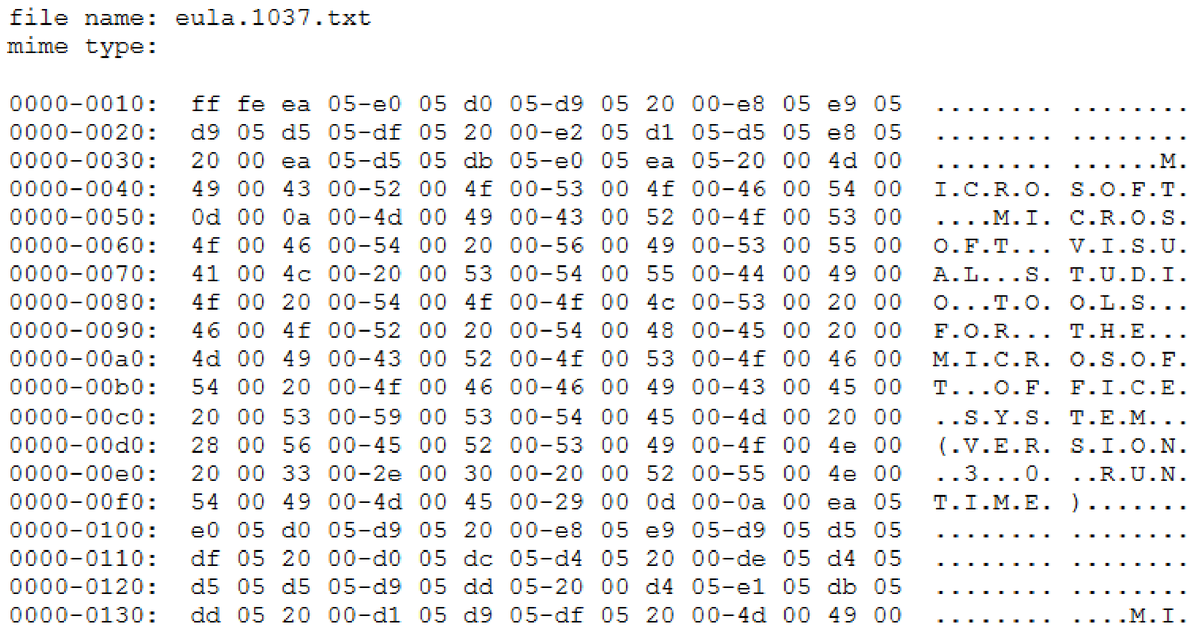
Now, double click on the file **eula.1033.txt** to open it in your default text viewer application (likely to be Notepad, Wordpad or Word on a Windows machine; TextEdit on a Mac). You may also use the free online application at **http://www.rapidtables.com/tools/notepad.htm.** Chances are, when **eula.1033.txt** opens in the text viewer, it will look “normal;” that is, you won’t see any dots or spaces between the letters of each word. That’s because your operating system (or the online text editor) is applying a code page that correctly interprets the Unicode data (likely UTF-8 or UTF-16) in the view presented to you.

Discussion Question: What difference might Unicode encoding make in framing searches for e-discovery?

Step 4: Foreign Language Encodings

Double click on the file **eula.1037.txt** to open it in your default text viewer application. When it opens, it should be in Hebrew with some scattered English text. If you see the Hebrew, it’s because your system is applying the correct Unicode character encoding to the data and not attempting to display it to you as ASCII text.

To see what it looks like when the wrong (ASCII) encoding is applied, return to the Online HexDump Utility at http://www.fileformat.info/tool/hexdump.htm and load eula.1037.txt. All you will be able to see in the right column will be the scattered English text. The Hebrew text will be replaced by dots. Like so:



```
file name: eula.1037.txt
mime type:

0000-0010:  ff fe ea 05-e0 05 d0 05-d9 05 20 00-e8 05 e9 05  ........ ........
0000-0020:  d9 05 d5 05-df 05 20 00-e2 05 d1 05-d5 05 e8 05  ........ ........
0000-0030:  20 00 ea 05-d5 05 db 05-e0 05 ea 05-20 00 4d 00  ........ ......M.
0000-0040:  49 00 43 00-52 00 4f 00-53 00 4f 00-46 00 54 00  I.C.R.O. S.O.F.T.
0000-0050:  0d 00 0a 00-4d 00 49 00-43 00 52 00-4f 00 53 00  ....M.I. C.R.O.S.
0000-0060:  4f 00 46 00-54 00 20 00-56 00 49 00-53 00 55 00  O.F.T... V.I.S.U.
0000-0070:  41 00 4c 00-20 00 53 00-54 00 55 00-44 00 49 00  A.L...S. T.U.D.I.
0000-0080:  4f 00 20 00-54 00 4f 00-4f 00 4c 00-53 00 20 00  O...T.O. O.L.S...
0000-0090:  46 00 4f 00-52 00 20 00-54 00 48 00-45 00 20 00  F.O.R... T.H.E...
0000-00a0:  4d 00 49 00-43 00 52 00-4f 00 53 00-4f 00 46 00  M.I.C.R. O.S.O.F.
0000-00b0:  54 00 20 00-4f 00 46 00-46 00 49 00-43 00 45 00  T...O.F. F.I.C.E.
0000-00c0:  20 00 53 00-59 00 53 00-54 00 45 00-4d 00 20 00  ..S.Y.S. T.E.M...
0000-00d0:  28 00 56 00-45 00 52 00-53 00 49 00-4f 00 4e 00  (.V.E.R. S.I.O.N.
0000-00e0:  20 00 33 00-2e 00 30 00-20 00 52 00-55 00 4e 00  ..3...0. ..R.U.N.
0000-00f0:  54 00 49 00-4d 00 45 00-29 00 0d 00-0a 00 ea 05  T.I.M.E. ).......
0000-0100:  e0 05 d0 05-d9 05 20 00-e8 05 e9 05-d9 05 d5 05  ........ ........
0000-0110:  df 05 20 00-d0 05 dc 05-d4 05 20 00-de 05 d4 05  ........ ........
0000-0120:  d5 05 d5 05-d9 05 dd 05-20 00 d4 05-e1 05 db 05  ........ ........
0000-0130:  dd 05 20 00-d1 05 d9 05-df 05 20 00-4d 00 49 00  ........ ....M.I.
```

Why? Because to maximize compatibility with single-byte ASCII text, Unicode also supports ASCII encoding; so, the ASCII viewer in the HexDump tool can see and correctly interpret the ASCII characters. However, the ASCII viewer can't make sense of double-byte encodings (i.e., the Hebrew text) and displays a dot instead.

Data Extraction and Document Filters

If ESI were like paper, you could open each item in its associated program (its *native application*), review the contents and decide whether the item is relevant or privileged. But, ESI is much different than paper documents in crucial ways:

- ESI collections tend to be exponentially more voluminous than paper collections
- ESI is stored digitally, rendering it unintelligible absent electronic processing
- ESI carries metainformation that is always of practical use and may be probative evidence
- ESI is electronically searchable while paper documents require laborious human scrutiny
- ESI is readily culled, filtered and deduplicated, and inexpensively stored and transmitted
- ESI and associated metadata change when opened in native applications

These and other differences make it impractical and risky to approach e-discovery via the piecemeal use of native applications as viewers. Search would be inconsistent and slow, and deduplication impossible. Too, you'd surrender all the benefits of mass tagging, filtering and production. Lawyers who learn that *native productions* are superior to other forms of production may mistakenly conclude that native production suggests use of native applications for review. Absolutely not! Native applications are <u>not</u> suited to e-discovery, and you shouldn't use them for review. E-discovery review tools are the only way to go.

To secure the greatest benefit of ESI in search, culling and review, we process ingested files to extract their text, embedded objects, and metadata. In turn, we normalize and tokenize extracted contents, add them to a database and index them for efficient search. These processing operations promote efficiency but impose penalties in terms of reduced precision and accuracy. It's a tradeoff demanding an informed and purposeful balancing of benefits and costs.

Returning to Programmer Ann and her efforts to fashion a new file format, Ann had a free hand in establishing the structural layout of her TaggedyAnn data files because she was also writing the software to read them. The ability to edit data easily is a hallmark of computing; so, programmers design files to be able to grow and shrink without impairing updating and retrieval of their contents. Files hold text, rich media (like graphics and video), formatting information, configuration instructions, metadata and more. All that disparate content exists as a sequence of hexadecimal characters. Some of it may reside at fixed offset addresses measured in a static number of bytes from the start of the file. But because files must be able to grow and shrink, fixed offset addressing alone won't cut it. Instead, files must supply dynamic directories of their contents or incorporate tags that serve as signposts for navigation.

When navigating files to extract contents, it's not enough to know where the data starts and ends, you must also know how the data's encoded. Is it ASCII text? Unicode? JPEG? Is it a date and time value or perhaps a bit flag where the numeric value serves to signal a characteristic or configuration to the program?

There are two broad approaches used by processing tools to extract content from files. One is to use the Application Programming Interface (API) of the application that created the file. The other is to turn to a published file specification or reverse engineer the file to determine where the data sought to be extracted resides and how it's encoded.

A software API allows "client" applications (*i.e.,* other software) to make requests or "calls" to the API "server" to obtain specific information and to ask the server to perform specific functions. Much like a restaurant, the client can "order" from a menu of supported API offerings without knowing what goes on in the kitchen, where the client generally isn't welcome to enter. Like a restaurant with off-menu items, the API may support undocumented calls intended for a limited pool of users.

For online data reposing in sites like Office 365, Dropbox or OneDrive, there's little choice but to use an API to get to the data; but for data in local files, using a native application's API is something of a last resort because APIs tend to be slow and constraining. Not all applications offer open APIs, and those that do won't necessarily hand off all data needed for e-discovery. For many years, a leading e-discovery processing tool required purchasers to obtain a "bootleg" copy of the IBM/Lotus Notes mail program because the secure structure of Notes files frustrated efforts to extract messages and attachments by any means but the native API.

An alternative to the native application API is the use of data extraction templates called **Document Filters.** Document filters lay out where content is stored within each filetype and how that content is encoded and interpreted. Think of them as an extraction template. Document filters can be based on a published file specification, or they can be painstakingly reverse engineered from examples of the data—a terrifically complex process that produces outcomes of questionable accuracy and consistency. Because document filters are challenging to construct and keep up to date for each of the hundreds of file types seen in e-discovery, few e-discovery processors build their own library of document filters. Instead, they turn to a handful of commercial and open source filters.

The leading commercial collection of document filters is Oracle's *Outside In*, which its publisher describes as "a suite of software development kits (SDKs) that provides developers with a comprehensive solution to extract, normalize, scrub, convert and view the contents of 600 unstructured file formats." *Outside In* quietly serves as the extraction and viewer engine behind many e-discovery review tools, a fact the sellers of those tools are often reluctant to concede; but

I suppose sellers of the Lexus ES aren't keen to note it shares its engine, chassis and most parts with the cheaper Toyota Avalon.

Aspose Pty. Ltd., an Australian concern, licenses libraries of commercial APIs, enabling software developers to read and write to, *e.g.*, Word documents, Excel spreadsheets, PowerPoint presentations, PDF files and multiple e-mail container formats. Aspose tools can both read from and write to the various formats, the latter considerably more challenging.

Hyland Software's *Document Filters* is another developer's toolkit that facilitates file identification and content extraction for 500+ file formats, as well as support for OCR, redaction and image rendering. Per Hyland's website, its extraction tools power e-discovery products from Catalyst and Reveal Software.

A fourth commercial product that lies at the heart of several e-discovery and computer forensic tools (*e.g.*, Relativity, LAW, Ringtail aka Nuix Discover and Access Data's FTK) is *dtSearch*, which serves as both content extractor and indexing engine.

On the open source side, Apache's *Tika* is a free toolkit for extracting text and metadata from over a thousand file types, including most encountered in e-discovery. *Tika* was a subproject of the open source Apache Lucene project, Lucene being an indexing and search tool at the core of several commercial e-discovery tools.

Beyond these five toolsets, the wellspring of document filters and text extractors starts to dry up, which means a broad swath of commercial e-discovery tools relies upon a tiny complement of text and metadata extraction tools to build their indices and front-end their advanced analytics.

In fact, most e-discovery tools seen in the last 15 years are proprietary wrappers around code borrowed or licensed from common sources for file identifiers, text extractors, OCR, normalizers, indexers, viewers, image generators and databases. Bolting these off-the-shelf parts together to deliver an efficient workflow and user-friendly interface is no mean feat.

But as we admire the winsome wrappers, we must remember that these products share the same DNA in spite of marketing efforts suggesting "secret sauces" and differentiation. More to the point, products built on the same text and metadata extractor share the same limitations and vulnerabilities as that extractor.

Recursion and Embedded Object Extraction

Just as an essential task in processing is to correctly identify content and apply the right decoding schema, a processing tool must extract and account for all the components of a file that carry potentially responsive information.

Modern productivity files like Microsoft Office documents are rich, layered containers called **Compound Files**. Objects like images and the contents of other file formats may be embedded and linked within a compound file. Think of an Excel spreadsheet appearing as a diagram within a Word document. Microsoft promulgated a mechanism supporting this functionality called **OLE** (pronounced "o-lay" and short for **O**bject **L**inking and **E**mbedding). OLE supports dragging and dropping content between applications and the dynamic updating of embedded content, so the Excel spreadsheet embedded in a Word document updates to reflect changes in the source spreadsheet file. A processing tool must be able to recognize an OLE object and extract and enumerate all the embedded and linked content.

A MIME e-mail is also a compound document to the extent it transmits multipart content, particularly encoded attachments. A processing tool must account for and extract every item in the e-mail's informational payload, recognizing that such content may nest like a Russian doll. An e-mail attachment could be a ZIP container holding multiple Outlook .PST mail containers holding e-mail collections that, in turn, hold attachments of OLE documents and other ZIP containers! The mechanism by which a processing tool explores, identifies, unpacks and extracts all embedded content from a file is called **recursion**. It's crucial that a data extraction tool be able to recurse through a file and loop itself to extract embedded content until there is nothing else to be found.

Family Tracking and Unitization: Keeping Up with the Parts

As a processing tool unpacks the embedded components of compound and container files, it must update the database with information about what data came from what file, a relationship called **unitization**. In the context of e-mail, recording the relationship between a transmitting message and its attachments is called **family tracking:** the transmitting message is the **parent object** and the attachments are **child objects**. The processing tool must identify and preserve metadata values applicable to the entire contents of the compound or container file (like system metadata for the parent object) and embedded metadata applicable to each child object. One of the most important metadata values to preserve and pair with each object is the object's custodian or source. Post-processing, every item in an e-discovery collection must be capable of being tied back to an originating file at time of ingestion, including its prior unitization and any parent-child relationship to other objects.

Exceptions Reporting: Keeping Track of Failures

It's rare that a sizable collection of data will process flawlessly. There will almost always be encrypted files that cannot be read, corrupt files, files in unrecognized formats or languages and files requiring optical character recognition (OCR) to extract text. A great many documents are not amenable to text search without special handling. Common examples of non-searchable documents are faxes and scans, as well as TIFF images and Adobe PDF documents lacking a text layer. A processing tool must track all exceptions and be capable of generating an **exceptions report** to enable counsel and others with oversight responsibility to act to rectify exceptions by, *e.g.*, securing passwords, repairing or replacing corrupt files and running OCR against the files. Exceptions resolution is key to a defensible e-discovery process.

Counsel and others processing ESI in discovery should broadly understand the exceptions handling characteristics of their processing tools and be competent to make necessary disclosures and answer questions about exceptions reporting and resolution. *Exceptions signify that evidence is missing; so, exceptions must be resolved or disclosed and defended.* As noted earlier, it's particularly perilous when a processing tool defaults to text stripping an unrecognized or misrecognized file because the tool may fail to flag a text-stripped file as an exception requiring resolution. Just because a tool succeeds in stripping some text from a file doesn't mean that all discoverable content was extracted.

Lexical Preprocessing of Extracted Text

Computers are excruciatingly literal. Computers cannot read. Computers cannot understand language in the way humans do. Instead, computers apply rules assigned by programmers to normalize, tokenize, and segment natural language, all instances of **lexical preprocessing**—steps to prepare text for parsing by other tools.

Normalization

ESI is numbers; numbers are precise. Variations in those numbers—however subtle to humans—hinder a computer's ability to equate information as humans do. Before a machine can distinguish words or build an index, we must massage the streams of text spit out by the document filters to ultimately increase recall; that is, to ensure that more documents are retrieved by search, even the documents we seek that don't exactly match our queries.

Variations in characters that human beings readily overlook pose big challenges to machines. So, we seek to minimize the impact of these variations through **normalization**. How we normalize data and even the order in which steps occur affect our ability to query the data and return correct results.

Character Normalization

Consider three characteristics of characters that demand normalization: **Unicode equivalency**, **diacriticals** (accents) and **case** (capitalization).

Unicode Normalization

In our discussion of ASCII encoding, we established that each ASCII character has an assigned, corresponding numeric value (*e.g.,* a capital "E" is 0100 0101 in binary, 69 in decimal and Ox45 in hexadecimal). But linguistically identical characters encoded in Unicode may be represented by *different* numeric values by virtue of accented letters having both precomposed and composite references. That means that you can use an encoding specific to the accented letter (a **precomposed** character) or you can fashion the character as a **composite** by pairing the encoding for the base letter with the encoding for the diacritical. For example, the Latin capital "E" with an acute accent (É) may be encoded as either **U+00C9** (a precomposed Latin capital letter E with acute accent) or as **U+0045** (Latin capital letter E) + **U+0301** (combining acute accent). Both will appear as É.

Easy enough to grasp, but there are a lot of accented variants of Latin letters, *e.g.*:

Ĕ Ḝ Ê Ê Ê̄ Ê̌ Ề Ế Ể Ễ Ệ Ẻ Ḙ Ě Ɇ Ė É Ė̃ Ẹ Ë È È̩ Ȅ É É̩ Ē È É Ẽ Ḛ Ȩ Ȩ́ Ȩ̃ Ȩ E̩ ȩ[

Surely, we don't want to have to account for *every* variation in which a character can be encoded in Unicode when fashioning a query in e-discovery! To obviate that burden, the Unicode Consortium promulgates normalization algorithms that produce a consistent ("normalized") encoding for each identical character. One version of the algorithm reduces all identical characters to a composed version, and another reduces all to a decomposed (composite) version. In e-discovery, we often seek to strip accents so see more of the latter.

Diacritical Normalization

Unicode normalization serves to ensure that the same canonical character is encoded in a consistent way. But often—especially in the United States—we want accented characters to be searchable whether a diacritical is employed or not. This requires normalizing the data to forge a false equivalency between accented characters and their non-accented ASCII counterparts. So, if you search for "resume" or "cafe," you will pick up instances of "resumé" and "café." As well, we must normalize ligatures like the German Eszett (ß) seen in the word "straße," or "street."

The major processing tools offer filters that convert alphabetic, numeric, and symbolic Unicode characters which are not in the first 127 ASCII characters (the "Basic Latin" Unicode block) into their reasonable ASCII equivalents, if any.

Case Normalization

The Latin alphabet is bicameral, meaning it employs upper- and lower-case letters to enhance meaning.[46] By contrast, languages such as Chinese, Arabic and Hebrew are unicameral and use no capitalization. Because people capitalize indiscriminately—particularly in e-mail and messaging—most often we want search queries to be case-insensitive such that DOE, Doe and doe all return the same hits. Other times, the ability to limit a search to a case-specific query is advantageous, such as by searching just DOE when your interest is the Department of Energy and search precision is more important than recall.

Just as processing tools can be configured to "fold" Unicode characters to ASCII equivalents, they can fold all letters to their upper- or lower-case counterparts, rendering an index that is case-

[46] *The terms upper- and lower-case derive from the customary juxtaposition of the shallow drawers or "cases" that held movable type for printing presses.*

insensitive. Customarily, normalization of case will require specification of a default language because of different capitalization conventions attendant to diverse cultures.

Impact of Normalization on Discovery Outcomes

Although all processing tools draw on a handful of filters and algorithms for normalization, processors don't implement normalization in the same sequence or with identical default settings. Accordingly, it's routine to see tools produce varying outcomes in culling and search because of differences in character normalization. Whether these differences are material or not depends upon the nature of the data and the inquiry; but any service provider and case manager should know how their tool of choice normalizes data.

In the sweep of a multi-million document project, the impact of normalization might seem trivial. Yet character normalization affects the whole collection and plays an outsize role in what's filtered and found. It's an apt reminder that a good working knowledge of processing equips e-discovery professionals to "normalize" *expectations,* especially expectations as to what data will be seen and searchable going forward. The most advanced techniques in analytics and artificial intelligence are no better than what emerges from processing. If the processing is off, it's fancy joinery applied to rotten wood.

Lawyers must fight for quality before review. Sure, review is the part of e-discovery most lawyers see and understand, so is the part many fixate on. As well, review is the costliest component of e-discovery and the one with cool tools. But here's the bottom line: *The most sophisticated MRI scanner won't save those who don't survive the trip to the hospital.* It's more important to have triage that gets people to the hospital alive than the best-equipped emergency room. Collection and processing are the EMTs of e-discovery. If we don't pay close attention to quality, completeness and process *before* review, review won't save us.

Time Zone Normalization

You needn't be an Einstein of e-discovery to appreciate that time is relative. Parsing a message thread, it's common to see e-mails from Europe to the U.S. prompt replies that, at least according to embedded metadata, appear to precede by hours the messages they answer. Time zones and daylight savings time both work to make it difficult to correctly order documents and communications on a consistent timeline. So, a common processing task is to normalize date and time values according to a single temporal baseline, often **Coordinated Universal Time** (UTC)—essentially Greenwich Mean Time—or to any other time zone the parties choose. The differential between the source time and **UTC offset** may then be expressed as plus or minus the numbers of hours separating the two (*e.g.,* UTC-0500 to denote five hours earlier than UTC).

West of London = UTC - X
East of London = UTC + X
STANDARD TIME ZONES OF THE WORLD
UTC-5
UTC+4
10
-9
-8
-7
-6
-5
-4
-3
-2
-1
UTC
+1
+2
+3
+4
+5
+6
+7
+8
+9
+10
RUSSIA
CHINA
BRAZIL
AUSTRALIA

Parsing and Tokenization

To this point, we've focused on efforts to identify a file's format, then extract its content and metadata—important tasks, because if you don't get the file's content out and properly decoded, you've nearly nothing to work with. But, getting data out is just the first step. Now, we must distill the extracted content into the linguistic components that serve to convey the file's informational payload; that is, we need to isolate the words within the documents and construct an index of those words to allow us to instantly identify or exclude files based on their lexical content.

There's a saying that anything a human being can do after age five is easy for a computer, but mastering skills humans acquire earlier is hard. Calculate pi to 31 trillion digits? Done! Read a Dr. Seuss book? Sorry, no can do.

Humans are good at spotting linguistic units like words and sentences from an early age, but computers must identify lexical units or "**tokens**" within extracted and normalized character strings, a process called "**tokenization**." When machines search collections of documents and data for keywords, they don't search the extracted text of the documents or data for matches; instead, they consult an index of words built from extracted text. Machines cannot read; instead, computers identify "words" in documents because their appearance and juxtaposition meet certain **tokenization rules.** These rules aren't uniform across systems or software. Many indices simply don't index short words (*e.g.,* two-letter words, acronyms and initializations). None index single letters or numbers.

Tokenization rules also govern such things as the handling of punctuated terms (as in a compound word like "wind-driven"), capitalization/case (will a search for "roof" also find "Roof?"), diacritical marks (will a search for "Rene" also find "René?") and numbers and single letters (will a search for "Clause 4.3" work? What about a search for "Plan B?"). Most people simply *assume* these searches will work. Yet, in many e-discovery search tools, they don't work as expected or don't work at all.

So, how do you train a computer to spot sentences and words? What makes a word a word and a sentence a sentence?

Languages based on Latin-, Cyrillic-, or Greek-based writing systems, such as English and European languages, are “segmented;” that is, they tend to set off (“delimit”) words by white space and punctuation. Consequently, most tokenizers for segmented languages base token boundaries on spacing and punctuation. That seems a simple solution at first blush, but one quickly complicated by hyphenated words, contractions, dates, phone numbers and abbreviations. How does the machine distinguish a word-break hyphen from a true or lexical hyphen? How does the machine distinguish the periods in the salutation “Mr.” or the initialization “G.D.P.R” from periods which signal the ends of sentences? In the realm of medicine and pharmacology, many words contain numbers, dashes and parentheses as integral parts. How could you defend a search for Ibuprofen if you failed to also seek instances of ***(RS)-2-(4-(2-methylpropyl)phenyl)propanoic acid***?

TEST TIP: Remember the difference between **normalization** and **tokenization**:

Normalization is the process of reformatting data to a standardized form, such as setting the date and time stamp of files to a uniform time zone or converting all content to the same character encoding. Normalization facilitates search and data organization.

Tokenization is a method of document parsing that identifies words ("tokens") to be used in a full-text index. Because computers cannot read as humans do but only see sequences of bytes, computers employ programmed tokenization rules to identify character sequences that constitute words and punctuation.

Again, tokenization rules aren’t uniform across systems, software or languages. Some tools are simply incapable of indexing and searching certain characters. These exclusions impact discovery in material ways. Several years ago, after contentious motion practice, a court ordered the parties to search a dataset using queries incorporating the term “20%.” No one was pleased to learn their e-discovery tools were incapable of searching for the percentage sign.

You cannot run a query in Relativity including the percentage sign (%) because Relativity uses *dtSearch* as an indexing tool and *dtSearch* has reserved the character “%” for another purpose. This is true no matter how you tweak the settings because the % sign simply cannot be added to the index and made searchable. When you run a search, you won’t be warned that the search is impossible; you’ll simply get no hits on any query requiring the % sign be found.

Using *dtSearch*/Relativity as another example, you can specify the way to process hyphens at the time an index is created, but you cannot change how hyphens are handled without re-indexing the collection. The default setting is to treat hyphens as spaces, but there are four alternative treatments.

From the *dtSearch* Help pages:

> The dtSearch Engine supports four options for the treatment of hyphens when indexing documents: spaces, searchable text, ignored, and "all three."
>
> *For most applications, treating hyphens as spaces is the best option.* Hyphens are translated to spaces during indexing and during searches. For example, if you index "first-class mail" and search for "first class mail", "first-class-mail", or "first-class mail", you will find the phrase correctly.
>
> **Values**

HyphenSettings Value	**Meaning**
dtsoHyphenAsIgnore	index "first-class" as "firstclass"
dtsoHyphenAsHyphen	index "first-class" as "first-class"
dtsoHyphenAsSpace	index "first-class" as "first" and "class"
dtsoHyphenAll	index "first-class" all three ways

…

The "all three" option has one advantage over treating hyphens as spaces: it will return a document containing "first-class" in a search for "firstclass". Otherwise, it provides no benefit over treating hyphens as spaces, and it has some significant disadvantages:

1. The "all three" option generates many extra words during indexing. For each pair of words separated by a hyphen, six words are generated in the index.
2. If hyphens are treated as significant at search time, it can produce unexpected results in searches involving longer phrases or words with multiple hyphens.

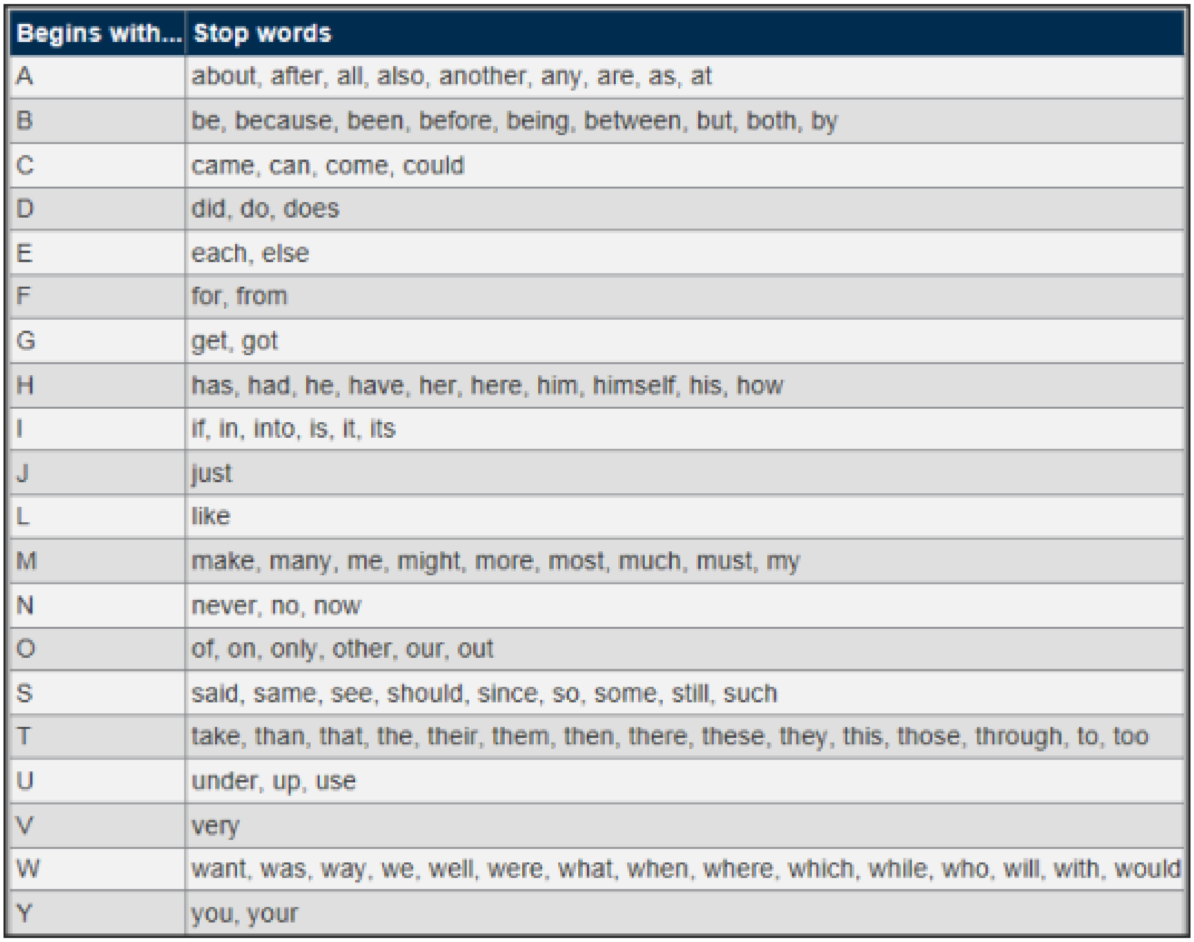
By default, *dtSearch* and the popular e-discovery review tool, Relativity, treat all the following characters as spaces:

!"#$&'()*+,./:;<=>?@[\5c]^`{|}~

Although several of the characters above can be made searchable by altering the default setting and reindexing the collection, the following characters CANNOT be made searchable in *dtSearch* and Relativity: ( ) * ? % @ ~ & : =

Stop Words

Some common "stop words" or "noise words" are excluded from an index when it's compiled. E-discovery tools typically exclude *dozens or hundreds* of stop words from indices. The table below lists 123 English stop words excluded by default in *dtSearch* and Relativity:



Begins with...	Stop words
A	about, after, all, also, another, any, are, as, at
B	be, because, been, before, being, between, but, both, by
C	came, can, come, could
D	did, do, does
E	each, else
F	for, from
G	get, got
H	has, had, he, have, her, here, him, himself, his, how
I	if, in, into, is, it, its
J	just
L	like
M	make, many, me, might, more, most, much, must, my
N	never, no, now
O	of, on, only, other, our, out
S	said, same, see, should, since, so, some, still, such
T	take, than, that, the, their, them, then, there, these, they, this, those, through, to, too
U	under, up, use
V	very
W	want, was, way, we, well, were, what, when, where, which, while, who, will, with, would
Y	you, your

Source: Relativity website, November 3, 2019

Relativity won't index punctuation marks, single letters or numbers. Nuix Discovery (formerly Ringtail) uses a similar English stop word list, except Nuix indexes the words "between," "does," "else," "from," "his," "make," "no," "so," "to," "use" and "want" and "does" where Relativity won't. Relativity indexes the words "an," "even," "further," "furthermore," "hi," "however," "indeed," "made," "moreover," "not" "or," "over," "she" and "thus" where Nuix won't. ***Does it make sense***

that both tools exclude "he" and "her," and both include "hers," but only Relativity excludes "his?"

Tools built on the open-source Natural Language Tool Kit won't index 179 English stop words. In other products, between 500 and 700 English stop words excluded. In one notorious instance, the two words that made up the company's own name were both stop words in their e-discovery system. They literally could not find their own name (or other stop words in queries they'd agreed to run)!

Remember, ***if it's not indexed, it's not searched***. Putting a query in quotes won't make any difference. No warning messages appear when you run a query including stop words, so it's the obligation of those running searches to acknowledge the incapability of the search. "No hits" is not the same thing as "no documents." If a party or counsel knows that the systems or searches used in e-discovery will fail to perform as expected, they should affirmatively disclose such shortcomings. If a party or counsel is uncertain whether systems or searches work as expected, they should find out by*, e.g.,* running tests to be reasonably certain.

No system is perfect, and perfect isn't the e-discovery standard. Often, we must adapt to the limitations of systems or software. But we must know what a system can't do before we can find ways to work around its limitations or set expectations consistent with actual capabilities, not magical thinking and unfounded expectations.

Building a Database and Concordance Index

This chapter is about processors. Databases (and viewers) belong to the realm of review tools and their features. However, a brief consideration of their interrelationship is useful.

The Review Database

At every step of processing, information derived from and about the items processed is continually handed off to a database. As the system ingests each file, a record of its name, size and system metadata values becomes part of the database. Sources—called "**custodians**" when they are individuals—are identified and comprise database fields. The processor calculates hash values and contributes them to the database. The tool identifies and extracts application metadata values from the processed information items, including, *inter alia*, authoring data for documents and subject, sender and recipient data for e-mail messages. The database also holds pointers to TIFF or PDF page images and to extracted text. The database is where items are **enumerated**, that is, assigned an item number that will uniquely identify each item in the processed collection. This is an identifier distinct from any Bates numbers subsequently assigned to items when produced.

The database lies at the heart of all e-discovery review tools. It's the recipient of much of the information derived from processing. But note, the database is <u>not</u> the index of text extracted from the processed items. The **concordance index**, the **database** and a third component, the **document viewer,** operate in so tightly coupled a manner that they seem like one.

A query of the index customarily triggers a return of information from the database about items "hit" by the query, and the contents of those items are, in turn, depicted in the viewer, often with hits highlighted. The perceived quality of commercial e-discovery review tools is a function of how seamlessly and efficiently these discrete functional components integrate to form a robust and intuitive user interface and experience.

Much like the file identification and content extraction tools discussed, e-discovery tool developers tend not to code databases from scratch but build atop a handful of open source or commercial database platforms. Notable examples are SQL Server and SQLite. Notwithstanding Herculean efforts of marketers to suggest differences, e-discovery tools tend to share the same or similar "database DNA." Users are none the wiser to the common foundations because the "back end" of discovery tools (the program's code and database operations layer) tends to be hidden from users and wrapped in an attractive interface.

The Concordance Index

The term "concordance" describes an alphabetical listing, particularly a mapping, of the important words in a text. Historically, scholars spent years painstakingly constructing concordances (or "full-text" indices) of Shakespeare's works or the Bible by hand. In e-discovery, software builds concordance indices to speed lexical search. While it's technically feasible to keyword search all documents in a collection, one after another (so-called "serial search"), it's terribly inefficient.[47]

Instead, the universal practice in e-discovery is to employ software to extract the text from information items, tokenize the contents to identify words and then construct a list of each token's associated document and location. Accordingly, *text searches in e-discovery don't search the evidence; they only search a concordance index of tokenized text*.

This is a crucial distinction because it means the quality of search in e-discovery is only as effective as the index is complete.

Indexing describes the process by which the data being is processed to form a highly efficient cross-reference lookup to facilitate rapid searching

Culling and Selecting the Dataset

Processing is not an end but a means by which potentially responsive information is exposed, enumerated, normalized and passed on for search, review and production. Although much culling and selection occurs in the search and review phase, the processing phase is an opportunity to reduce data volumes by culling and selecting by defensible criteria.

[47] Computer forensic examiners still use serial searches when the corpus is modest and when employing Global Regular Expressions (GREP searches) to identify patterns conforming to, *e.g.*, social security or credit card numbers.

Now that the metadata is in a database and the collection has been made text searchable by creation of a concordance index, it's feasible to filter the collection by, *e.g.,* date ranges, file types, Internet domains, file size, custodian and other objective characteristics. We can also cull the dataset by **immaterial item suppression**, **de-NISTing** and **deduplication**, all discussed *infra*.

The crudest but most common culling method is keyword and query filtering; that is, lexical search. Lexical search and its shortcomings will be addressed in later chapters, but it should be clear by now that the quality of the processing bears materially on the ability to find what we seek through lexical search. No search and review process can assess the content of items missed or malformed in processing.

Immaterial Item Suppression

E-discovery processing tools must be able to track back to the originating source file for any information extracted and indexed. Every item must be catalogued and enumerated, including each container file and all contents of each container. Still, in e-discovery, we've little need to search or produce the wrappers if we've properly expanded and extracted the contents. The wrappers are immaterial items.

Immaterial items are those extracted for forensic completeness but having little or no intrinsic value as discoverable evidence. Common examples of immaterial items include the folder structure in which files are stored and the various container files (like ZIP, RAR files and other containers, *e.g.,* mailbox files like Outlook PST and MBOX, and forensic disk image wrapper files like .E0x or .AFF) that tend to have no relevance apart from their contents.

Accordingly, it's handy to be able to suppress immaterial items once we extract and enumerate their contents. It's pointless to produce a ZIP container if its contents are produced, and it's perilous to do so if some contents are non-responsive or privileged.

De-NISTing

De-NISTing is a technique used in e-discovery and computer forensics to reduce the number of files requiring review by excluding standard components of the computer's operating system and off-the-shelf software applications like Word, Excel and other parts of Microsoft Office. Everyone has this digital detritus on their systems—things like Windows screen saver images, document templates, clip art, system sound files and so forth. It's the stuff that comes straight off the installation disks, and it's just noise to a document review.

Eliminating this noise is called "de-NISTing" because those noise files are identified by matching their cryptographic hash values (*i.e.,* digital fingerprints, explanation to follow) to a huge list of software hash values maintained and published by the National Software Reference Library, a branch of the National Institute for Standards and Technology (**NIST**). The NIST list is free to download, and pretty much everyone who processes data for e-discovery and computer forensic examination uses it.

The value of de-NISTing varies according to the makeup of the collection. It's very effective when ESI has been collected indiscriminately or by forensic imaging of entire hard drives (including operating system and executable files). De-NISTing is of limited value when the collection is composed primarily of user-created files and messages as distinguished from system files and executable applications. As a rule, the better focused the e-discovery collection effort (*i.e.,* the more targeted the collection), the smaller the volume of data culled via de-NISTing.

Cryptographic Hashing (AGAIN?!?!):

We spend considerable time in this class learning that all ESI is just a bunch of numbers. We muddle through readings and exercises about Base2 (binary), Base10 (decimal), Base16 (hexadecimal) and Base64 and about the difference between single-byte encoding schemes (like ASCIII) and double-byte encoding schemes (like Unicode). It may seem like a wonky walk in the weeds; but it's time well spent when you snap to the crucial connection between numeric encoding and our ability to use math to cull, filter and cluster data. It's a necessary precursor to gaining Proustian "new eyes" for ESI.

Because ESI is just a bunch of numbers, we can use algorithms (mathematical formulas) to distill and compare those numbers. Every student of electronic discovery learns about cryptographic hash functions and their usefulness as tools to digitally fingerprint files in support of identification, authentication, exclusion and deduplication. When I teach law students about hashing, I tell them that hash functions are published, standard mathematical algorithms into which we input digital data of arbitrary size and the hash algorithm spits out a bit string (again, just a sequence of numbers) of fixed length called a "hash value." Hash values *almost* exclusively correspond to the digital data fed into the algorithm (termed "the message") such that the chance of two different messages sharing the same hash value (called a "hash collision") is exceptionally remote. But because it's *possible*, we can't say each hash value is truly "unique."

Using hash algorithms, any volume of data—from the tiniest file to the contents of entire hard drives and beyond—can be *almost* uniquely expressed as an alphanumeric sequence. In the case of the MD5 hash function, data is distilled to a value written as 32 hexadecimal characters (0-9 and A-F). It's hard to understand until you've figured out Base16; but, those 32 characters represent 340 *trillion, trillion, trillion* different possible values (2^{128} or 16^{32}).

Hash functions are one-way calculations, meaning you can't reverse ("invert") a hash value and ascertain the data corresponding to the hash value in the same way that you can't decode a human fingerprint to deduce an individual's eye color or IQ. *It identifies, but it doesn't reveal.* Another key feature of hashing is that, due to the so-called “avalanche effect” characteristic of a well-constructed cryptographic algorithm, when the data input changes even slightly, the hash value changes dramatically, meaning there’s no discernable relationship between inputs and outputs. Similarity between hash values doesn’t signal any similarity in the data hashed.

There are lots of different hash algorithms, and different hash algorithms generate different hash values for the same data. That is, the hash value for the phrase “Mary had a little lamb” will be the following in each of the following hash algorithms:

MD5:	e946adb45d4299def2071880d30136d4
SHA-1:	bac9388d0498fb378e528d35abd05792291af182
SHA-256:	efe473564cb63a7bf025dd691ef0ae0ac906c03ab408375b9094e326c2ad9a76

It’s identical data, *but it prompts different hashes using different algorithms*. Conversely, identical data will generate identical hash values when using the *same* hash function. Freely published hash functions are available to all, so if two people (or machines) anywhere use the same hash function against data and generate matching hash values, their data is identical. If they get different hash values, they can be confident the data is different. The differences may be trivial in practical terms, but any difference suffices to produce markedly different hash values.

Let’s dig down a bit here and explore the operations behind calculating an MD5 hash value. If you really don’t care, just skip ahead to deduplication.

A widely used hash function is the Message Digest 5 (MD5) hash algorithm circulated in 1992 by MIT professor Ron Rivest as Requests for Comments 1321. Requests for Comments or RFCs are a way the technology community circulates proposed standards and innovations generally relating to the Internet. MD5 has been compromised in terms of its immunity to hash collisions in that’s it’s feasible to generate different inputs that generate matching MD5 hashes; however, MD5’s flaws minimally impact its use in e-discovery where it remains a practical and efficient way to identify, deduplicate and cull datasets.

When I earlier spoke of a hash algorithm generating a hash value of “fixed length,” that fixed length for MD5 hashes is 128 bits (16 bytes) or 128 ones and zeroes in binary or Base2 notation. That’s a vast number space. It’s 340,282,366,920,938,463,463,374,607,431,768,211,455 possibilities in our familiar decimal or Base10 notation. It’s also unwieldy, so we shorten MD5 hash values to a 32-character Base16 or hexadecimal (“hex”) notation. It’s the same numeric value conveniently expressed in a different base or “radix,” so it requires only one-fourth as many characters to write the number in hex notation as in binary notation.

That 32-*character* MD5 hash value is built from four 32-*bit* calculated values that are concatenated, that is, positioned end to end to form a 128-bit sequence or “string.” Since we can write a 32-bit number as eight hexadecimal characters, we can write a 128-bit number as four concatenated 8-character hex values forming a single 32-character hexadecimal hash. Each of those four 32-bit values are the product of 64 calculations (four rounds of 16 operations) performed on each 512-bit chunk of the data being hashed while applying various specified constants. After each round of calculations, the data shifts within the array in a practice called left bit rotation, and a new round of calculations begins. The entire hashing process starts by padding the message data to ensure it neatly comprises 512-bit chunks and by initializing the four 32-bit variables to four default values.

No, OF COURSE I won't test you on that last paragraph!

Despite the complexity of these calculations, it's possible to contrive hash collisions where different data generate matching MD5 hash values. Accordingly, the cybersecurity community have moved away from MD5 in applications requiring collision resistance, such as digital signatures.

You may wonder why MD5 remains in wide use if it's "broken" by engineered hash collisions. Why not simply turn to more secure algorithms like SHA-256? Some tools and vendors have done so, but a justification for MD5's survival is that the additional calculations required to make alternate hash functions more secure consume more time and computing resources. Too, most tasks in e-discovery built around hashing—*e.g.,* deduplication and De-NISTing—don't demand strict protection from engineered hash collisions. For e-discovery, MD5 "ain't broke," so there's little cause to fix it.

Deduplication

Processing information items to calculate hash values supports several capabilities, but probably none more useful than deduplication.

Near-Deduplication

A modern hard drive holds trillions of bytes, and even a single Outlook e-mail container file typically comprises billions of bytes. Accordingly, it's easier and faster to compare 32-character/16 byte "fingerprints" of voluminous data than to compare the data itself, particularly as the comparisons must be made repeatedly when information is collected and processed in e-discovery. In practice, each file ingested and each item extracted is hashed, and its hash value is compared to the hash values of items previously ingested and extracted to determine if the file or item has been seen before. The first file is sometimes called the "pivot file," and subsequent files with matching hashes are suppressed as duplicates, and the instances of each duplicate and certain metadata is typically noted in a deduplication or "occurrence" log.

When the data is comprised of loose files and attachments, a hash algorithm tends to be applied to the full contents of the files. Notice that I said to "*contents*." Some data we associate with files is not actually stored inside the file but must be gathered from the file system of the device storing the data. Such "system metadata" is not contained within the file and, thus, is not included in the calculation when the file's content is hashed. A file's name is perhaps the best example of this. Recall that even slight differences in files cause them to generate different hash values. But, since a file's name is not typically housed within the file, you can change a file's name without altering its hash value.

So, the ability of hash algorithms to deduplicate depends upon whether the numeric values that serve as building blocks for the data differ from file to file. Keep that firmly in mind as we consider the many forms in which the informational payload of a document may manifest.

A Word .DOCX document is constructed of a mix of text and rich media encoded in Extensible Markup Language (XML), then compressed using the ubiquitous ZIP compression algorithm. It's a file designed to be read by Microsoft Word.

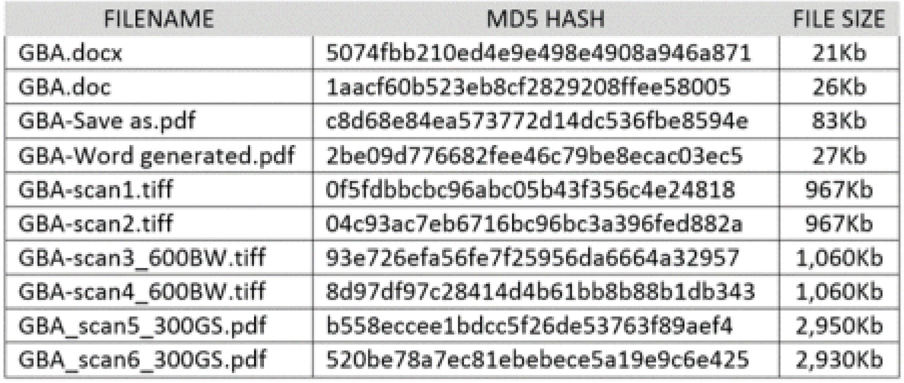
When you print the "same" Word document to an Adobe PDF format, it's reconstructed in a *page description language* specifically designed to work with Adobe Acrobat. It's structured, encoded and compressed in an entirely different way than the Word file and, as a different format, carries a different binary header signature, too.

When you take the printed version of the document and scan it to a Tagged Image File Format (TIFF), you've taken a picture of the document, now constructed in still another different format—one designed for TIFF viewer applications.

To the uninitiated, they are all the "same" document and might look pretty much the same printed to paper; but as ESI, their structures and encoding schemes are radically different. Moreover, even files generated in the same format may not be *digitally* identical when made at separate times. For example, no two optical scans of a document will produce identical hash values because there will always be some variation in the data acquired from scan to scan. Slight differences perhaps; but, any difference at all in content is going to frustrate the ability to generate matching hash values.

Opinions are cheap. Testing is truth. To illustrate this, I created a Word document of the text of Lincoln's Gettysburg Address. First, I saved it in the latest .DOCX Word format. Then, I saved a copy in the older .DOC format. Next, I saved the Word document to a .PDF format, using both the Save as PDF and Print to PDF methods. Finally, I printed and scanned the document to TIFF and PDF. Without shifting the document on the scanner, I scanned it several times at matching and differing resolutions.

I then hashed all the iterations of the "same" document. As the table below demonstrates, none of them matched hash-wise, not even the successive scans of the paper document:



FILENAME	MD5 HASH	FILE SIZE
GBA.docx	5074fbb210ed4e9e498e4908a946a871	21Kb
GBA.doc	1aacf60b523eb8cf2829208ffee58005	26Kb
GBA-Save as.pdf	c8d68e84ea573772d14dc536fbe8594e	83Kb
GBA-Word generated.pdf	2be09d776682fee46c79be8ecac03ec5	27Kb
GBA-scan1.tiff	0f5fdbbcbc96abc05b43f356c4e24818	967Kb
GBA-scan2.tiff	04c93ac7eb6716bc96bc3a396fed882a	967Kb
GBA-scan3_600BW.tiff	93e726efa56fe7f25956da6664a32957	1,060Kb
GBA-scan4_600BW.tiff	8d97df97c28414d4b61bb8b88b1db343	1,060Kb
GBA_scan5_300GS.pdf	b558eccee1bdcc5f26de53763f89aef4	2,950Kb
GBA_scan6_300GS.pdf	520be78a7ec81ebebece5a19e9c6e425	2,930Kb

Thus, file hash matching—the simplest and most defensible approach to deduplication—won't serve to deduplicate the "same" document when it takes different forms or is made optically at separate times.

Now, here's where it can get confusing. If you copied any of the electronic *files* listed above, the duplicate files would hash match the source originals and would handily deduplicate by hash. Consequently, multiple copies of the same electronic files will deduplicate, but that is because the files being compared have the same *digital* content. But we must be careful to distinguish the identicality seen in multiple iterations of the same file from the pronounced differences seen when we generate different electronic versions at different times from the same content. One notable exception seen in my testing was that successively saving the same Word document to a PDF format in the same manner sometimes generated identical PDF files. It didn't occur consistently (*i.e.,* if enough time passed, changes in metadata in the source document triggered differences prompting the calculation of different hash values); but it happened, so is worth mentioning.

Here, a quick primer on deduplication of e-mail might be useful.

Mechanized deduplication of e-mail data can be grounded on three basic approaches:

1. Hashing the entire message as a file (*i.e.,* a defined block of data) containing the e-mail messages and comparing the resulting hash value for each individual message file. If they match, the files hold the same data. This tends not to work for e-mail messages exported as files because, when an e-mail message is stored as a file, messages that we regard as identical in common parlance (such as identical message bodies sent to multiple recipients) are not identical in terms of their byte content. The differences tend to reflect either variations in transmission seen in the message header data (the messages having traversed different paths to reach different recipients) or variations in time (the same message containing embedded time data when exported to single message storage formats as discussed above with respect to the .MSG format).

2. Hashing segments of the message using the same hash algorithm and comparing the hash values for each corresponding segment to determine relative identicality. With this approach, a hash value is calculated for the various parts of a message (*e.g.,* Subject, To, From, CC, Message Body, and Attachments) and these values are compared to the hash values calculated against corresponding parts of other messages to determine if they match. This method requires exclusion of those parts of a message that are certain to differ (such as portions of message headers containing server paths and unique message IDs) and normalization of segments, so that contents of those segments are presented to the hash algorithm in a consistent way.
3. Textual comparison of segments of the message to determine if certain segments of the message match to such an extent that the messages may be deemed sufficiently "identical" to allow them to be treated as the same for purposes of review and exclusion. This is much the same approach as (2) above, but without the use of hashing to compare the segments.

Arguably, a fourth approach entails a mix of these methods.

All these approaches can be frustrated by working from differing forms of the "same" data because, from the standpoint of the tools which compare the information, the forms are significantly different. Thus, if a message has been "printed" to a TIFF image, the bytes that make up the TIFF image bear no digital resemblance to the bytes comprising the corresponding e-mail message, any more than a photo of a rose smells or feels like the rose.

In short, changing forms of ESI changes data, and changing data changes hash values. Deduplication by hashing requires the same source data and the same, consistent application of algorithms. This is easy and inexpensive to accomplish, but it requires a compatible workflow to ensure that evidence is not altered in processing to in ways that might prevent the application of simple and inexpensive mechanized deduplication.

When parties cannot deduplicate e-mail, the reasons will likely be one or more of the following:

1. They are working from different forms of the ESI
2. They are failing to consistently exclude inherently non-identical data (like message headers and IDs) from the hash calculation
3. They are not properly normalizing the message data (such as by ordering all addresses alphabetically without aliases)
4. They are using different hash algorithms
5. They are not preserving the hash values throughout the process; or
6. They are changing the data.

Other Processing Tasks

This chapter addresses the core functions of ESI processing in e-discovery, but there are many other processing tasks that make up today's powerful processing tools. Some examples include:

Foreign Language Detection

Several commercial and open-source processing tools support the ability to recognize and identify foreign language content, enabling selection of the right filters for text extraction, character set selection and diacritical management. Language detection also facilitates assigning content to native speakers for review.

Entropy Testing

Entropy testing is a statistical method by which to identify encrypted files and flag them for special handling.

Decryption

Some processing tools support use of customizable password lists to automate decryption of password-protected items when credentials are known.

Bad Extension Flagging

Most processing tools warn of a mismatch between a file's binary signature and its extension, potentially useful to resolve exceptions and detect data hiding.

Color Detection

When color conveys information, it's useful to detect such usage and direct color-enhanced items to production formats other than grayscale TIFF imaging.

Hidden Content Flagging

It's common for evidence, especially Microsoft Office content, to incorporate relevant content (like collaborative comments in Word documents and PowerPoint speaker notes) that won't appear in the production set. Flagging such items for special handing is a useful way to avoid missing that discoverable (and potentially privileged) content.

N-Gram and Shingle Generation

Increasingly, advanced analytics like predictive coding aid the review process and depend upon the ability to map document content in ways that support algorithmic analysis. N-gram generation and text shingling are text sampling techniques that support latent-semantic analytics.

Optical Character Recognition (OCR)

OCR is the primary means by which text stored as imagery, thus lacking a searchable text layer (*e.g.,* TIFF images, JPGs and PDFs) can be made text searchable. Some processing tools natively support optical character recognition, and others require users to run OCR against exception files in a separate workflow then re-ingest the content accompanied by its OCR text.

Virus Scanning

Files collected in e-discovery may be plagued by malware, so processing may include methods to quarantine afflicted content via virus scanning.

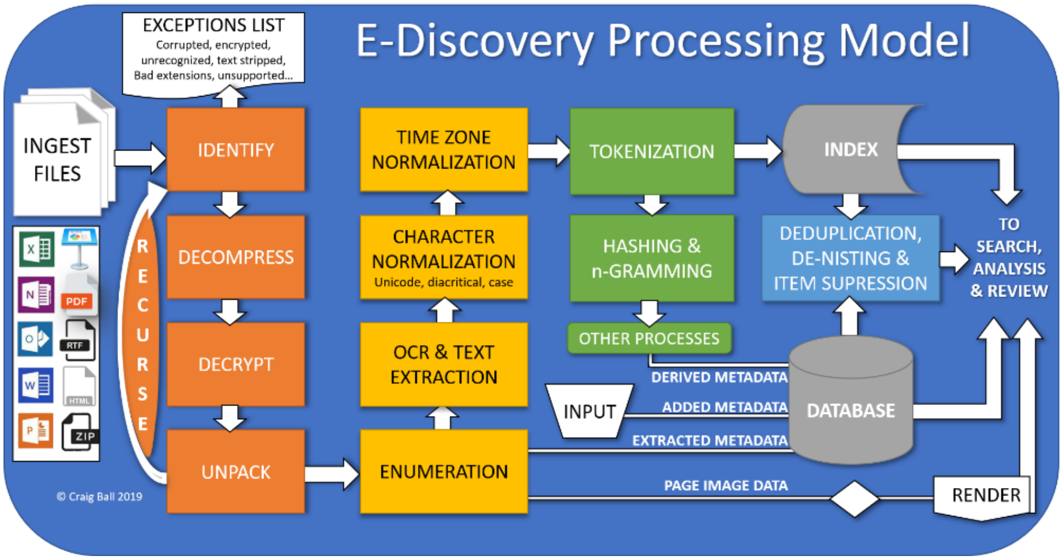
Production Processing

Heretofore, we've concentrated on processing before search and review, but there's typically a processing workflow that follows search and review: **Production Processing**. Some tools and workflows convert items in the collection to imaged formats (TIFF or PDF) before review; in others, imaging is obviated by use of a viewer component of the review tool. If not imaged before review, the e-discovery tool may need to process items selected for production and redaction to imaged formats suited to production and redaction.

Further in conjunction with the production process, the tool will generate a load file to transmit extracted metadata and data describing the organization of the production, such as pointers to TIFF images and text extractions. Production processing will also entail assignment of Bates numbers to the items produced and embossing of Bates numbers and restrictive language (*i.e.*, "Produced Subject to Protective Order").

Illustration 1: E-Discovery Processing Model

Larger version next page.



E-Discovery Processing Model

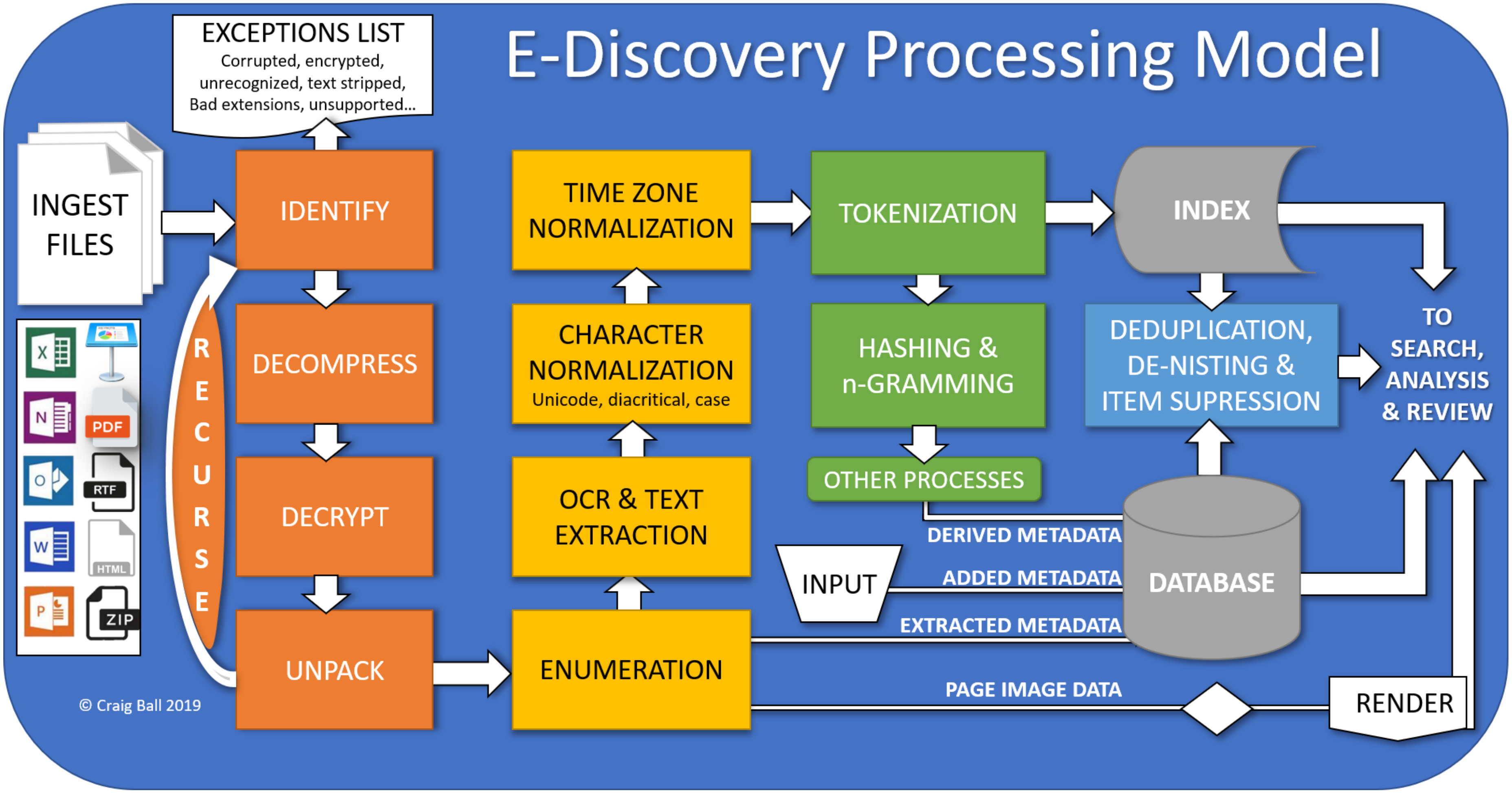
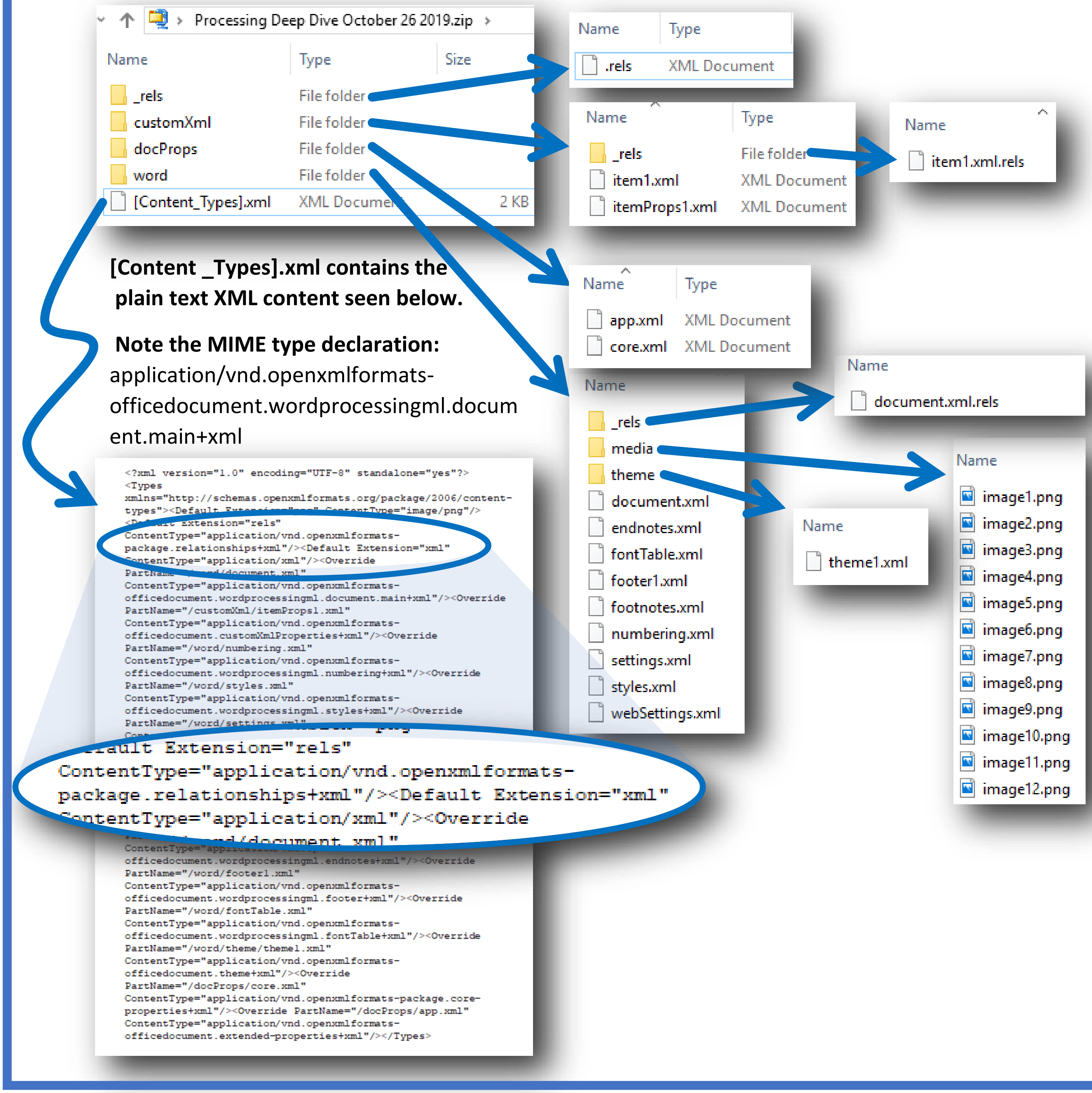


Illustration 2: Anatomy of a Word DOCX File

By changing a Word document's extension from .DOCX to .ZIP, you can "trick" Windows into decompressing the file and sneak a peek into its internal structure and contents. Those contents little resemble the document you'd see in Word; but, as you peruse the various folders and explore their contents, you'll find the text, embedded images, formatting instructions and other components Word assembles to compose the document.

The root level of the decompressed file (below left) contains four folders and one XML file.



Processing Deep Dive October 26 2019.zip

Name	Type	Size
_rels	File folder	
customXml	File folder	
docProps	File folder	
word	File folder	
[Content_Types].xml	XML Document	2 KB

Name	Type
.rels	XML Document

Name	Type
_rels	File folder
item1.xml	XML Document
itemProps1.xml	XML Document

Name
item1.xml.rels

Name	Type
app.xml	XML Document
core.xml	XML Document

Name
_rels
media
theme
document.xml
endnotes.xml
fontTable.xml
footer1.xml
footnotes.xml
numbering.xml
settings.xml
styles.xml
webSettings.xml

Name
document.xml.rels

Name
theme1.xml

Name
image1.png
image2.png
image3.png
image4.png
image5.png
image6.png
image7.png
image8.png
image9.png
image10.png
image11.png
image12.png

[Content _Types].xml contains the plain text XML content seen below.

Note the MIME type declaration: application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types
xmlns="http://schemas.openxmlformats.org/package/2006/content-
types"><Default Extension="png" ContentType="image/png"/>
<Default Extension="rels"
ContentType="application/vnd.openxmlformats-
package.relationships+xml"/><Default Extension="xml"
ContentType="application/xml"/><Override
PartName="/word/document.xml"
ContentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.document.main+xml"/><Override
PartName="/customXml/itemProps1.xml"
ContentType="application/vnd.openxmlformats-
officedocument.customXmlProperties+xml"/><Override
PartName="/word/numbering.xml"
ContentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.numbering+xml"/><Override
PartName="/word/styles.xml"
ContentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.styles+xml"/><Override
PartName="/word/settings.xml"
```

```
Default Extension="rels"
ContentType="application/vnd.openxmlformats-
package.relationships+xml"/><Default Extension="xml"
ContentType="application/xml"/><Override
PartName="/word/document.xml"
```

```
ContentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.endnotes+xml"/><Override
PartName="/word/footer1.xml"
ContentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.footer+xml"/><Override
PartName="/word/fontTable.xml"
ContentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.fontTable+xml"/><Override
PartName="/word/theme/theme1.xml"
ContentType="application/vnd.openxmlformats-
officedocument.theme+xml"/><Override
PartName="/docProps/core.xml"
ContentType="application/vnd.openxmlformats-package.core-
properties+xml"/><Override PartName="/docProps/app.xml"
ContentType="application/vnd.openxmlformats-
officedocument.extended-properties+xml"/></Types>
```

Processing Glossary

Term	Definition
ASCII	American Standard Code for Information Interchange (ASCII) is a plain text character encoding standard where seven- or eight-bit integers correspond to 128 or 256 characters and codes for electronic storage and communication. The eight-bit pairings are often mistakenly referred to as *Extended ASCII*. The 128 characters in 7-bit ASCII encoding correspond to 95 printable characters (a-z, A-Z, 0-9 and punctuation) and 33 non-printable control codes, e.g., carriage return, line feed, tab and bell. The 256 ASCII characters enabled by 8-bit integers are for various purposes, *e.g.*, foreign language characters and line drawing symbols.
Bates Numbers	Sequential numeric identifiers imprinted on document pages or assigned to files during the discovery process. Bates Numbers typically include a prefix to identify the producing party or matter as well as a numeric value (*e.g.*, DEF_000000001).
Binary File Signature	Also known as "file header signature," "binary header signature" or "magic number." Typically, the first few bytes of data in a file identifies the format of the data contained therein. For example, ZIP compressed files begin with Hex 504B (or the initials PK in ASCII). Most JPG image files begin with Hex FF D8 FF E0.
Case Normalization	Improves search recall by adding information to an index that searches for terms with lowercase characters and identifies its uppercase counterpart and vice versa. For example, a search for Rice will also find instances of RICE and rice.
Character Normalization	Seeks to minimize the impact of variations in alphanumeric characters often overlooked by human beings but posing a challenge to machines. This may include Case Normalization, Diacritical Normalization and Unicode Normalization.
Chain of Custody	The procedures employed to protect and document the acquisition, handling and storage of evidence to demonstrate these activities did not alter or corrupt evidentiary integrity.

Compression	The storage or transmission of data in a reduced size by using technology to eliminate redundancy ("lossless compression") or by removing non-essential details (such as, picture elements in a JPEG or inaudible components of audio). Compression permits more efficient storage, sometimes at the cost of reduced fidelity ("lossy compression"). ZIP, RAR and TAR are common lossless compression formats in eDiscovery.
Container File	A file that holds or transports other files, *e.g.*, compressed container files (.ZIP and .RAR) and email container files (.PST and .MBOX). Container file content is "unpacked" or “exploded” during processing enabling the container file to be suppressed as immaterial once fully extracted.
Corruption	Damage to the integrity of a file that impacts its ability to be processed. File corruption may be caused by, *e.g.,* network transmission errors, software glitches, physical damage to storage media (i.e., bad sectors) or use of an incompatible decoding tool.
Custodian	The individuals or entities who hold, or have the right to control, records and information.
DAT File	A delimited load file used in conjunction with Concordance-formatted productions. A .DAT file includes a header row of field identifiers that corresponds to the data that follows. Each field is separated ("delimited") by a character ("delimiter") that signals the division of fields.
Deduplication	The identification and suppression of identical copies of messages or documents in a data set based upon the items' hash values or other criteria.
DeNIST	The use of hash values to identify, suppress and/or remove commercial software from a data collection. The hash values are maintained by the National Institute of Standards and Technology (NIST) in its National Software Reference Library (NSRL).
Diacritical Normalization	Improves search recall by adding to an index terms with diacritics (*e.g.*, accented characters) so as to locate counterparts without diacritics. For example, a search for "résumé" would also locate instances of resume and vice versa.

DTSearch	A content extraction, indexing and text search tool licensed to and at the heart of several leading e-discovery and computer forensic tools (*e.g.,* Relativity, LAW, Ringtail (now Nuix Discover) and Access Data's FTK).
Encoding	The process of converting electronically stored and transmitted information from one form to another. Character encoding maps alphanumeric characters into numeric values, typically notated as binary or hexadecimal numbers. ASCII and Unicode are examples of character encoding.
Encryption	The process of encoding data to unintelligible ciphertext to prevent access without the proper decryption key (*e.g.*, password).
ESI	Electronically Stored Information (ESI) as defined by Federal Rule of Civil Procedure 34(a)(1)(A), includes "writings, drawings, graphs, charts, photographs, sound recordings, images, and other data or data compilations—stored in any medium from which information can be obtained either directly or, if necessary, after translation by the responding party into a reasonably usable form."
Exception Reporting	This process of identifying items which fail during processing. Exceptions may include encrypted files that cannot be read, corrupt files, files in unrecognized formats or languages, and files that require optical character recognition (OCR) for text extraction.
Family Group	In the context of an email, a transmitting message (parent object) and its attachments (child objects).
File Header Signature	Also known as a "binary header signature," "binary file signature" or "magic number." Typically, the first few hex bytes of data in a file identifies the format of the data within the file. For example, ZIP compressed files begin with Hex 504B (or the initials PK in ASCII). Most JPG image files begin with Hex FF D8 FF E0.
Filtering	The process of culling files from a data set based on characteristics such as, file type, date and size. In e-discovery, files are filtered to suppress multiple copies of the same item (deduplication), irrelevant system files (deNISTing), immaterial container files after content extraction and by lexical search (filtering by keywords).

Forensic Image	An exact, verified copy of electronic media. Forensic imaging produces a hash-authenticated, sector-by-sector ("bitstream") copy of electronic media that can be restored for analysis. This process is typically used to preserve active data, unallocated clusters and file slack space.
Hash	A "digital fingerprint" of data or "message digest," generated by a one-way cryptographic algorithm (*e.g.*, MD5, SHA-1, SHA-256) and recorded as a hexadecimal character string, *e.g.* 13bfb1528002a68d94249c4ffb09359f. The potential of two different files having matching hash values is so remote that hash value comparisons serve as effective tools for file authentication, file exclusion (DeNISTing) and data deduplication.
Identification	In e-discovery, the mechanism by which a processing tool determines the structure and encoding of a file based upon the file's header signature and filename extension.
IM	Instant Message (IM) is a form of real-time text communication over the Internet typically expressed in conversation form. IM can involve communications between two people or larger groups, who sometimes communicate in “rooms.”
Image Format	Images initially referred to the output from document scanning but can also refer to files rendered directly from native files. These files are created to emulate a printed page. In e-discovery, the most common image formats are Tagged Image File Format (TIFF), Portable Document Format (PDF) and JPEG. "Rendering" is the processing step where ESI is converted to image formats.
Index	A data structure that improves the speed of search for data retrieval. E-discovery employs full text indexing of processed data to speed search and to reduce storage space.
Ingestion	The act of loading data into an application for processing.
Keyword	Search term used to query an index or database.

Language Detection	Recognition and identification of foreign language content that enables selection of appropriate filters for text extraction, character set selection and diacritical management. Language detection also facilitates assigning foreign language content to native speakers for review.
Load File	An ancillary file used in e-discovery to transmit, system and application metadata, extracted text, Bates numbers and structural information describing the production. Load files accompany folders holding native, text and image files and provide essential information about the files being transmitted.
MD5	Message Digest 5 (MD5) is a common cryptographic hash algorithm used for file authentication, file exclusion (DeNISTing) and data deduplication.
Metadata	Data describing the characteristics of other data. File metadata may be System Metadata (*e.g.,* file name, size and date last modified, accessed or created are stored outside the file) or Application Metadata (*e.g.,* last printed date or amount of editing time stored within the file). The term metadata can also include human judgments about a file, e.g. hot or privileged, or information about the file, e.g. from, to, subject, sent date.
MIME	Multipurpose Internet Mail Extensions (MIME) refers to a two-part, hierarchical method of classification for electronic files. MIME Types (also known as Media Types) classify files within one of ten types: *application, audio, image, message, multipart, text, video, font, example* and *model*. Each type is divided into subtypes with sufficient granularity to describe all common variants within the type. For example, the MIME Type of a PDF file is "application/pdf," a .DOCX file is "application/vnd.openxmlformats-officedocument.wordprocessingml.document," and a TIFF image file is "image/tiff." The Internet Assigned Numbers Authority (IANA) is a standards organization that registers new types and subtypes in the MIME Type taxonomy.
Media Type	Alternate term for MIME Type, see MIME.

Native Format	In the context of software applications, native format refers to the file format which an application creates and uses by design—generally the default, unprocessed format of a file when collected from the original source, *e.g.,* Microsoft Word stores documents as .DOCX files, their native format.
NSRL	The National Software Reference Library (NSRL) is maintained by the National Institute of Standards and Technology (NIST), an agency of the U.S. Department of Commerce. The data published by the NSRL (principally hash values of commercial software) is used to rapidly identify and eliminate known files, such as operating system and application files.
Noise Words	Common terms purposefully excluded from a searchable index to conserve storage space and improve performance. Also known as "stop words."
Normalization	The process of reformatting data to a standardized form, such as setting the date and time stamp of files to a uniform time zone or converting all content to the same character encoding. Normalization facilitates search and data organization.
OCR or Optical Character Recognition	The use of software to identify alphanumeric characters in static images (*i.e.,* TIFF or PDF files) to facilitate text extraction and electronic search. OCR programs typically create matching text files that are used for text search with the accompanying images.
Processing	Encompasses the steps required to extract text and metadata from information items and to build a searchable index. ESI processing tools perform five common functions: (1) decompress, unpack and fully explore (*i.e., recurse)* ingested items; (2) identify and apply templates (filters) to encoded data to parse (interpret) contents and extract text, embedded objects, and metadata; (3) track and hash items processed, enumerate and unitize all items, and track failures; (4) normalize and tokenize text and data and create an index and database of extracted information; and (5) cull data by file type, date, lexical content, hash value, and other criteria.
Recursion	The mechanism by which a processing tool explores, identifies, unpacks and extracts all embedded content from a file, repeating the recursive process as many times as needed to achieve full extraction.

Request for Comment (RFC)	The longstanding, informal circulation of proposed protocols and standards among computer scientists, engineers and others interested in the development of the Internet and other networks. RFCs define the structure of email messages and attachments for transmission via the Internet.
SHA	Secure Hash Algorithm (SHA) (SHA-1, SHA-256) is a family of cryptographic hash algorithms used for file authentication, file exclusion (DeNISTing) and data deduplication.
SMS	Short Message Service (SMS) is a communication protocol that enables mobile devices to exchange text messages up to 160 characters in length.
Stop Words	Common terms purposefully excluded from a searchable index to conserve storage space and improve performance. Also known as "noise words."
System Files	The program and driver files crucial to the overall function of a computer's operating and file systems. Because system files are not user-created, they may be excluded from a collection of potentially responsive data by deNISTing.
Targeted Collection	A technique used to reduce overcollection of ESI by marshaling potentially responsive data based on data characteristics (such as, file type, date, folder location, keyword search, etc.) as opposed to duplicating the entire contents of a storage device (*e.g.,* by imaging).
Threading	Collection and organization of messaging as a chronologically ordered conversation.
Tika	An open-source toolkit for extracting text and metadata from over one thousand file types, including most encountered in e-discovery. Tika was a subproject of the open-source Apache Lucene project. Lucene is an indexing and searching tool at the core of several commercial e-discovery applications.
Time Zone Normalization	The recasting of time values of ESI--particularly of e-mail collections--to a common temporal baseline, often Coordinated Universal Time (UTC) or another time zone the parties designate.

Tokenization	A method of document parsing that identifies words ("tokens") to be used in a full-text index. Because computers cannot read as humans do but only see sequences of bytes, computers employ programmed tokenization rules to identify character sequences that constitute words and punctuation. Western languages typically use spaces and punctuation to identify word (or token) breaks. Because other languages, *e.g.* Chinese, Japanese and Korean, do not use these methods to break characters into words, l tokenization software ensures that words and other tokens are indexed properly for search.
Unicode	An international, multibyte encoding scheme for text, symbols, emoji and control codes. Unicode 13.0 offers 154 encoding schemes or scripts comprising 143,859 characters. Unicode was developed to overcome the limits of the single byte ASCII encoding scheme that lacked the capacity to encode foreign language characters and other symbols needed for international writing and communication. Unicode is now the standard for Western and international text encoding.
Unicode Normalization	Improves search recall by adding information to an index that locates Unicode characters encoded in multiple ways when searching with any counterpart encoding. Linguistically identical characters encoded in Unicode (so-called "canonical equivalents") may be represented by different numeric values by virtue of accented letters having both precomposed (é) and composite references (e + ◌́). Unicode normalization replaces equivalent sequences of characters so that any two texts that are canonically equivalent will be reduced to the same sequence of searchable code called the "normalization form" or "normal form" of the original text.
UTF-8	Unicode Transformation Format (character encoding 8) or UTF-8 is the most widely used Unicode encoding, employing one byte for standard English letters and symbols (making UTF-8 backwards compatible with ASCII), two bytes for additional Latin and Middle Eastern characters, and three bytes for Asian characters. Additional characters may be represented using four bytes.

❁ Exercise 10: Metadata: File Table Data

GOALS: The goals of this exercise are for the student to:

1. Distinguish between system metadata and application metadata;
2. Explore the Windows Master File Table; and
3. Understand that, because system metadata is not stored within the file, you don't preserve system metadata by simply copying the contents of the file; you must grab its system metadata from the file table, too.

OUTLINE: Students will create a file in Notepad and search within and without the file for metadata.

This easy exercise is a belated return to metadata, but purposefully, because we need to nail down that crucial distinction between system and application metadata and what each holds and take a quick peek at encoded file times.

Background

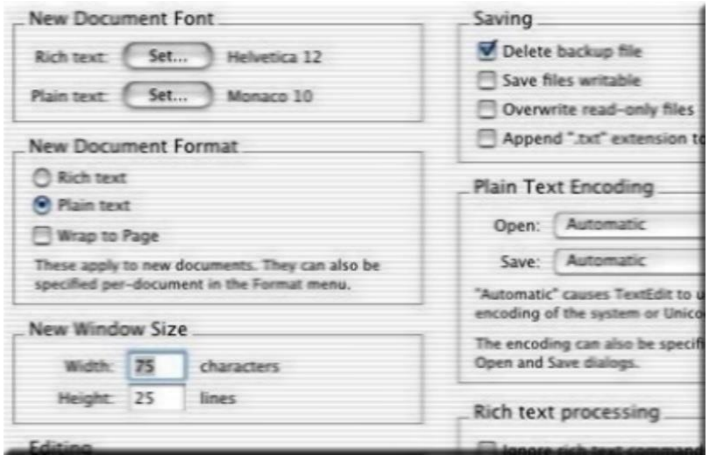
Once again we will use the HexDump utility at http://www.fileformat.info/tool/hexdump.htm, to view every byte in a file as ASCII text and hex values. So, when you know a file has attendant metadata that you can't see within the four corners of the file, it must be *somewhere*. Where does that metadata come from?

Step 1: Create a simple text file

In Windows: On your Desktop, right click on an open area select New>Text Document. Name the file "**me.txt**," then open the file you just created and type your full name. Save the file, then close it. Double click the file to re-open it and confirm that your name appears in the document named me.txt on your Desktop.

In MacOS: You will use the Mac default text editor called TextEdit to create a plain text (ASCII) file. But, since the Text Edit program creates Rich Text (RTF) formats by default, you must first modify some program settings:

a. Open TextEdit.
b. Choose Preferences from the TextEdit application menu.
c. Click the Plain Text radio button for New Document Format.
d. Be sure the checkbox for "Wrap to Page" is deselected.
e. Close the Preferences box.



New Document Font
Rich text: Set... Helvetica 12
Plain text: Set... Monaco 10

New Document Format
○ Rich text
● Plain text
□ Wrap to Page
These apply to new documents. They can also be specified per-document in the Format menu.

New Window Size
Width: 75 characters
Height: 25 lines

Editing

Saving
☑ Delete backup file
□ Save files writable
□ Overwrite read-only files
□ Append ".txt" extension t

Plain Text Encoding
Open: Automatic
Save: Automatic
"Automatic" causes TextEdit to u
encoding of the system or Unico
The encoding can also be specif
Open and Save dialogs.

Rich text processing
□ Ignore rich text command

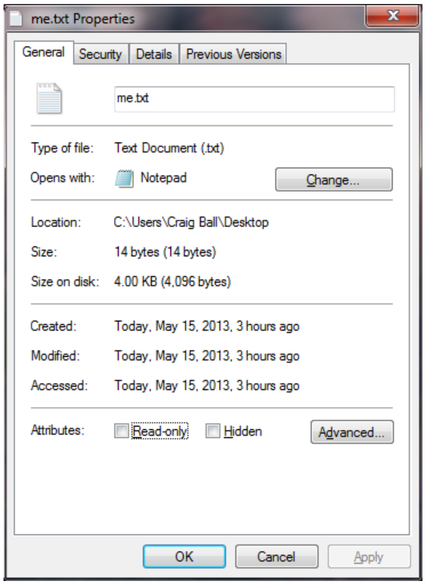
Create a new file using TextEdit and type just your full name in the file. Save the file as "me.txt" to your Desktop and close it. Re-open the me.txt file and confirm that your name appears in the document.

Step 2: Check the Properties

In Windows: Right click on your me.txt file and select "Properties."

In MacOS: Right click and select Get Info

Note the file's size and its size on disk. The first reflects the actual byte count for the data needed to store your name (including spaces). The size on disk is the total size of the cluster(s) allocated to storing the file. On my Windows machine, the drive is logically divided into 4 kilobyte clusters, so every file occupies at least 4KB on disk as seen in the figure at right (but see the discussion of resident MFT data below).



me.txt Properties

me.txt

Type of file: Text Document (.txt)

Opens with: Notepad

Location: C:\Users\Craig Ball\Desktop

Size: 14 bytes (14 bytes)

Size on disk: 4.00 KB (4,096 bytes)

Created: Today, May 15, 2013, 3 hours ago

Modified: Today, May 15, 2013, 3 hours ago

Accessed: Today, May 15, 2013, 3 hours ago

Attributes: □ Read-only □ Hidden

Step 3: Dump the Hex

Using your web browser, go to the Online HexDump Utility at http://www.fileformat.info/tool/hexdump.htm and click "choose File." Using the selection box that will appear, navigate to the file you just created called "me.txt." Click "Open." Now click the blue "Dump" button on the Online HexDump Utility page. You should see something like this (but with *your* name, of course):

```
file name: me.txt
mime type:

0000-000e:  59 6f 75 72-20 66 75 6c-6c 20 4e 61-6d 65        Your.ful l.Name
```

Step 3: Carefully examine the complete contents of the file

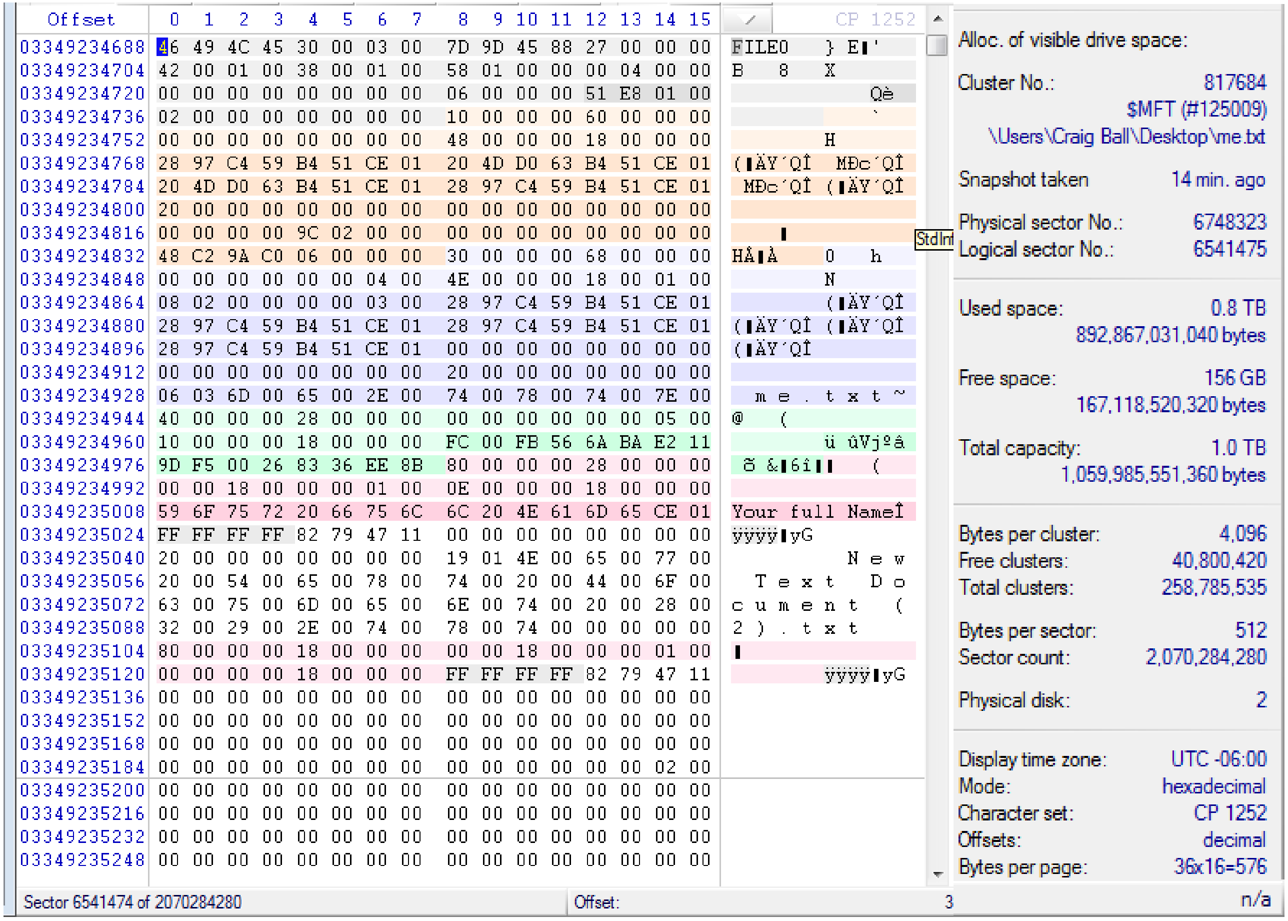
Look at the hex. Look at the text. Do you see any data within the file other than your name? Do you see any file path (location) data? Do you see any date or time data? *Do you even see the file's name within the file?* Every file has system metadata, *so where's the metadata if it's not in the file? It's in the file table: the MFT in Windows or the Catalog File in an Apple system.*

Plumbing the Windows MFT

MFT stands for **Master File Table.** On a Windows system, the MFT is like a library card catalog, storing information about the location of the "book" (file) and describing some of its characteristics (system metadata). The MFT is where most system metadata reside, in contrast to *application* metadata, which resides within the file it describes and moves with the file when copied.

The MFT is made up of numerous 1,024-byte entries, each describing a file or folder stored on the media. The image below is a screenshot of the MFT entry for the **me.txt** file on my Windows Desktop. Like all MFT entries, it begins with FILE0, and after some weirdly encoded stuff, you'll see the name (me.txt) stored in code page 1252 (which if you remember our processing discussion is Microsoft's version of ISO 8859-1, a/k/a Latin 1). Note the spaces between the letters of the file name, which tells us it is **double byte encoded**, that is, in a Unicode-compatible way.

Master File Table Entry for me.txt



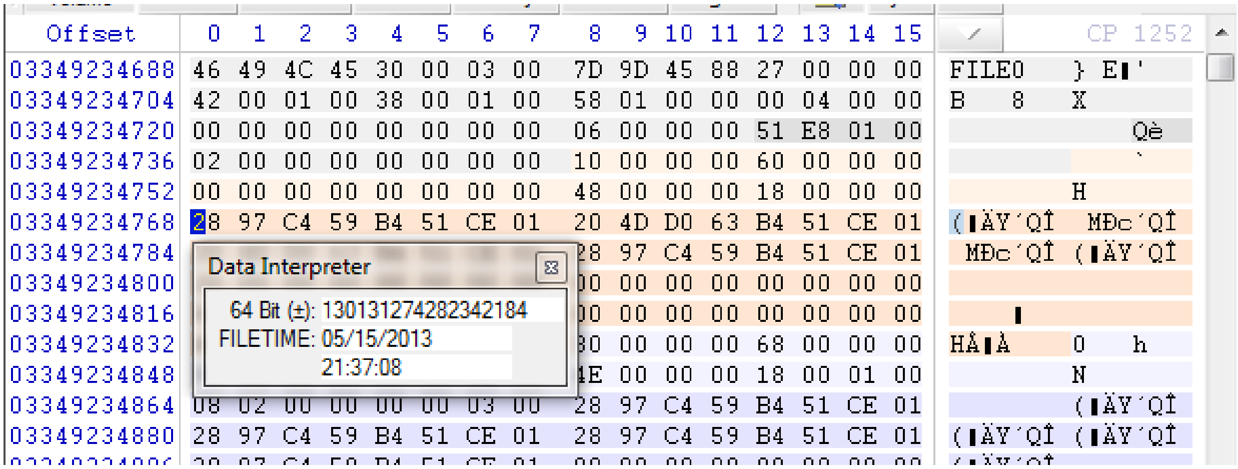
```
Offset       0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15   CP 1252
03349234688 46 49 4C 45 30 00 03 00  7D 9D 45 88 27 00 00 00  FILE0   } E▌'
03349234704 42 00 01 00 38 00 01 00  58 01 00 00 00 04 00 00  B   8   X
03349234720 00 00 00 00 00 00 00 00  06 00 00 00 51 E8 01 00              Qè
03349234736 02 00 00 00 00 00 00 00  10 00 00 00 60 00 00 00              `
03349234752 00 00 00 00 00 00 00 00  48 00 00 00 18 00 00 00          H
03349234768 28 97 C4 59 B4 51 CE 01  20 4D D0 63 B4 51 CE 01  (▌ÄY´QÎ  MÐc´QÎ
03349234784 20 4D D0 63 B4 51 CE 01  28 97 C4 59 B4 51 CE 01   MÐc´QÎ (▌ÄY´QÎ
03349234800 20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
03349234816 00 00 00 00 9C 02 00 00  00 00 00 00 00 00 00 00      ▌
03349234832 48 C2 9A C0 06 00 00 00  30 00 00 00 68 00 00 00  HÂ▌À    0   h
03349234848 00 00 00 00 00 00 04 00  4E 00 00 00 18 00 01 00          N
03349234864 08 02 00 00 00 00 03 00  28 97 C4 59 B4 51 CE 01          (▌ÄY´QÎ
03349234880 28 97 C4 59 B4 51 CE 01  28 97 C4 59 B4 51 CE 01  (▌ÄY´QÎ (▌ÄY´QÎ
03349234896 28 97 C4 59 B4 51 CE 01  00 00 00 00 00 00 00 00  (▌ÄY´QÎ
03349234912 00 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00
03349234928 06 03 6D 00 65 00 2E 00  74 00 78 00 74 00 7E 00    m e . t x t ~
03349234944 40 00 00 00 28 00 00 00  00 00 00 00 00 00 05 00  @   (
03349234960 10 00 00 00 18 00 00 00  FC 00 FB 56 6A BA E2 11          ü ûVjºâ
03349234976 9D F5 00 26 83 36 EE 8B  80 00 00 00 28 00 00 00   õ &▌6î▌▌    (
03349234992 00 00 18 00 00 00 01 00  0E 00 00 00 18 00 00 00
03349235008 59 6F 75 72 20 66 75 6C  6C 20 4E 61 6D 65 CE 01  Your full NameÎ
03349235024 FF FF FF FF 82 79 47 11  00 00 00 00 00 00 00 00  ÿÿÿÿ▌yG
03349235040 20 00 00 00 00 00 00 00  19 01 4E 00 65 00 77 00            N e w
03349235056 20 00 54 00 65 00 78 00  74 00 20 00 44 00 6F 00    T e x t   D o
03349235072 63 00 75 00 6D 00 65 00  6E 00 74 00 20 00 28 00  c u m e n t   (
03349235088 32 00 29 00 2E 00 74 00  78 00 74 00 00 00 00 00  2 ) . t x t
03349235104 80 00 00 00 18 00 00 00  00 00 18 00 00 00 01 00  ▌
03349235120 00 00 00 00 18 00 00 00  FF FF FF FF 82 79 47 11          ÿÿÿÿ▌yG
03349235136 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
03349235152 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
03349235168 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
03349235184 00 00 00 00 00 00 00 00  00 00 00 00 00 00 02 00
03349235200 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
03349235216 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
03349235232 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
03349235248 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
Sector 6541474 of 2070284280          Offset:          3
```

Alloc. of visible drive space:

Cluster No.:	817684
	$MFT (#125009)
	\Users\Craig Ball\Desktop\me.txt
Snapshot taken	14 min. ago
Physical sector No.:	6748323
Logical sector No.:	6541475
Used space:	0.8 TB
	892,867,031,040 bytes
Free space:	156 GB
	167,118,520,320 bytes
Total capacity:	1.0 TB
	1,059,985,551,360 bytes
Bytes per cluster:	4,096
Free clusters:	40,800,420
Total clusters:	258,785,535
Bytes per sector:	512
Sector count:	2,070,284,280
Physical disk:	2
Display time zone:	UTC -06:00
Mode:	hexadecimal
Character set:	CP 1252
Offsets:	decimal
Bytes per page:	36x16=576
	n/a

An interesting aspect of the MFT is that if the contents of a file are sufficiently small (less than about 750 bytes), the operating system doesn't *really* create a file at all. Instead, it stores the contents *right in the MFT* and just *pretends* to create a discrete file. Because the me.txt file holds so little content, we can see its content ("Your full Name") stored right in the MFT entry (beginning FILE0).

The MFT also stores date and time values reflecting when the file was Modified, Accessed and Created. You can't see them because they are encoded in an extraordinary way. Windows file times are stored as (and I absolutely LOVE this) a value equal to the number of 100 nanosecond intervals since January 1, 1601. Thus, if you look at the hex content from the MFT entry, the sixth line down begins with these eight bytes: 0x2897C459B451CE01. This is a 64-bit numeric value equivalent to the decimal 130,131,274,282,342,184. It also happens to *equal the number of 100 nanosecond intervals between January 1, 1601 and May 15, 2013 @21:37:08 UTC, when I created the file.*



```
   Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15           CP 1252
03349234688  46 49 4C 45 30 00 03 00  7D 9D 45 88 27 00 00 00   FILE0   } E▌'
03349234704  42 00 01 00 38 00 01 00  58 01 00 00 00 04 00 00   B   8   X
03349234720  00 00 00 00 00 00 00 00  06 00 00 00 51 E8 01 00               Qè
03349234736  02 00 00 00 00 00 00 00  10 00 00 00 60 00 00 00               `
03349234752  00 00 00 00 00 00 00 00  48 00 00 00 18 00 00 00           H
03349234768  28 97 C4 59 B4 51 CE 01  20 4D D0 63 B4 51 CE 01   (▌ÄY´QÎ  MÐc´QÎ
03349234784                           28 97 C4 59 B4 51 CE 01    MÐc´QÎ (▌ÄY´QÎ
03349234800                           00 00 00 00 00 00 00 00
03349234816                           00 00 00 00 00 00 00 00         ▌
03349234832                           30 00 00 00 68 00 00 00   HÅ▌À    0    h
03349234848                           4E 00 00 00 18 00 01 00           N
03349234864  08 02 00 00 00 00 03 00  28 97 C4 59 B4 51 CE 01           (▌ÄY´QÎ
03349234880  28 97 C4 59 B4 51 CE 01  28 97 C4 59 B4 51 CE 01   (▌ÄY´QÎ (▌ÄY´QÎ
```

Data Interpreter

64 Bit (±): 130131274282342184
FILETIME: 05/15/2013
21:37:08

NOTE: You've nothing you need to turn in for this Exercise.

Discussion Points to Ponder (NOT homework):

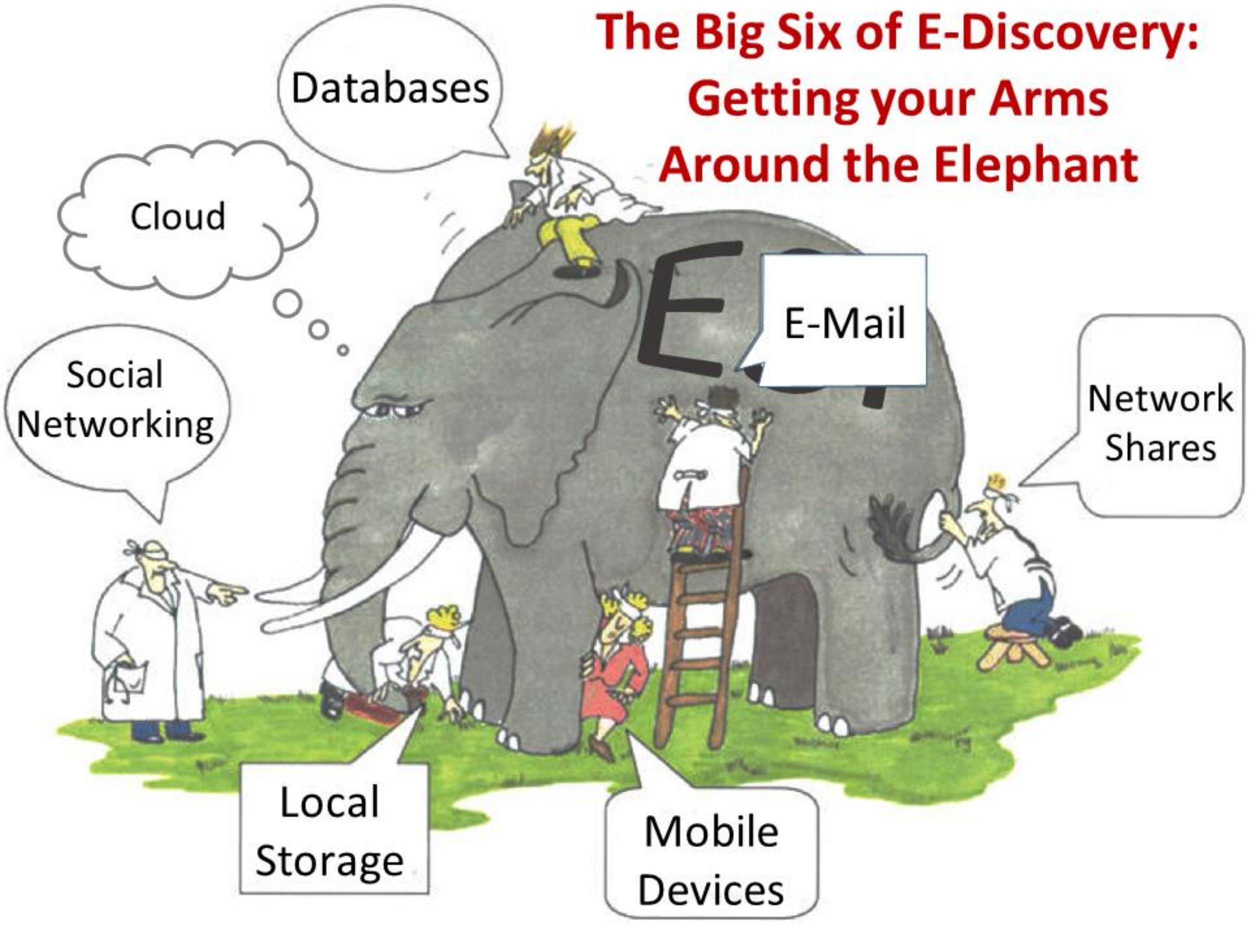
1. If the contents of the file me.txt reside in the MFT, why does Properties state that the file is taking up 4KB of space on disk?
2. When we copy a file from one media to another (as might occur when collecting ESI for processing), what MFT metadata routinely follows the file to its destination? Why these fields? What metadata is lost unless overt steps are taken to collect it? Does the destination medium have its own Master File Table? What data is lost when the file tables of the source and target media are incompatible?

The Big Six: Getting your Arms around the ESI Elephant

Many cultures share the parable of the six blind men describing an elephant. The one who grabbed the tail likened the elephant to a snake. The blind man who grabbed the trunk said, “no, more like a tree branch,” and the one with his arms around the elephant’s leg said, “you’re both wrong, an elephant is like a tree trunk.” The man touching the ear opined that the elephant was like a large leaf, and the blind man at the tusk said, “you’re all crazy. An elephant is like a spear.” None of them understood the true nature of an elephant because they failed to consider all its aspects.

VITAL VOCABULARY
Microsoft Exchange
Journaling Server
Media Rotation
Key Custodian
Structured Data
Query Language
Schema

In e-discovery, too, we cannot grasp the true nature of potentially responsive data until we touch many parts of the ESI elephant.



There are no forms or checklists that can take the place of understanding electronic evidence any more than a Polish phrasebook will equip you to try a case in Gdańsk. But there are a few rules of thumb that, *applied thoughtfully*, will help you get your arms around the ESI elephant. Let’s start with the Big Six and work through some geek speak as we go.

The Big Six…Plus
Without knowing anything about IT systems, you can safely assume there are at least six principal sources of digital evidence that may yield responsive ESI:

1. Key Custodians' E-Mail (Sources: server, local, archived and cloud)

Corporate computer users will have a complement of e-mail under one or more **e-mail aliases** (i.e., shorthand addresses) stored on one or more **e-mail servers**. These servers may be physical hardware managed by IT staff or **virtual machines** leased from a **cloud provider,** either running mail server software, most likely applications called **Microsoft Exchange** or **Lotus Domino.** A third potential source is a **Software as a Service (SaaS)** offering from a cloud provider, an increasingly common and important source**.** Webmail may be as simple as a single user's Gmail account or, like the Microsoft Office 365 product, a complete replication of an enterprise e-mail environment, sometimes supporting e-discovery preservation and search capabilities.

Users also tend to have a different, but overlapping complement of e-mail stored on desktops, laptops and handheld devices they've regularly used. On desktops and laptops, e-mail is found **locally** (on the user's hard drive) in **container files** with the file extensions **.pst** and **.ost** for Microsoft Outlook users or **.nsf** for Lotus Notes users. Finally, each user may be expected to have a substantial volume of **archived e-mail** spread across several on- and offline sources, including backup tapes, **journaling servers** and local archives on workstations and in network storage areas called **shares** (discussed below).

These locations are the "*where*" of e-mail, and it's crucial to promptly pin down "where" to ensure that your clients (or your opponents) don't overlook sources, especially any that may spontaneously disappear over time through **purges** (automatic deletion) or backup media **rotation** (reuse by overwriting).

Your goal here is to determine for each key custodian what they have in terms of:

- *Types of messages* (did they retain both Sent Items and Inbox contents? Have they retained messages as they were foldered by users?);
- *Temporal range of messages* (what are the earliest dates of e-mail messages, and are there significant gaps?); and
- *Volume* (numbers of messages and attachments versus total gigabyte volume—not the same thing).

Now, you're fleshing out the essential *"who, what, when, where and how"* of ESI.

2. Key Custodians' Documents and Data: Network Shares

Apart from e-mail, custodians generate most work product in the form of **productivity documents** like Microsoft Word documents, Excel spreadsheets, PowerPoint presentations and the like. These may be stored locally, *i.e.*, in a folder on the C: or D: drive of the user's computer (local storage, see below). More often, corporate custodians store work product in an area reserved to them on a network **file server** and **mapped** to a drive letter on the user's local machine. The user sees a lettered drive indistinguishable from a local drive, except that all data resides on the server, where it can be regularly backed up. This is called the user's **network share** or **file share**.

Just as users have file shares, work groups and departments often have network storage areas that are literally "shared" among multiple users depending upon the access privileges granted to them by the network administrator. These shared areas are, at once, everyone's data and no one's data because it's common for custodians to overlook **group shares** when asked to identify their data repositories. Still, these areas must be assessed and, as potentially relevant, preserved, searched and produced. Group shares may be **hosted** on company servers or “in the cloud," which is to say, in storage space of uncertain geographic location, leased from a service provider and accessed via the Internet. Enterprises employ virtual workspaces called **deal rooms** or **work rooms** where users "meet" and collaborate in cyberspace. Deal rooms have their own storage areas and other features, including message boards and communications tools--they’re like Facebook for business.

3. Mobile Devices: Phones, Tablets, IoT

Look around you in any airport, queue, elevator and waiting room or on any street corner. Chances are many of the people you see are looking at the screen of a mobile device. According to the U.S. Center for Disease Control, more than 41% of American households have no landline phone, relying on wireless service alone. For those between the ages of 25 and 29, two-thirds are wireless-only. Per an IDC report sponsored by Facebook, four out of five people start using their smartphones within 15 minutes of waking up and, for most, it’s the very first thing they do, ahead of brushing their teeth or answering nature’s call.

The Google Play app store holds 2,1 million apps and the Apple App Store supplies roughly the same number accounting for some 200 billion downloads. All these apps push, pull or store some data, and many of them surely contain data relevant to litigation. More people access the internet via phones than all other devices combined. Yet, in e-discovery, litigants often turn a blind eye to the content of mobile devices, sometimes rationalizing that whatever is on the phone or tablet must be replicated somewhere else. It’s no; and if you’re going to make such a claim, you’d best be prepared to back it up with solid metrics (such as by comparing data residing on mobile devices against data secured from other sources routinely collected and processed in e-discovery).

The bottom line is: if you’re not including the data on phones and tablets, you’re surely missing relevant, unique and often highly probative information.

4. Key Custodians' Documents and Data: Local Storage

Enterprises employ network shares to ensure that work product is backed up on a regular basis; but, despite a company’s best efforts to shepherd custodial work product into network shares, users remain bound and determined to store data on local, physical media, including local laptop and desktop hard drives, external hard drives, thumb drives, optical disks, camera media and the like. In turn, custodians employ idiosyncratic organizational schemes or abdicate organization altogether, making their My Documents folder a huge hodgepodge of every document they've ever created or collected.

Though it's expedient to assume that no unique, potentially-responsive information resides in local storage, it's rarely a sensible or defensible assumption absent document efforts to establish that the no-local-storage policy and the local storage reality are one-and-the-same.

5. Social Networking Content

The average Facebook user visits the site 14 times daily and spends 40 minutes looking at Facebook content. That's the average; so, if you haven't visited today, some poor soul must give Facebook 80 minutes and 28 visits. Perhaps because we believe we are sharing with "friends" or simply because nothing is private anymore, social networking content is replete with astonishingly candid photos, confessions, rants, hate speech, statements against interest and a host of other information that is evidence in the right case. Experts often blog or tweet. Spouses stray on dating and hook up sites like Tinder and Hinge. Corporations receive kudos and complaints via a variety of social portals. If you aren't asking about social networking content, you're missing a lot of elephant!

6. Databases (server, local and cloud)

From Access databases on desktop machines to enterprise databases running multinational operations (think UPS or Amazon.com), databases of every stripe are embedded throughout every company. Other databases are leased or subscribed to from third parties via the cloud (think Salesforce.com or Westlaw). Databases hold so-called **structured data**, a largely meaningless distinction when one considers that most of data stored within databases is unstructured, and much of what we deem unstructured data, like e-mail, is housed in databases. The key is recognizing that databases exist and must be interrogated to obtain the responsive information they hold.

The initial goal for e-discovery is to identify the databases and learn what they do, who uses them and what types and ranges of data they hold. Then, determine what standard reports they can generate in what formats. If standard reports aren't sufficient to meet the needs in discovery, inquire into the databases **schema** (*i.e.,* its structure) and determine what **query language** the database supports to explore how data can be extracted.

PLUS. Cloud Sources

The Big Six probably deserve to be termed the Big Seven by the meteoric rise of the cloud as both a repository for replicated content and a burgeoning source of relevant and unique ESI in its own right. For now, it's Six Plus because the Cloud touches so many of the other six and because it's evolving so quickly that it's likely to ultimately differentiate into several distinct sources of unique, discoverable ESI. Whether we consider the shift of corporate applications and IT infrastructure to leased cloud environments like Amazon Web Services and Microsoft Azure or the tendency of individuals to store data in tools like Box, Dropbox, Google Drive, Microsoft OneDrive, Apple's iCloud, Slack, Salesforce and others, the cloud is more than merely an adjunct to the Big Six sources when seeking to identify and preserve potentially responsive ESI.

As well, scanned paper records made searchable by Optical Character Recognition (OCR) tools remain a not-to-be-overlooked source of discoverable evidence.

The Big Six Plus don't cover the full range of ESI, but they encompass *most* potentially responsive data in *most* cases. A few more thoughts worth nailing to your forehead:

Pitfalls and Sinkholes

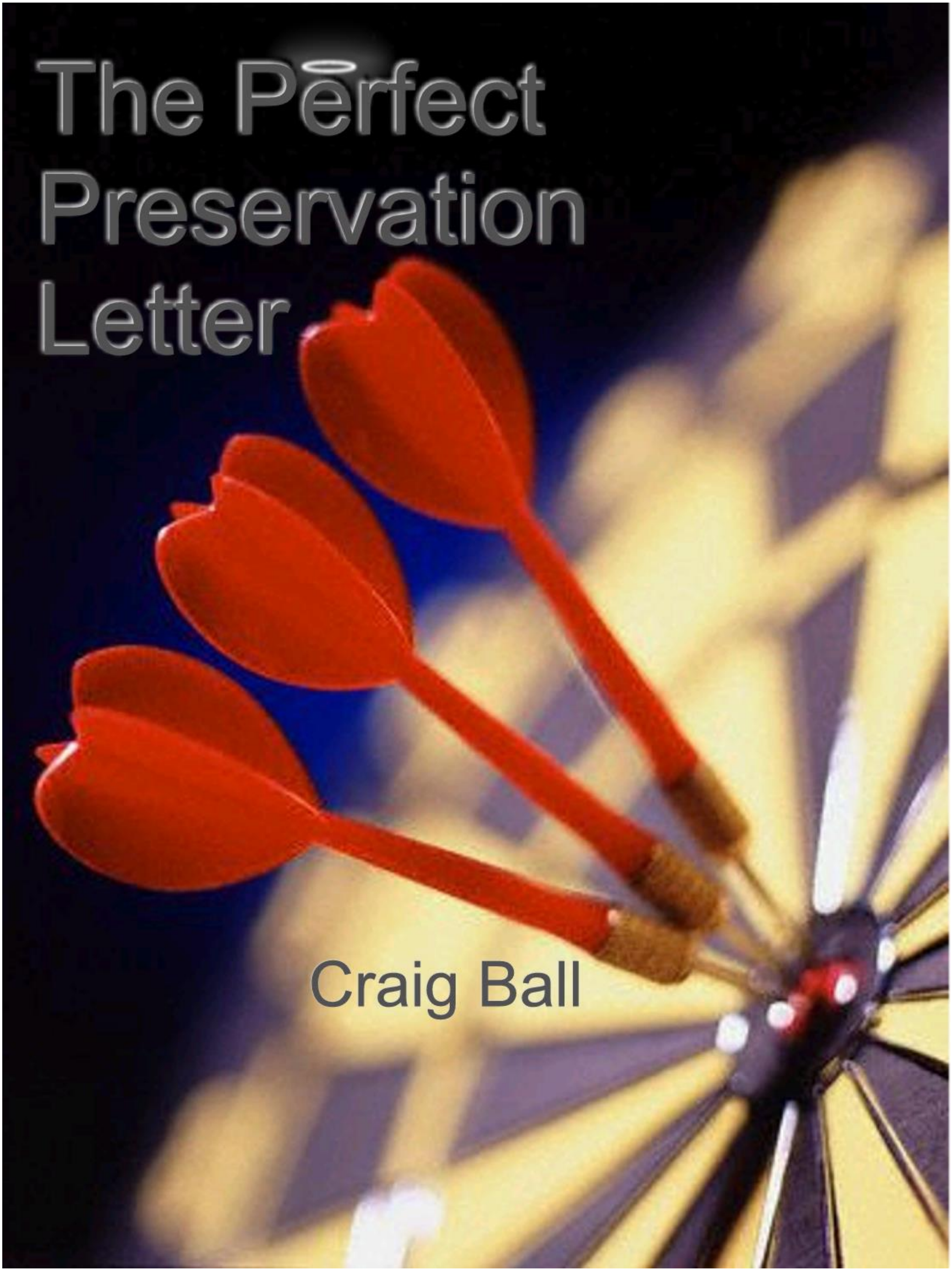
Few organizations preserve all legacy data (information no longer needed in day-to-day operations); however, most retain large swaths of legacy data in backups, archives and mothballed systems. Though a party isn't obliged to electronically search or produce all its potentially responsive legacy data when to do so would entail undue burden or cost, courts nonetheless tend to require parties resisting discovery to ascertain what they have and quantify and prove the burden and cost to search and produce it. This is an area where litigants often fail.

A second pitfall is that lawyers too willingly accept "it's gone" when a little wheedling and tenacity would reveal that the information exists and is not even particularly hard to access. It's an area where lawyers must be vigilant because litigation is regarded as a sinkhole by most everyone except the lawyers. Where ESI is concerned, custodians and system administrators assume too much, do too little or simply say whatever will make the lawyers go away.

Lather, Rinse and Repeat

So long as potentially responsive data is properly preserved, it's not necessary or desirable in a high-volume ESI case to seek to secure all potentially relevant data in a single e-discovery foray. It's more effective to divide and conquer. First, collect, examine and produce the most relevant and accessible ESI from what I like to call the über-key custodians; then, use that information to guide subsequent discovery requests. Research from the NIST TREC Legal Track proves that a two-tiered e-discovery effort produces markedly better results when the parties use the information gleaned from the first tier to inform their efforts through the second.

The Perfect Preservation Letter



Craig Ball

The Perfect Preservation Letter
Craig Ball

Well, I was drunk the day my Mom got outta prison, and I went to pick her up in the rain;
But before I could get to the station in my pickup truck, she got runned over by a damned old train.

From *"You Never Even Called Me by My Name" (a/k/a "The Perfect Country and Western Song")*
By Steve Goodman, performed by David Allan Coe

Outlaw musician David Allan Coe sings of how no country and western song can be "perfect" unless it talks of Mama, trains, trucks, prison and getting drunk. Likewise, no digital evidence preservation letter can be "perfect" unless it clearly identifies the materials requiring protection, educates your opponent about preservation options and lays out the consequences of failing to preserve the evidence. *You won't find the perfect preservation letter in any formbook.* You must custom craft it from a judicious mix of clear, technically astute terminology and *fact-specific* direction. It compels broad retention while asking for no more than the essentials. It rings with reasonableness. Its demands are *proportionate to the needs of the case,* and it keeps the focus of e-discovery where it belongs: *on relevance*. This article discusses features of an effective, efficient preservation letter and offers suggestions as to how it can be drafted and deployed.

VITAL VOCABULARY
Preservation Letter
Proportionate
Anticipation of Litigation
Sanctions
Intent to Deprive
System Metadata
Application Metadata
Backup Tape
Computer Forensics
Forensic Image
Bitstream

THE ROLE OF THE PRESERVATION LETTER

You can read the Federal Rules of Civil Procedure from cover to cover and you'll find no mention of preservation letters. So why invest effort creating the perfect preservation letter? Doesn't every lawyer know the law and rules prohibiting destruction of evidence apply to electronically stored information just like any other evidence? Don't all litigators ensure clients take reasonable steps to preserve information in anticipation of litigation and discovery? Fifteen years after amendment of the Federal Rules on these points and countless published decisions post-*Zubulake v. UBS Warburg*, 220 F.R.D. 212 (S.D.N.Y. 2003), the answer remains a sad, resounding "NO." *You cannot rely upon the competence and training of opposing counsel when it comes to electronic evidence.* Too many litigators and in-house counsel remain clueless and careless about information systems. The reality of electronic discovery is it starts off as the responsibility of those who don't understand the technology and ends up as the responsibility of those who don't understand the law. A well-drafted preservation letter helps bridge this knowledge gap.

> The reality of electronic discovery is it starts off as the responsibility of those who don't understand the technology and ends up as the responsibility of those who don't understand the law.

At bottom, the preservation letter reminds parties to preserve evidence, *to act,* so evidence doesn’t disappear. But the preservation letter also serves as the linchpin of claims for spoliation, helping establish the requisite intent to deprive and conscious disregard for the duty to preserve. The more plainly and practically you convey what evidence must be retained, the greater your client’s access to justice when an opponent loses or destroys it.

The more plainly and practically you convey what evidence must be retained, the greater your client’s access to justice when an opponent loses or destroys it.

THE RULES OF CIVIL PROCEDURE

Though serving a preservation letter isn’t a formal component of civil discovery procedures, it’s a wise precursor to the obligations imposed by the federal, state and local rules of procedure imposing discovery “meet and confer” obligations. Rule 26 of the Federal Rules of Civil Procedure requires litigants “discuss any issues about preserving discoverable information, *Fed. R. Civ. P. Rule 26*, and “any issues about disclosure, discovery, or preservation of electronically stored information, including the form or forms in which it should be produced.˝ *Fed. R. Civ. P. Rule 26(f)(3)*. By compelling early consideration of the nature and scope of potentially relevant evidence, often before litigation has begun, the preservation letter serves to frame the agenda for conferences to follow.

The preservation letter plays a key role in a court’s consideration of whether a party acted in bad faith in connection with the irreparable loss of data that should have been preserved. Rule 37(e) of the Federal Rules of Civil Procedure states:

> **Failure to Preserve Electronically Stored Information.**
> If electronically stored information that should have been preserved in the anticipation or conduct of litigation is lost because a party failed to take reasonable steps to preserve it, and it cannot be restored or replaced through additional discovery, the court:
> (1) upon finding prejudice to another party from loss of the information, may order measures no greater than necessary to cure the prejudice; or
> (2) only upon finding that the party acted with the intent to deprive another party of the information’s use in the litigation may:
> (A) presume that the lost information was unfavorable to the party;
> (B) instruct the jury that it may or must presume the information was unfavorable to the party; or
> (C) dismiss the action or enter a default judgment.

Assessment of intent turns on the subjective awareness of the party failing to preserve evidence. The preservation letter helps establish such awareness, proving a party destroying evidence knew of its discoverability and purposefully disregarded it. A clear and instructive preservation letter that serves to educate your opponent isn't just a professional courtesy; it compels recognition of the duty to intervene to prevent data loss and makes it harder to assert ignorance as a defense.

A clear and instructive preservation letter that serves to educate your opponent isn't just a professional courtesy; it compels recognition of the duty to intervene to prevent data loss and makes it harder to assert ignorance as a defense.

What is Electronic Evidence Preservation?

When evidence was on paper, preserving it was simple: We set the original or a copy aside, confident that it would come out of storage as it went in. Absent destructive forces or tampering, paper stays the same. But despite lawyers' archaic ardor for paper, modern information is born *digitally* and stored *digitally*. Little of it is ever printed save for short-term convenience and then discarded.

Preserving electronically stored information (ESI) poses unique challenges because:

- Touching ESI changes it
- Digital evidence is ill-suited to printing
- ESI must be interpreted to be used
- Storage media are fragile and dynamic, changing all the time
- Digital storage media are disposable and recyclable

TOUCHING ESI CHANGES IT

Route a document through a dozen hands and, aside from a little finger grime or the odd coffee stain, the document won't be changed by moving, copying or reading it. But, open the same document in Microsoft Word, or copy it to a thumb drive, and you've irretrievably changed the document's *system metadata*, the data-about-data metrics, like a document's creation date, that may be evidence in its own right. Open the document in its native application (e.g., Microsoft Word) and embedded *application metadata* values are irreparably altered.

Even the medium employed to copy or transmit data may play a role in altering its metadata. Back when it was common to use recordable optical disks to transfer or produce ESI, few appreciated that merely copying a file from a Windows computer to a recordable CD-R stripped the file of time values. Hard drives, floppy disks, thumb drives and optical media all use different file system architecture such that the CD-R doesn't offer a structure capable of storing all Windows time metadata. Where the Windows NTFS file system offers three "slots" for storing file dates (i.e., Modified, Accessed and Created), the CD-R's Joliet file structure supplies just one. With nowhere to go, temporal metadata is jettisoned in the CD recording process, and the missing metadata misreported on the destination system. Similar incongruities may impact the ability to store long filenames as well as the precision of time values. When ESI is evidence, such differences matter.

DIGITAL EVIDENCE IS ILL-SUITED TO PRINTING

Much modern evidence doesn't lend itself to paper. For example, a spreadsheet displays values derived from embedded formulae, but you can't embed those formulae in paper and see the calculated values. In large databases, information occupies expansive grids that wouldn't fit on a printed page. Sound and video evidence can't make the leap to paper and allocating a full sheet of paper to every text message is insanely wasteful and cumbersome. So, preserving on paper has ceased to be a practical option.

ESI MUST BE INTERPRETED TO BE USED

If legible and in a familiar language, a paper document conveys information directly to the reader. A literate person can interpret an alphabet, aided by blank spaces and a few punctuation marks. It's a part of our grade school "programming." All digital data are just streaming information denoted as ones and zeroes. For these streams of data to convey anything intelligible to humans, the data must be interpreted by a computer using specialized programming called "interfaces" and "applications." Without the right interface and application—sometimes even without the correct *version* of an interface or application—data is wholly inaccessible or may be inaccurately presented. Successfully preserving data may entail preserving legacy applications capable of correctly interpreting the data as well as legacy computing environments—hardware and software—capable of running the applications. Operator's manuals and the schema laying out a database's architecture may be needed as well.

STORAGE MEDIA ARE FRAGILE AND DYNAMIC

If your great grandfather put a letter in a folder a century ago, chances are good that apart from signs of age, you could pull it out today and read it. But changes in storage technology and instant obsolescence have already rendered fifteen-year-old digital media largely inaccessible absent considerable effort and expense. How many of us still have a computer capable of reading an optical disk, let alone a floppy disk? Data stored on back up tapes and other magnetic and optical media fades and disappears over time like the contents of once-common thermal fax paper. Disks expected to last a century are turning up illegible in a few years. Back up tapes stretch a bit each time they are used and are sensitive to poor storage conditions. Long-term data preservation will entail either the emergence of re durable media or a relentless effort to migrate and re-migrate legacy data to new media.

DIGITAL STORAGE MEDIA ARE DISPOSABLE AND RECYCLABLE

By and large, paper is not recycled for information storage; at least not in a way we'd confuse its prior use as someone's Last Will & Testament with its reincarnation as a cardboard carton. By contrast, a hard drive is constantly changing and recycling its contents. A later version of a document may overwrite—and by so doing, destroy—an earlier draft, and storage space released by the deletion of one file may well be re-used for storage of another. This is in sharp contrast to paper preservation, where you can save a revised printout of a document without affecting—and certainly not obliterating-- a prior printed version.

Clearly, successful preservation of digital data isn't as simple as copying something and sticking it in a folder; but your opponent may not appreciate the planning and effort digital preservation requires. When that's the case, the requesting party is at a crossroads: Do you seek to educate the producing party or its counsel about how and why to properly preserve digital evidence, or do you keep mum in hopes that an advantage will flow from your opponent's ineptitude? That is, do you want the evidence or the sanction?

Setting an opponent up for a spoliation sanction is a fool's errand; most of the time, you'll want the evidence.

THE DUTY TO PRESERVE

At what point does the duty to preserve evidence arise? When the lawsuit is filed? Upon receipt of a preservation letter? When served with a request for production?

The duty to preserve evidence may arise before—and certainly arises without—a preservation letter. In fact, the duty can arise long before. A party's obligation to preserve evidence is generally held to arise when the party knows or has reason to know that evidence may be relevant to future litigation. This "reasonable anticipation of litigation" standard means that any person or company who should see a claim or lawsuit on the horizon must act, *even before a preservation letter or lawsuit has materialized*, to cease activities likely to destroy electronic or tangible evidence and must take affirmative steps to preserve such evidence.

The duty to preserve evidence may arise before—and certainly arises without—a preservation letter.

Thus, the preservation letter is but one of several events sufficient to trigger the duty to preserve evidence, *but the preservation letter is an explicit, decisive trigger.* Often, the preservation letter's arrival marks the moment parties awaken to their duty to determine what evidence exists and what must be retained.

Often, the preservation letter's arrival marks the moment parties awaken to their duty to determine what evidence exists and what must be retained.

BALANCE, REASONABLENESS AND PROPORTIONALITY

I've seen producing parties sneer in contempt at preservation letters when they should consider them a gift. A well-crafted preservation demand is well-nigh a checklist of sources and forms of potentially relevant ESI. Does it too-often overreach? Certainly, because most are drafted by lawyers knowing little-or-nothing about an opponent's information systems. Apprehension and ignorance foster everything-but-the-kitchen-sink requests; the perfect preservation letter esteems the "how" and "how much" issues faced by the other side.

A preservation letter seeking everything and a pony or serving to paralyze an opponent's operations won't see compliance or enforcement. Absent evidence of misconduct (e.g., overt destruction of evidence), a court won't sanction a party for failing to comply with a preservation letter so onerous that no one dare turn on their computer for fear of spoliation! For a preservation letter to work, it must be reasonable on its face.

A preservation letter seeking everything and a pony or serving to paralyze an opponent's operations won't see compliance or enforcement.

Take Note: If your goal is to keep the other side from destroying relevant evidence, any judge will support you in that effort *if your demands aren't cryptic, overbroad or unduly burdensome*. In a word: *proportionate.*

If it could be accomplished with paper evidence, judges expect a corollary accomplishment with electronic evidence. Still, digital is different, and some of the ways we approach paper discovery just won't fly for electronic evidence. For example, using the term "any and all" in a request for digital evidence is a red flag for potential over breadth. Demanding that an opponent retain "any and all electronic communications" is nonsense. After all, phone conversations are electronic communications, and it's unlikely that, outside a regulated environment like a retail brokerage, a court would require a litigant to record all calls, though a judge shouldn't hesitate to compel *retention* of recordings (think Zoom meetings) *when conferences are already recorded and relevant.* If what you want preserved is e-mail, or text messaging or social networking content, *spell it out.* Your opponent may squawk, but at least the battle lines will be drawn on specific evidentiary items your opponent may destroy instead of fighting about vague language" The risk to this approach is that your opponent may fail to preserve what you haven't specified. Fear not! To the extent the evidence destroyed was relevant and material, an omnibus request to retain information items bearing on the claims made the basis of the claim will catch it.

Remember: the preservation letter neither creates the duty to preserve *nor constrains it*. Parties must still think for themselves. If the evidence was relevant and discoverable, its intentional destruction is spoliation, even if you didn't cite it in your preservation demand.

Remember: the preservation letter neither creates the duty to preserve nor constrains it. Parties must still think for themselves. If the evidence was relevant and discoverable, its intentional destruction is spoliation, even if you didn't cite it in your preservation demand.

PRESERVATION ESSENTIALS

First and foremost, a perfect preservation letter must seek to halt routine business practices geared to the destruction of potential evidence. It might call for an end to automatic purging of messages, repurposing of drives, overwriting of logs, scheduled destruction of back up media, sale, gift or destruction of computer systems and, (especially if you know computer forensics may come into play) running "privacy" software.. A lot of digital evidence disappears because of a lack of communication ("legal forgot to tell IT") or of individual initiative ("this is MY e-mail and I can delete it if I want to"). So, be sure to highlight the need to effectively communicate retention obligations to those with hands-on access to systems and suggest steps to forestall personal delete-o-thons. **Remember:** When you insist that communications about preservation obligations ***reach*** every custodian of discoverable data and that such communications stress the importance of the duty to preserve, you are demanding no more than the law requires. See, *e.g., Zubulake, supra.*

Next, get fact specific! Focus on items specifically bearing on the claim or suit, like relevant business units, activities, practices, procedures, time intervals, jurisdictions, facilities and key players (a/k/a "custodians"). Here, follow the "who, what, when, where and how" credo of good journalism. Preservation letters are more than a boilerplate form into which you pack every synonym for document and computer. If your preservation letter boils down to "save everything about anything by everyone, everywhere at any time," it's time to re-draft it because not only will no trial court enforce it, many will see it as discovery abuse.

The preservation letter's leading role is to educate your opponent about the many forms of relevant electronic evidence and the importance of taking prompt, affirmative steps to be sure that evidence remains accessible. Educating the other side isn't a noble undertaking—it's sound strategy. Spoliation is frequently defended on the basis of ignorance; *e.g.,* "Your honor, we had no idea that we needed to do that," and your goal is to slam the door on the "it was an oversight" excuse. Doing so entails more than just reciting a litany of storage media to be preserved--you've got to educate, clearly and concisely.

Don't be so focused on electronic evidence that you fail to direct your opponent to retain the old-fashioned paper variety. Finally, remember that *turnabout is fair play*. Don't expect to hold your opponent to a standard of preservation your client won't meet. Your opponent may face a greater *burden* to preserve a larger volume or variety of relevant information, but *their duty to preserve is no greater than yours*.

Don't expect to hold your opponent to a standard of preservation your client won't meet.

THE NATURE OF THE CASE

As documentary discovery typically follows service of a complaint, parties know what a dispute is about by the time the first request arrives. But a pre-suit preservation letter may be your opponent's first inkling they face litigation. Don't assume those receiving your preservation letter know what the dispute is about: *spell it out for them*. Supply sufficient information about the claim to allow a reasonable person reading the preservation letter to understand what evidence may be relevant. Names of key players, dates, business units, office locations, causes of action and events will all be weighed in deciding what's relevant and must be retained. The more you elucidate, the less likely you are to hear, "*If you wanted Madison's text messages, why didn't you mention Madison in the preservation letter?*"

WHEN TO SEND A PRESERVATION LETTER

The conventional wisdom is that preservation letters should go out as soon as you can identify potential defendants. But there may be compelling reasons to delay sending a preservation letter.

For example, when you face opponents who won't hesitate to destroy evidence, a preservation letter is just the starting gun and blueprint for a delete-o-thon. Instead, consider seeking a temporary restraining order or appointment of a discovery master (but recognize that the Comments to the proposed Rules amendments strongly discourage entry of *ex parte* preservation orders). Deferring the letter may be wise when your investigation is ongoing, and the service of a preservation letter will cause the other side to hire a lawyer or trigger work product privileges running from the anticipation of litigation. There may even be circumstances where you ***want*** your opponent's routine, good faith destruction of information to continue, such as where information unfavorable to your position will be lost in the usual course of business.

WHO GETS THE LETTER?

If counsel hasn't appeared for your opponent, to whom should you direct your perfect preservation letter? Here, the best advice is erring on the side of as many appropriate persons as possible. Certainly, if an individual will be the target of the action, he or she should receive the preservation letter. However, if you know of others who may hold potential evidence (such as a spouse, accountant, employer, banker, customers and business associates), it's smart to serve a *tailored* preservation demand on them, making clear that you are seeking preservation of physical and electronic records in their possession pertaining to the matters made the basis of the contemplated action. Some litigants use the preservation letter to put pressure customers lost to or solicited by a competitor-defendant. ***Beware such tactics!*** The preservation letter isn't a discovery mechanism expressly countenanced by the rules of procedure, so its misuse as an instrument of intimidation may not be privileged and could provoke a counterclaim based of libel or tortuous interference.

If the other side is a corporation, a directive to the wrong person may be ignored or be late in reaching those capable of putting a litigation holds in place. Consequently, if no counsel has appeared, it's wise to direct preservation letters to several within the organization, including, *inter alia*, the Chief Executive Officer, General Counsel, Director of Information Technologies and perhaps even the Head of Corporate Security and registered agent for service of process. You may want to copy other departments, facilities or business units.

Consider who is most likely to *unwittingly* destroy evidence and be certain that person receives a preservation letter. Sending a preservation letter to a person likely to destroy evidence *intentionally* is a different story. The letter may operate as the triggering event to spoliation, so you may need to balance the desire to give notice against the potential for irretrievable destruction.

Of course, preservation letters, like any important notice, should be dispatched in a way enabling you to prove receipt, even if that means via certified mail, return receipt requested.

HOW MANY PRESERVATION LETTERS?

Turning to the obligatory litigation-as-war metaphor, is a preservation letter best delivered as a single giant salvo across the opponent's bow, or might it instead be more effectively launched as several targeted blows? It's common to dispatch a single, comprehensive request, but might it instead be wiser to present your demands in a *series* of focused requests, broken out by, *e.g.*, type of digital medium, issues, business units, or the roles of key players? Your preservation letter may be destined to be an exhibit to a motion, so when the time comes to seek sanctions for a failure to preserve evidence, wouldn't it be more compelling to direct the court to a lean, specific preservation notice than a bloated beast? Consider supplementing a "master" preservation notice with specific notices directed at key players as the matter proceeds. It's difficult to claim, "*We didn't realize you wanted Elizabeth's Facebook content*" when Elizabeth got her very own, custom-tailored preservation letter.

SPECIFYING FORM OF PRESERVATION

The Federal Rules of Civil Procedure permit a requesting party to specify the form or forms in which the requesting party wants electronic evidence produced. Often, there's no additional trouble or expense for the producing party to generate one format over another and there may be occasions where a non-native production format is preferred, such as when evidence must be redacted to remove privileged content. *But should the preservation letter specify the form in which the data should be preserved?* Generally, not. Your preservation letter should not demand preservation in forms other than those used in the ordinary course of business. However, when your specification operates to *ease* the cost or burden to the producing party or otherwise *help* the producing party fulfill its preservation obligation, an alternate format might be *suggested*.

Your preservation letter should not demand preservation in forms other than those used in the ordinary course of business.

SPECIAL CASES: BACK UP TAPES, COMPUTER FORENSICS AND METADATA

The e-discovery wars rage in the mountains of e-mail and flatlands of Excel spreadsheets, but nowhere is the battle so pitched as at the front lines and flanks called *back up tapes, computer forensics and metadata*. These account for much of the bloodshed and so deserve special consideration in a preservation letter.

BACK UP TAPES

In the "capture the flag" e-discovery conflicts waged years ago, waged, the primary objective was often your opponent's server backup tapes or, more particularly, forcing their retention and restoration. Backup systems have but one legitimate purpose, being the retention of data required to get a business information system "back up" on its feet in the event of disaster. To this end, a business need retain disaster recovery data for a brief interval since there are few instances where a business would wish to re-populate its information systems with stale data. Because only the latest data has much utility in a well-designed backup system, the tapes containing the oldest backed-up information are typically recycled. This practice is "tape rotation," and the interval between use and reuse of a tape or set of tapes is the "rotation cycle" or "rotation interval."

Ideally, the contents of a backup system would be entirely cumulative of the active "online" data on the servers, workstations, laptops and other devices that make up a network. But, because businesses entrust the power to destroy data to every computer user--including those motivated to make evidence disappear--backup tapes are often the only evidence containers beyond the reach of those with the incentive to destroy or fabricate evidence. Going way back to Col. Oliver

North's deletion of e-mail subject to subpoena in the 1980's Iran-Contra affair, it's long been the backup systems that ride to truth's rescue with "smoking gun" evidence.

Another reason backup tape lay at the epicenter of early e-discovery disputes was that many organizations used to retain back up tapes long after they lost their usefulness for disaster recovery. When data has been deleted from the active systems, the stale backup tapes are a means by which the missing pieces of the evidentiary puzzle can be restored.

In organizations with many servers, backup systems are complex, hydra-headed colossi. There may be no simple one-to-one correspondence between a server and a user, and most tape backup systems structure stored data differently from active data on the server, complicating restoration and exploration. Volume, complexity and the greater time it takes to access tape compared to disk all contribute to the potentially high cost of targeting backup tapes in discovery. Compelling a large organization to interrupt its tape rotation, set aside back up tapes and purchase a fresh set can carry a princely price tag, but if the tapes aren't preserved, deleted data may be gone forever. That's been the Hobson's choice[48] of e-discovery.

A preservation letter should target just the backup media likely to contain deleted data relevant to the issues in the case—a feat easier said than done. Whether by Internet research, contact with former employees or consultation with other lawyers who've plowed the same ground, seek to learn all you can about the architecture of the active and backup systems. The insight gleaned from such an effort may allow for a more narrowly tailored preservation request or justify a much broader one.

The responding party need not preserve evidence that is merely cumulative, so once established that data has not been deleted and all relevant information still exists on the servers, the backup tapes should be released to rotation. Again, this is harder than it sounds because it requires three elements often absent from the adversarial process: ***communication, cooperation and trust*****.** Hopefully, the adoption of compulsory meet-and-confer sessions in state courts will force litigants to focus on e-discovery issues sufficiently early to stem unnecessary costs by narrowing the breadth of preservation efforts to just those actions or items most likely to yield discoverable data.

DRIVE IMAGING

Data deleted from a personal computer isn't gone. On electromagnetic ("spinning") hard drives, the operating system simply releases the space the deleted data occupies for reuse and treats the space as available for reuse. Deletion rarely erases data. In fact, there are three and *only* three ways that information's destroyed on personal computer:

There are three and *only* three ways that information's destroyed on a personal computer

[48] Thomas Hobson was a British stable keeper in the mid-1600s whose policy was that you either took the horse nearest the stable door or he wouldn't rent you a horse. "Hobson's choice" has come to mean an illusory alternative. Back up tapes are problematic, but the unacceptable alternative is letting evidence disappear.

1. Completely overwriting the deleted data on magnetic media (*e.g.,* floppy disks, tapes or conventional hard drives) with new information.
2. Strongly encrypting the data and then "losing" the encryption key; or,
3. Physically damaging the media to such an extent that it cannot be read.

Computer forensics is the science that, *inter alia,* resurrects deleted data. Because operating systems turn a blind eye to deleted data (or at least that which has gone beyond the realm of the Recycle Bin), a copy of a drive made by ordinary processes won't retrieve the deleted data. Computer forensic scientists use specialized tools and techniques to copy every sector on a drive, including those holding deleted information. When the stream of data containing each bit on the media (the so-called "bitstream") is duplicated to a sequence of files, it's called a "drive image" or "forensic image." Computer forensic tools analyze and extract data from images.

In routine computer operation, deleted data is overwritten by random re-use of the space it occupies or by system maintenance activities; consequently, the ability to resurrect deleted data with computer forensics erodes over time. *When the potential for discovery from deleted files on personal computers is an issue*, a preservation letter may specify that the computers on which the deleted data reside should be removed from service and shut down or imaged in a forensically sound manner. Such a directive might read:

> **Act to Prevent Spoliation**
> You should take affirmative steps to prevent anyone with access to your data, systems, accounts and archives from seeking to modify, destroy or hide potentially relevant ESI wherever it resides (such as by deleting or overwriting files, using data shredding and erasure applications, re-imaging, damaging or replacing media, encryption, compression, steganography or the like).
>
> **System Sequestration or Forensically Sound Imaging** **[*When Implicated*]**
> As an appropriate and cost-effective means of preservation, you should remove from service and securely sequester the systems, media, and devices housing potentially relevant ESI of the following persons:
>
> **[NAME KEY PLAYERS MOST DIRECTLY INVOLVED IN CAUSE]**
>
> In the event you deem it impractical to sequester systems, media and devices, we believe that the breadth of preservation required, coupled with the modest number of systems implicated, dictates that forensically sound imaging of the systems, media and devices of those named above is expedient and cost effective. As we anticipate the need for forensic examination of one or more of the systems and the presence of relevant evidence in forensically significant areas of the media, we demand that you employ forensically sound ESI preservation methods. Failure to use such methods poses a significant threat of spoliation and data loss.
>
> "Forensically sound ESI preservation" means duplication of all data stored on the

evidence media while employing a proper chain of custody and using tools and methods that make no changes to the evidence and support authentication of the duplicate as a true and complete bit- for-bit image of the original. The products of forensically sound duplication are called, *inter alia,* "bitstream images" of the evidence media. A forensically sound preservation method guards against changes to metadata evidence and preserves all parts of the electronic evidence, including deleted evidence within "unallocated clusters" and "slack space."

Be advised that a conventional copy or backup of a hard drive does not produce a forensically sound image because it captures only active data files and fails to preserve forensically significant data existing in, e.g., unallocated clusters and slack space.

Further Preservation by Imaging

With respect to the hard drive, thumb drives, phones, tablets and storage devices of each of the persons named below and of each person acting in the capacity or holding the job title named below, demand is made that you immediately obtain, authenticate and preserve forensically sound images of the storage media in any computer system (including portable and personal computers, phones and tablets) used by that person during the period from _______ 20___ to _______, 20___, as well as recording and preserving the system time and date of each such computer.

[NAMES, JOB DESCRIPTIONS OR JOB TITLES]

Once obtained, each such forensically sound image should be labeled to identify the date of acquisition, the person or entity acquiring the image and the system and medium from which it was obtained. Each such image should be preserved without alteration and authenticated by hash value.

METADATA

Metadata, the "data about data" created by computer operating systems and applications, may be critical evidence in your case, and its preservation requires prompt and decisive action. Information stored and transmitted electronically is tracked by the system where it resides and by the applications that create and use it.

For example, a Microsoft Word document is comprised of information you can see (*e.g.,* the text of the document and the data revealed when you look at the document's "Properties" in the File menu), as well as information you don't always see like tracked changes, collaborative comments, revision histories and other data the program only displays on request). This *application* metadata is stored within the document file and moves with the file when it is copied or transmitted. Likewise, the computer system on which the document resides keeps a record of when the file was created, accessed and modified, as well as the size, name and location of the file. This *system* metadata is *not* stored within the document. So, when a file is copied or transmitted—as when it's uploaded or copied to thumb drive for production—potentially relevant and discoverable system metadata is lost or changed. Absent proper steps to protect metadata, it's constantly at peril of loss or alteration.

Metadata is not a crucial evidence in all matters, but it's always enormously important to culling and managing electronic evidence, and to assessing integrity and authenticity. Metadata proves when a document or record was created, altered, copied or deleted. If you reasonably anticipate that metadata will be important—and that's so often the case—you should specifically direct the other side to preserve relevant metadata evidence and warn of the risks threatening its loss and corruption. Because most lawyers have a spotty appreciation of the variety and utility of system and application metadata, the perfect preservation letter defines metadata and informs your opponent where to find it, the actions that damage it and, if possible, the mechanisms by which it should be preserved. *It pays to be specific.* Although specificity is challenging when we know nothing about an opponent's ESI usage, for most of the information deployed in discovery (e.g., e-mail, texts, documents, spreadsheets and presentations), we *CAN* anticipate the metadata of the most common forms and applications. For example, if you know you will need, say, the *Message ID* and *In-Reply-To* metadata fields to thread e-mail, demand that those fields be preserved.

For further information about metadata, see "*Beyond Data about Data: the Litigators Guide to Metadata,*" *infra* and at http://www.craigball.com/metadata.pdf.

DOES IT REALLY MAKE A DIFFERENCE?

Are you prepared to let relevant evidence disappear without a fight? ***No!***
Can the perfect preservation letter really make *that* much difference? ***Yes!***

The preservation letter demands your best effort for a host of reasons. It's the basis of your opponent's first impression of you and your case. A well-drafted preservation letter speaks volumes about your savvy, focus and preparation. A poorly drafted, scattergun missive suggests a lazy formbook attorney who's given little thought to where the case is going or what evidence is required. A letter that demonstrates close attention to detail and preemptively slams the door on cost-shifting and "innocent" spoliation bespeaks a force to be reckoned with. The artful preservation letter serves as a blueprint for meet and confer sessions and a touchstone for efforts to remedy destruction of evidence.

Strategically, the preservation letter forces your opponent to weigh potential costs and business disruption early, often before a lawsuit. If it triggers a litigation hold, everyone from the board room to the mail room may learn of the claim and be obliged to take immediate action. It may serve as the starting gun for a reckless rush to destroy evidence or trigger a move toward amicable resolution. But done right, ***the one thing it won't be is ignored*****.**

APPENDIX: EXEMPLAR PRESERVATION DEMAND TO OPPONENT

What follows *isn't* the perfect preservation letter for *your unique* case, so ***don't deploy it as a form***. Instead, use it as a ***drafting aid*** to flag issues unique to relevant electronic evidence, and tailor your preservation demand proportionately, *scaled to the unique issues, parties, and systems in your case.*

(Download as a Word document **here**)

Demand for Preservation of Electronically Stored Information and Other Evidence

I write as counsel for [Plaintiff(s)] [Defendant(s)] to advise you of [a claim for damages and other relief against you] growing out of the following matters (hereinafter this "cause"):

[DESCRIPTION OF MATTER, INCLUDING ACTORS, EVENTS, DATES, LOCATIONS, CLAIMS/DEFENSES]

We demand that you preserve documents, tangible things, and electronically stored information potentially relevant to the issues and defenses in this cause. As used in this document, "you" and "your" refers to **[NAME OF OPPONENT]**, and its predecessors, successors, parents, subsidiaries, divisions and affiliates, officers, directors, agents, attorneys, accountants, employees, partners Assigns and other persons occupying similar positions or performing similar functions.

You must anticipate that information responsive to discovery resides on your current and former computer systems, phones and tablets, in online repositories and on other storage media and sources (including voice- and video recording systems, Cloud services and social networking accounts).

Electronically stored information (hereinafter "ESI") should be afforded the broadest possible meaning and includes (*by way of example and not as an exclusive list*) potentially relevant information electronically, magnetically, optically, or otherwise stored as and on:

- **Digital communications** (*e.g.*, e-mail, voice mail, text messaging, WhatsApp, SIM cards)
- **E-Mail Servers** (*e.g.,* Microsoft 365, Gmail, and Microsoft Exchange databases)
- **Word processed documents** (*e.g.,* Microsoft Word, Apple Pages or Google Docs files/drafts)
- **Spreadsheets and tables** (*e.g.,* Microsoft Excel, Google Sheets, Apple Numbers)
- **Presentations** (*e.g.,* Microsoft PowerPoint, Apple Keynote, Prezi)
- **Social Networking Sites** (*e.g.,* Facebook, Twitter, Instagram, LinkedIn, Reddit, Slack, TikTok)
- **Online ("Cloud") Repositories** (*e.g.,* Drive, OneDrive, Box, Dropbox, AWS, Azure)
- **Databases** (*e.g.,* Access, Oracle, SQL Server data, SAP)
- **Backup and Archival Files** (*e.g.,* Veritas, Zip, Acronis, Carbonite)
- **Contact and Customer Relationship Management Data** (*e.g.,* Salesforce, Outlook, MS Dynamics)
- **Online Banking, Credit Card, Retail and other Relevant Account Records**
- **Accounting Application Data** (*e.g.,* QuickBooks, NetSuite, Sage)
- **Image and Facsimile Files** (*e.g.*, .PDF, .TIFF, .PNG, .JPG, .GIF., HEIC images)
- **Sound Recordings** (*e.g.,* .WAV and .MP3 files)
- **Video and Animation** (*e.g.,* Security camera footage, .AVI, .MOV, .MP4 files)
- **Calendar, Journaling and Diary Application Data** (*e.g.,* Outlook PST, Google Calendar, blog posts)

- **Project Management Application Data**
- **Internet of Things (IoT) Devices and Apps** *(e.g.*, Amazon Echo/Alexa, Google Home, Fitbit)
- **Computer Aided Design/Drawing Files**
- **Online Access Data** (e.g., Temporary Internet Files, Web cache, Google History, Cookies)
- **Network Access and Server Activity Logs**

ESI resides not only in areas of electronic, magnetic, and optical storage media reasonably accessible to you, but also in areas you may deem *not* reasonably accessible. You are obliged to *preserve* potentially relevant evidence from *both* sources of ESI, even if you do not anticipate *producing* such ESI or intend to claim it is confidential or privileged from disclosure.

The demand that you preserve both accessible and inaccessible ESI is reasonable and necessary. Pursuant to the rules of civil procedure, you must identify all sources of ESI you decline to produce and demonstrate to the court why such sources are not reasonably accessible. For good cause shown, the court may order production of the ESI, even if it is not reasonably accessible. Accordingly, you must preserve ESI that you deem inaccessible so as not to preempt the court's authority.

Preservation Requires Immediate Intervention

You must act immediately to preserve potentially relevant ESI, including, without limitation, information with the *earlier* of a Created or Last Modified date on or after [**DATE**] through the date of this demand and continuing thereafter, concerning:

1. The events and causes of action described [**above**] **[in the Complaint] [in the Answer]**
2. ESI you may use to support claims or defenses in this case
3. ….

Adequate preservation of ESI requires more than simply refraining from efforts to delete, destroy or dispose of such evidence. You must intervene to prevent loss due to routine operations or active deletion by employing proper techniques and protocols to preserve ESI. *Many routine activities serve to irretrievably alter evidence and constitute unlawful spoliation of evidence.*

Preservation requires action.

Nothing in this demand for preservation of ESI should be read to limit or diminish your concurrent common law and statutory obligations to preserve documents, tangible things and other potentially relevant evidence.

Suspension of Routine Destruction

You are directed to immediately initiate a litigation hold for potentially relevant ESI, documents and tangible things and to act diligently and in good faith to secure and audit compliance with such litigation hold. You are further directed to immediately identify and modify or suspend features of your information systems and devices that, in routine operation, operate to cause the loss of potentially relevant ESI. Examples of such features and operations may include:

- Purging the contents of e-mail and messaging repositories by age, quota, or other criteria
- Using data or media wiping, disposal, erasure or encryption utilities or devices
- Overwriting, erasing, destroying, or discarding backup media
- Re-assigning, re-imaging or disposing of systems, servers, devices or media
- Running "cleaner" or other programs effecting wholesale metadata alteration
- Releasing or purging online storage repositories or non-renewal of online accounts
- Using metadata stripper utilities
- Disabling server, packet, or local instant messaging logging
- Executing drive or file defragmentation, encryption, or compression programs

Guard Against Deletion

You should anticipate the potential that your officers, employees, or others may seek to hide, destroy or alter ESI. You must act to prevent and guard against such actions. Especially where company machines were used for Internet access or personal communications, you should anticipate that users may seek to delete or destroy information they regard as personal, confidential, incriminating or embarrassing, and in so doing, they may also delete or destroy potentially relevant ESI. This concern is not unique to you. It's simply conduct that occurs with such regularity that any custodian of ESI and their counsel must anticipate and guard against its occurrence.

Preservation of Backup Media

You are directed to preserve complete backup media sets (including differentials and incremental backups) that may contain unique communications and ESI of the following custodians for all dates during the below-listed intervals:

[CUSTODIAN] [INTERVAL*, e.g.,* 1/1/20 through 7/15/20]

Act to Prevent Spoliation

You should take affirmative steps to prevent anyone with access to your data, systems, accounts and archives from seeking to modify, destroy or hide potentially relevant ESI wherever it resides (such as by deleting or overwriting files, using data shredding and erasure applications, re-imaging, damaging or replacing media, encryption, compression, steganography or the like).

System Sequestration or Forensically Sound Imaging [When Implicated]

As an appropriate and cost-effective means of preservation, you should remove from service and securely sequester the systems, media, and devices housing potentially relevant ESI of the following persons:

[NAME KEY PLAYERS MOST DIRECTLY INVOLVED IN CAUSE]

In the event you deem it impractical to sequester systems, media and devices, we believe that the breadth of preservation required, coupled with the modest number of systems implicated, dictates that forensically sound imaging of the systems, media and devices of those named above is expedient and cost effective. As we anticipate the need for forensic examination of one or more of the systems and the presence of relevant evidence in forensically significant areas of the media, we demand that you employ forensically sound ESI preservation methods. Failure to use such methods poses a significant threat of spoliation and data loss.

“Forensically sound ESI preservation” means duplication of all data stored on the evidence media while employing a proper chain of custody and using tools and methods that make no changes to the evidence and support authentication of the duplicate as a true and complete bit-for-bit image of the original. The products of forensically sound duplication are called, *inter alia,* “bitstream images” of the evidence media. A forensically sound preservation method guards against changes to metadata evidence and preserves all parts of the electronic evidence, including deleted evidence within “unallocated clusters” and “slack space.”

Be advised that a conventional copy or backup of a hard drive does not produce a forensically sound image because it captures only active data files and fails to preserve forensically significant data existing in, e.g., unallocated clusters and slack space.

Further Preservation by Imaging

With respect to the hard drive, thumb drives, phones, tablets and storage devices of each of the persons named below and of each person acting in the capacity or holding the job title named below, demand is made that you immediately obtain, authenticate and preserve forensically sound images of the storage media in any computer system (including portable and personal computers, phones and tablets) used by that person during the period from _______ 20___ to _______, 20___, as well as recording and preserving the system time and date of each such computer.

[NAMES, JOB DESCRIPTIONS OR JOB TITLES]

Once obtained, each such forensically sound image should be labeled to identify the date of acquisition, the person or entity acquiring the image and the system and medium from which it was obtained. Each such image should be preserved without alteration and authenticated by hash value.

Preservation in Native Forms

You should anticipate that ESI, including but not limited to e-mail, documents, spreadsheets, presentations, and databases, will be sought in the form or forms in which it is ordinarily maintained (*i.e.,* native form). Accordingly, you should preserve ESI in such native forms, and you should not employ methods to preserve ESI that remove or degrade the ability to search the ESI by electronic means or that make it difficult or burdensome to access or use the information.

You should additionally refrain from actions that shift ESI from reasonably accessible media and forms to less accessible media and forms if the effect of such actions is to make such ESI not reasonably accessible.

Metadata

You should anticipate the need to disclose and produce system and application metadata and act to preserve it. System metadata is information describing the history and characteristics of other ESI. This information is typically associated with tracking or managing an electronic file and often includes data reflecting a file's name, size, custodian, location and dates of creation and last modification. Application metadata is information automatically included or embedded in electronic files, but which may not be apparent to a user, including deleted content, draft language, commentary, tracked changes, speaker notes, collaboration and distribution data and dates of creation and printing. For electronic mail, metadata includes all header routing data and Base 64 encoded attachment data, in addition to the To, From, Subject, Received Date, CC and BCC header fields.

Metadata may be overwritten or corrupted by careless handling or improper preservation, including by carelessly copying, forwarding, or opening files.

Servers

With respect to servers used to manage e-mail (*e.g.,* Microsoft 365, Microsoft Exchange, Lotus Domino) and network storage (often called a "network share"), the complete contents of each relevant custodian's network share and e-mail account should be preserved. There are several cost-effective ways to preserve the contents of a server without disrupting operations. If you are uncertain whether the preservation method you plan to employ is one that we will deem sufficient, please contact the undersigned.

Home Systems, Laptops, Phones, Tablets, Online Accounts, Messaging Accounts and Other ESI Sources

Though we expect that you will act swiftly to preserve data on office workstations and servers, you should also determine if any home or portable systems or devices may contain potentially relevant data. To the extent that you have sent or received potentially relevant e-mails or created or reviewed potentially relevant documents away from the office, you must preserve the contents of systems, devices and media used for these purposes (including not only potentially relevant data from portable and home computers, but also from external storage drives, thumb drives, CD- R/DVD-R disks and the user's phone, tablet, voice mailbox or other forms of ESI storage.). Similarly, if you used online or browser-based e-mail and messaging accounts or services (such as Gmail, Yahoo Mail, Microsoft 365, Apple Messaging, WhatsApp or the like) to send or receive potentially relevant messages and attachments, the contents of these account mailboxes and messages should be preserved.

Ancillary Preservation

You must preserve documents and other tangible items that may be required to access, interpret or search potentially relevant ESI, including manuals, schema, logs, control sheets, specifications, indices, naming protocols, file lists, network diagrams, flow charts, instruction sheets, data entry forms, abbreviation keys, user ID and password rosters and the like.

You must preserve passwords, keys and other authenticators required to access encrypted files or run applications, along with the installation disks, user manuals and license keys for applications required to access the ESI.

If needed to access or interpret media on which ESI is stored, you must also preserve cabling, drivers, and hardware. This includes tape drives, readers, DBMS other legacy or proprietary devices and mechanisms.

Paper Preservation of ESI is Inadequate
As hard copies do not preserve electronic searchability or metadata, they are not an adequate substitute for, or cumulative of, electronically stored versions. If information exists in both electronic and paper forms, you should preserve both forms.

Agents, Attorneys and Third Parties
Your preservation obligation extends beyond ESI in your care, possession or custody and includes ESI in the custody of others that is subject to your direction or control. Accordingly, you must notify any current or former agent, attorney, employee, custodian and contractor in possession of potentially relevant ESI to preserve such ESI to the full extent of your obligation to do so, and you must take reasonable steps to secure their compliance.

Preservation Protocols
We are desirous of working with you to agree upon an acceptable protocol for forensically sound preservation and can supply a suitable protocol if you will furnish an inventory and description of the systems and media to be preserved. Alternatively, if you promptly disclose the preservation protocol you intend to employ, we can identify any points of disagreement and resolve them. A successful and compliant ESI preservation effort requires expertise. If you do not currently have such expertise at your disposal, we urge you to engage the services of an expert in electronic evidence and computer forensics so that our experts may work cooperatively to secure a balance between evidence preservation and burden that's fair to both sides and acceptable to the court.

Do Not Delay Preservation
I'm available to discuss reasonable preservation steps; however, *you should not defer preservation steps pending such discussions if ESI may be lost or corrupted because of delay.* Should your failure to preserve potentially relevant evidence result in the corruption, loss, or delay in production of evidence to which we are entitled, such failure would constitute spoliation of evidence, and we will not hesitate to seek sanctions.

Confirmation of Compliance
Please confirm by [**DATE**], that you have taken the steps outlined in this letter to preserve ESI and tangible documents potentially relevant to this action. If you have not undertaken the steps outlined above, or have taken other actions, please describe what you have done to preserve potentially relevant evidence and what you will not do. Else we will rely upon you to complete the preservation sought herein.

❁ Exercise 11: Compiling a Checklist of Sources

GOALS: The goals of this exercise are for the student to:

1. Develop a checklist of potential data sources that are candidates for legal preservation;
2. Explore published ESI checklists; and
3. Identify sources missing from published checklists, improving organization and utility.

Next week, for Exercise 3, you will compile a data map of your own digital footprint. The key takeaways from the data mapping exercise are a deeper appreciation of the variety of sources documenting our digital lives and how many of these sources we overlook. Most students are surprised by how little they know about the native forms and data volumes of their own information, particularly that reposed in third-parties. As well, student data maps are idiosyncratic, rife with different units, source names and descriptors. In an enterprise setting, data maps for e-discovery must be consistent and complete for all custodians.

So, what might help you prepare to do a more complete, consistent job in identifying sources of evidence? ***How about a checklist of sources?***

There are loads of e-discovery checklists extant, geared to assist lawyers and litigation support personnel in the identification and preservation of discoverable information. Some are comprehensive; most are spotty. Older checklists tend to omit important new sources like Slack or Instagram. For example, former President Trump's lawyers "borrowed" the verbatim text of a preservation letter contained in an article I published a dozen years ago without updating it to include later-developed sources. The ironic upshot is that the Twitter President failed to include tweets in his preservation demand. (*See*, https://craigball.net/2018/01/05/the-sincerest-form-of-flattery/).

Assignment: You are to develop the best checklist of potential data sources that are candidates for legal preservation—good enough that you'll want to keep it to use in your own work. It does NOT have to be entirely original, and I encourage you to start with the best existing checklist(s) you can find and make improvements. Certainly, you should seek to add missing sources, but you may also see a better way to organize and present the content. Remember that *this checklist is one of sources not methods* and, crucially, *it should serve to enhance the completeness of the personal data map you will create*. Thus, generic checklist items like "Online sources" or "Local storage" won't be sufficiently granular.

You may consult any written and online resources, including Google, PACER filings, journal articles, blogs, Lexis or Westlaw and form books. You may also seek input and guidance from practicing attorneys, judges, professors, IT personnel, consultants, vendors or others; anyone or anything SO LONG AS you <u>do</u> <u>not</u> present the work of anyone else as your own without attribution. You are welcome to borrow liberally from print or online sources (including published forms); but <u>you must give full and proper attribution to such sources</u>. If you present much of someone else's work as your work product *without* proper attribution, I will consider your submission plagiarized.

This is not a test of your retyping skill; so, if the task requires a lot of retyping from a form, consider supplying the source form and separately (and clearly) specifying the changes and additions. That said, please don't include a bunch of redundant material. *I wouldn't expect this task to take much more than two hours done sparingly or three done well*. Clear, well-organized and thorough are important; "pretty" not so much.

The checklist should be useful for consideration of individuals and their personal data resources as well as for entities and their systems and databases. This exercise entails a mix of research and independent thought to ensure that the checklist you compile suffices for both.

Remember: Borrow as much as you like but credit your sources faithfully. Also, technology changes rapidly. If your source checklist is from ten years ago, what's missing from such an old roster of sources?

Luddite Lawyer's Guide to Computer Backup Systems

Backup is the Rodney Dangerfield of the e-discovery world. It gets no respect. Or, maybe it's Milton, the sad sack with the red stapler from the movie, *Office Space*. Backup is pretty much ignored...until headquarters burns to the ground or it turns out the old tapes in the basement hold the only copy of the all-important TPS reports demanded in discovery.

Would you be surprised to learn that backup is the hottest, fastest moving area of information technology? Consider the:

- Migration of data to the "cloud" *(Minsk! Why's our data in Minsk?);*
- Explosive growth in hard drive capacities *(Four terabytes! On a desktop?);*
- Ascendency of virtual machines *(Isn't that the title of the next Terminator movie?);* and
- Increased reliance on replication *(D2D2T? That's the cute Star Wars droid, right?).*

If you don't understand how backup systems work, you can't reliably assess whether discoverable data exists or how much it will cost in terms of sweat and coin to access, search and recover that data.

The Good and Bad of Backups

Ideally, the contents of a backup system would be entirely cumulative of the active "online" data on the servers, workstations and laptops that make up a network. But because businesses entrust the power to alter and destroy data to every computer user--including those motivated to make evidence disappear—and because companies configure systems to purge electronically stored information as part of records retention programs, backup tapes may prove to be the only source of evidence beyond the reach of those who've failed to preserve evidence and who have an incentive to destroy or fabricate it. Going back as far as 1986 and Col. Oliver North's deletion of e-mail subject to subpoena

<u>Vital Vocabulary</u>

Look for these key terms:

- *disaster recovery*
- *full backup*
- *differential backup*
- *incremental backup*
- *tape restoration*
- *tape rotation*
- *legacy tapes*
- *replication*
- *drive imaging*
- *bitstream*
- *backup set*
- *backup catalog*
- *tape log*
- *linear serpentine*
- *virtual tape library*
- *D2D2T*
- *RAID*
- *striping*
- *parity*
- *hash value*
- *single-instance storage*
- *non-native restoration*
- *Cloud backup*

in the Reagan-era Iran-Contra affair, it's long been backup systems that ride to truth's rescue with "smoking gun" evidence.

Backup tapes can also be fodder for pointless fishing expeditions mounted without regard for the cost and burden of turning to backup media, or targeted prematurely in discovery, before more accessible data sources have been exhausted.

Grappling with Backup Tapes

Backup tapes are made for ***disaster recovery***, i.e., picking up the pieces of a damaged or corrupted data storage system. Some call backups "snapshots" of data, and like a photo, backup tapes capture only what's in focus. To save time and space, backups typically ignore commercial software programs that can be reinstalled in the event of disaster, so ***full backups*** typically focus on all *user created* data. ***Incremental backups*** grab just what's been created or changed since the last full or incremental backup. Together, they put Humpty-Dumpty back together again in a process called ***tape restoration*.**

Tape is cheap, durable and portable, the last important because backups need to be stored away from the systems at risk. Tape is also slow and cumbersome, downsides discounted because it's so rarely needed for restoration.

Because backup systems have but one legitimate purpose--being the retention of data required to get a business information system "back up" on its feet after disaster--a business only needs recovery data covering a brief interval. No business wants to replicate its systems as they existed six months or even six weeks before a crash. Thus, *in theory*, older tapes are supposed to be recycled by overwriting them in a practice called ***tape rotation*.**

But, as theory and practice are rarely on speaking terms, companies may keep backup tapes long past (sometimes *years* past) their usefulness for disaster recovery and often beyond the IT department's ability to access tapes created with obsolete software or hardware. These ***legacy tapes*** are business records—sometimes the last surviving copy—but are afforded little in the way of *records management*. Even businesses that overwrite tapes every two weeks replace their tape sets from time to time as faster, bigger options hit the market. The old tapes are frequently set aside and forgotten in offsite storage or a box in the corner of the computer room.

Like the DeLorean in "Back to the Future," legacy tapes allow you to travel back in time. It doesn't take 1.2 million gigawatts of electricity, just lots of cabbage.

Duplication, Replication and Backup

We save data from loss or corruption via one of three broad measures: duplication, replication and backup.

Duplication is the most familiar--protecting the contents of a file by making a copy of the file to another location. If the copy is made to another location on the same medium (e.g., another folder on the hard drive), the risk of corruption or overwriting is reduced. If the copy is made to another medium (another hard drive), the risk of loss due to media failure is reduced. If the copy is made to a distant physical location, the risk of loss due to physical catastrophe is reduced.

You may be saying, “Wait a second. Isn’t backup just a form of duplication?” To some extent, it is; and certainly, duplication is the most common “backup” method used on a personal computer. But, true enterprise backup injects other distinctive elements, the foremost being that enterprise backups are not user-initiated but occur systematically, untied to the whims and preferences of individual users.

Replication is duplication without discretion. That is, the contents of one storage medium are periodically or continuously mirrored to another storage medium. Replication may be as simple as RAID 1 mirroring of two local hard drives (where one holds exactly the same data as the other) or as elaborate as keeping a distant data operations center on standby, ready to go into service in the event of a catastrophe.

Unlike duplication and replication, backup involves (reversible) alteration of the data and logging and cataloging of content. Typically, backup entails the use of software or hardware that compresses and encrypts data. Further, backup systems are designed to support iteration, e.g., they manage the scheduling and scope of backup, track the content and timing of backup “sets” and record the allocation of backup volumes across multiple devices or media.

Major Elements of Backup Systems

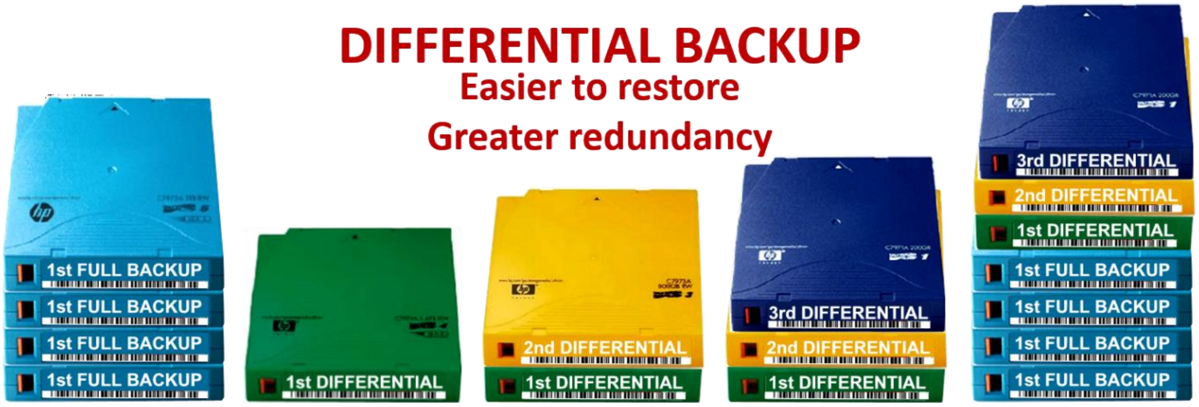
Understanding backups requires an appreciation of the three major elements of a backup system: the source data, the target data (“backup set”) and the catalog.

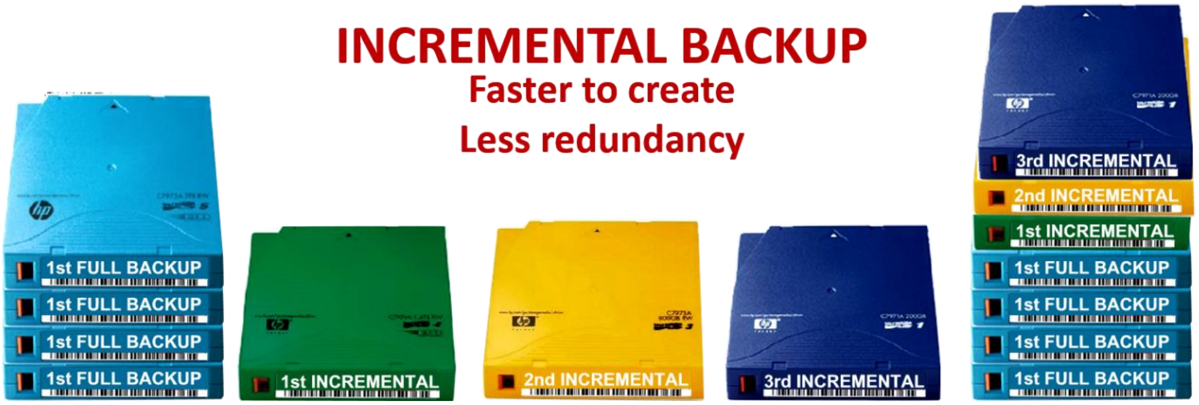
1. Source Data (Logical or Physical) Though users tend to think of the source data as a collection of files, backup may instead be drawn from the broader, logical divisions of a storage medium, called “partitions,” “volumes” and “folders.” ***Drive imaging,*** a specialized form of backup employed by IT specialists and computer forensic examiners, may draw from below the logical hierarchy of a drive, collecting a “bitstream” of the drive’s contents reflecting the contents of the medium at the physical level. The bitstream of the medium may be stored in a single large file, but more often it’s broken into manageable, like-sized “chunks” of data to facilitate more flexible storage.

2. Backup Set (Physical or Logical, Full or Changed-File) A ***backup set*** may refer to a *physical* collection of *media* housing backed up data, i.e., the collective group of magnetic tape cartridges required to hold the data, or the “set” may reference the *logical* grouping of *files* (and associated catalog) which collectively comprise the backed up data. Compare, "*those three LTO tape cartridges*" to "*the backup of the company's Microsoft Exchange Mail Server*."

Backup sets further divide between what can be termed "full backups" and "changed-file backups." As you might expect, full backups tend to copy everything present on the source (or at least "everything" that has been defined as a component of the full backup set) where changed-file backups duplicate items that have been added or altered since the last full backup.

The changed-file components further subdivide into **incremental backups**, **differential backups** and **delta block-level backups**. The first two identify changed files based on either the status of a file's archive bit or a file's created and modified date values. The essential difference is that every differential backup duplicates files added or changed since the last *full* backup, where incremental backups duplicate files added or changed since the last *incremental* backup. The delta block-level method examines the contents of a file and stores only the *difference*s between the version of the file contained in the full backup and the modified version. This approach is trickier, but it permits the creation of more compact backup sets and accelerates backup and restoration.





<u>3. Backup Catalog vs. Tape Log</u> Unlike duplication and replication, where generally no record is kept of the files moved or their characteristics, the creation and maintenance of a catalog is a key element of enterprise backup. The ***backup catalog*** tracks*, inter alia,* the source and metadata of each file or component of the backup set as well as the location of the element within the set. The catalog delineates the quantity of target media and identifies and sequences each tape or disk required for restoration. Without a catalog setting out the logical organization of the data as stored, it would be impossible to distinguish between files from different sources having the same names or to extract selected files without restoration of all of the backed up data.

Equally important is the catalog's role in facilitating single instance backup of identical files. Multiple computers—especially those within the same company—store many files with identical names, content and metadata. It's a waste of time and resources to backup multiple iterations of identical data, so the backup catalog makes it possible to store just a single instance of such files and employ placeholder "stubs" or pointers to track all locations to which the file should be restored.

Obviously, *lose* the catalog, and it's tough to put Humpty Dumpty back together again.

It's important to distinguish the catalog--a detailed digital record that, if printed, would run to hundreds of pages or more--from the ***tape log,*** which is typically a simple listing of backup events and dates, machines and tape identifier. *See*, *e.g.,* the sample page of a tape log attached as Appendix A.

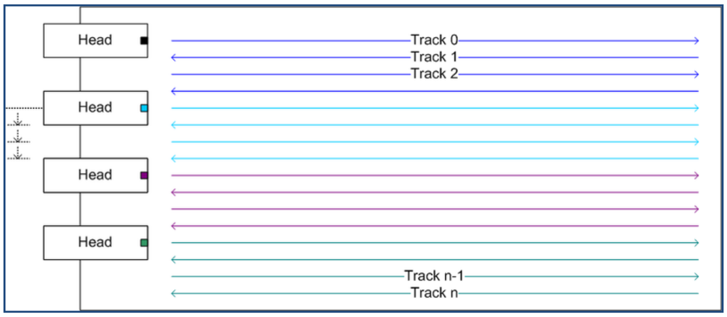
Backup Media: Tape and Disk-to-Disk

Tape Backup

Though backup tape seems almost antique, tape technology has adapted well to modern computing environments. The IBM 3420 reel-to-reel backup tapes that were a computer room staple in the 1970s and '80s employed 240 feet of half-inch tape on 10.5-inch reels. These tapes were divided into 9 tracks of data and held a then-impressive 100 megabytes of information traveling at 1.2 megabytes per second.

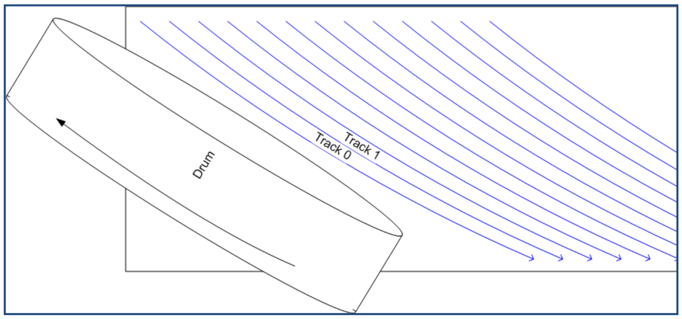
Today's LTO-8 tapes are housed in a 4-inch square LTO cartridge less than an inch thick and feature 3,150 feet of half-inch tape divided into 6,656 tracks holding 12 terabytes of information transferring at 300 megabytes per second.

That's 740 times as many tracks, 250 times faster data transfer and *100,000 times greater* data storage capability in a far smaller package.



Older readers may recall “auto-reverse” tape transport mechanisms, which eliminated the need to eject and turn over an audiocassette to play the other side. Many modern backup tapes use a scaled-up version of that back-and-forth or ***linear serpentine*** recording scheme. “Linear” because it stores data in parallel tracks running the length of the tape, and “serpentine” because its path snakes back-and-forth like a mountain road. Thirty-two of the LTO-8 cartridge’s 6,656 tracks are read or written as the tape moves past the heads, so it takes 208 *back-and-forth passes* or “wraps” to read or write the full contents of a single LTO-8 cartridge.

That’s about 124 *miles* of tape passing the heads!



An alternate recording scheme employed by SAIT-2 tape systems employs a ***helical recording*** system that writes data in parallel tracks running diagonally across the tape, much like a household VCR. Despite a slower transfer rate, helical recording also achieves 800GB of storage capacity on 755 feet of 8mm tape housed in a compact cartridge like that used in handheld video cameras. Development of SAIT tape technology was abandoned in 2006 and Sony stopped selling SAIT in 2010; so, they aren’t seen much beyond tape archives.

Why is Tape So Slow?

Clearly, tape is a pretty remarkable technology that’s seen great leaps in speed and capacity. The latest tapes on the market can reportedly outstrip the ability of a hard drive to handle their throughput.

Still, even the best legal minds have yet to find loopholes in those pesky laws of physics.

All that serpentine shuttling back and forth over 124 miles of tape is a mechanical process. It occurs at a glacial pace relative to the speed with which computer circuits move data.

Further, backup restoration is often an incremental process. Reconstructing reliable data sets may require data from multiple tapes to be combined. Add to the mix the fact that as hard drive capacities have exploded, tape must store more and more information to keep pace. Gains in performance are offset by growth in volume.

How Long to Restore?

Several years ago, the big Atlanta tape house, eMag Solutions, LLC, weighed in on the difference between the time it *should* take to restore a backup tape considering just its capacity and data transfer rate versus the time it *really* takes considering the following factors that impact restoration:

- Tape format;
- Device interface, i.e., SCSI or fiber channel;
- Compression;
- Device firmware;
- The number of devices sharing the bus;
- The operating system driver for the tape unit;
- Data block size (large blocks fast, small blocks slow);
- File size (with millions of small files, each must be cataloged);
- Processor power and adapter card bus speed;
- Tape condition (retries eat up time);
- Data structure (e.g., big database vs. brick level mailbox accounts);
- Backup methodology (striped data? multi server?).

The following table reflects eMag's reported experience:

Drive Type	**Native cartridge capacity**	**Drive Native Data Transfer Speed**67F[49]	**Theoretical Minimum Data Transfer Time**	**Typical Real World Data Transfer Time**
DLT7000	35GB	3MB/sec	3.25 Hrs	6.5 Hrs
DLT8000	40GB	3MB/sec	3.7 Hrs	7.4 Hrs
LTO1	100GB	15MB/sec	1.85 Hrs	4.0 Hrs
LTO2	200GB	35MB/sec	1.6 Hrs	6.0 Hrs
SDLT 220	110GB	11MB/sec	2.8 Hrs	6.0 Hrs
SDLT 320	160GB	16MB/sec	2.8 Hrs	6.0 Hrs

[49] " *How Long Does it Take to Restore a Tape*," eMag blog, 7/17/2009 at http://tinyurl.com/tapetime, Some of these transfer rate values are at variance with manufacturer's stated values, but they are reported here as published by eMag.

The upshot is that it takes *about twice as long* to restore a tape under real world conditions than the media's stated capacity and transfer rate alone would suggest. Just to generate a catalog for a tape, the tape must be read in its entirety. Consequently, it's not feasible to deliver 3,000 tapes to a vendor on Friday and expect a catalog to be generated by Monday. The *price* to do the work has dropped dramatically, but the *time* to do the work has not.

Extrapolating from this research, we can conceive a formula to estimate the real world time to restore a set of backup tapes of consistent drive type and capacity, and considering that, employing multiple tape drives, tapes may be restored simultaneously:

$$\textbf{Real World Transfer Time (in Hours)} = \frac{\textbf{Native Cartridge Capacity (in GB)}}{\textbf{1.8 x Drive Native Transfer Speed}}$$

Applying this to a LTO-7 tape:

$$\frac{\textbf{Native Cartridge Capacity (in GB)}}{\textbf{1.8 x Transfer Speed (in MB/s)}} = \frac{\textbf{6 TB}}{\textbf{1.8 x 300}} = \frac{\textbf{6,000}}{\textbf{540}} = \textbf{11.1 hours}$$

Of course, this is merely a rule-of-thumb for a single tape. As you seek to apply it to a large-scale data restoration, be sure to factor in other real world factors impacting speed, such as the ability to simultaneously use multiple drives for restoration, the need to swap tapes and replace target drives, to clean and align drive mechanisms, the working shifts of personnel, weekend and holidays, time needed for recordkeeping, for resolving issues with balky tapes and for steps taken in support of quality assurance.

Common Tape Formats

The LTO tape format is the clear winner of the tape format wars, having eclipsed all contenders save the disk and cloud storage options that now threaten to end tape's enduring status as the leading backup medium. As noted, the LTO-8 format natively holds 12.0 terabytes of data at a transfer rate of 360 megabytes per second. *These values are expected to continue to double roughly every two years through 2020,* with 24TB/60TB LTO9 expected out before the end of the year. Tape use is down, but not out—not for some time.

Too, the dusty catacombs beneath Iron Mountain still brim with all manner of legacy tape formats that will be drawn into e-discovery fights for years to come. Here are some of the more common formats seen in the last 30 years and their characteristics:

Name	Format	A/K/A	Length	Width	Capacity (GB)	Transfer Rate (MB/sec)
DLT 2000	DLT3	DLT	1200 ft	1/2”	10	1.25
DLT 2000 XT	DLT3XT	DLT	1828 ft	1/2”	15	1.25
DLT 4000	DLT 4	DLT	1828 ft	1/2”	20	1.5
DLT 7000	DLT 4	DLT	1828 ft	1/2”	35	5
DLT VS-80	DLT 4	TK-88	1828 ft	1/2”	40	3
DLT 8000	DLT 4	DLT	1828 ft	1/2”	40	6
DLT-1	DLT 4	TK-88	1828 ft	1/2”	40	3
DLT VS-160	DLT 4	TK-88	1828 ft	1/2”	80	8
SDLT-220	SDLT 1		1828 ft	1/2”	110	10
DLT V4	DLT 4	TK-88	1828 ft	1/2”	160	10
SDLT-320	SDLT 1		1828 ft	1/2”	160	16
SDLT 600	SDLT 2		2066 ft	1/2”	300	36
DLT-S4	DLT-S4	DLT Sage	2100 ft	1/2”	800	60
DDS-1	DDS-1	DAT	60M	4mm	1.3	.18
DDS-1	DDS-1	DAT	90M	4mm	2.0	.18
DDS-2	DDS-2	DAT	120M	4mm	4	.60
DDS-3	DDS-3	DAT	125M	4mm	12	1.1
DDS-4	DDS-4	DAT	150M	4mm	20	3
DDS-5	DAT72	DAT	170M	4mm	36	3
DDS-6	DAT160	DAT	150M	4mm	80	6.9
M1	AME	Mammoth	22M	8mm	2.5	3
M1	AME	Mammoth	125M	8mm	14	3
M1	AME	Mammoth	170M	8mm	20	3
M2	AME	Mammoth 2	75M	8mm	20	12
M2	AME	Mammoth 2	150M	8mm	40	12
M2	AME	Mammoth 2	225M	8mm	60	12
Redwood	SD3	Redwood	1200 ft	1/2”	10/25/50	11
TR-1		Travan	750 ft	8mm	.40	.25
TR-3		Travan	750 ft	8mm	1.6	.50
TR-4		Travan	740 ft	8mm	4	1.2
TR-5		Travan	740 ft	8mm	10	2.0
TR-7		Travan	750 ft	8mm	20	4.0

Name	Format	A/K/A	Length	Width	Capacity (GB)	Transfer Rate (MB/sec)
AIT 1	AIT		170M	8mm	25	3
AIT 1	AIT		230M	8mm	35	4
AIT 2	AIT		170M	8mm	36	6
AIT 2	AIT		230M	8mm	50	6
AIT 3	AIT		230M	8mm	100	12
AIT 4	AIT		246M	8mm	200	24
AIT 5	AIT		246M	8mm	400	24
Super AIT 1	AIT	SAIT-1	600M	8mm	500	30
Super AIT 2	AIT	SAIT-2	640M	8mm	800	45
3570 B	3570b	IBM Magstar MP		8mm	5	2.2
3570 C	3570c	IBM Magstar MP		8mm	5	7
3570 C	3570c XL	IBM Magstar MP		8mm	7	7
IBM3592	3592	3592	609m	1/2”	300	40
T9840A	Eagle		886 ft	1/2”	20	10
T9840B	Eagle		886 ft	1/2”	20	20
T9840C	Eagle		886 ft	1/2”	40	30
T9940A			2300 ft	1/2”	60	10
T9940B			2300 ft	1/2”	200	30
T10000	T10000	STK Titanium		1/2”	500	120
T10000B	T10000B			1/2”	1000	120
T10000C	T10000C			1/2”	5000	240
T10000D	T10000D			1/2”	8500	252
Ultrium	Ultrium	LTO 1	609M	1/2”	100	15
Ultrium	Ultrium	LTO 2	609M	1/2”	200	40
Ultrium	Ultrium	LTO 3	680M	1/2”	400	80
Ultrium	Ultrium	LTO 4	820M	1/2”	800	120
Ultrium	Ultrium	LTO 5	846M	1/2”	1,500	140
Ultrium	Ultrium	LTO 6	846M	1/2”	2,500	160
Ultrium	Ultrium	LTO 7	960M	1/2”	6,000	300
Ultrium	Ultrium	LTO 8	960M	1/2”	12,000	360
Ultrium	Ultrium	LTO9 (Q4 2020)	?	1/2”	24,000	708

Disk-to-Disk Backup

Tapes are stable, cheap and portable—a natural media for moving data in volumes too great to transmit by wire without consuming excessive bandwidth and disrupting network traffic. But strides in deduplication and compression technologies, joined by drops in hard drive costs and leaps in hard drive capacities, have eroded the advantages of tape-based transfer and storage.

When data sets are deduplicated to unique content and further trimmed by compression, much more data resides in much less drive space. With cheaper, bigger drives flooding the market, hard drive storage capacity has grown to the point that disk backup intervals are on par with the routine rotation intervals of tape systems (e.g., 8-16 weeks), Consequently, disk-to-disk backup options once considered too expensive or disruptive are feasible.

Hard disk arrays can now hold months of disaster recovery data at a cost that competes favorably with tape. Thus, tape is ceasing to be a disaster recovery medium and is instead being used solely for long-term data storage; that is, as a place to migrate disk backups for purposes *other than* disaster recovery, *i.e.*, archival.

Of course, the demise of tape backup has been confidently predicted for years, even while the demand for tape continued to grow. But for the first time, the demand curve for tape has begun to head south.

D2D (for Disk-to-Disk) backup made its appearance wearing the sheep's clothing of tape. In order to offer a simple segue from the 50-year dominance of tape, the first disk arrays were designed to emulate tape drives so that existing software and programmed backup routines needn't change. These are ***virtual tape libraries*** or *VTL*s.

As D2D supplants tape for backup, the need remains for a stable, cheap and portable medium for long-term retention of archival data--the stuff too old to be of value for disaster recovery but comprising the digital annals of the enterprise. This need continues to be met by tape, a practice that has given rise to a new acronym: ***D2D2T*,** for Disk-to-Disk-to-Tape. By design, tape now holds the company's archives, which ensures the continued relevance of tape backup systems to e-discovery.

Essential Technologies: Compression and Deduplication

Along with big, cheap hard drives and RAID redundancy, compression and deduplication have made cost-effective disk-to-disk backup possible. But compression and deduplication are important for tape, too, and bear further mention.

Compression

The design of backup systems is driven by considerations of speed and cost. Perhaps surprisingly, the speed and expense with which an essential system can be brought back online after failure is less critical than the speed and cost of each backup. The reason for this is that (hopefully) failure is a rare occurrence whereas backup is (or should be) frequent and routine. Certainly, no one would seriously contend that restoring a failed system from a morass of magnetic tape is the fastest, cheapest way to rebuild a failed system. No, the advantage of tape is its relatively low cost per gigabyte to store data, not to restore it.

Electrons move much faster than machines. The slowest parts of any backup systems are the mechanical components: the spinning reels, moving heads and the human beings loading and unloading tape transports. One way to maximize the cost advantage and efficiency of tape is to increase the density of data that can be stored per inch of tape. The more you can store per inch, the fewer tapes to be purchased and loaded and the fewer miles of tape to pass by the read-write heads.

Because electrons move speed-of-light faster than mechanical parts of backup systems, a lot of computing power can be devoted to restructuring data in ways that it fits more efficiently on tape or disk. For example, if a horizontal line on a page were composed of one hundred dashes, it takes up less space to describe the line as “100 dashes” or 100- than to actually type out 100 dashes. Of course, it would take some time to count the dashes, determine there were precisely 100 of them and ensure the shorthand reference “100 dashes” doesn’t conflict with some other part of the text; but, these tasks can be accomplished by digital processors in infinitely less time than that required to spin a reel of tape to store the difference between the data and its shorthand reference.

This is the logic behind data compression; that is, the use of computing power to re-express information in more compact ways to achieve higher transfer rates and consume less storage space. Compression is an essential, ubiquitous technology. Without it, there would be no YouTube, Netflix, streaming music and video, DVRs, HD digital cameras, Internet radio and much else that we prize in the digital age.

And without compression, you’d need a whole lot more time, tape and money to back up a computer system.

While compression schemes for files tend to comprise a fairly small number of published protocols (e.g., Zip, LZH), compression algorithms for backup have tended to be proprietary to the backup software or hardware implementing them and to change from version-to-version. Because of this, undertaking the restoration of legacy backup tapes entails more than simply finding a compatible tape drive and determining the order and contents of the tapes. You may also need particular software to decompress the data.

Deduplication

Companies that archive backup tapes may retain years of tapes, numbering in the hundreds or thousands. Because each full backup is a snapshot of a computer system at the time it's created, there is a substantial overlap between backups. An e-mail in a user's Sent Items mailbox may be there for months or years, so every backup replicates that e-mail, and restoration of every backup adds an identical copy to the material to be reviewed. Restoration of a year of monthly backups would generate 12 copies of the same message, thereby wasting reviewers' time, increasing cost and posing a risk of inconsistent treatment of identical evidence (as occurs when one reviewer flags a message as privileged, but another decides it's not). The level of duplication between ne backup to the next is often as high as 90%.

Consider, too, how many messages and attachments are dispatched to all employees or members of a product team. Across an enterprise, there's a staggering level of repetition.

Accordingly, an essential element of backup tape restoration is deduplication; that is, using computers to identify and cull identical electronically stored information before review. Deduplicating within a single custodian's mailboxes and documents is called ***vertical deduplication***, and it's a straightforward process. However, corporate backup tapes aren't geared to single users. Instead, business backup tapes hold messages and documents for multiple custodians storing identical messages and documents. Restoration of backup tapes generates duplicates within individual accounts (vertically) and across multiple users (horizontally). Deduplication of messages and documents across multiple custodians is called (not surprisingly) ***horizontal deduplication*****.**

Horizontal deduplication significantly reduces the volume of information to be reviewed and minimizes the potential for inconsistent characterization of identical items; however, it can make it impossible to get an accurate picture of an individual custodian's data collection because many constituent items may be absent, eliminated after being identified as identical to another user's items.

Consequently, deduplication plays two crucial roles when backup sets are used as a data source in e-discovery. First, deduplication must be deployed to eliminate the substantial repetition from one backup iteration to the next; that is, to eliminate that 90% overlap mentioned above. Second, deduplication is useful in reducing the cost and burden of review by eliminating vertical and horizontal repetition within and across custodians.

Modern backup systems are designed to deduplicate ESI *before* it's stored; that is, to eliminate all but a single instance of recurring content, hence the name, ***single-instance storage*****.** Using a method called *in-line deduplication,* a unique digital fingerprint or *hash value* is calculated for each file or data block as it's stored and that hash value is added to a list of stored files. Before being stored, each subsequent file or data block has its hash value checked against the list of stored files.

If an identical file has already been stored, the duplicate is not added to the backup media but, instead, a pointer or stub to the duplicate is created. An alternate approach, called *post-process deduplication*, works in a similarly, except that all files are first stored on the backup medium, then analyzed and selectively culled to eliminate duplicates.

Data Restoration

Clearly, data in a backup set is a bit like the furniture at Ikea: It's been taken apart and packed tight for transport and storage. But, when that data is needed for e-discovery--it must be reconstituted and reassembled. It starts to take up a lot of space again. That restored data has to go *somewhere,* usually t*o* a native computing environment just like the one from which it came.

But the system where it came from may be at capacity with new data or not in service anymore. Historically, small and mid-size companies lacked the idle computing capacity to effect restoration without a significant investment in equipment and storage. Larger enterprises devote more stand-by resources to recovery for disaster recovery and may have had alternate environments ready to receive restored data, but those resources had to be at the ready in the event of emergency. It was often unacceptably risky to dedicate them, even briefly, to electronic discovery.

The burden and cost of recreating a restoration platform for backup data was a major reason why backup media came to be emblematic of ESI deemed "not reasonably accessible." But while the inaccessibility presumption endures, newer technology has largely eliminated the need to recreate a native computing environment in order to restore backup tapes. Today, when a lawyer or judge opines that "backups are not reasonably accessible, *per se*," you can be sure they haven't looked at the options in several years.

Non-Native Restoration

A key enabler of low-cost access to tapes and other backup media has been the development of software tools and computing environments that support ***non-native restoration****.* Non-native restoration dispenses with the need to locate copies of particular backup software or to recreate the native computing environment from which the backup was obtained. It eliminates the time, cost and aggravation associated with trying to reconstruct a sometimes decades-old system. All major vendors of tape restoration services offer non-native restoration options, and it's even possible to purchase software facilitating in-house restoration of tape backups to non-native environments.

Perhaps the most important progress has been made in the ability of vendors both to generate comprehensive indices of tape contents and extract specific files or file types from backup sets. Consequently, it's often feasible for a vendor to, e.g., acquire just certain types of documents for particular custodians without the need to restore all data in a backup. In some situations, backups are simply not that much harder or costlier to deal with in e-discovery than active data, and they're occasionally the smarter *first* resort in e-discovery.

Going to the Tape *First*?

Perhaps due to the *Zubulake*68F[50] opinion or the commentary to the 2006 amendments to the Federal Rules of Civil Procedure,69F[51] e-discovery dogma is that backup tapes are the costly, burdensome recourse of last resort for ESI.

Pity. Sometimes backup tapes are the *easiest, most cost-effective* source of ESI.

For example, if the issue in the case turns on e-mail communications between Don and Elizabeth during the last week of June of 2007, but Don's no longer employed and Elizabeth doesn't keep all her messages, what are you going to do? If these were messages that should have been preserved, you could pursue a forensic examination of Elizabeth's computer (cost: $5,000-$10,000) or collect and search the server accounts and local mail stores of 50 other employees who might have been copied on the missing messages (cost: $25,000-$50,000).

Or, you could go to the backup set for the company's e-mail server from July 1 and recover just Don's or Elizabeth's mail stores (cost: $1,000-$2,500).

The conventional wisdom would be to fight any effort to go to the tapes, but the numbers show that, on the right facts, it's both faster and cheaper to do so.

Sampling

Sampling backup tapes entails selecting parts of the tape collection deemed most likely to yield responsive information and restoring and searching only those selections before deciding whether to restore more tapes. Sampling backup tapes is like drilling for oil: You identify the best prospects and drill exploratory wells. If you hit dry holes, you pack up and move on. But if a well starts producing, you keep on developing the field.

The size and distribution of the sample hinges on many variables, among them the breadth and organization of the tape collection, relevant dates, fact issues, business units and custodians, resources of the parties and the amount in controversy. Ideally, the parties can agree on a sample size or they can be encouraged to arrive at an agreement through a mediated process.

Because a single backup may span multiple tapes, and because recreation of a full backup may require the contents of one or more incremental or differential backup tapes, sampling of backup

[50] Zubulake v. UBS Warburg, 217 F.R.D. 309 (S.D.N.Y. 2003

[51] Fed R. Civ. P. 26(b)(2)(B).

tapes should be thought of as the selection of data snapshots at intervals rather than the selection of tapes. Sensible sampling necessitates access to and an understanding of the tape catalog. Understanding the catalog likely requires explanation of both the business system hardware (*e.g., What is the SQL Server's purpose?*) and the logical arrangement of data on the source machines *(e.g., What's stored in the Exchange Data folder?*). Parties should take pains to ensure that each sample is complete for a selected date or interval; that is, the number of tapes shouldn't be arbitrary but should fairly account for the totality of information captured in a single relevant backup event.

Backup and the Cloud

Nowhere is the observation that "*the Cloud changes everything*" more apt than when applied to backups. Microsoft, Amazon, Rackspace, Google and a host of other companies are making it practical and cost-effective to eschew local backups in favor of backing up data securely over the internet to leased repositories in the Cloud. The cost per gigabyte is literally pennies now and, if history is a guide, will continue to decrease to staggeringly low rates as usage explodes.

The incidence of adoption of cloud computing and storage among corporate IT departments is enormous and, assuming no high-profile gaffes, will accelerate with the availability of high bandwidth network connections and as security concerns wane.

But the signal impact of the Cloud won't be as a medium for backup of corporate data but to obviate *any* need for user backup. As data and corporate infrastructure migrate to the cloud, backup will cease to be a customer responsibility and will occur entirely behind-the-scenes as a perennial responsibility of the cloud provider. The cloud provider will likely fulfill that obligation via a mix of conventional backup media (e.g., tape) and redundancy across far-flung regional datacenters. But no matter. *How the cloud provider handles its backup responsibility will be no concern of the customer so long as the system maintains uptime availability.*

Welcome to the Future

Back in 2009, Harvard Law professor Lawrence Lessig observed, "*We are not going back to the twentieth century. In a decade, most Americans will not even remember what that century was like*."[52] That decade has passed, yet much of what lawyers know about enterprise disaster recovery systems harkens back to a century twenty-two years gone. If we seek the *information* of the twentieth century, it's likely to come from backup tapes.

> We are not going back to the twentieth century. In a decade, a majority of Americans will not even remember what that century was like.
>
> **Lawrence Lessig, 2009**

[52] Lawrence Lessig, *Against Transparency*, The New Republic, October 9, 2009.

Backup tapes won’t play a significant role in e-discovery in the twenty-first century, if only because the offline backup we knew--dedicated to disaster recovery and accreted grandfather-father-son[53] rotation schemes--is fast giving way to data repositories nearly as accessible as our own laptops. The distinction between inaccessible backups and accessible active data stores will soon be just a historical curiosity, like selfie sticks or Jared Kushner. Instead, we will turn our attention to a panoply of electronic archives encompassing tape, disk and "cloud" components. The information we once pulled from storage and extracted tape-by-tape will simply be available all the time until someone acts to make it go away. Our challenge won't be in restoring information, but in making sense of it.

[53] Grandfather-father-son describes the most common rotation scheme for backup media. The last daily "son" backup graduates to "father" status at the end of each week. Weekly "father" backups graduate to "grandfather" status at the end of each month. Grandfather backups are often stored offsite long past their utility for disaster recovery.

TEN PRACTICE TIPS FOR BACKUPS IN CIVIL DISCOVERY

1. **Backup ≠ Inaccessible.** Don't expect to exclude the content of backups from the scope of discovery if you haven't laid the foundation to do so. Fed. R. Civ. P. 26(b)(2)(B) requires parties identify sources deemed not reasonably accessible because of undue burden or cost. ***Be prepared to prove the cost and burden through reliable metrics and testimony.***
2. ***Determine if your client:***
   • Routinely restores backup tapes to, *e.g.*, insure the system is functioning properly or as a service to those who have mistakenly deleted files;
   • Restored the backup tapes other matters or uses them as an archive;
   • Has the system capacity and in house expertise to restore the data;
   • Has the capability to search the tapes for responsive data?
3. **Don't blindly pull tapes for preservation.** Backup tapes don't exist in a vacuum but as part of an information system. An effectively managed system incorporates labeling, logging and tracking of tapes, permitting reliable judgments to be made about what's on particular tapes insofar as tying contents to business units, custodians, machines, data sets and intervals. It's costly to have to process tapes just to establish their contents. ***Always* preserve associated backup catalogues when you preserve tapes.**
4. **Be prepared to put forward a sensible sampling protocol in lieu of wholesale restoration.**
5. **Test and sample backups to determine if they hold responsive, material and unique ESI.** Judges are unlikely to force you to restore backup tapes when sensible sampling regiments demonstrate that the effort is likely to yield little of value. Backup tapes are like drilling for oil: *After a few dry holes, it's time to find a new prospect*.
6. **Be prepared to show that the relevant data on tapes is available from more accessible sources.** Sampling, testing and expert testimony help here.
7. **Know the limits of backup search capabilities.** Most backup tools have search capabilities; however, few of these are up to the task of e-discovery. Can the tool search within all common file types and compressed and container file formats?
8. **Appearances matter!** What would the Judge think if she walked through your client's tape storage area? Does it look like a dumping ground?
9. **If using a cloud-based backup system, consider bringing your e-discovery tools to the data in the Cloud instead of spending days getting the data out.**
10. **Backup tape is for disaster recovery.** If it's too stale to use to bring the systems back up, why keep it? Get rid of it!

Appendix 1: Exemplar Backup Tape Log

Tape No.	Sess. ID	Host Name	Backup Date/Time	Size in Bytes	Session Type
ABC 001	37	EX1	8/1/2007 6:15	50,675,122,176	Exchange 200x
ABC 001	38	EX1	8/1/2007 8:28	337,707,008	System state
ABC 001	39	MGT1	8/1/2007 8:29	6,214,713,344	files incremental or differential
ABC 001	40	MGT1	8/1/2007 8:45	5,576,392,704	SQL Database Backup
ABC 001	41	SQL1	8/1/2007 8:58	10,004,201,472	files incremental or differential
ABC 001	42	SQL1	8/1/2007 9:30	8,268,939,264	SQL Database Backup
ABC 001	43	SQL1	8/1/2007 9:52	272,826,368	System state
ABC 005	2	EX1	8/14/2007 18:30	51,735,363,584	Exchange 200x
ABC 005	3	EX1	8/14/2007 20:35	338,427,904	System state
ABC 005	4	MGT1	8/14/2007 20:38	6,215,368,704	files incremental or differential
ABC 005	5	MGT1	8/14/2007 20:53	5,677,776,896	SQL Database Backup
ABC 005	6	SQL1	8/14/2007 21:06	10,499,260,416	files incremental or differential
ABC 005	7	SQL1	8/14/2007 21:38	8,322,023,424	SQL Database Backup
ABC 005	8	SQL1	8/14/2007 21:57	273,022,976	System state

ABC 002	207	NT1	8/15/2007 20:19	31,051,481,088	loose files
ABC 002	18	NT1	8/16/2007 8:06	47,087,616,000	loose files
ABC 014	9	EX1	8/17/2007 6:45	52,449,443,840	Exchange 200x
ABC 014	10	EX1	8/17/2007 8:53	337,969,152	System state
ABC 014	11	MGT1	8/17/2007 8:54	6,215,368,704	files incremental or differential
ABC 014	12	MGT1	8/17/2007 9:09	5,698,748,416	SQL Database Backup
ABC 014	13	SQL1	8/17/2007 9:22	10,537,009,152	files incremental or differential
ABC 014	14	SQL1	8/17/2007 9:47	8,300,986,368	SQL Database Backup
ABC 014	15	SQL1	8/17/2007 10:08	272,629,760	System state
ABC 003	16	NT1	8/18/2007 6:15	46,850,179,072	loose files
ABC 003	17	NT1	8/18/2007 9:26	44,976,308,224	loose files
ABC 004	19	NT1	8/21/2007 6:16	46,901,690,368	loose files
ABC 004	20	NT1	8/21/2007 9:30	44,742,868,992	loose files
ABC 009	30	EX1	8/22/2007 8:52	53,680,603,136	Exchange 200x
ABC 009	31	EX1	8/22/2007 11:01	348,782,592	System state

ABC 009	32	MGT1	8/22/2007 11:03	6,215,434,240	files incremental or differential
ABC 009	33	MGT1	8/22/2007 11:18	5,715,722,240	SQL Database Backup
ABC 009	34	SQL1	8/22/2007 11:31	10,732,371,968	files incremental or differential
ABC 009	35	SQL1	8/23/2007 4:08	8,362,000,384	SQL Database Backup
ABC 009	36	SQL1	8/23/2007 4:33	272,629,760	System state
ABC 011	44	NT1	8/23/2007 6:16	46,938,193,920	loose files
ABC 011	45	NT1	8/23/2007 9:32	44,611,403,776	loose files

Databases in E-Discovery

Years ago, when I set out to write this chapter on databases in electronic discovery, I went to the literature to learn prevailing thought and ensure I wasn't treading old ground. What I found surprised me.

I found there's next to no literature on the topic! What little authority exists makes brief mention of flat file, relational and enterprise databases, notes that discovery from databases is challenging and then flees to other topics.72F[54] A few commentators mention *In re Ford Motor Co.,*73F[55] the too-brief 2003 decision reversing a trial court's order allowing a plaintiff to root around in Ford's databases with nary a restraint. Although the 11th Circuit cancelled that fishing expedition, they left the door open for a party to gain access to an opponent's databases on different facts, such as where the producing party fails to meet its discovery obligations.

Vital Vocabulary

Look for these key terms:

- *Table*
- *Field*
- *Record*
- *Flat File Database*
- *Relational Database*
- *DBMS*
- *Primary Key*
- *Foreign Key*
- *Constraints*
- *Structured Query Language*
- *Schema*
- *Data Dictionary*
- *ERD*
- *Field Mapping*

The constant counsel offered by any article touching on databases in e-discovery is "get help." That's good advice, but not always feasible or affordable.

Because databases run the world, we can't avoid them in e-discovery.

We must know enough about how they work to deal with them when the case budget or time constraints make hiring an expert impossible. We need to know how to identify and preserve databases, and we must learn how to gather sufficient information about them to frame and respond to discovery about databases.

Databases run the world

You can't surf the 'net, place a phone call, swipe your parking access card, use an ATM, charge a meal, buy groceries, secure a driver's license, book a flight or get admitted to an emergency room without a database making it happen.

[54] Happily, since I first published, others have waded in and produced more practical scholarship. Here are links to two recent, thoughtful publications on the topic:
Requests for Production of Databases: Documents v. Data, by Christine Webber and Jeff Kerr
The Sedona Conference Database Principles Addressing the Preservation & Production of Databases & Database Information in Civil Litigation

[55] 345 F.3d 1315 (11th Cir. 2003)

Databases touch our lives all day, every day. Our computer operating systems and e-mail applications are databases. The spell checker in our word processor is a database. Google and Yahoo search engines are databases. Westlaw and Lexis, too. Craigslist. Amazon.com. E-Bay. Facebook. All big honkin' databases.

Yet, when it comes to e-discovery, we tend to fix our attention on documents, without appreciating that most electronic evidence exists only as a flash mob of information assembled and organized on the fly from a dozen or thousand or million discrete places. In our zeal to lay hands on documents instead of data, we make discovery harder, slower and costlier. Understanding databases and acquiring the skills to peruse and use their contents gets us to the evidence better, faster and cheaper.

Databases are even changing the way we think about discovery. Historically, parties weren't obliged to *create* documents for production in discovery; instead, you produced what you had on file. Today, documents don't exist until you generate them. Tickets, bank statements, websites, price lists, phone records and register receipts are all just *ad hoc* reports generated by databases. Documents don't take tangible form until you print them out, and more and more, only the tiniest fraction of documents—one-tenth of one percent—will emerge as ink on paper, obliging litigants to be adept at both crafting queries to elicit responsive data and mastering ways to interpret and use the data stream that emerges.

Introduction to Databases

Most of us use databases with no clue how they work. Take e-mail, for example. Whether you know it or not, each e-mail message you view in Outlook or through your web browser is a report generated by a database query and built of select fields of information culled from a complex dataset. It's then presented to you in a user-friendly arrangement determined by your e-mail client's capabilities and user settings.

That an e-mail message is not a single, discrete document is confusing to some. The data segments or "fields" that make up an e-mail are formatted with such consistency from application-to-application and appear so similar when we print them out that we mistake e-mail messages for fixed documents. But each is really a customizable report from the database called your e-mail.

When you see a screen or report from a database, you experience an assemblage of information that "feels" like a document, but the data that comes together to create what you see are often drawn from different sources within the database and from different systems, locations and formats, all changing moment to moment.

Understanding databases begins with mastering some simple concepts and a little specialized terminology. Beyond that, the distinction between your e-mail database and Google's is mostly marked by differences in scale, optimization and security.

Constructing a Simple Database

If you needed a way to keep track of the cases on your docket, you'd probably begin with a simple table of columns and rows written on a legal pad. You'd start listing your clients by name. Then, you might list the names of other parties, the case number, court, judge and trial date. If you still had room, you'd add addresses, phone numbers, settlement demands, insurance carriers, policy numbers, opposing counsel and so on.

In database parlance, you've constructed a **"table,"** and each separate information item you entered (e.g., name, address, court) is called a **"field."** The group of items you assembled for each client (probably organized in columns and arranged in a row to the right of each name) is collectively called a **"record."** Because the client's name is the field that governs the contents of each record, it would be termed the **"key field."**

Pretty soon, your table would be unwieldy and push beyond the confines of a sheet of paper. If you added a new matter or client to the table and wanted it to stay in alphabetical order by client name, you'd probably have to rewrite the list.

So, you might turn to index cards. Now, each card is a "record" and lists the information (the "fields") pertinent to each client. It's easy to add cards for new clients and re-order them by client name. Then, sometimes you'd want to order matters by trial date or court. To do that, you'd either need to extract specific data from each card to compile a report, re-sort the cards, or maintain three sets of differently ordered cards, one by name, one by trial date and a third by court.

Your cards comprise a database of three tables. They are still deemed tables even though you used a card to hold each record instead of a row. One table uses client name as its key field, another uses the trial date and the third uses the court. Each of these three sets of cards is a **"flat file database,"** distinguished by the characteristic that all the fields and records (the cards) comprise a single file (i.e., each a deck of cards) with no relationships or links between the various records and fields except the table structure (the order of the deck and the order of fields on the cards).

Of course, you'd need to keep all cards up-to-date as dates, phone numbers and addresses change. When a client has more than one matter, you'd have to write all the same client data on multiple cards and update each card, one-by-one, trying not to overlook any card. What a pain!

So, you'd automate, turning first to something like a spreadsheet. Now, you're not limited by the dimensions of a sheet of paper. When you add a new case, you can insert it anywhere and re-sort the list by name, court or trial date. You're not bound by the order in which you entered the information, and you can search electronically.

Though faster and easier to use than paper and index cards, your simple spreadsheet is still just a table in a flat file database. You must update every field that holds the same data when that data changes (though "find and replace" functions make this more efficient and reliable), and when you want to add, change or extract information, you have to open and work with the entire table.
What you need is a system that allows a change to *one* field to update *every* field in the database with the same information, not only within a single table but across *all* tables in the database. You need a system that identifies the relationship between common fields of data, updates them when needed and, better still, uses that common relationship to bring together more related information. Think of it as adding rudimentary intelligence to a database, allowing it to "recognize" that records sharing common fields likely relate to common information. Databases that do this are called **"relational databases,"** and they account for most of the databases used in business today, ranging from simple, inexpensive tools like Microsoft Access or Intuit QuickBooks to enormously complex and costly "enterprise-level" applications marketed by Oracle and SAP.74F[56]

To be precise, only the tables of data are the "database," and the software used to create, maintain and interrogate those tables is called the **Database Management System** or **DBMS**. In practice, the two terms are often used interchangeably.

Relational Databases
Let's re-imagine your case management system as a relational database. You'd still have a table listing all clients organized by name. On this CLIENTS table, each client record includes name, address and case number(s). Even if a client has multiple cases in your office, there is still just a single table listing:

CLIENTS

CLT_LAST	CLT_FIRST	ST_ADD	CITY	STATE	ZIP	CASE_NO
Ballmer	Steven	3832 Hunts Point Rd.	Hunts Point	WA	98004	001, 005
Chambers	John	5608 River Way	Buena Park	CA	90621	002
Dell	Michael	3400 Toro Canyon Rd.	Austin	TX	78746	003, 007
Ellison	Lawrence	745 Mountain Home Rd.	Woodside	CA	94062	004
Gates	William	1835 73rd Ave. NE	Medina	WA	98039	001, 005
Jobs	Steven	460 Mountain Home Rd.	Woodside	CA	94062	006, 009
Palmisano	Samuel	665 Pequot Ave.	Southport	CT	06890	007

[56] One of the most important and widely used database applications, MySQL, is open source; so, while great fortunes have been built on relational database tools, the database world is by no means the exclusive province of commercial software vendors.

It’s essential to keep track of cases and upcoming trials, so you create another table called CASES:

CASES

CASE_NO	TRL_DATE	MATTER	TYPE	COURT
001	2011-02-14	U.S. v. Microsoft	Antitrust	FDDC-1
002	2012-01-09	EON v Cisco	Patent	FEDTX-2
003	2011-02-15	In re: Dell	Regulatory	FWDTX-4
004	2011-05-16	SAP v. Oracle	Conspiracy	FNDCA-8
005	2012-01-09	Microsoft v. Yahoo	Breach of K	FWDWA-6
006	2010-12-06	Apple v. Adobe	Antitrust	FNDCA-8
007	2011-10-31	Dell v. Travis County	Tax	TX250
008	null	Hawkins v. McGee	Med Mal	FUSSC
009	2011-12-05	Jobs v. City of Woodside	Tax	CASMD09

You also want to stay current on where your cases will be tried and the presiding judge, so you maintain a COURTS table for all the matters on your docket:

COURTS

COURT	JUDGE	FED_ST	JURISDICTION
FNDCA-8	Laporte	FED	Northern District of California (SF)
FDDC-1	Kollar-Kotelly	FED	USDC District of Columbia
FWDTX-4	Sparks	FED	Western District of Texas
TX250	Dietz	STATE	250th JDS, Travis County, TX
CASMD09	Parsons	STATE	San Mateo Superior Court, CA
FEDTX-2	Ward	FED	Eastern District of Texas
FWDWA-6	Jones	FED	Western District of Washington
FUSSC	Hand	FED	United States Supreme Court

As we look at these three tables, note that each has a unique key field called the “**primary key**” for that table.75F[57] For the CLIENTS table, the primary key is the client’s last name.76F[58] The primary key is the trial date for the TRIAL_DATES table and it’s a unique court identifier for the COURTS table. The essential characteristic of a primary key is that it cannot repeat within the table for which it serves as primary key, and a properly-designed database will prevent a user from creating duplicate primary keys.

[57] Tables can have more than one primary key.

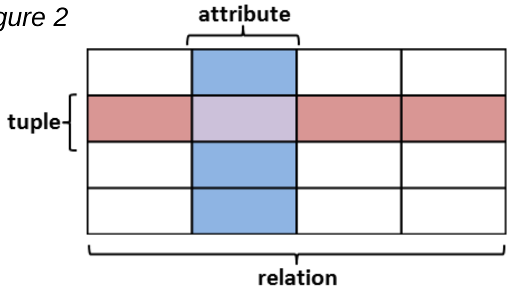
[58] In practice, a last name would be a poor choice for a primary key in that names tend not to be unique—certainly a law firm could expect to have multiple clients with the same surname.

Many databases simply assign a unique primary key to each table row, either a number or a non-recurring value built from elements like the first four letters of a name, first three numbers in the address, first five letters in the street name and the Zip code. For example, an assigned key for Steve Ballmer derived from data in the CLIENTS table might be BALL383HUNTS98004. The primary key is used for indexing the table to make it more efficient to search, sort, link and perform other operations on the data.

Tuples and Attributes

Now, we need to introduce some new terminology because the world of relational databases has a language all its own. Dealing with the most peculiar term first, the contents of each row in a table is called a **"tuple,"** defined as an ordered list of elements.77F[59] In the COURTS table above, there are seven tuples, each consisting of four elements. These elements, ordered as columns, are called **"attributes,"** and what we've called tables in the flat file world are termed **"relations"** in relational databases. Put another way, *a relation is defined as a set of tuples that have the same attributes* (See Figure 1).

Figure 2

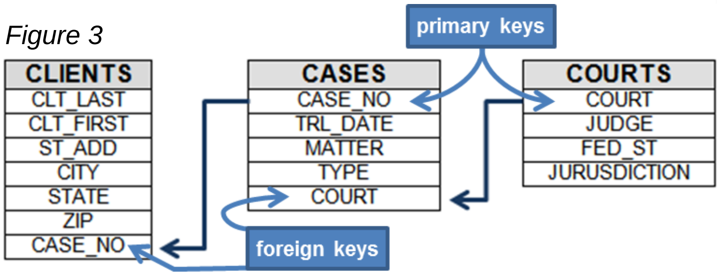


The magic happens in a relational database when tables are **"joined"** (much like the cube in Figure 2)78F[60] by referencing one table from another.79F[61] This is done by incorporating the primary key in the table referenced as a **"foreign key"** in the referencing table. The table referenced is the **"parent table,"** and the referencing table is the **"child table"** in this joining of the two relations.

Figure 1

In Figure 3, COURTS is the parent table to CASES with respect to the primary key field, "COURT." In the CASES table, the foreign key for the field COURT points back to the COURTS table, assuring that the

Figure 3



[59] Per Wikipedia, the term "tuple" originated as an abstraction of the sequence: single, double, triple, quadruple, quintuple, sextuple, septuple, octuple...n-tuple. The unique 0-tuple is called the null tuple. A 1-tuple is called a "singleton," a 2-tuple is a "pair" and a 3-tuple is a "triple" or "triplet." The *n* can be any positive integer. For example, a complex number can be represented as a 2-tuple, a quaternion can be represented as a 4-tuple, an octonion can be represented as an octuple (mathematicians use the abbreviation "8-tuple"), and a sedenion can be represented as a 16-tuple. I include this explanation to remind readers why many of us went to law school instead of studying computer science.

[60] Although unlike the cube, a relational database is not limited to just three dimensions of attachment.

[61] The term "relation" is so confounding here, I will continue to refer to them as tables.

most current data will populate the field. In turn, the CLIENTS table employs a foreign key relating to the CASE_NO attribute in the CASE table, again assuring that the definitive information populates the attribute in the CLIENTS table.

Remember that what you are seeking here is to ensure that you do not build a database with inconsistent data, such as conflicting client addresses. Data conflicts are avoided in relational databases by allowing the parent primary key to serve as the definitive data source. So, by pointing each child table to that definitive parent via the use of foreign keys, you promote so-called **"referential integrity"** of the database. Remember, also, that while a primary key must be unique to the parent table, it can be used as many times as desired when referenced as a foreign key. As in life, parents can have multiple children, but a child can have but one set of (biological) parents.

Field Properties and Record Structures

When you were writing case data on your index cards, you were unconstrained in terms of the information you included. You could abbreviate, write dates as words or numeric values and include as little or as much data as the space on the card and intelligibility allowed. But for databases to perform properly, the contents of fields should conform to certain constraints to insure data integrity. For example, you wouldn't want a database to accept four or ten letters in a field reserved for a Zip code. Neither should the database accept duplicate primary keys or open a case without including the name of a client. If a field is designed to store only a U.S. state, then you don't want it to accept "Zambia" or "female." You also don't want it to accept "Noo Yawk."

Accordingly, databases are built to enforce specified field property requirements. Such properties may include:

1. **Field size**: limiting the number of characters that can populate the field or permitting a variable length entry for memos;
2. **Data type**: text, currency, integer numbers, date/time, e-mail address and masks for phone numbers, Social security numbers, Zip codes, etc.;
3. **Unique fields**: Primary keys must be unique. You typically wouldn't want to assign the same case number to different matters or two Social Security numbers to the same person.
4. **Group or member lists**: Often fields may only be populated with data from a limited group of options (e.g., U.S. states, salutations, departments and account numbers);
5. **Validation rules**: To promote data integrity, you may want to limit the range of values ascribed to a field to only those that makes sense. A field for a person's age shouldn't accept negative values or (so far) values in excess of 125. A time field should not accept "25:00pm" and a date field designed for use by Americans should guard against European date notation. Credit card numbers must conform to specific rules, as must Zip codes and phone numbers; or

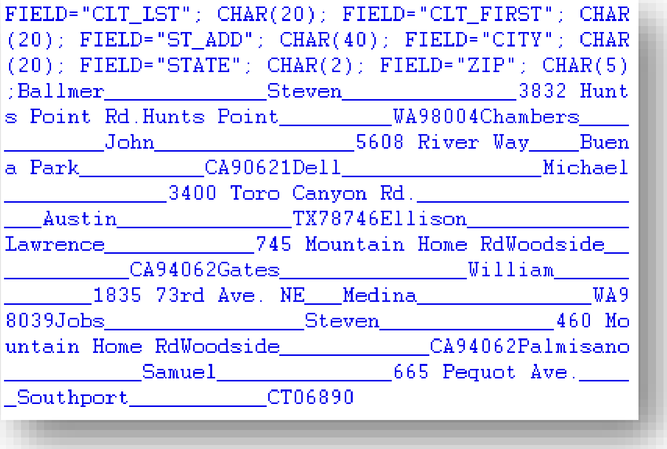
6. **Required data**: The absence of certain information may destroy the utility of the record, so certain fields are made mandatory (e.g., a car rental database may require input of a valid driver's license number).

You'll appreciate why demanding production of the raw tables in a database may be an untenable approach to e-discovery when you consider how databases store information. When a database populates a table, it's stored in either **fixed length** or **variable length** fields.

Fixed-Length Field Records

Fixed length fields are established when the database is created, and it's important to appreciate that the data is stored as long sequences of data that may, to the untrained eye, simply flow together in one incomprehensible blob. A fixed length field record may begin with information setting out information concerning all of the fields in the record, such as each field's name (e.g., COURT), followed by its data type (e.g., alphanumeric), length (7 characters) and format (e.g., only values matching a specified list of courts).

Figure 4

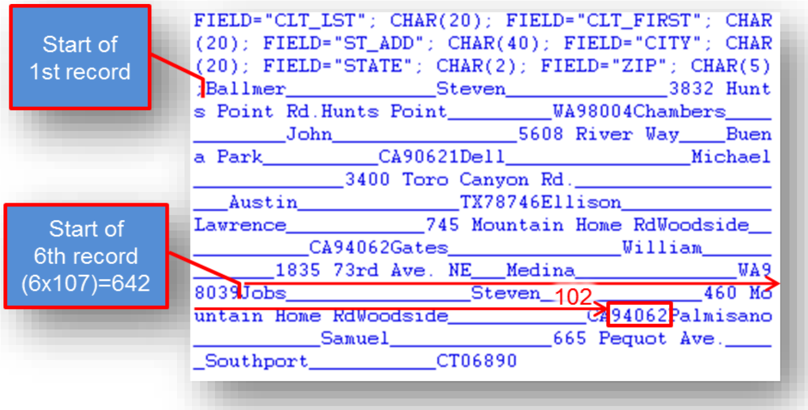


```
FIELD="CLT_LST"; CHAR(20); FIELD="CLT_FIRST"; CHAR
(20); FIELD="ST_ADD"; CHAR(40); FIELD="CITY"; CHAR
(20); FIELD="STATE"; CHAR(2); FIELD="ZIP"; CHAR(5)
;Ballmer_____________Steven______________3832 Hunt
s Point Rd.Hunts Point_________WA98004Chambers____
________John________________5608 River Way____Buen
a Park__________CA90621Dell________________Michael
_____________3400 Toro Canyon Rd._________________
___Austin______________TX78746Ellison_____________
Lawrence____________745 Mountain Home RdWoodside__
__________CA94062Gates_______________William______
_______1835 73rd Ave. NE___Medina______________WA9
8039Jobs________________Steven______________460 Mo
untain Home RdWoodside____________CA94062Palmisano
___________Samuel______________665 Pequot Ave.____
_Southport___________CT06890
```

A fixed length field record for a simplified address table might look like **Figure 4**.

Note how the data is one continuous stream. The name, order and length of data allocated for each field is defined at the beginning of the string in all those "FIELD=" and CHAR(x) statements, such that the total length of each record is 107 characters. To find a given record in a table, the database software simply starts accessing data for that record at a distance (also called an "offset") from the start of the table equal to the number of records times the total length allocated to each record.

Figure 5



```
FIELD="CLT_LST"; CHAR(20); FIELD="CLT_FIRST"; CHAR
(20); FIELD="ST_ADD"; CHAR(40); FIELD="CITY"; CHAR
(20); FIELD="STATE"; CHAR(2); FIELD="ZIP"; CHAR(5)
;Ballmer_____________Steven______________3832 Hunt
s Point Rd.Hunts Point_________WA98004Chambers____
________John________________5608 River Way____Buen
a Park__________CA90621Dell________________Michael
_____________3400 Toro Canyon Rd._________________
___Austin______________TX78746Ellison_____________
Lawrence____________745 Mountain Home RdWoodside__
__________CA94062Gates_______________William______
_______1835 73rd Ave. NE___Medina______________WA9
8039Jobs________________Steven______________460 Mo
untain Home RdWoodside____________CA94062Palmisano
___________Samuel______________665 Pequot Ave.____
_Southport___________CT06890
```

So, as shown in **Figure 5**, the fourth record starts 428 characters from the start of the first record. In turn, each field in the record starts a fixed number of characters from the start of the record. If you wanted to extract the late Steve Jobs' Zip code from the exemplar table, the Jobs address record is the 6th record, so it starts 642 characters (or bytes) from the start of the first record and the Zip code field begins 102 characters from the start of the sixth record (20+20+40+20+2), or 744 bytes from the start of the first record. This sort of offset retrieval is tedious for humans, but it's a cinch for computers.

Variable-Length Field Records

One need only recall the anxiety over the Y2K threat to appreciate why fixed length field records can be problematic. Sometimes, the space allocated to a field proves insufficient in unanticipated ways, or you may simply need to offer the ability to expand the size of a record on-the-fly. Databases employ variable length field records whose size can change from one record to the next. Variable length fields employ **pointer fields** that seamlessly redirect data retrieval to a designated point in the memo file where the variable length field data begins (or continues). The database software then reads from the memo file until it encounters an end-of-file marker or another pointer to a memo location holding further data.

Forms, Reports and Query Language

Now that you've glimpsed the ugly guts of database tables, you can appreciate why databases employ database management software to enter, update and retrieve data. Though DBMS software serves many purposes geared to indexing, optimizing and protecting data, the most familiar role of DBMS software is as a user interface for forms and reports.

There's little difference between forms and reports except that we tend to call the interface used to input and modify data a "form" and the interface to extract data a "report." Both are simply user-friendly ways to implement commands in **"query languages."**

Query language is the term applied to the set of commands used to retrieve information from a database. The best known and most widely used of these is called **SQL** (for **Structured Query Language**, officially 'ess-cue-ell,' but most everyone calls it "sequel"). SQL is a computer language, but different from computer languages like Java or C++ that can be used to construct applications, SQL's sole purpose is the creation, management and interrogation of databases.

Though the moniker "query language" might lead anyone to believe that its raison d'être is to get data out of databases, in fact, SQL handles the heavy lifting of database creation and data insertion, too. SQL includes subset command sets for data control (DCL), data manipulation (DML) and data definition (DDL). SQL syntax is beyond the scope of this paper, but the following snippet of code will give you a sense of how SQL is used to create a table like the case management tables discussed above:

```
CREATE TABLE COURTS
        (COURT varchar(7), PRIMARY KEY,
        JUDGE varchar(18),
        FED_ST varchar(5),
        JURISDICTION varchar (40));
CREATE TABLE CASES
        (CASE_NO int IDENTITY(1,1)PRIMARY KEY,
        TRL_DATE
        MATTER varchar (60),
        TYPE varchar (40)
        COURT varchar(7));
```

In these few lines, the COURTS and CASES tables are created, named and ordered into various alphanumeric fields of varying specified lengths. Two primary keys are set and one key, CASE_NO, is implemented so as to begin with the number 1 and increment by 1 each time a new case is added to the CASES table.

Who Owns SQL?

In fact, nobody “owns” SQL, but several giant software companies, notably Oracle and Microsoft, have built significant products around SQL and produced their own proprietary dialects of SQL. When you hear someone mention “SQL Server,” they’re talking about a Microsoft product, but Microsoft doesn’t own SQL; it markets a database application that’s compatible with SQL.

SQL has much to commend it, being both simple and powerful; but, even the simplest computer language is too much for the average user. So, databases employ graphical user interfaces (GUIs) to put a friendly face on SQL. When you enter data into a form or run a search, you’re simply triggering a series of pre-programmed SQL commands.

In e-discovery, if the standard reports supported by the database are sufficiently encompassing and precise to retrieve the information sought, great! You’ll have to arrive at a suitable form of production and perhaps wrangle over scope and privilege issues; but, the path to the data is clear.

However, because most companies design their databases for operations not litigation, very often, the standard reporting capabilities won’t retrieve the types of information required in discovery. In that event, you’ll need more than an SQL doctor on your team; you’ll also need a good x-ray of the databases to be plumbed.

Schemas, Data Dictionaries, System Catalogs, and ERDs,

The famed database administrator, Leo Tolstoy, remarked, “Great databases are all alike, every ordinary database is ordinary in its own way.” Although it’s with tongue-in-cheek that I invoke Tolstoy’s famous observation on happy and unhappy families, it’s apt here and means that you can only assume so much about the structure of an unfamiliar database. After that, you need the manual and a map.

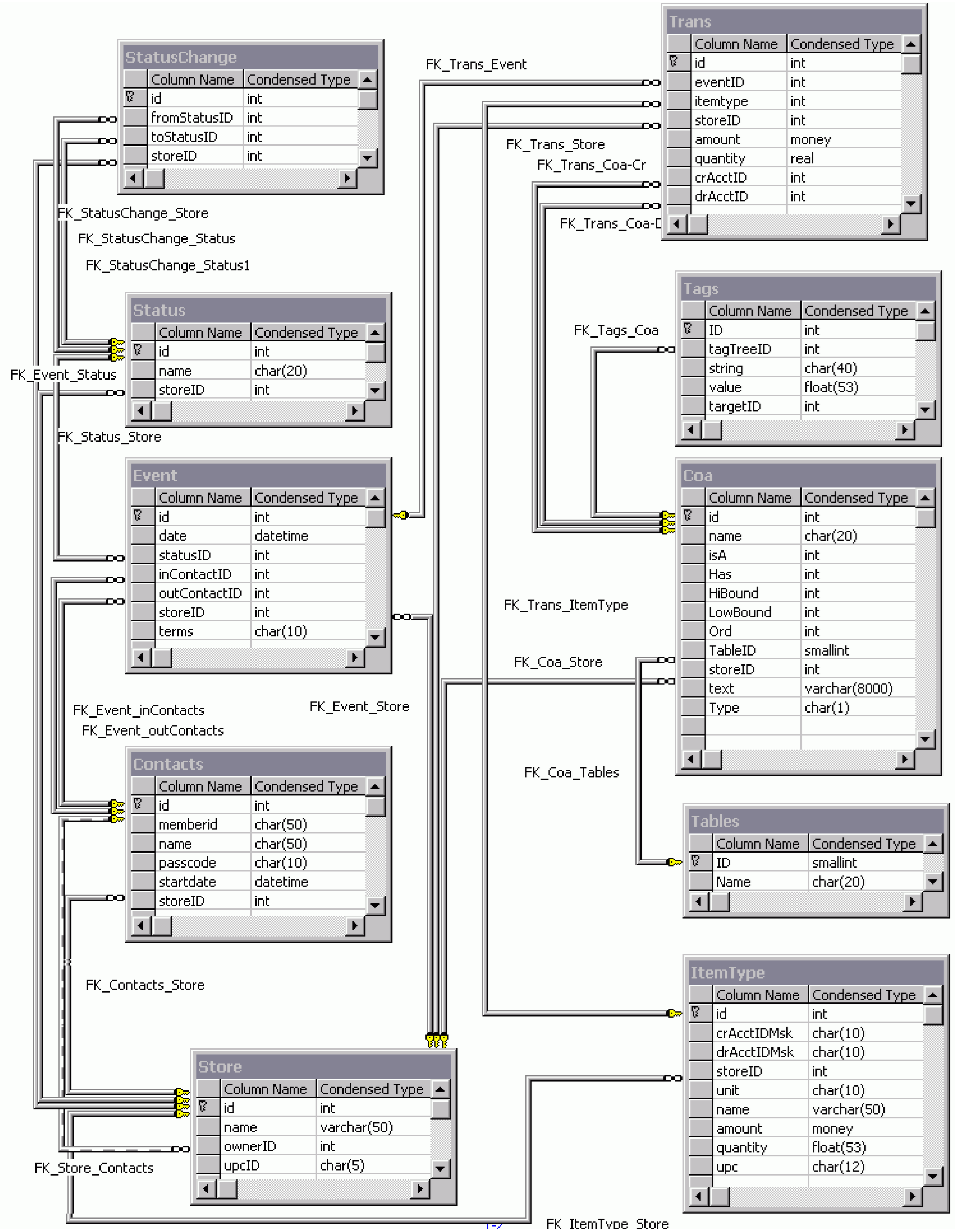
.

In the lingo of database land, the “map” is the database’s **schema,** and it’s housed in the system’s **data dictionary**. It may be the system’s **logical schema**, detailing how the database is designed in terms of its table structures, attributes, fields, relationships, joins and views. Or, it could be its **physical schema**, setting out the hardware and software implementation of the database on machines, storage devices and networks. As Tolstoy might have said, “A logical schema explains death; but it won’t tell you where the bodies are buried.”

Information in a database is mostly gibberish without the metadata that gives it form and function. In an SQL database, the compendium of all that metadata is called the **system catalog.** In practice, the terms system catalog, schema and data dictionary seem to be used interchangeably—they are all—in essence--databases storing information about the metadata of a database. The most important lesson to derive from this discussion is that there is a map—or one can be easily generated—so get it!

Unlike that elusive Loch Ness monster of e-discovery, the “enterprise data map,” the schemas of databases tend to exist and are usually maps; that is, graphical depictions of the database structures. **Entity-Relationship Modeling** (**ERM**) is a system and notation used to lay out the conceptual and logical schema of a relational database. The resulting diagrams (akin to flow charts) are called **Entity-Relationship Diagrams** or **ERDs (Figure 6, next page)**.

Figure 6: ERD of Database Schema



StatusChange

Column Name	Condensed Type
id	int
fromStatusID	int
toStatusID	int
storeID	int

Status

Column Name	Condensed Type
id	int
name	char(20)
storeID	int

Event

Column Name	Condensed Type
id	int
date	datetime
statusID	int
inContactID	int
outContactID	int
storeID	int
terms	char(10)

Contacts

Column Name	Condensed Type
id	int
memberid	char(50)
name	char(50)
passcode	char(10)
startdate	datetime
storeID	int

Store

Column Name	Condensed Type
id	int
name	varchar(50)
ownerID	int
upcID	char(5)

Trans

Column Name	Condensed Type
id	int
eventID	int
itemtype	int
storeID	int
amount	money
quantity	real
crAcctID	int
drAcctID	int

Tags

Column Name	Condensed Type
ID	int
tagTreeID	int
string	char(40)
value	float(53)
targetID	int

Coa

Column Name	Condensed Type
id	int
name	char(20)
isA	int
Has	int
HiBound	int
LowBound	int
Ord	int
TableID	smallint
storeID	int
text	varchar(8000)
Type	char(1)

Tables

Column Name	Condensed Type
ID	smallint
Name	char(20)

ItemType

Column Name	Condensed Type
id	int
crAcctIDMsk	char(10)
drAcctIDMsk	char(10)
storeID	int
unit	char(10)
name	varchar(50)
amount	money
quantity	float(53)
upc	char(12)

Two Lessons from the Database Trenches

The importance of securing the schema, manuals, data dictionary and ERDs was borne out by my experience serving as Special Master for Electronically Stored Information. in a drug product liability action involving thousands of plaintiffs, I was tasked to expedite discovery from as many as 60 different enterprise databases, each more sprawling and complex than the next. The parties were at loggerheads, and serious sanctions were in the offing.

The plaintiffs insisted the databases would yield important evidence. Importantly, plaintiffs' team included support personnel technically astute enough to get deeply into the weeds with the systems. Plaintiffs were willing to narrow the scope of their database discovery to eliminate those that were unlikely to be responsive and to narrow the scope of their requests. But, to do that, they'd need to know the systems.

For each system, we faced the same questions:

i. What does the database do?
ii. What is it built on?
iii. What information does it hold?
iv. What content is relevant, responsive and privileged?
v. What forms does it take?
vi. How can it be searched effectively; using what query language?
vii. What are its reporting capabilities?
viii. What form or forms of production will be functional, searchable and cost-effective?

It took a three-step process to turn things around. First, the plaintiffs were required to do their homework, and the defense supplied the curriculum. That is, the defense was required to furnish documentation concerning the databases. First, each system had to be identified. The defense prepared a spreadsheet detailing, *inter alia*:

- Names of systems
- Applications;
- Date range of data;
- Size of database;
- User groups; and
- Available system documentation (including ERDs and data dictionaries).

This enabled plaintiffs to prioritize their demands to the most relevant systems. I directed the defendants to furnish operator's manuals, schema information and data dictionaries for the most relevant systems.

The second step was ordering that narrowly-focused meet-and-confer sessions be held between technical personnel for both sides. These were conducted by telephone, and the sole topic of each was one or more of the databases. The defense was required to make knowledgeable personnel available for the calls and plaintiffs were required to confine their questions to the nuts-and-bolts of the databases at issue.

When the telephone sessions concluded, Plaintiffs were directed to serve their revised request for production from the database. In most instances, the plaintiffs had learned enough about the databases that they were actually able to propose SQL queries to be run.

This would have been sufficient in most cases, but this case was especially contentious. The decisive step needed to resolve the database discovery logjam was a meeting in the nature of a mediation over which I would preside. In this proceeding, counsel and technical liaison, joined by the database specialists, would meet face-to-face over two days. We would work through each database and arrive at specific agreements concerning the scope of discovery for each system, searches run, sample sizes employed and timing and form of production. The devil is in the details, and the goal was to nail down every detail.

It took two such sessions, but in the end, disputes over databases largely ceased, the production changed hands smoothly, and the parties could refocus on the merits.

The heroes in this story are the technical personnel who collaborated to share information and find solutions when the lawyers could see only contentions. The lesson: **Get the geeks together, and then get out of their way.**

Lesson Two

In a recent case where I served as special master, the Court questioned the adequacy of defendants' search of their databases. The defendants used many databases to run their far-flung operations, ranging from legacy mainframe systems housed in national data centers to homebrew applications cobbled together using Access or Excel. But whether big or small, I found with disturbing regularity that the persons tasked to query the systems for responsive data didn't know how to use them or lacked the rights needed to access the data they were obliged to search.

The takeaway: **Never assume that a DBMS query searches all of the potentially responsive records, and never assume that the operator knows what they are doing.**

Database systems employ a host of techniques to optimize performance and protect confidentiality. For example

- Older records may be routinely purged from the indices;
- Users may lack the privileges within the system to access all the potentially responsive records;
- Queries may be restricted to regions or business units;
- Tables may not be joined in the particular ways needed to gather the data sought.

Any of these may result in responsive data being missed, even by an apparently competent operator.

Establishing operator competence can be challenging, too. Ask a person tasked with running queries if they have the requisite DBMS privileges required for a comprehensive search, and they're likely to give you a dirty look and insist they do. In truth, they probably don't know. What they have are the privileges they need to do their job day-to-day; but those may not be nearly sufficient to elicit all of the responsive information the system can yield.

How do you preserve a database in e-discovery?

Talk to even tech-savvy lawyers about preserving databases, and you'll likely hear how database are gigantic and dynamic or how incomprehensibly risky and disruptive it is to mess with them. The lawyer who responds, "Don't be ridiculous. We're not preserving our databases for your lawsuit," isn't protecting her client.

Or, opposing counsel may say, "Preserve our databases? Sure, no problem. We back up the databases all the time. We'll just set aside some tapes." This agreeable fellow isn't protecting his client either. When it comes time to search the data on tape, Mr. Congeniality may learn that his client has no ability to restore the data without displacing the server currently in use, and restoration doesn't come quick or cheap.

What both lawyers should have said is, "Let me explain what we have and how it works. Better yet, let's get our technical advisors together. Then, we'll try to work out a way to preserve what you really need in a way you can use it. If we can't agree, I'll tell you what my client will and won't do, and you can go to the judge right away, if you think we haven't done enough."

Granted, this conversation almost never occurs for a host of reasons. Counsel may have no idea what the client has or how it works. Or the duty to preserve attaches before an opposing counsel emerges. Or counsel believes that cooperation is anathema to zealous advocacy and wants only to scorch the Earth.

In fact, it's not that daunting to subject most databases to a defensible litigation hold, if you understand how the database works and exert the time and effort required to determine what you're likely to need preserved.

Databases are dynamic by design, but not all databases change in ways that adversely impact legal hold obligations. Many databases—particularly accounting databases—are accretive in design. That is, they add new data as time goes on, but do not surrender the ability to thoroughly search data that existed in prior periods. For accretive databases, all counsel may need to do is ascertain and ensure that historical data isn't going anywhere for the life of the case.

Creating snapshots of data stores or pulling a full backup set for a relevant period is a sensible backstop to other preservation efforts, as an "if all else fails" insurance policy against spoliation. If the likelihood of a lawsuit materializing is remote or if there is little chance that the tapes preserved will ultimately be subjected to restoration, preservation by only pulling tapes may prove sufficient and economical. But, if a lawsuit is certain and discovery from the database(s) is likely, the better approach is to identify ways to either duplicate and/or segregate the particular dynamic data you'll need or export it to forms that won't unduly impair searchability and utility. That is, you want to keep the essential data reasonably accessible and shield it from changes that will impair its relevance and probative value.

If the issue in litigation is temporally sensitive—e.g., wholesale drug pricing in 2018 or reduction in force decisions in 2019—you'll need to preserve the responsive data before the myriad components from which it's drawn, and the filters, queries and algorithms that govern how it's communicated, change. You'll want to retain the ability to generate the reports that should be reasonably anticipated and not lose that ability because of an alteration in some dynamic element of the reporting process.

Forms of Production

In no other corner of e-discovery are litigants quite so much as the dog that caught the car than when dealing with databases. Data from specialized and enterprise databases often don't play well with off-the-shelf applications; not surprising, considering the horsepower and high cost of the systems tasked to run these big iron applications. Still, there is always a way.

Sometimes a requesting party demands a copy of an entire database, often with insufficient consideration of what such a demand might entail were it to succeed. If the database is built in Access or on other simple platforms, it's feasible to acquire the hardware and software licenses required to duplicate the producing party's database environment sufficiently to run the application. But, if the data sets are so large as to require massive storage resources or are built on an enterprise-level DBMS like Oracle or SAP, mirroring the environment is almost out of the question. I say "almost" because the emergence of Infrastructure-as-a-Service Cloud computing options promises to make it possible for mere mortals to acquire enterprise-level computing power for short stints

A more likely production scenario is to narrow the data set by use of filters and queries, then either export the responsive date to a format that can be analyzed in other applications (e.g., exported as extensible markup language (XML), comma separated values (CSV) or in another delimited file) or run reports (standard or custom) and ensure that the reporting takes a form that, unlike paper printouts, lends itself to electronic search.

Before negotiating a form of production, investigate the capabilities of the DBMS. The database administrator may not have had occasion to undertake a data export and so may have no clue what an application can do much beyond the confines of what it does every day. It's the rare DBMS that can't export delimited data. Next, have a proposed form of production in mind and, if possible, be prepared to instruct the DBMS administrator how to secure the reporting or export format you seek,

Remember that the resistance you experience in seeking to export to electronic formats may not come from the opposing party of the DBMS administrator. More often, an insistence on reports being produced as printouts or page images is driven by the needs of opposing counsel. In that instance, it helps to establish that the export is feasible as early as possible.

As with other forms of e-discovery, be careful not to accept production in formats you don't want because, like-it-or-not, many Court give just one bite at the production apple. If you accept it on a paper or as TIFF images for the sake of expediency, you often close the door on re-production in more useful forms.

Even if the parties can agree upon an electronic form of production, it's nevertheless a good idea to secure a test export to evaluate before undertaking a high volume export.

Closing Thoughts

When dealing with databases in e-discovery, requesting parties should avoid the trap of "You have it. I want it." Lawyers who'd never be so foolish as to demand the contents of a file room will blithely insist on production of the "database." For most, were they to succeed in such a foolish quest, they'd likely find themselves in possession of an obscure collection of inscrutable information they can't possibly use.

Things aren't much better on the producing party's side, where counsel routinely fail to explore databases in e-discovery on the theory that, if a report hasn't been printed out, it doesn't have to be created for the litigation. Even when they do acknowledge the duty to search databases, few counsel appreciate how pervasively embedded databases are in their clients' businesses, and fewer still possess the skills needed to translate an amorphous request for production into precise, effective queries.

Each is trading on ignorance, and both do their clients a disservice.

But these are the problems of the past and, increasingly, there's cause for cautious optimism in how lawyers and litigants approach databases in discovery. Counsel are starting to inquire into the existence and role of databases earlier in the litigation timeline and are coming to appreciate not only how pervasive databases are in modern commerce, but how inescapable it is that they take their place as important sources of discoverable ESI.

More on Databases in Discovery

I loathe the practice of law from forms but bow to its power. Lawyers love forms; so, to get lawyers to use more efficient and precise prose in their discovery requests, we can't just harangue them to do it; we've "got to put the hay down where the goats can get it." To that end, here is some language to consider when seeking information about databases and when serving notice of the deposition of corporate designees (*e.g.,* per Rule 30(b)(6) in Federal civil practice or Rule 199(b)(1) of the Texas Rules of Civil Procedure):

For each database or system that holds potentially responsive information, we seek the following information to prepare to question the designated person(s) who, with reasonable particularity, can testify on your behalf about information known to or reasonably available to you concerning:

1. **The standard reporting capabilities of the database or system, including the nature, purpose, structure, appearance, format and electronic searchability of the information conveyed within each standard report (or template) that can be generated by the database or system or by any overlay reporting application;**
2. **The enhanced reporting capabilities of the database or system, including the nature, purpose structure, appearance, format and electronic searchability of the information conveyed within each enhanced or custom report (or template) that can be generated by the database or system or by any overlay reporting application;**
3. **The flat file and structured export capabilities of each database or system, particularly the ability to export to fielded/delimited or structured formats in a manner that faithfully reflects the content, integrity and functionality of the source data;**
4. **Other export and reporting capabilities of each database or system (including any overlay reporting application) and how they may or may not be employed to faithfully reflect the content, integrity and functionality of the source data for use in this litigation;**
5. **The structure of the database or system to the extent necessary to identify data within potentially responsive fields, records and entities, including field and table names, definitions, constraints and relationships, as well as field codes and field code/value translation or lookup tables.**
6. **The query language, syntax, capabilities and constraints of the database or system (including any overlay reporting application) as they may bear on the ability to identify, extract and export potentially responsive data from each database or system;**
7. **The user experience and interface, including datasets, functionality and options available for use by persons involved with the PROVIDE APPROPRIATE LANGUAGE RE THE ACTIVITIES PERTINENT TO THE MATTERS MADE THE BASIS OF THE SUIT;**

8. **The operational history of the database or system to the extent that it may bear on the content, integrity, accuracy, currency or completeness of potentially responsive data;**

9. **The nature, location and content of any training, user or administrator manuals or guides that address the manner in which the database or system has been administered, queried or its contents reviewed by persons involved with the PROVIDE APPROPRIATE LANGUAGE RE THE ACTIVITIES PERTINENT TO THE MATTERS MADE THE BASIS OF THE SUIT;**

10. **The nature, location and contents of any schema, schema documentation (such as an entity relationship diagram or data dictionary) or the like for any database or system that may reasonably be expected to contain information relating to the PROVIDE APPROPRIATE LANGUAGE RE THE ACTIVITIES PERTINENT TO THE MATTERS MADE THE BASIS OF THE SUIT;**

11. **The capacity and use of any database or system to log reports or exports generated by, or queries run against, the database or system where such reports, exports or queries may bear on the PROVIDE APPROPRIATE LANGUAGE RE THE ACTIVITIES PERTINENT TO THE MATTERS MADE THE BASIS OF THE SUIT;**

12. **The identity and roles of current or former employees or contractors serving as database or system administrators for databases or systems that may reasonably be expected to contain (or have contained) information relating to the PROVIDE APPROPRIATE LANGUAGE RE THE ACTIVITIES PERTINENT TO THE MATTERS MADE THE BASIS OF THE SUIT; and**

13. **The cost, burden, complexity, facility and ease with which the information within databases and systems holding potentially responsive data relating to the PROVIDE APPROPRIATE LANGUAGE RE THE ACTIVITIES PERTINENT TO THE MATTERS MADE THE BASIS OF THE SUIT; may be identified, preserved, searched, extracted and produced in a manner that faithfully reflects the content, integrity and functionality of the source data.**

Yes, this is the dread "discovery about discovery;" but, it's a necessary precursor to devising query and production strategies for databases. If you don't know what the database holds or the ways in which relevant and responsive data can be extracted, you are at the mercy of opponents who will give you data in unusable forms or give you nothing at all.

Remember, these are not magic words. I just made them up, and there's plenty of room for improvement. If you borrow this language, please take time to understand it, and particularly strive to know *why* you are asking for what you demand. Supplying the information requires effort that should be expended in support of a genuine and articulable need for the information. If you don't need the information or know what you plan to do with it, don't ask for it.

These few questions were geared to the feasibility of extracting data from databases so that it stays utile and complete. Enterprise databases support a raft of standardized reporting capabilities:

"screens" or "reports" run to support routine business processes and decision making. An insurance carrier may call a particular report the "Claims File;" but, it is not a discrete "file" at all. It's a predefined template or report that presents a collection of data extracted from the database in a consistent way. Lots of what we think of as sites or documents are really reports from databases. Your Facebook page? It's a report. Your e-mail from Microsoft Outlook? Also, a report.

In addition to supplying a range of standard reports, enterprise databases can be queried using enhanced reporting capabilities ("custom reports") and using overlay reporting tools–commercial software "sold separately" and able to interrogate the database in order to produce specialized reporting or support data analytics. A simple example is presentation software that generates handsome charts and graphics based on data in the database. The presentation software didn't come with the database. It's something they bought (or built) to "bolt on" for enhanced/overlay reporting.

Although databases are queried using a "query language," users needn't dirty their hands with query languages because queries are often executed "under the hood" by the use of those aforementioned standardized screens, reports and templates. Think of these as pre-programmed, pushbutton queries. There is usually more (and often much more) that can be gleaned from a database than what the standardized reports supply, and some of this goes to the integrity of the data itself. In that case, understanding the query language is key to fashioning a query that extracts what you need to know, both *within* the data and *about* the data.

As importantly as learning what the database can produce is understanding what the database does or does not display to end users. These are the *user experience* (UX) and *user interface* (UI). Screen shots may be worth a thousand words when it comes to understanding what the user saw or what the user might have done to pursue further intelligence.

Enterprise and commercial databases tend to be big and expensive. Accordingly, most are well documented in manuals designed for administrators and end users. When a producing party objects that running a query is burdensome, the manuals may make clear that what you seek is no big deal to obtain.

One feature that sets databases apart from many others forms of ESI is the critical importance of the fielding of data. **Preserving the fielded character of data is essential to preserving its utility and searchability.** "Fielding data" means that information is stored in locations dedicated to holding just that information. Fielding data serves to separate and identify information so you can search, sort and cull using just that information. It's a capability we take for granted in databases but that is often crippled or eradicated when data is produced in e-discovery. **Be sure that you consider the form of production and ensure that the fielded character of the data produced will not be lost, whether supplied as a standard report or as a delimited export.**

Fielding data isn't new. We did it back when data was stored as paper documents. Take a typical law firm letter: the letterhead identifies the firm, the date below the letterhead is understood to be the date sent. A *Re:* line follows, denoting matter or subject, then the addressee, salutation, etc. The recipient is understood to be named at the start of the letter and the sender at the bottom. These conventions governing where to place information are vital to our ability to understand and organize conventional correspondence.

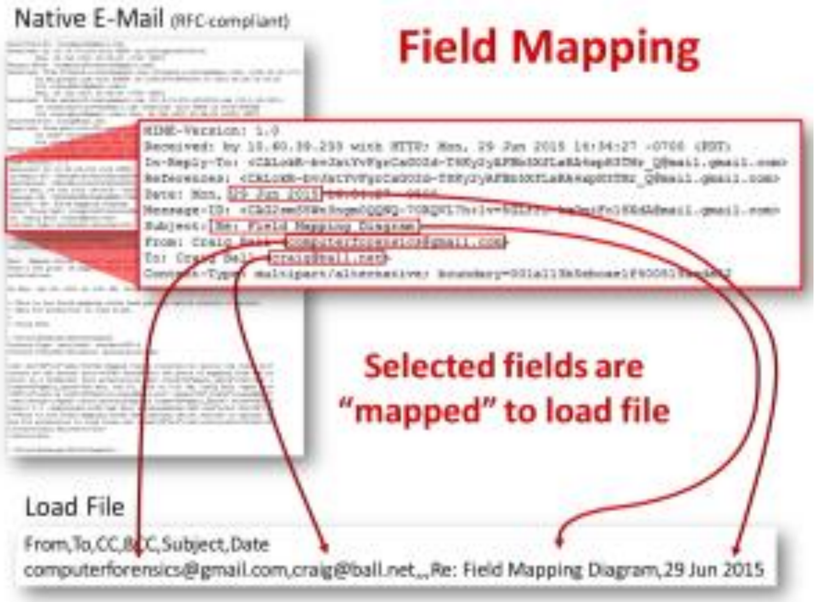
Similarly, all the common productivity file types encountered in e-discovery (Microsoft Office formats, PDF and e-mail) employ fielding to abet utility and functionality. Native "documents" are natively fielded; that is, a file's content is structured to ensure that particular pieces of information reside in defined locations within the file. This structure is understood and exploited by the native application and by tools designed to avail themselves of the file architecture.

We act inconsistently, inefficiently and irrationally when we deal with fielded information in e-discovery. In contrast to just a few years ago, only the most Neanderthal counsel now challenges the need to produce the native fielding of spreadsheet data. Accordingly, production of spreadsheets in native forms has evolved to become routine and (largely) uncontentious. To get to this point, workflows were modified, Bates numbering procedures were tweaked, and despite dire predictions, none of it made the sky fall. We can and must do the same with PowerPoint presentations and Word documents.

"*What's vice today may be virtue tomorrow*," wrote novelist (and jurist) Henry Fielding.

Now, take e-mail. All e-mail is natively fielded data, and the architecture of e-mail messages is established by published standards called RFCs—structural conventions that e-mail applications and systems must embrace to ensure that messages can traverse any server. The RFCs define placement and labeling of the sender, recipients, subject, date, attachments, routing, message body and other components of every e-mail that transits the Internet.

But when we produce e-mail in discovery, the "accepted" practice is to deconstruct each message and produce it in a cruder fielded format that's incompatible with the RFCs and unrecognizable to any e-mail tool or system. Too, the production is almost always incomplete compared to the native content.


Native E-Mail (RFC compliant)
Field Mapping
MIME-Version: 1.0
Received: by 10.60.30.233 with HTTP; Mon, 29 Jun 2015 14:34:27 -0700 (PDT)
In-Reply-To:
References:
Date: Mon, 29 Jun 2015
Message-ID:
Subject: Re: Field Mapping Diagram
From: Craig Ball computerforensics@gmail.com
To: Craig Ball craig@ball.net
Content-Type: multipart/alternative;
Selected fields are "mapped" to load file
Load File
From,To,CC,BCC,Subject,Date
computerforensics@gmail.com,craig@ball.net,,,Re: Field Mapping Diagram,29 Jun 2015

The deconstruction of fielded data is accomplished by a process called **Field Mapping**. The contents of fields within the native source are extracted and inserted into a matrix that may assign the same name to the field as accorded by the native application or rename it to something else altogether. Thus, the source data is “mapped’ to a new name and location. At all events, the mapped fields never mirror the field structure of the source file.

Ever? No, *never.*

The jumbled fielding doesn’t entirely destroy the ability to search within fields or cull and sort by fielded content; but it requires lawyers to rent or buy tools that can re-assemble and read the restructured data in order to search, sort and review the content. And again, information in the original is often omitted, not because it’s privileged or sensitive, but because…well, um, er, *we just do it that way, dammit*!

But the information that’s omitted, surely that’s useless metadata, right?

Interestingly, no. In fact, the omitted information significantly aids our ability to make sense of the production, such as the fielded data that allows messages to be organized into conversational threads (*e.g.*, In-Reply-To, References and Message-ID fields) and the fielded data that enables messages to be correctly ordered across time zones and daylight savings time (*e.g.,* UTC offsets).

“Why do producing parties get to recast and omit this useful information,” you ask? The industry responds: "*These are not the droids you’re looking for.*" "*Hey, is that Elvis?*" "*No Sedona for you!*"

The real answer is that counsel, and especially requesting counsel, are asleep at the wheel. Producing parties have been getting away with this nonsense, unchallenged, for so long, they’ve come to view it as a birthright. But reform is coming, at the glacial pace for which we lawyers are justly reviled, I mean *revered*.

E-discovery standards have indeed evolved to acknowledge that e-mail must be supplied with some fielding preserved; but there is no sound reason to produce e-mail with shuffled or omitted fields. It doesn't cost more to be faithful to the native or near-native architecture or be complete in supplying fielded content; in fact, producing parties pay *more* to degrade the production, and what emerges costs more to review.

Perhaps the hardest thing for lawyers and judges to appreciate is the importance fielding plays in culling, sorting and search.

- It’s efficient to be able to cull and sort files *only* by certain dates.
- It’s efficient to be able to search *only* within e-mail recipients.
- It’s efficient to be able to distinguish Speaker Notes within a PowerPoint or filter by the Author field in a Word document.

Preserving the fielded character of data makes that possible. Preserving the fielded data *and* the native file architecture allows use of a broad array of tools against the data, where restructuring fielded data limits its use to only a handful of pricey tools that understand peculiar and proprietary production formats.

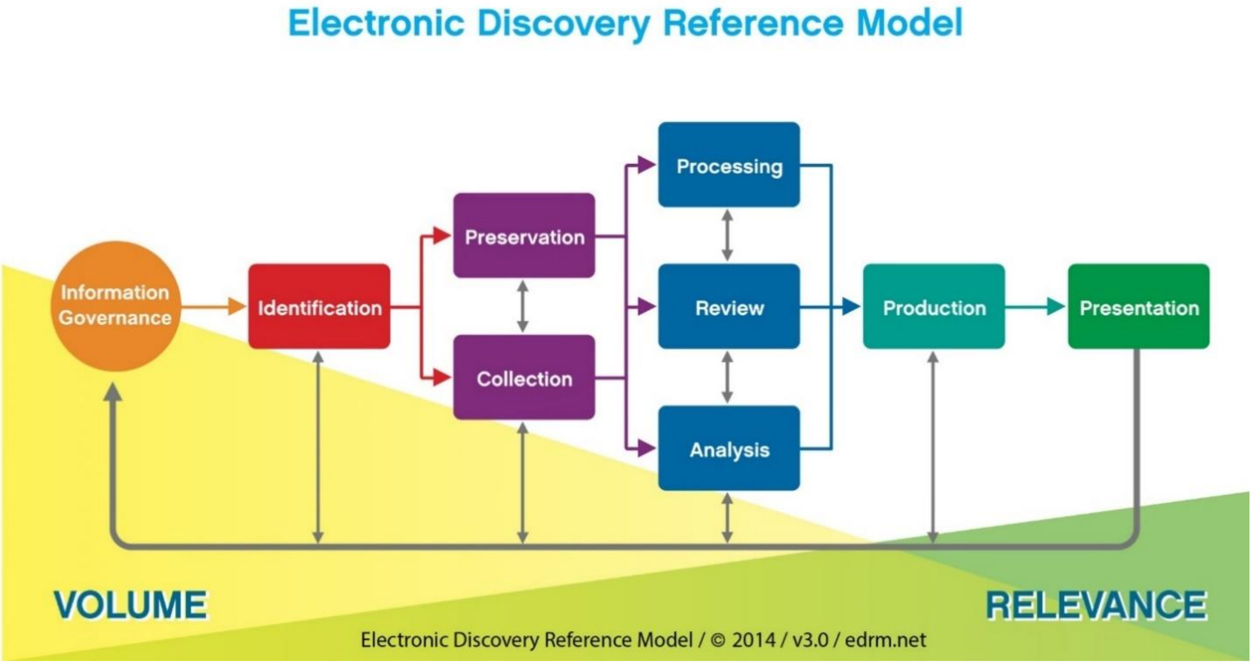
It’s not enough for producing parties to respond, “*But, you can reassemble the kit of data we produce to make it work somewhat like the original evidence*.” In truth, you often can't, and you shouldn't have to try.

It ties back to the Typewriter Generation mentality that keeps us thinking about “documents” and seeking to define everything we seek as a "document." Most information sought in discovery today is not a purposeful precursor to something that will be printed. Most modern evidence is data, *fielded* data. Modern productivity files aren’t blobs of text, they're ingenious little *databases*. *Powerful. Rich. Databases.* Their native content and architecture are key to their utility and efficient searchability in discovery. Get the fielding right, and functionality follows.

Seeking discovery from databases is a key capability in modern litigation, and it’s not easy for the technically challenged (although it’s probably a whole lot easier than your opponent claims). Getting the proper data in usable forms demands careful thought, tenacity and more-than-a-little homework. Still, anyone can do it, alone with a modicum of effort, or aided by a little expert assistance.

The Next Arc of Exercises

Let’s return to the EDRM:



Young lawyers serving large firms tend not to see electronic evidence until tasked with review, and then what they see has undergone a host of transformative operations. All that comes before—identification, preservation, collection and processing—is the province of e-discovery experts, in-house personnel and litigation support specialists. Potentially-relevant evidence seems to miraculously appear before lawyers’ eyes, but the reality is that dozens or hundreds of decisions and processes determine if relevant evidence sees the light of day or if the case devolves into costly and frustrating discovery disputes. To competently defend those decisions and processes, lawyers must understand them. So, the workbook exercises entail tasks like forensic imaging of evidence media and processing and producing ESI, that lawyers seldom *do* but are often called upon to *explain* and *defend* in court.

Vital Vocabulary

Chain of Custody
Targeted Collection
Forensic Imaging
Active Data Areas
Unallocated Clusters
Slack Space
Public Cloud

Identification, Preservation and Collection

We’ve looked at systems and media that house ESI and data and metadata from those sources. *ESI is dynamic.* Messages come and go via automated processes. Social networking sites update ceaselessly. Logs overwrite cyclically. Mobile devices push and pull data 24/7. Corporate mail systems typically purge the contents of users’ inboxes every 90-120 days unless the user or the IT department interdict that deletion. *Data is always in flux and in motion.* So, doing nothing, or doing something too slowly, is calculated to cause the loss of information.

You can't act to preserve ESI you don't know exists, and you certainly can't collect ESI you haven't preserved if it's lost or altered by the time you go looking for it. Too, collecting ESI for discovery and as evidence demands more exacting processes than we use to guard against hard drive failure. Demonstrating the authenticity of ESI as evidence entails protecting the **integrity of the evidenc**e–data *and* metadata--and proving a proper **chain of custody**. No ESI should be produced in discovery or offered into evidence absent the ability to trace it back to its origins and demonstrate it wasn't changed, or if changed, how and why?

Lawyers collect ESI in discovery to support processing, search, review and production. It's expensive and challenging. In a perfect world, we wouldn't need to collect ESI because the systems housing the evidence would fully support forensic preservation, search and review. Perhaps not surprisingly, supporting the needs of litigation doesn't drive software development; so currently, few information systems do.

Accordingly, the e-discovery process entails *collecting* ESI, and collection tends to employ two techniques: **targeted collection** and **forensic imaging**. Targeted collection is the identification and duplication of potentially relevant ESI according to specific characteristics of the files and folders in which it resides. The collection from Madison's Windows laptop of all Word documents and Excel spreadsheets in the Documents folder having last modified dates between 1/1/2020 and 12/31/2021 is an example of a targeted collection.

Forensic imaging entails the duplication of the *complete* contents of a storage medium, typically encompassing the readily-accessible **active data areas** and the inaccessible, forensically-significant regions of the medium like **unallocated clusters** and **slack space**.

Targeted collection and forensic imaging each has its proponents. Settling upon the optimum method for collection entails balancing pros and cons at a point in litigation when much remains uncertain respecting the facts and issues in the case.

Targeted collection tends to reduce data volumes, with a commensurate reduction in costs to process and host ESI. Decreased volume also means less data to search and review, prompting greater savings. That's a big plus.

But the savings sought from targeted collection must be weighed against the risk of leaving relevant and responsive ESI behind and the expenditures required to scope and carry out targeted collection. Someone must *choose* what will file types, intervals and locations are targeted in a targeted collection; forensic imaging requires only the identification of sources to be imaged.

If it turns out that a file type or location was missed in targeted collection, or the date interval of the files collected proved inadequate, targeted collection prompts a costly do-over--assuming sources and data haven't been lost or changed over time.

Forensic imaging ensures that all the content on a source remains available, intact and at-hand. If issues of spoliation crop up, forensic imaging supports forensic analysis where targeted collection often does not. Notwithstanding it's belt-and-suspenders advantages, forensic imaging entails the use of specialized tools and skills that carry their own costs and consequences. Forensically preserving the full contents of storage media doesn't obviate the need to separate wheat from chaff; it defers that effort. When more data means more money expended, forensic imaging often proves the superior means of preservation but not the most cost-effective approach to collection.

The cost-benefit equation changes when the data to be preserved and collected resides in the **Public Cloud** (*i.e.,* on servers shared using the public Internet and managed by third-party service providers like Amazon Web Services and Microsoft's Azure). Forensic artifacts like deleted files in unallocated clusters don't exist in the public cloud, making targeted collection the only option. As well, public cloud services may support preservation-in-place by simply ticking a box in Settings (as can be done with email and SharePoint content in the SaaS application Microsoft 365).

In your exercises, you will forensically image the content of an evidence thumb drive and do a targeted collection of data as a precursor to ingesting and processing ESI in a commercial e-discovery tool. You'll cull and search an ESI collection then configure and generate a production set including load files and Bates numbering. In practice, these tasks are done for you. When things go wrong—as they often do—having done them, even just once, will better equip you to make things right.

❁ Exercise 12: Data Mapping

GOALS: The goals of this exercise are for the student to:

1. Consider the breadth and complexity of potentially discoverable information;
2. Appreciate the detailed metrics attendant to building a utile data map; and
3. Develop a data map of your data footprint.

VITAL VOCABULARY

Productivity Files
Information Governance
Possession/Custody/Control
Data Mapping
Trust But Verify
SharePoint

All of us live in a world that's rich with digital data streams. We leave countless electronic trails in our wake in the form of **electronically stored information** or **ESI**.[62] Our business work product (e.g., letters, reports, memos, financial reports and marketing material), manifests as discrete **productivity files**[63] which, when paired with electronic communications, *e.g.,* e-mail and other messaging, tends to account for the bulk of data preserved, pursued and produced in discovery.

Back when business work product took paper forms, standardized mechanisms like folders, drawers, cabinets and file rooms supported our ability to preserve and find information. As the physical organization of information waned and evolved with the shift to personal, network and mobile computing, information that once existed only on paper now takes many forms, in many iterations and as fragments splayed across many repositories and media.

The consequence of this sea change in **information governance**[64] has been that people and companies generally have a poor appreciation of the nature, quantity and form of the ESI in their

[62] As amended in 2006, Rules 26 and 34 of the Federal Rules of Civil Procedure (FRCP) employ the phrase "electronically stored information" but wisely do not define same in recognition of technology's ability to outpace law. The phrase supplants the prior use of "data compilations" and per the Rules' Comments "includes any type of information that is stored electronically." The *Guidelines For State Trial Courts Regarding Discovery of Electronically-Stored Information* defines electronically-stored information as "any information created, stored, or best utilized with computer technology of any type. It includes but is not limited to data; word-processing documents; spreadsheets; presentation documents; graphics; animations; images; e-mail and instant messages (including attachments); audio, video, and audiovisual recordings; voicemail stored on databases; networks; computers and computer systems; servers; archives; back-up or disaster recovery systems; discs, CD's, diskettes, drives, tapes, cartridges and other storage media; printers; the Internet; personal digital assistants; handheld wireless devices; cellular telephones; pagers; fax machines; and voicemail systems."

[63] The term "productivity files" refers to the common Microsoft Office application files most often seen in business settings (.doc or .docx Word documents, .xls or .xlsx Excel spreadsheets, .ppt or .pptx PowerPoint presentations; the "dotted" three- or four-letter references being the file extension) and Adobe .pdf Portable Document Files.

[64] The Gartner consulting firm defines Information Governance as "the specification of decision rights and an accountability framework to encourage desirable behavior in the valuation, creation, storage, use, archival and deletion of information. It includes the processes, roles, standards and metrics that ensure the effective and efficient use of information in enabling an organization to achieve its goals."

possession, custody or control.[65] A common thread in cases where courts have punished parties or counsel for e-discovery failures has been the failure of parties or counsel to know what ESI they had and in what forms, what **custodians[66]** held it, where they stored it and what risks of alteration or disposal affected the ESI.

Before you can preserve, review or produce ESI, you must first know what you have, where you have it, the forms it takes and how much of it you've got. The process by which parties and counsel build inventories of potentially relevant ESI is called **data mapping**.

Introduction to Data Mapping

"Data mapping" encompasses methods used to facilitate and memorialize the identification of ESI, an essential prerequisite to everything in the EDRM east of Information Governance.

Data mapping is an unfortunate moniker because it suggests the need to generate a graphical representation of ESI sources, leading many to assume a data map is synonymous with those Visio-style network diagrams Information Technology (IT) departments use to depict, *inter alia,* hardware deployments and IP addresses.

Unless created expressly for e-discovery, few companies have any diagram approaching what's required to serve as a sufficient data map for e-discovery. Neither network diagrams from IT nor retention schedules from Records and Information Management (RIM) are alone sufficient to serve as an EDD data map, but they contribute valuable information, useful clues to where relevant ESI resides.

Thus, a data "map" isn't often a map or diagram, though both are useful ways to organize the information. A data map is likely a list, table, spreadsheet or database. I tend to use Excel spreadsheets because it's easier to run totals. Corporations may use specialized software specifically designed to track a data-mapping and -preservation effort. A data map may also be a narrative. Whatever the form, clients rarely have a data map lying around. It's got to be constructed, often from scratch.

What your data map looks like matters less than the information it contains. Again, don't let the notion of a "map" mislead. The data map is as much about *what* as *where*. If the form chosen enables you to quickly and clearly access the information needed to implement defensible preservation, reliably project burden, guide collection and accurately answer questions at both the meet and confer and in court, then it's the right form, even if it isn't a pretty picture.

[65] FRCP 26(a)(1)(A), and sometimes articulated as *care*, custody or control

[66] Though the term "records custodian" is customarily defined as the person responsible for, or the person with administrative control over, granting access to an organization's documents or electronic files while protecting the data as defined by the organization's security policy or its standard IT practices, the term tends to be accorded a less precise definition in e-discovery and is best thought of as anyone with possession, custody or control of ESI, including a legal right or *practical ability* to access same. *See, e.g., In re NTL, Inc. Securities Litigation*, 244 F.R.D. 179, 195 (S.D.N.Y. 2007).

Scope
The duty to identify ESI is the most encompassing obligation in e-discovery. Think about it: You can't act to preserve sources you haven't found. You certainly can't collect, review or produce them. The Federal Rules of Civil Procedure expressly impose a duty to identify all potentially responsive sources of information deemed "not reasonably accessible" that won't be searched; so even if you won't search potentially responsive ESI, you're bound to identify it.

A "data map" might be better termed an "Information Inventory." It's akin to the inventories that retail merchants undertake to know what's on their shelves by description, quantity, location and value.

Creating a competent data map is also akin to compiling a history of:

- Human resources and careers (after all, cases are still mostly about people);
- Information systems and their evolution; and
- Projects, facilities and tools.

A data map spans both logical and physical sources of information. Bob's e-mail is a logical collection that may span multiple physical media. Bob's hard drive is a physical collection that may hold multiple logical sources. Logical and physical sources may overlap, but they are rarely the same.

As needed, a data map might encompass:

- **Custodian and/or source of information**;
- **Location**;
- **Physical device or medium**;
- **Currency of contents**;
- **Volume** (e.g., in bytes);
- **Numerosity** (e.g., how many messages and attachments?)
- **Time span** (including intervals and significant gaps)
- **Purpose** (How is the ESI resource tasked?);
- **Usage** (Who uses the resource and when?);
- **Form**; and
- **Fragility** (What are the risks it may go away?).

This isn't an exhaustive list because the facets change with the nature of the sources inventoried. To wit, you map different data for e-mail than for databases.

A data map isn't a mindless exercise in minutiae. The level of detail must conform to the likely relevance and materiality of the information; so, you must adapt the inventory to the issues in the case.

Tips for Better Data Mapping

Custodial interviews (i.e., questioning persons who hold data) are an essential component of a sound data map methodology; but, custodial interviews are an unreliable (and occasionally even counterproductive) facet of data mapping, too. Custodians will know a lot about their data that will be hard to ferret out except by questioning them. Custodians will not know (or will misstate) a lot about their data leaving gaps to be filled though, e.g., search and sampling.

Do not become so wedded to a checklist when conducting custodial interviews that you fail to listen closely and use common sense. When a custodian claims they have no thumb drives or web mail accounts, don't just move on. *It's just not so.* When a custodian claims they've never used a home computer for work, don't believe it without eliciting a reason to trust their statement. Remember: custodians want you *out of their stuff and out of their hair.* Even those acting in complete good faith will say what promotes that end. Trust, *but verify*.

Don't be so intent on minimizing sources that you foster reticence. If you really want to find ESI, use open-ended language that elicits candor." Avoid leading questions like, "*You didn't take any confidential company data home in violation of policy, did you*?" That's unlikely to elicit, "*Sure, I did!*" Offer an incentive to disclose; "*It would really help us if you had your e-mail from 2018*".

Legacy hardware and media grow invisible, even when right under your nose. A custodian no longer sees the old CPU in the corner. The IT tech no longer sees the box under the desk filled with backup tapes. You must bring Proustian "new eyes" to the effort, and not be reluctant to say, "What's in there?" or "Let me see please." Don't be blind leading the blind!

> **The real voyage of discovery consists not in seeking new landscapes, but in having new eyes.**
>
> **Marcel Proust**

Companies don't buy costly systems and software and expense it. They must amortize the cost over time and maintain amortization and depreciation schedules. Accordingly, the accounting department's records can be a ready means to identify systems, mobile devices and even pricey software applications that are paths to ESI sources.

Three Pressing Points to Ponder

- **Accountability is key every step of the way.** If someone says, "that's gone," be sure to note who made the representation and test its accuracy. Get their skin in the game. Ultimately, building the data map needs to be one person's hands on, "buck stops here" responsibility, and that person needs to give a hot damn about the quality of their work. Make it a boots-on-the-ground duty devolving on someone with the ***ability, curiosity, diligence and access*** to get the job done.

- **Where you start matters less than when and with whom.** Don't dither! Dive in the deep end! Go right to the über key custodians and start digging. Get eyes on offices, storerooms, closets, servers and C: drives. Go where the evidence leads!

- **Just because your data map can't be perfect doesn't mean it can't be great.** Don't fall into the trap of thinking that, because no data mapping effort can be truly complete and current, the quality of the data map doesn't matter. Effective data mapping is the bedrock on which any sound e-discovery effort is built.

Quest for E-Discovery: Creating a Corporate Data Map

Adapted from the article, *Quest for E-Discovery: Creating a Data Map*, by Ganesh Vednere, Manager with Capgemini Financial Services in New York

1. **Get a list of all systems – and be prepared for a few surprises.** Begin the process by creating a list of all systems that exist in the company. This is easier said than done, as in many cases, IT does not even have a full list of all systems. Sure, they usually have a list of systems, but don't take that as the final list! Due diligence involves talking to business process owners, employees, and contractors, which often brings to light hidden systems, utilities, and home-grown applications that were unbeknownst to IT. Ensure that all types of systems are covered, e.g. physical servers, virtual servers, networks, externally hosted systems, backups (including tapes), archival systems, and desktops, etc. Pay special attention to emails, instant messaging, core business systems, collaboration software, and file shares, etc.

2. **Document system information.** After the list of all systems is known, gather as much information about each as possible. This exercise can be performed with the help of system infrastructure teams, application support teams, development teams, and business teams. Here are some types of information that can be gathered: system name, description, owner, platform type, location; is it a home grown-package, and does it store both structured and unstructured data; system dependencies (i.e., what systems are dependent on it and what systems does it depend on); business processes supported, business criticality of the system, security and access controls, format of data stored, format of data produced, reporting capabilities, how/where the system is hosted; backup process and schedule, archival process and schedule, whether data is purged or not; if purged, how often and what data gets purged; how many users, is there external access allowed (outside of the company firewall), are retention policies applied, what are the audit-trail capabilities, what is the nature of data stored, e.g. confidential data, nonpublic personal information, or still others.

3. **Get a list of business processes.** Inventory the list of business processes and map it to the system list obtained in the step above to ensure that all the distinct types of ESI are documented. The list of business processes is also useful during the discovery process, when

one can leverage the list to hone in on a particular type of ESI and obtain information about how it was generated, who owned the data, how the data was processed, how it was stored, and so on. A list of business processes can also be useful when assessing information flows.

4. **Develop a list of roles, groups, and users (custodians).** Obtain the organizational chart and determine the roles and groups across the business and the business processes. Document the process custodians and map out who had privileges to do what. Understand the human actors in the information lifecycle flow.

5. **Document the information flow across the entire organization.** Determine where critical pieces of information got initiated, how the information was/is manipulated, what systems touch the information, who processes the information, what systems depend on the information, and so on. Understanding the flow of information is key to the data mapping/discovery process.

6. **Determine how email is stored, processed, and consumed.** Given the sizable percentage of business information and business records that reside in email, special attention needs to be placed on email ESI. Typically, email is the first thing that opposing counsel go after, so determining whether email retention and disposition policies are consistently enforced will be key to proving good faith. There are many automated tools that will enable you to create email maps, link threads of conversation, heuristically perform relevancy search, extract underlying metadata, and so on. Before deciding to buy the best-of-breed solution, however, perform due diligence on existing email processes. Understand how employees are using email. Are they creating local archives (.PST files), are they storing emails on a network or a cloud repository, are they disposing of them at the end of retention periods, are they using personal emails to conduct official business, and so on? Identify deficiencies and violations in email policies before the opposing counsel does.

7. **Identify use of collaboration tools.** SharePoint will have the lion's share of the collaboration space in many organizations, but even then, you must ensure that all other tools-–whether they are social networking tools (e.g., Slack), Web-based tools, or home-grown tools–-are included in the data-mapping process. You need to carefully document the types of information being stored on each of these tools. Sometimes company information has a nasty habit of being found in the most unlikely of places. Wherever possible work with compliance, information management, or records management groups to establish usage policies to prevent runaway viral growth of these tools. If the organization already has thousands of unmanaged SharePoint sites, work with IT and business to institute governance controls to prevent further runaway growth.

8. **Don't forget offsite storage.** After inventorying and mapping all systems, one would think the job is done. Alas, there is more work ahead. Offsite storage is an often-under-appreciated aspect of the discovery process. It is quite reasonable to assume that there might be substantial evidence stored offsite which might become incriminating later. Offsite

storage may contain boxes or tapes full of records whose existence was somehow never properly documented, with the result that they cannot be located unless someone opens the box or attempts to recover the tape data. These records continue to live well past their onsite cousins. This means the organization continues to have the record in backup tapes (or paper) and other formats that it purportedly claimed to have destroyed. The search for records in offsite storage is made more complicated if the offsite storage process did not create detailed indices about the contents. If there are tapes labeled “2018 Backup Y: Drive,” then it may become quite an arduous task to determine what information is really contained in those tapes. Nevertheless, the journey must be started. It could involve anything from a full-scale review of all tapes, followed by reclassifying and re-filing the tapes, to perhaps a review of just the offsite storage manifests. It could also involve a search for critical information or a clean-up of the last three years’ worth of tapes, and so on.

What a Data Map Should Look Like

The form and format of data maps differ widely by industry type, organizational size, geography, regulatory environment, business processes, and more. While each organization's data map may look different, there are several key elements essential to any good data map:

- **Looks Matter.** How the data map looks is key to its usability, relevance, and presentability. A good data map will be organized either functionally or hierarchically with various data points organized around key subject lines. Typically, it would consist of rows of data with columns of attributes for each data set. The size of the map is entirely dependent upon the organization, but at a minimum, each one should contain information about people, process and systems.

- **A format that supports change.** Data maps are subject to frequent change and thus choosing a format that allows updates to be made in a painless manner is critical. In the initial stages significant volumes of data need to be entered, so start with a format that supports quick data entry, such as Excel, and subsequently migrate to a longer-term format that supports searching, reporting, and quick retrieval, such as a database. Do not overcomplicate either the form or the format. Bottom line: "Keep it Simple."

- **Emphasize the quality of content.** Data map designers tend to "over engineer" the document and set themselves up for a process that involves gathering numerous data values for each entry in the map. Instead, by honing in on only those columns that truly add value to the document, the process of collecting, collating and organizing the information for it becomes more manageable. For each column in the data map, collect as much accurate information as possible. For the "location" column, for instance, enumerate both primary and secondary locations, if there is one. A system may store the last 10 years of data online (primary storage location) with legacy data archived in a data archival system, tape, or offsite location. All locations should be reflected on the data map.

- **Access and Storage.** Data are typically considered a "record" under record retention rules and therefore all the requirements of good records management would apply. Unless explicitly prohibited, access to the data map can be granted to various groups and roles within an organization. The rationale is that the data map contains critical information that should be accessible broadly rather than available only to some individuals. Most of these individuals, however, would get "read-only" access to it. Accordingly, a view of the data map should be placed on a more widely accessible storage location while the data map itself can be controlled via the appropriate database or file system controls.

- **Maintaining the Data Map.** Ensuring that the data map stays accurate is vital to the relevance and long-term viability of it. A cross-functional team comprised of business, IT, and compliance that is sponsored by legal should be setup to maintain it. A data map administrator who performs the edits and controls access should also be established, and an appropriate chain of custody should be established such that when the data map administrator leaves the organization, the right handoffs take place. Data map updates should generally be done on an annual basis, but also in response to significant organizational events, as well as compliance and regulatory changes, or revamping of IT systems and processes. The update process should be a collaborative effort and not just a "do we have to do this" exercise.

- **Using the Data Map.** One would think that once created, the data map would be widely used and referenced by all departments for various purposes. Surprisingly, this is not always the case. The data map simply becomes a "checkbox" that gets relegated to a paralegal in the litigation group. Why isn't business, IT, or compliance using the data map, after all the time and effort spent creating it? The answer may lie in the perception that the document is only for e-discovery and not useful for day-to-day operations. While that may be partially true, the data map is indeed a lot more versatile and useful. It can be used for everything from IT portfolio rationalization to IT asset management and business process improvement. It is therefore incumbent upon the data map team to undertake suitable efforts and means to publicize, communicate, and demonstrate how it can be and is useful to various cross functions within the organization.

Exercise 12: Mapping your own Data

This is an important exercise. It's the first chance to gauge your level of thought and diligence in mapping your informational footprint. If you cannot fathom your own discoverable corpus of data, how can you assist clients to meet their obligation to identify and preserve their data? Students who have performed poorly on this exercise did so by overgeneralization and lack of exploration. In e-discovery, it's not enough to say the form of data is "electronic." Responding "unknown" respecting form or volume is insufficient if a little digging would prompt a more enlightened response.

Scenario:

You are the client in this scenario, and you've been sued in federal court. Your lawyer tells you the court has ordered all parties to preserve information, whether on paper or stored electronically. She instructs you to create a list of every source of ESI "*in your custody or possession or subject to your control*" where you've stored information, or others have stored information for you, in the *last four years* including every medium you've employed to regularly communicate in writing over the same time period. She adds, "*Don't forget phones and those thumb drive thingies; and be sure to include online stuff and mail, financial data and social networking and work-related data because the other side might subpoena that stuff from third-parties, like your bank and mobile phone company who may have it even if you don't have access anymore. I'm sure the other side will try to prove you missed something, so be very thorough.*"

Your lawyer explains that "reasonably accessible" information includes any information that you have in your custody as well as that which you routinely access or use, or that you could access and use. To make your job easier, your lawyer supplies a spreadsheet for your use in helping construct a data map. You protest that you should only have to deal with what's relevant and that's not clear from the claims; but your lawyer is adamant that you should NOT be selective in identifying the full reach of your digital footprint "D*o your best,*" she adds, "*but remember, this judge is pretty serious about e-discovery, and we don't want to lose the case because we failed to list something the other side might find out about later. Don't worry about deciding what's relevant or privileged, that's my job; but I need complete information on the spreadsheet.*"

Assignment: Complete the spreadsheet (data map) as your lawyer directed.

Please note:

1. This is the scenario. It's about you, not someone else or your current or prior employer. It's personal. You're not the lawyer here. You can't fire your lawyer or persuade her that, by your reading of the law, her request to you is overbroad and unduly burdensome. Neither can you respond that, without knowing what the case is about, you can't comply. *The uncertainty respecting scope and relevance is not an oversight here*. The nature of the request closely parallels the paucity of guidance and lack of restraint commonly seen in practice when lawyers frame legal hold instructions.

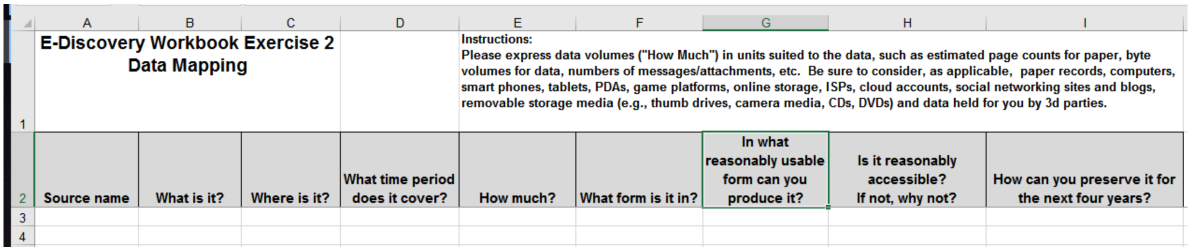
2. Once more, you are the client here. You will be bearing the cost of preservation and are spending your own hard-earned cash; so, consider how you should *balance* the obligation to preserve against the cost of your contemplated method. Unless you're wealthy and wasteful, don't assume that some expert will image everything for you. Also, remember that evidence can be both inculpatory *and* exculpatory. You may be barred from introducing information that helps you, if you fail to identify it in a timely way.

3. The duty to preserve encompasses more than what is in one's custody. It can include material that is subject to your control. So, consider whether you have access to data in another's custody (like your bank or Cloud service provider) or in the care or custody of a person or entity with whom you are in contractual privity or over which you have some practical control (like an attorney, CPA, doctor, family member or close friend).

4. The goal of the exercise is not to invade your privacy. You are mapping your own data because it's easiest for you. Mapping your own data doesn't require you to reach out to others as a lawyer must do. If identifying a genuine source seems too intrusive, feel free to change the name. That is, if you don't want to list that you have a Gmail account, you can call it something like "web mail account #1." If you don't want anyone to know you once used MySpace or Second Life, you can call them Social Networking Sites 1 and 2. X-Box Live can be "Online Gaming Community." Again, the purpose is not to intrude upon private matters but to promote your learning to map data sources accurately, *thoroughly* and in cogent ways that facilitate meeting obligations to identify, preserve, search and produce evidence in discovery.

5. Different sources demand different solutions; so, don't imagine that all sources can be defensibly preserved by pat solutions like, "I won't delete it" or "I'll have it forensically imaged." Some will. Some won't. Ponder options, consequences and cost. Also, "electronic" or "digital" isn't what we mean by "form" in e-discovery. What's the native form the data occupies? Do you even know? Can you find out with a little exploration?

6. An Excel spreadsheet may be downloaded from HTTP://craigball.com/Exercise_3_E-Discovery_Data_Mapping.xls. A cross-platform template is also available online via Google Docs for those who prefer to work that way: **http://tinyurl.com/datamap2.**



E-Discovery Workbook Exercise 2
Data Mapping

Instructions:
Please express data volumes ("How Much") in units suited to the data, such as estimated page counts for paper, byte volumes for data, numbers of messages/attachments, etc. Be sure to consider, as applicable, paper records, computers, smart phones, tablets, PDAs, game platforms, online storage, ISPs, cloud accounts, social networking sites and blogs, removable storage media (e.g., thumb drives, camera media, CDs, DVDs) and data held for you by 3d parties.

Source name	What is it?	Where is it?	What time period does it cover?	How much?	What form is it in?	In what reasonably usable form can you produce it?	Is it reasonably accessible? If not, why not?	How can you preserve it for the next four years?

7. The time required to complete the assignment will vary depending upon the number and variety of sources and your ability to ascertain the required metrics. *If it takes more than several hours, you're overdoing it.* If it takes under an hour or two, chances are you haven't

considered all sources or collected enough metrics. You need not consider paper document sources but instead concentrate your efforts on *electronically* stored information.

8. For this exercise, you are free to seek information from any source so long as you do not delegate the core effort to anyone else. You need not contact others (e.g., employers, schools, family) for specific metrics.

9. If you have questions, e-mail them to me at craig@ball.net.

10. As you work on this project, please reflect on the pervasiveness and variety of digital information; then, consider what might be required to data map a government agency or a corporation facing a class action or regulatory inquiry. Observe how little you may know about the nature and extent of data others hold for and about you. Further, be sensitive to any reluctance you feel about disclosing information and the thought and time required to marshal the data. How might such feelings in your clients and their employees impede a thorough and accurate data mapping effort? What strategies might attorneys employ to elicit information and prevent clients from falsely checking “none” on a questionnaire or furnishing incomplete data?

Mastering E-Mail in Discovery

Introduction

Get the e-mail! It's long been the war cry in e-discovery. It's a recognition of e-mail's enduring importance and ubiquity. We go after e-mail because it accounts for the majority of business communications and because, despite years of cautions and countless headlines tied to e-mail improvidence, e-mail users still let their guards down and reveal plainspoken truths they'd never put in a memo.

If you're on the producing end of a discovery request, you not only worry about what the messages say, but also whether you and your client can find, preserve and produce all responsive items. Questions like these *should* keep you up nights:

- Will the client simply conceal damning messages, leaving counsel at the mercy of an angry judge or disciplinary board?
- Will employees seek to rewrite history by deleting "their" e-mail from company systems?
- Will the searches employed prove reliable and be directed to the right digital venues?
- Will review processes unwittingly betray privileged or confidential communications?

Meeting these challenges begins with understanding e-mail technology well enough to formulate a sound, defensible strategy. For requesting parties, it means grasping the technology well enough to assess the completeness and effectiveness of your opponent's e-discovery efforts.

Not Enough Eyeballs

Futurist Arthur C. Clarke said, "Any sufficiently advanced technology is indistinguishable from magic." E-mail, like television or refrigeration, is one of those magical technologies we use every day without really knowing how it works. "It's magic to me, your Honor," won't help you when the e-mail pulls a disappearing act. Judges expect you to pull that e-mail rabbit out of your hat.

A lawyer managing electronic discovery is obliged to do more than just tell their clients to "produce the e-mail." The lawyer must endeavor to understand the client's systems and procedures, as well as ask the right questions of the right personnel. Too, counsel must know when he or she isn't getting trustworthy answers. That's asking a lot, but virtually all business documents are born digitally and only a tiny fraction are ever printed.[67] Hundreds of billions of e-mails traverse the Internet *daily,* far more than telephone and postal traffic combined,[68] and the average business

[67] Extrapolating from a 2003 updated study compiled by faculty and students at the School of Information Management and Systems at the University of California at Berkeley. http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/ (visited 5/18/2013)

[68] http://www.radicati.com/wp/wp-content/uploads/2013/04/Email-Statistics-Report-2013-2017-Executive-Summary.pdf (visited 5/26/2016)

person sends and receives roughly 123 e-mails daily. And the e-mail volumes continue to grow even as texting and other communications channels have taken off.

Neither should we anticipate a significant decline in users' propensity to retain their e-mail. Here again, it's too easy and, at first blush, too cheap to expect users to selectively dispose of e-mail and still meet business, litigation hold and regulatory obligations. Our e-mail is so twisted up with our lives that to abandon it is to part with our personal history.

This relentless growth isn't happening in just one locale. E-mail lodges on servers, cell phones, laptops, home systems, thumb drives and in the cloud. Within the systems, applications and devices we use to store and access e-mail, most users and even most IT professionals don't know where messages lodge or exactly how long they hang around.

Test Your E.Q.

Suppose opposing counsel serves a preservation demand or secures an order compelling your client to preserve electronic messaging. Are you assured that your client can and will faithfully back up and preserve responsive data? Even if it's practicable to capture and set aside the current server e-mail stores of key custodians, are you *really* capturing all or even most of the discoverable communications? How much is falling outside your net, and how do you assess its importance?

Here are a dozen questions you should be able to confidently answer about your client's communication systems:

1. What messaging environment(s) does your client employ? Microsoft Exchange, IBM Domino, Office 365 or something else?
2. Do *all* discoverable electronic communications come in and leave via the company's e-mail server?
3. Is the e-mail system configured to support synchronization with local e-mail stores on laptops and desktops?
4. How long have the current e-mail client and server applications been used?
5. What are the message purge, retention, journaling and archival settings for each key custodian?
6. Can your client disable a specific custodian's ability to delete messages?
7. Does your client's backup or archival system capture e-mail stored on individual user's hard drives, including company-owned laptops?
8. Where are e-mail container files stored on laptops and desktops?
9. How should your client collect and preserve relevant web mail?
10. Do your clients' employees use home machines, personal e-mail addresses or browser-based e-mail services like Gmail for discoverable business communications?

11. Do your clients' employees use instant messaging on company computers or over company-owned networks?
12. Does your client permit employee-owned devices to access the network or e-mail system?

If you are troubled that you can't answer these questions, you should be; but know you're not alone. Despite decades of dealing with e-mail in discovery, most lawyers still can't. And if you're a lawyer, don't delude yourself that these are someone else's issues, *e.g.,* your litigation support people or IT expert. These are your issues when it comes to dealing with the other side and the court about the scope of e-discovery.

Staying Out of Trouble

Fortunately, the rules of discovery don't require you to do the impossible. All they require is diligence, reasonableness and good faith. To that end, you must be able to establish that you and your client acted swiftly, followed a sound plan, and took such action as reasonable minds would judge adequate to the task. It's also important to keep the lines of communication open with the opposing party and the court, seeking agreement with the former or the protection of the latter where fruitful. I'm fond of quoting Oliver Wendell Holmes' homily, "Even a dog knows the difference between being stumbled over and being kicked." Judges, too, have a keen ability to distinguish error from arrogance. There's no traction for sanctions when the failure to produce electronic evidence occurred despite good faith and due diligence.

…And You Could Make Spitballs with It, Too

Paper discovery enjoyed a self-limiting aspect because businesses tended to allocate paper records into files, folders and cabinets according to persons, topics, transactions or periods of time. The space occupied by paper and the high cost to create, manage and store paper records served as a constant impetus to cull and discard them, or even to avoid creating them in the first place. By contrast, the ephemeral character of electronic communications, the ease of and perceived lack of cost to create, duplicate and distribute them and the very low direct cost of data storage have facilitated a staggering and unprecedented growth in the creation and retention of electronic evidence. At 123 e-mails per day, a company employing 100,000 people could find itself storing almost 4*.5 billion* e-mails annually.

Did You Say *Billion*?

But volume is only part of the challenge. Unlike paper records, e-mail tends to be stored in massive data blobs. My e-mail comprises almost 25 gigabytes of data and contains over 100,000 messages, many with multiple attachments covering virtually every aspect of my life and many other people's lives, too. In thousands of those e-mails, the subject line bears only a passing connection to the contents as "Reply to" threads strayed further and further from the original topic. E-mails meander through disparate topics or, by absent-minded clicks of the "Forward" button, lodge in my inbox dragging with them, like toilet paper on a wet shoe, the unsolicited detritus of other people's business.

To respond to a discovery request for e-mail on a topic, I'd either need to skim/read a horrific number of messages or I'd have to naively rely on keyword search to flush out all responsive material. If the request for production implicated material I no longer kept on my current computer or web mail collections, I'd be forced to root around through a motley array of archival folders, old systems, obsolete disks, outgrown hard drives, ancient backup tapes (for which I currently have no tape reader) and unlabeled CDs. Ugh!

Net Full of Holes

I'm just one guy. What's a company to do when served with a request for "all e-mail" on a matter in litigation? Surely, I mused, someone must have found a better solution than repeating the tedious and time-consuming process of accessing individual e-mail servers at far-flung locations along with the local drives of all key players' computers?

In researching this text, I contacted colleagues in both large and small electronic discovery consulting groups, inquiring about "the better way" for enterprises, and was struck by the revelation that, if there was a better mousetrap, they hadn't discovered it either. Uniformly, we recognized such enterprise-wide efforts were gargantuan undertakings fraught with uncertainty and concluded that counsel must somehow seek to narrow the scope of the inquiry—either by data sampling, use of advanced analytics or through limiting discovery according to offices, regions, time span, business sectors or key players. Trying to capture *everything,* enterprise-wide, is trawling with a net full of holes.

New Tools

The market has responded in recent years with tools that either facilitate search of remote e-mail stores, including locally stored messages, from a central location (*i.e.,* enterprise search) or which agglomerate enterprise-wide collections of e-mail into a single, searchable repository (*i.e.,* e-mail archiving), often reducing the volume of stored data by so-called "single instance deduplication," rules-based journaling and other customizable features.

These tools, especially enterprise archival and advanced analytics termed "TAR" or "Predictive Coding," promise to make it easier, cheaper and faster to search and collect responsive e-mail, but they're costly and complex to implement. Neither established standards nor a leading product has emerged. Further, it remains to be seen whether the practical result of a serial litigant employing an e-mail archival system is that they—for all intents and purposes--end up keeping every message for every employee and becoming increasingly dependent upon fraught electronic search to cull wheat from chaff.

E-Mail Systems and Files

The “behind-the-firewall” corporate and government e-mail environment is dominated by two well-known, competitive product pairs: Microsoft Exchange Server and its Outlook e-mail client and IBM Lotus Domino server and its Lotus Notes client. A legacy environment called Novell GroupWise occupies a negligible third place, largely among government users.

Increasingly, corporate and government e-mail environment no longer live behind-the-firewall but are ensconced in the Cloud. Cloud products such as Google Apps and Microsoft Office 365 now account for an estimated 20-25% market shares, with Microsoft claiming that 4 out of 5 Fortune 500 companies use Office 365.

When one looks at personal and small office/home office business e-mail, it’s rare to encounter LOCAL server-based systems. Here, the market belongs to Internet service providers (*e.g.,* the major cable and telephone companies) and web mail providers (*e.g.,* Gmail and Yahoo! Mail). Users employ a variety of e-mail client applications, including Microsoft Outlook, Apple Mail and, of course, their web browsers and webmail. This motley crew and the enterprise behemoths are united by common e-mail *protocols* that allow messages and attachments to be seamlessly handed off between applications, providers, servers and devices.

Mail Protocols

Computer network specialists are always talking about this “protocol” and that “protocol.” Don’t let the geek-speak get in the way. An *application protocol* or API is a bit of computer code that facilitates communication between applications, *i.e.,* your e-mail client and a network like the Internet. When you send a snail mail letter, the U.S. Postal Service’s “protocol” dictates that you place the contents of your message in an envelope of certain dimensions, seal it, add a defined complement of address information and affix postage to the upper right-hand corner of the envelope adjacent to the addressee information. Only then can you transmit the letter through the Postal Service’s network of post offices, delivery vehicles and postal carriers. Omit the address, the envelope or the postage—or just fail to drop it in the mail—and Grandma gets no Hallmark this year! Likewise, computer networks rely upon protocols to facilitate the transmission of information. You invoke a protocol—*Hyper Text Transfer Protocol*—every time you type *http://* at the start of a web page address.

Incoming Mail: POP, IMAP, MAPI and HTTP E-Mail

Although Microsoft Exchange Server rules the roost in enterprise e-mail, it’s by no means the most common e-mail system for the individual and small business user. If you still access your personal e-mail from your own Internet Service Provider, chances are your e-mail comes to you from your ISP’s e-mail server in one of three ways: POP3, IMAP or HTTP, the last commonly called web- or browser-based e-mail. Understanding how these three protocols work—and differ—helps in identifying where e-mail can be found.

POP3 (Post Office Protocol, version 3) is the oldest and was once the most common of the three approaches and the one most familiar (by function, if not by name) to users of the Windows Mail, Outlook Express and Eudora e-mail clients. But, it's rare to see many people using POP3 e-mail today. Using POP3, you connect to a mail server, download copies of all messages and, unless you have configured your e-mail client to leave copies on the server, the e-mail is deleted on the server and now resides on the hard drive of the computer you used to pick up mail. Leaving copies of your e-mail on the server seems like a great idea as it allows you to have a backup if disaster strikes and facilitates easy access of your e-mail, repeatedly, from different computers. However, few ISPs afforded unlimited storage space on their servers for users' e-mail, so mailboxes quickly became "clogged" with old e-mails, and the servers started bouncing new messages. As a result, POP3 e-mail typically resides only on the local hard drive of the computer used to read the mail and on the backup system for the servers which transmitted, transported and delivered the messages. In short, POP is locally-stored e-mail that supports some server storage; but, again, this once dominant protocol is little used anymore.

IMAP (Internet Mail Access Protocol) functions in much the same fashion as most Microsoft Exchange Server installations in that, when you check your messages, your e-mail client downloads just the headers of e-mail it finds on the server and only retrieves the body of a message when you open it for reading. Else, the entire message stays in your account on the server. Unlike POP3, where e-mail is searched and organized into folders locally, IMAP e-mail is organized and searched on the server. Consequently, the server (and its backup tapes) retains not only the messages but also the way the user *structured* those messages for archival.

Since IMAP e-mail "lives" on the server, how does a user read and answer it without staying connected all the time? The answer is that IMAP e-mail clients afford users the ability to synchronize the server files with a local copy of the e-mail and folders. When an IMAP user reconnects to the server, local e-mail stores are updated (synchronized) and messages drafted offline are transmitted. So, to summarize, IMAP is server-stored e-mail, with support for synchronized local storage.

A notable distinction between POP3 and IMAP e-mail centers on where the "authoritative" collection resides. Because each protocol allows for messages to reside both locally ("downloaded") and on the server, it's common for there to be a difference between the local and server collections. Under POP3, the *local* collection is deemed authoritative whereas in IMAP the *server* collection is authoritative. But for e-discovery, the key point is that the contents of the local and server e-mail stores can and do *differ.*

MAPI (Messaging Application Programming Interface) is the e-mail protocol at the heart of Windows and Microsoft's Exchange Server applications. Simple MAPI comes preinstalled on Windows machines to provide basic messaging services. A more sophisticated version of MAPI

(Extended MAPI) is installed with Microsoft Outlook and Exchange. Like IMAP, MAPI e-mail is typically stored on the server and not necessarily on the client machine. The local machine may be configured to synchronize with the server mail stores and keep a copy of mail on the local hard drive (typically in a Personal Storage file with the extension .PST or an Offline Synchronization file with the extension .OST), but this is user- and client application-dependent. Though it's rare (especially for laptops) for there to be no local e-mail stores for a MAPI machine, it's nonetheless possible and companies have lately endeavored to do away with local e-mail storage on laptop and desktop computers. When machines are configured to bar creation of local PST and OST files, e-mail won't be found on the local hard drive except to the extent fragments may turn up through computer forensic examination.

HTTP (Hyper Text Transfer Protocol) mail, or web-based/browser-based e-mail, dispenses with the local e-mail client and handles all activities on the server, with users managing their e-mail using their Internet browser to view an interactive web page. Although most browser-based e-mail services support local POP3 or IMAP synchronization with an e-mail client, most users have no local record of their browser-based e-mail transactions except for messages they've affirmatively saved to disk or portions of e-mail web pages which happen to reside in the browser's cache (*e.g.,* Internet Explorer's Temporary Internet Files folder). Gmail and Yahoo! Mail are popular examples of browser-based e-mail services, although many ISPs (including all the national providers) offer browser-based e-mail access in addition to POP and IMAP connections.

The protocol used to carry e-mail is not especially important in electronic discovery except to the extent that it signals the most likely place where archived and orphaned e-mail can be found. Companies choose server-based e-mail systems (*e.g.,* IMAP and MAPI) for two principal reasons. First, such systems make it easier to access e-mail from various locations and machines. Second, it's easier to back up e-mail from a central location. Because IMAP and MAPI systems store e-mail on the server, the backup system used to protect server data can yield a mother lode of server e-mail.

Depending upon the backup procedures used, access to archived e-mail can prove a costly and time-consuming task or a relatively easy one. The enormous volume of e-mail residing on backup tapes and the potentially prohibitive cost to locate and restore that e-mail makes discovery of archived e-mail from backup tapes a major bone of contention between litigants. In fact, most reported cases addressing cost-allocation in e-discovery seem to have been spawned by disputes over e-mail on server backup tapes.

Outgoing Mail: SMTP and MTA

Just as the system that brings water into your home works in conjunction with a completely different system that carries wastewater away, the protocol that delivers e-mail to you is completely different from the one that transmits your e-mail. Everything discussed in the preceding paragraphs concerned the protocols used to *retrieve* e-mail from a mail server.

Yet another system altogether, called SMTP for *Simple Mail Transfer Protocol*, takes care of outgoing e-mail. SMTP is indeed a remarkably simple protocol and doesn't even require authentication, in much the same way as anyone can anonymously drop a letter into a mailbox. A server that uses SMTP to route e-mail over a network to its destination is called an MTA for *Message Transfer Agent*. Examples of MTAs you might hear mentioned by IT professionals include Sendmail, Exim, Qmail and Postfix. Microsoft Exchange Server is an MTA, too. In simplest terms, an MTA is the system that carries e-mail between e-mail servers and sees to it that the message gets to its destination. Each MTA reads the code of a message and determines if it is addressed to a user in its domain and, if not, passes the message on to the next MTA after adding a line of text to the message identifying the route to later recipients. If you've ever set up an e-mail client, you've probably had to type in the name of the servers handling your outgoing e-mail (perhaps *SMTP.yourISP.com*) and your incoming messages (perhaps *mail.yourISP.com* or *POP.yourISP.com*).

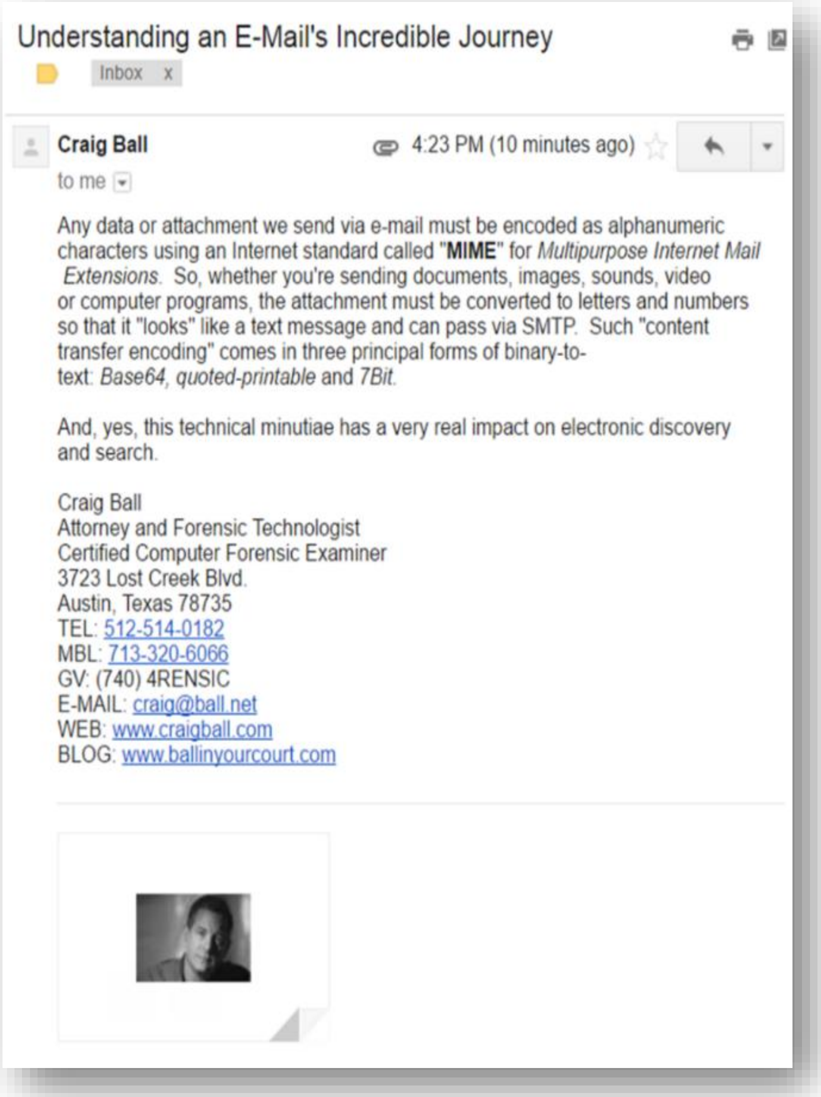
Anatomy of an E-Mail

Now that we've waded through the alphabet soup of protocols managing the movement of an e-mail message, let's look inside the message itself. Considering the complex systems on which it lives, an e-mail is astonishingly simple in structure. The Internet protocols governing e-mail transmission require electronic messages to adhere to rigid formatting, making individual e-mails easy to dissect and understand. The complexities and headaches associated with e-mail don't really attach until the e-mails are stored and assembled into databases and local stores.

An e-mail is just a plain text file. Though e-mail can be "tricked" into carrying non-text binary data like application files (*i.e.,* a Word document) or image attachments (*e.g.,* GIF or JPEG files), this piggybacking requires binary data be *encoded into text* for transmission. Consequently, even when transmitting files created in the densest computer code, *everything in an e-mail is plain text*.

E-Mail Autopsy: Tracing a Message's Incredible Journey

The image below left is an e-mail I sent to computerforensics@gmail.com from my alias craig@ball.net using my Gmail account craigball@gmail.com. A tiny JPG photograph was attached. A user might see the e-mail below left and mistakenly assume that what they see is all there is. Far from it! The image below right contains the source code of the e-mail.[69] Viewed in its "true" and complete format, it's too long to legibly appear on one page. So, let's dissect it by looking at its constituent parts: message header, message body and encoded attachment



Understanding an E-Mail's Incredible Journey

Craig Ball

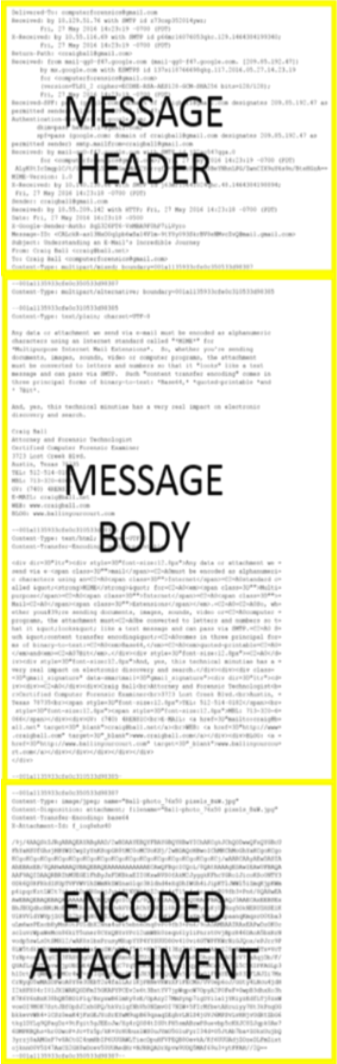
4:23 PM (10 minutes ago)

to me

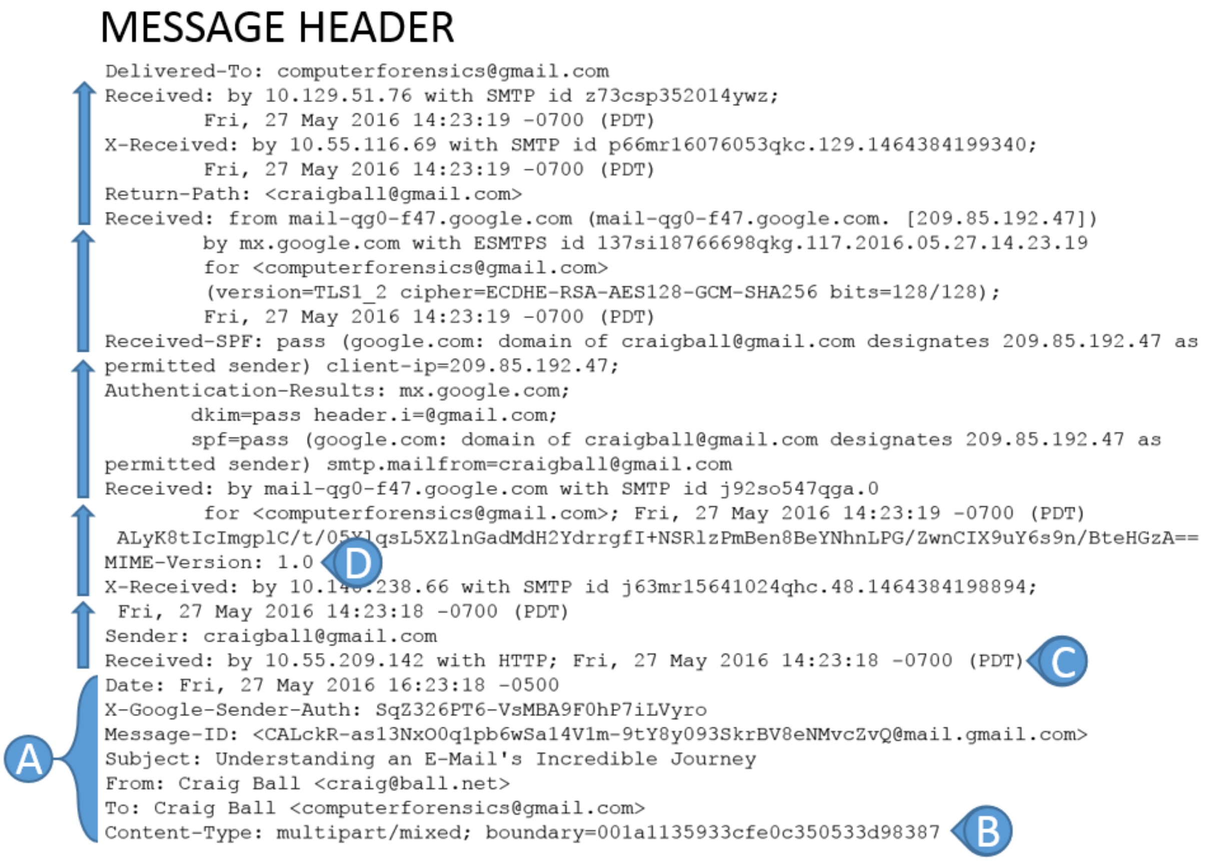
Any data or attachment we send via e-mail must be encoded as alphanumeric characters using an Internet standard called "**MIME**" for *Multipurpose Internet Mail Extensions*. So, whether you're sending documents, images, sounds, video or computer programs, the attachment must be converted to letters and numbers so that it "looks" like a text message and can pass via SMTP. Such "content transfer encoding" comes in three principal forms of binary-to-text: *Base64*, *quoted-printable* and *7Bit*.

And, yes, this technical minutiae has a very real impact on electronic discovery and search.

Craig Ball
Attorney and Forensic Technologist
Certified Computer Forensic Examiner
3723 Lost Creek Blvd.
Austin, Texas 78735
TEL: 512-514-0182
MBL: 713-320-6066
GV: (740) 4RENSIC
E-MAIL: craig@ball.net
WEB: www.craigball.com
BLOG: www.ballinyourcourt.com



[69] While viewing a Gmail message, you can display the source code for a message by selecting "Show original" from the message options drop-down menu. By default, Outlook makes only some encoded header content readily viewable at message Properties—the complete source code of incoming e-mail is not recorded absent a system Registry edit, which is not a casual operation!



MESSAGE HEADER

```
Delivered-To: computerforensics@gmail.com
Received: by 10.129.51.76 with SMTP id z73csp352014ywz;
        Fri, 27 May 2016 14:23:19 -0700 (PDT)
X-Received: by 10.55.116.69 with SMTP id p66mr16076053qkc.129.1464384199340;
        Fri, 27 May 2016 14:23:19 -0700 (PDT)
Return-Path: <craigball@gmail.com>
Received: from mail-qg0-f47.google.com (mail-qg0-f47.google.com. [209.85.192.47])
        by mx.google.com with ESMTPS id 137si18766698qkg.117.2016.05.27.14.23.19
        for <computerforensics@gmail.com>
        (version=TLS1_2 cipher=ECDHE-RSA-AES128-GCM-SHA256 bits=128/128);
        Fri, 27 May 2016 14:23:19 -0700 (PDT)
Received-SPF: pass (google.com: domain of craigball@gmail.com designates 209.85.192.47 as
permitted sender) client-ip=209.85.192.47;
Authentication-Results: mx.google.com;
       dkim=pass header.i=@gmail.com;
       spf=pass (google.com: domain of craigball@gmail.com designates 209.85.192.47 as
permitted sender) smtp.mailfrom=craigball@gmail.com
Received: by mail-qg0-f47.google.com with SMTP id j92so547qga.0
        for <computerforensics@gmail.com>; Fri, 27 May 2016 14:23:19 -0700 (PDT)
 ALyK8tIcImgplC/t/05Y1qsL5XZlnGadMdH2YdrrgfI+NSRlzPmBen8BeYNhnLPG/ZwnCIX9uY6s9n/BteHGzA==
MIME-Version: 1.0
X-Received: by 10.140.238.66 with SMTP id j63mr15641024qhc.48.1464384198894;
 Fri, 27 May 2016 14:23:18 -0700 (PDT)
Sender: craigball@gmail.com
Received: by 10.55.209.142 with HTTP; Fri, 27 May 2016 14:23:18 -0700 (PDT)
Date: Fri, 27 May 2016 16:23:18 -0500
X-Google-Sender-Auth: SqZ326PT6-VsMBA9F0hP7iLVyro
Message-ID: <CALckR-as13NxO0q1pb6wSa14V1m-9tY8y093SkrBV8eNMvcZvQ@mail.gmail.com>
Subject: Understanding an E-Mail's Incredible Journey
From: Craig Ball <craig@ball.net>
To: Craig Ball <computerforensics@gmail.com>
Content-Type: multipart/mixed; boundary=001a1135933cfe0c350533d98387
```

In an e-mail header, each line beginning with "Received" or X-Received" represents the transfer of the message between two e-mail servers. The transfer sequence is reversed chronologically such that those closest to the top of the header were inserted after those that follow, and the topmost line reflects delivery to the recipient's e-mail server and account, in this instance, *computerforensics@gmail.com*. As the message passes through intervening hosts, each adds its own identifying information along with the date and time of transit.

The area of the header labeled **(A)** contains the parts of the message designating the sender, addressee, date, time and subject line of the message. These are the only features of the header most recipients ever see. Note that the 24-hour message time has been recast as to a 12-hour format when shown in Gmail.

In the line labeled "Date," both the date and time of transmittal are indicated. The time indicated is 16:23:18, and the "-0500" which follows denotes the time difference between the sender's local time (the system time on my computer in New Orleans, Louisiana during daylight savings time) and Coordinated Universal Time (UTC), roughly equivalent to Greenwich Mean Time. As the offset from UTC was minus five hours on May 27, 2016, we deduce that the message was sent from a machine set to Central Daylight Time, giving some insight into the sender's location. Knowing the originating computer's time and time zone can occasionally prove useful in demonstrating fraud or fabrication.

E-mail must adhere to structural conventions. One of these is the use of a Content-Type declaration and setting of content boundaries, enabling systems to distinguish the message header region from the message body and attachment regions. The line labeled **(B)** advises that the message will be "multipart/mixed," indicating that there will be multiple constituents to the item (*i.e.,* header/message body/attachment), and that these will be encoded in different ways, hence "mixed." To prevent confusion of the boundary designator with message text, a complex sequence of characters is generated to serve as the content boundary. The first boundary, declared as "001a1135933cfe0c350533d98387," serves to separate the message header from the message body and attachment. It also signals the end of the message.

The message was created and sent using Gmail web interface; consequently, the first hop **(C)** indicates that the message was transmitted using HTTP and first received by IP (Internet Protocol) address 10.55.209.142 at 14:23:18 -0700 (PDT). Note that the server marks time in Pacific Daylight Time, suggesting it may be located on the west coast. The message is immediately handed off to another IP address 10.140.238.66 using Simple Mail Transfer Protocol, denoted by the initials SMTP. Next, we see another SMTP hand off to Google's server named "mail-qg0-f47.google.com" and so on until delivery to my account, *computerforensics@gmail.com*.

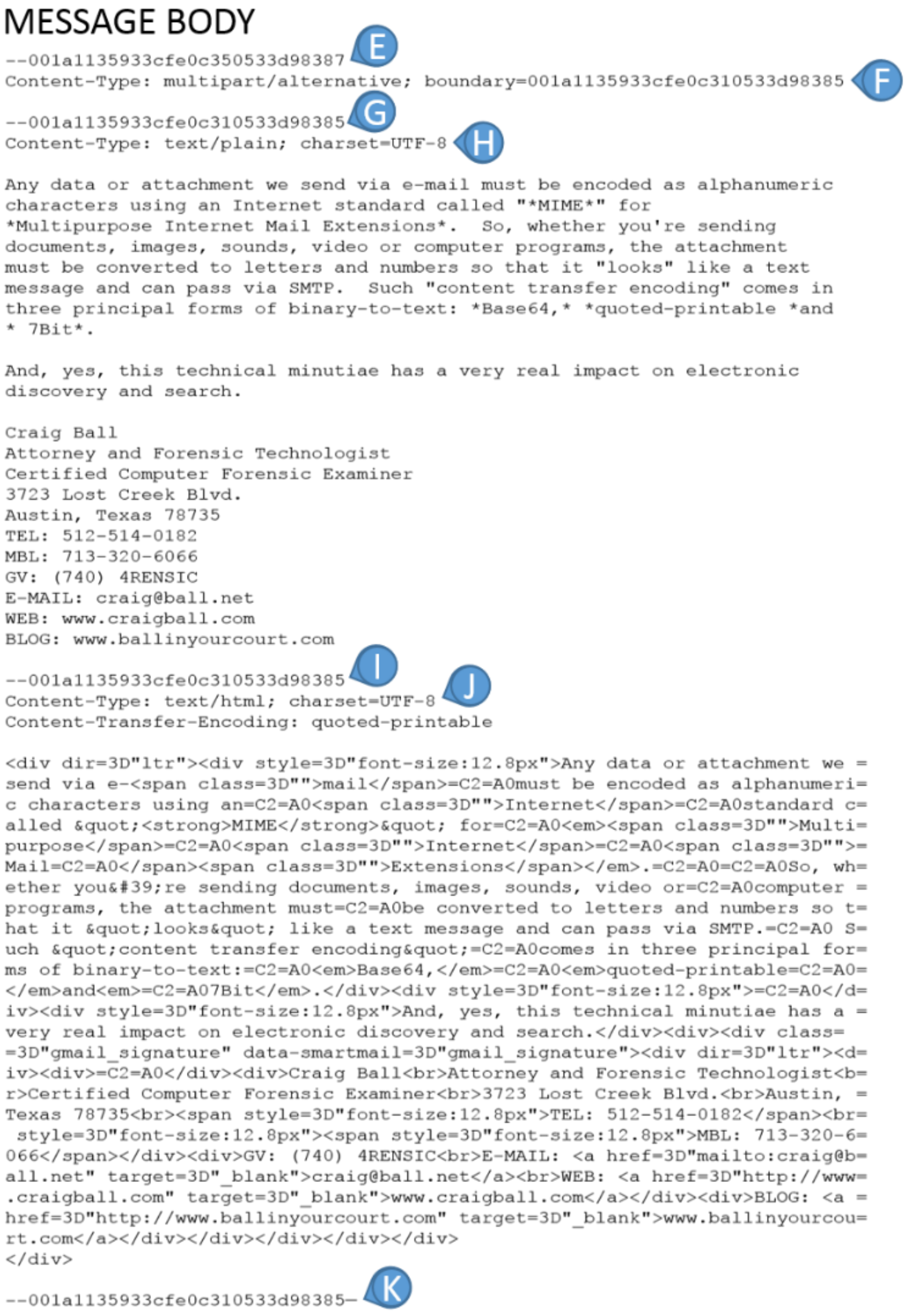
In the line labeled **(D),** the message header declares the message as being formatted in MIME (MIME-Version: 1.0).[70] Ironically, there is no other version of MIME than 1.0; consequently, trillions of e-mails have dedicated vast volumes of storage and bandwidth to this useless version declaration.

Proceeding to dissect the message body seen on the next page, at line **(E)**, we see our first boundary value (--001a1135933cfe0c350533d98387) serving to delineate the transition from header to message body. At line **(F)**, another Content-Type declaration advises that this segment of the message will be multipart/alternative (the alternatives being plain text or HTML) and a second boundary notation is declared as 001a1135933cfe0c350533d98385. Note that the first boundary ends in 387 and the second in 385. The second boundary is used at **(G)** to mark the start of the first alternative message body, declared as text/plain at line **(H)**.in plain text.

We then see the second boundary value used at line **(I)** to denote the start of the second alternative message body, and the Content-Type declared to be text/html at line **(J).** The second boundary

[70] MIME, which stands for Multipurpose Internet Mail Extensions, is a seminal Internet standard that supports non-US/ASCII character sets, non-text attachments (e.g., photos, video, sounds and machine code) and message bodies with multiple parts. Virtually all e-mail today is transmitted in MIME format.

notation is then used to signal the conclusion of the multipart/alternative content.

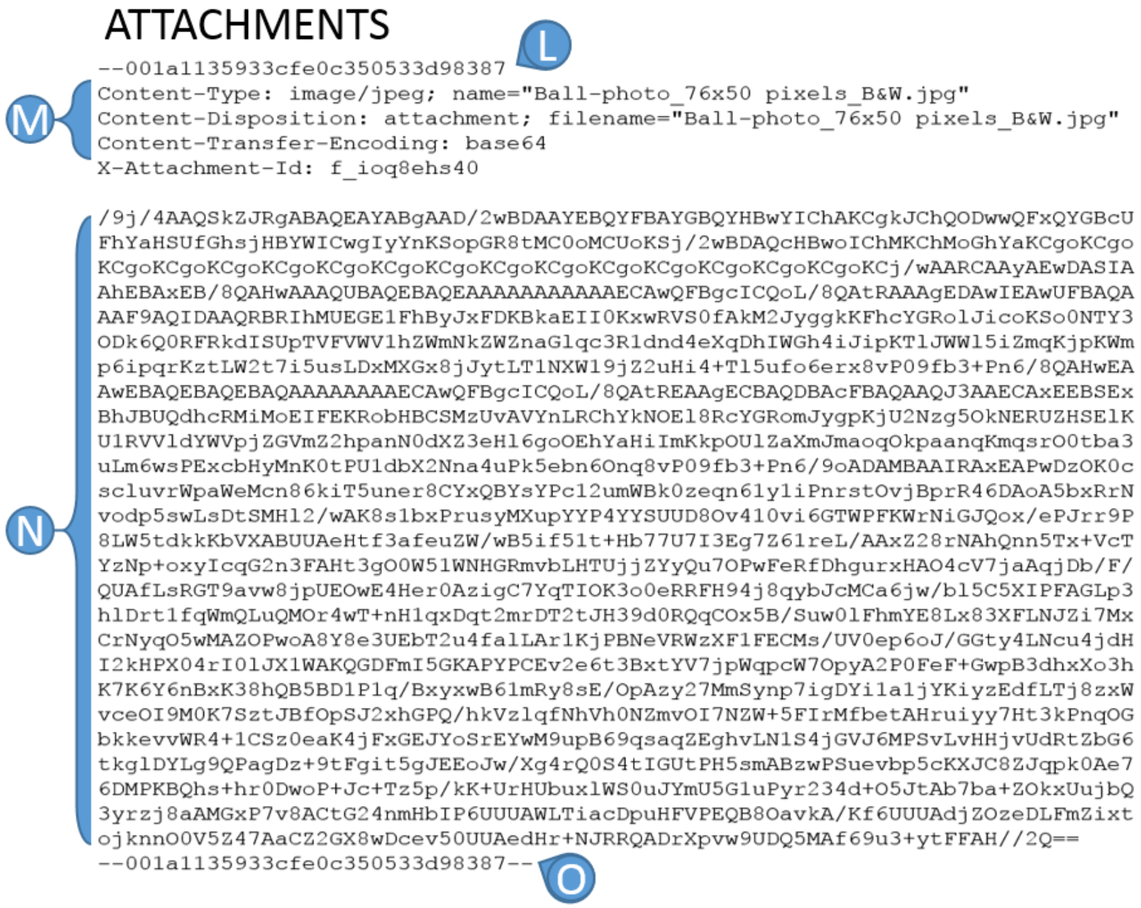


MESSAGE BODY

```
--001a1135933cfe0c350533d98387
Content-Type: multipart/alternative; boundary=001a1135933cfe0c310533d98385

--001a1135933cfe0c310533d98385
Content-Type: text/plain; charset=UTF-8

Any data or attachment we send via e-mail must be encoded as alphanumeric
characters using an Internet standard called "*MIME*" for
*Multipurpose Internet Mail Extensions*.  So, whether you're sending
documents, images, sounds, video or computer programs, the attachment
must be converted to letters and numbers so that it "looks" like a text
message and can pass via SMTP.  Such "content transfer encoding" comes in
three principal forms of binary-to-text: *Base64,* *quoted-printable *and
* 7Bit*.

And, yes, this technical minutiae has a very real impact on electronic
discovery and search.

Craig Ball
Attorney and Forensic Technologist
Certified Computer Forensic Examiner
3723 Lost Creek Blvd.
Austin, Texas 78735
TEL: 512-514-0182
MBL: 713-320-6066
GV: (740) 4RENSIC
E-MAIL: craig@ball.net
WEB: www.craigball.com
BLOG: www.ballinyourcourt.com

--001a1135933cfe0c310533d98385
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

<div dir=3D"ltr"><div style=3D"font-size:12.8px">Any data or attachment we =
send via e-<span class=3D"">mail</span>=C2=A0must be encoded as alphanumeri=
c characters using an=C2=A0<span class=3D"">Internet</span>=C2=A0standard c=
alled &quot;<strong>MIME</strong>&quot; for=C2=A0<em><span class=3D"">Multi=
purpose</span>=C2=A0<span class=3D"">Internet</span>=C2=A0<span class=3D"">=
Mail=C2=A0</span><span class=3D"">Extensions</span></em>.=C2=A0=C2=A0So, wh=
ether you&#39;re sending documents, images, sounds, video or=C2=A0computer =
programs, the attachment must=C2=A0be converted to letters and numbers so t=
hat it &quot;looks&quot; like a text message and can pass via SMTP.=C2=A0 S=
uch &quot;content transfer encoding&quot;=C2=A0comes in three principal for=
ms of binary-to-text:=C2=A0<em>Base64,</em>=C2=A0<em>quoted-printable=C2=A0=
</em>and<em>=C2=A07Bit</em>.</div><div style=3D"font-size:12.8px">=C2=A0</d=
iv><div style=3D"font-size:12.8px">And, yes, this technical minutiae has a =
very real impact on electronic discovery and search.</div><div><div class=
=3D"gmail_signature" data-smartmail=3D"gmail_signature"><div dir=3D"ltr"><d=
iv><div>=C2=A0</div><div>Craig Ball<br>Attorney and Forensic Technologist<b=
r>Certified Computer Forensic Examiner<br>3723 Lost Creek Blvd.<br>Austin, =
Texas 78735<br><span style=3D"font-size:12.8px">TEL: 512-514-0182</span><br=
 style=3D"font-size:12.8px"><span style=3D"font-size:12.8px">MBL: 713-320-6=
066</span></div><div>GV: (740) 4RENSIC<br>E-MAIL: <a href=3D"mailto:craig@b=
all.net" target=3D"_blank">craig@ball.net</a><br>WEB: <a href=3D"http://www=
.craigball.com" target=3D"_blank">www.craigball.com</a></div><div>BLOG: <a =
href=3D"http://www.ballinyourcourt.com" target=3D"_blank">www.ballinyourcou=
rt.com</a></div></div></div></div></div>
</div>

--001a1135933cfe0c310533d98385–
```

I didn’t draft the message in *either* plain text or HTML formats, but my e-mail service thoughtfully did both to ensure that my message won’t confuse recipients using (incredibly old) e-mail software unable to display the richer formatting supported by HTML. For these recipients, the plain text version gets the point across, albeit sans the bolding, italics, hyperlinks and other embellishments of the HTML version.

Turning to the last segment of the message, we see, at **(L)**, the transition between the message body and the attachment segments commemorated by our old friend 387, the first boundary notation.



ATTACHMENTS

```
--001a1135933cfe0c350533d98387
Content-Type: image/jpeg; name="Ball-photo_76x50 pixels_B&W.jpg"
Content-Disposition: attachment; filename="Ball-photo_76x50 pixels_B&W.jpg"
Content-Transfer-Encoding: base64
X-Attachment-Id: f_ioq8ehs40
```

/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAYEBQYFBAYGBQYHBwYIChAKCgkJChQODwwQFxQYGBcU
FhYaHSUfGhsjHBYWICwgIyYnKSopGR8tMC0oMCUoKSj/2wBDAQcHBwoIChMKChMoGhYaKCgoKCgo
KCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCj/wAARCAAyAEwDASIA
AhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQA
AAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3
ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWm
p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEA
AwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSEx
BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElK
U1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3
uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDzOK0c
scluvrWpaWeMcn86kiT5uner8CYxQBYsYPc12umWBk0zeqn61y1iPnrstOvjBprR46DAoA5bxRrN
vodp5swLsDtSMHl2/wAK8s1bxPrusyMXupYYP4YYSUUD8Ov410vi6GTWPFKWrNiGJQox/ePJrr9P
8LW5tdkkKbVXABUUAeHtf3afeuZW/wB5if51t+Hb77U7I3Eg7Z61reL/AAxZ28rNAhQnn5Tx+VcT
YzNp+oxyIcqG2n3FAHt3gO0W51WNHGRmvbLHTUjjZYyQu7OPwFeRfDhgurxHAO4cV7jaAqjDb/F/
QUAfLsRGT9avw8jpUEOwE4Her0AzigC7YqTIOK3o0eRRFH94j8qybJcMCa6jw/bl5C5XIPFAGLp3
hlDrt1fqWmQLuQMOr4wT+nH1qxDqt2mrDT2tJH39d0RQqCOx5B/Suw0lFhmYE8Lx83XFLNJZi7Mx
CrNyqO5wMAZOPwoA8Y8e3UEbT2u4FESs/UV0ep6oJ/GGty4LNcu4jdH
I2kHPX04rI0lJX1WAKQGDFmI5GKAPYPCEv2e6t3BxtYV7jpWqpcW7OpyA2P0FeF+GwpB3dhxXo3h
K7K6Y6nBxK38hQB5BD1Pl1q/BxywB61mRy8sE/OpAzy27MmSynp7igDYi1a1jYKiyzEdfLTj8zxW
vceOI9M0K7SztJBfOpSJ2xhGPQ/hkVzlqfNhVh0NZmvOI7NZW+5+Fr6ZR+Ne3kiB3Pnd7dt5ZGnbGm
bkkevvWR4+1CSz0eaK4jFxGEJYosrEYwM9upB69qsaqZEghvLN1S4jGVJ6MPSvLvHHjvUdRtZbG6
tkglDYLg9QPagzDz+hr0DwoP+Jc+Tz5p5p/kK+UrHUbuxJYmU5G1uPyr234Ab+ZOkxUujbQ
6DMPKBQhs+hr0DwoP+Jc+Tz5p/kK+UrHUbuxlWS0uJYmU5G1uPyr234d+O5JtAb7ba+ZOkxUujbQ
3yrzj8aAMGxP7v8ACtG24nmHbIP6UUUAWLTiacDpuHFVPEQB8OavkA/KHUAdjZOzeDLFmZixt
ojknnO0V5Z47AaCZ2GX8wDcev50UUAedHr+NJFAH//2Q==

```
--001a1135933cfe0c350533d98387--
```

At **(M)**, we see another declaration of Content-Type, now as an image in the JPEG format common to digital photography. The "name" segment identifies the item encoded and the Content-Disposition designates how the item is to be handled on delivery; here, as an attachment to be assigned the same filename when decoded at its destination. But where is the JPG photo?

Recall that to travel as an e-mail attachment, binary content (like photos, sound files, video or machine codes) must first be converted to plain text characters. Thus, the photograph has been encoded to a format called **base64**, which substitutes 64 printable ASCII characters (A–Z, a–z, 0–9, + and /) for any binary data or for foreign characters, like Cyrillic or Chinese, that can be represented by the Latin alphabet.[71] Note the declaration in **(M)**, "**Content-Transfer-Encoding: base64**."

[71] A third common transfer encoding is called "quoted-printable" or "QP encoding." It facilitates transfer of non-ASCII 8-bit data as 7-bit ASCII characters using three ASCII characters (the "equals" sign followed by two hexadecimal characters: 0-9 and A-F) to stand in for a byte of data Quoted-printable is employed where the content to be encoded is

Accordingly, the attached JPEG photograph with the filename “Ball-photo_76x50 pixels_B&W.jpg,” has been encoded from non-printable binary code into those 26 lines of apparent gibberish comprising nearly 2,000 plain text characters **(N).** It’s now able to traverse the network as an e-mail, yet easily be converted back to binary data when the message reaches its destination.

Figure 3

```
/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAYEBQYFBAYGBQYHBwYIChAKCgkJChQODwwQFxQYGBcU
FhYaHSUfGhsjHBYWICwgIyYnKSopGR8tMC0oMCUoKSj/2wBDAQcHBwoIChMKChMoGhYaKCgoKCgo
KCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCj/wAARCAAyAEwDASIA
AhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQA
AAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3
ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWm
p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEA
AwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSEx
BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElK
U1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3
uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDzOK0c
scluvrWpaWeMcn86kiT5uner8CYxQBYsYPc12umWBk0zeqn61y1iPnrstOvjBprR46DAoA5bxRrN
vodp5swLsDtSMHl2/wAK8s1bxPrusyMXupYYP4YYSUUD8Ov410vi6GTWPFKWrNiGJQox/ePJrr9P
8LW5tdkkKbVXABUUAeHtf3afeuZW/wB5if51t+Hb77U7I3Eg7Z61reL/AAxZ28rNAhQnn5Tx+VcT
YzNp+oxyIcqG2n3FAHt3gO0W51WNHGRmvbLHTUjjZYyQu7OPwFeRfDhgurxHAO4cV7jaAqjDb/F/
QUAfLsRGT9avw8jpUEOwE4Her0AzigC7YqTIOK3oO0eRRFH94j8qybJCMCa6jw/b15C5XIPFAGLp3
hlDrt1fqWmQLuQMOr4wT+nH1qxDqt2mrDT2tJH39d0RQqCOx5B/SuwOlFhmYE8Lx83XFLNJZi7Mx
CrNyqO5wMAZOPwoA8Y8e3UEbT2u4faiLAr1KjPBNeVRWzXF1FECMs/UV0ep6oJ/GGty4LNcu4jdH
I2kHPX04rIOlJX1WAKQGDFmI5GKAPYPCEv2e6t3BxtYV7jpWqcW7OpyA2POFeF+GwpB3dhxXo3h
K7K6Y6nBxK38hQB5BD1P1q/BxyRy8sE/OpAzy27MmSynp7igDyP1alj1YKiyzEdfLTj8zxW
vceOI9MOK7SztJBfOpSJ2xhGPQ/hkvz1qfNhvh0NZmvOI7NZW+5FIrMfbetAHruiyy7Ht3kPnqOG
bkkevWR4+1CSz0eaK4jFxGEJYoSrEYwM9upBqGV6MPSvLvVHHjvUdRtZG6
tkglDYLg9QPagDz+rQ0S4tIGUtPH5smAbzPEQ09TFgit5gJEEoJw/Xg4rQ0S4tIGUtPH5smAbzwPSuevbp5cKXJC8ZJqpk0Ae7
6DMPKBQhs+hr0DwoP+Jc+Tz5Tz0kK+UrHUbxlWS0uJYmU5UzbuUjw8aRGlu1uPyr234Ab7ba+ZOkxUujbQ
3yrzj8aAMGxP7v8ACtG24nmHbIP6UUUAWLkVpEQB8OavkA/Kf6UUUAdjZOzeDLmFmdFmzez1DLFmZix
ojknnO0V5Z47AaCZ2GX8wDcev5OUUAedHr+NJRRQADrXpvw9u9UDQ5MAf69u3+ytFFAH//2Q==
```

Finally, the message transmission concludes with the first boundary notation at **(O)**.

The lesson from this is that what you see displayed in your e-mail client application isn’t really the e-mail. It’s an *arrangement* of selected *parts* of the message, frequently modified in some respects from the native message source that traversed the network and Internet and, as often, supplemented by metadata (like message flags, contact data and other feature-specific embellishments) unique to your software and setup. What you see handily displayed as a discrete attachment is encoded into the message body. The time assigned to the message is calculated relative to your machine’s time and DST settings. Even the sender’s name may be altered based upon the way your machine and contact’s database are configured. What you see is not always what you get (or got).

Hashing and Deduplication

The ability to “fingerprint” data using hash algorithms makes it possible to identify identical files without the necessity of examining their content. If the hash values of two files are identical, the files are identical. As previously discussed, this file-matching ability allows hashing to be used to deduplicate collections of electronic files before review, saving money and minimizing the potential for inconsistent decisions about privilege and responsiveness for identical files.

Although hashing is a useful and versatile technology, it has a few shortcomings. Because the tiniest change in a file will alter that file’s hash value, hashing is of little value in comparing files that have

predominantly ASCII text coupled with some non-ASCII items. Its principal advantage is that it allows the encoded data to remain largely intelligible to readers.

any differences, even if those differences have no bearing on the substance of the file. Applied to e-mail, we understand from our e-mail “autopsy” that messages contain unique identifiers, time stamps and routing data that would frustrate efforts to compare one complete message to another using hash values. Looking at the message as a whole, multiple recipients of the same message have different versions insofar as their hash values.

Consequently, deduplication of e-mail messages is accomplished by calculating hash values for selected segments of the messages and comparing those segment values. Thus, hashing e-mails for deduplication will omit the parts of the header data reflecting, *e.g.,* the message identifier and the transit data. Instead, it will hash just the data seen in, *e.g.,* the To, From, Subject and Date lines, message body and encoded attachment. If these match, the message can be said to be *practically* identical.

By hashing particular segments of messages and selectively comparing the hash values, it’s possible to gauge the *relative* similarity of e-mails and perhaps eliminate the cost to review messages that are *inconsequentially* different. This concept is called “near deduplication.” It works, but it’s important to be aware of exactly what it’s excluding and why. It’s also important to advise your opponents when employing near deduplication and ascertain whether you’re mechanically excluding evidence the other side deems relevant and material.

Hash deduplication of e-mail is tricky. Time values may vary, along with the apparent order of attachments. These variations, along with minor formatting discrepancies, may serve to prevent the exclusion of items defined as duplicates. When this occurs, be certain to delve into the reasons *why* apparent duplicates aren’t deduplicating, as such errors may be harbingers of a broader processing problem.

Local E-Mail Storage Formats and Locations

Suppose you’re faced with a discovery request for a client’s e-mail and there’s no budget or time to engage an e-discovery service provider or ESI expert?

Where are you going to look to find stored e-mail, and what form will it take?

"Where's the e-mail?" It's a simple question, and one answered too simply and often wrongly by, "It's on the server" or "The last 60 days of mail is on the server and the rest is purged." Certainly, much e-mail will reside on the server, but <u>most</u> e-mail is elsewhere; and it's never all gone in practice, notwithstanding retention policies. The true location and extent of e-mail depends on systems configuration, user habits, backup procedures and other hardware, software and behavioral factors. This is true for mom-and-pop shops, for large enterprises and for everything in-between.

Going to the server isn’t the wrong answer. It’s just not the entire answer. In a matter where I was tasked to review e-mails of an employee believed to have stolen proprietary information, I went first to the company’s Microsoft Exchange e-mail server and gathered a lot of unenlightening e-mail. Had I stopped there, I would've missed the Hotmail traffic in the Temporary Internet Files folder and the Short Message Service (SMS) exchanges in the smartphone synchronization files. I’d have overlooked the Microsoft Outlook archive file (archive.pst) and offline synchronization file (Outlook.ost) on the employee’s laptop, collectively holding thousands more e-mails, including some “smoking guns” absent from the server. These are just some of the many places e-mails without counterparts on the server may be found. Though an exhaustive search of every nook and cranny may not be required, you need to know your options in order to assess feasibility, burden and cost.

E-mail resides in some or all of the following venues, grouped according to relative accessibility:

Easily Accessible:

E-Mail Server: E-mail residing in active files on enterprise servers: MS Exchange e.g., (.edb, .stm, .log files), Office 365, Lotus Notes (.nsf files).

File Server: E-mail saved as individual messages or in container files on a user’s network file storage area (“network share”).

Desktops and Laptops: E-mail stored in active files on local or external hard drives of user workstation hard drives (e.g., .pst, .ost files for Outlook and .nsf for Lotus Notes), laptops (.ost, .pst, .nsf), mobile devices, and home systems, particularly those with remote access to networks.

OLK system subfolders holding viewed attachments to Microsoft Outlook messages, including deleted messages.

Mobile devices: An estimated 65% of e-mail messages were opened using mobile phones and tablets in Q4 2015. As many of these were downloaded to a local mail app, they reside on the device and do not necessarily lose such content when the same messages are deleted from the server. E-mail on mobile devices is readily accessible to the user, but poses daunting challenges for preservation and collection in e-discovery workflows.

Nearline e-mail: Optical "juke box" devices, backups of user e-mail folders.

Archived or journaled e-mail: e.g., HP Autonomy Zantaz Enterprise Archive Solution, EMC EmailXtender, NearPoint Mimosa, Symantec Enterprise Vault.

Accessible, but Often Overlooked:

E-mail residing on non-party servers: ISPs (IMAP, POP, HTTP servers), Office 365, Gmail, Yahoo! Mail, Hotmail, etc.

E-mail forwarded and cc'd to external systems: Employee forwards e-mail to self at personal e-mail account.

E-mail threaded as text behind subsequent exchanges.

Offline local e-mail stored on removable media: External hard drives, thumb drives and memory cards, optical media: CD-R/RW, DVD-R/RW, floppy drives, zip drives.

Archived e-mail: Auto-archived or saved under user-selected filename.

Common user "flubs": Users experimenting with export features unwittingly create e-mail archives.

Legacy e-mail: Users migrate from e-mail clients "abandoning" former e-mail stores. Also, e-mail on mothballed or re-tasked machines and devices.

E-mail saved to other formats: PDF, .tiff, .txt, .eml, .msg, etc.

E-mail contained in review sets assembled for other litigation/compliance purposes.

E-mail retained by vendors or third- parties (e.g., former service provider or attorneys)

Paper print outs.

Less Accessible:

Offline e-mail on server backup tapes and other media.

E-mail in forensically accessible areas of local hard drives and re-tasked/reimaged legacy machines: deleted e-mail, internet cache, unallocated clusters.

The levels of accessibility above speak to practical challenges to ease of access, not to the burden or cost of review. The burden continuum isn’t a straight line. That is, it may be less burdensome or costly to turn to a small number of less accessible sources holding relevant data than to broadly search and review the contents of many accessible sources. Ironically, it typically costs much more to process and review the contents of a mail server than to undertake forensic examination of a key player’s computer; yet, the former is routinely termed “reasonably accessible” and the latter not.

The issues in the case, key players, relevant time periods, agreements between the parties, applicable statutes, decisions and orders of the court determine the extent to which locations must be examined; however, the failure to diligently identify relevant e-mail carries such peril that caution should be the watchword. Isn't it wiser to invest more effort to know exactly what the client has—even if it's not reasonably accessible and will not be searched or produced—than concede at the sanctions hearing the client failed to preserve and produce evidence it didn't know it because no one looked?

Looking for E-Mail 101

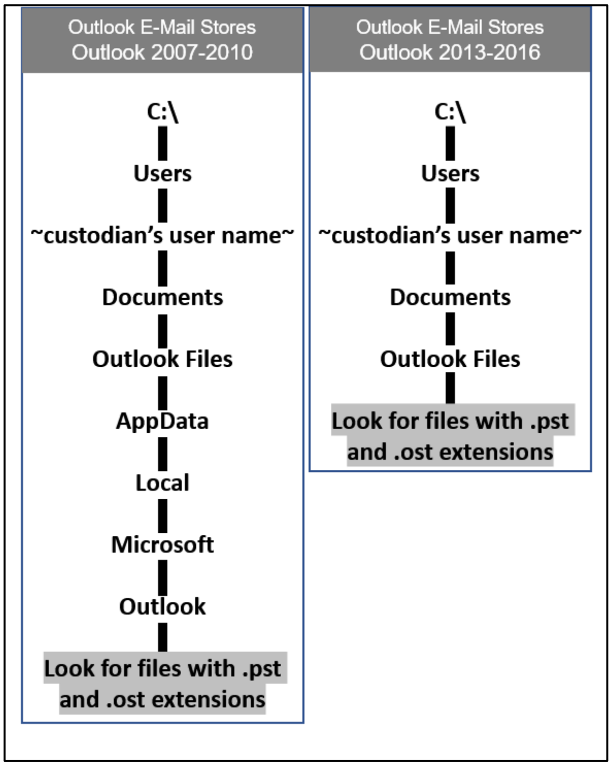
Because an e-mail is just a text file, individual e-mails could be stored as discrete text files. But that's not a very efficient or speedy way to manage many messages, so you'll find that most e-mail client software doesn't do that. Instead, e-mail clients employ proprietary database files housing e-mail messages, and each of the major e-mail clients uses its own unique format for its database. Some programs encrypt the message stores. Some applications merely display e-mail housed on a remote server and do not store messages locally (or only in fragmentary way). The only way to know with certainty if e-mail is stored on a local hard drive is to look for it.

Merely checking the e-mail client's settings is insufficient because settings can be changed. Someone not storing server e-mail today might have been storing it a month ago. Additionally, users may create new identities on their systems, install different client software, migrate from other hardware or take various actions resulting in a cache of e-mail residing on their systems without their knowledge. *If they don't know it's there, they can't tell you it's not*. On local hard drives, you've simply got to know what to look for and where to look…*and then you've got to look for it.*

For many, computer use has been a decades-long adventure. One may have first dipped her toes in the online ocean using browser-based e-mail or an AOL account. Gaining computer-savvy, she may have signed up for broadband access or with a local ISP, downloading e-mail with Netscape Messenger or Microsoft Outlook Express. With growing sophistication, a job change or new technology at work, the user may have migrated to Microsoft Outlook or Lotus Notes as an e-mail client, then shifted to a cloud service like Office 365. Each of these steps can orphan a large cache of e-mail, possibly unbeknownst to the user but still fair game for discovery. Again, you've simply got to know what to look for and where to look.

One challenge you'll face when seeking stored e-mail is that every user's storage path is different. This difference is not so much the result of a user's ability to specify the place to store e-mail—which few do, but which can make an investigator's job more difficult when it occurs—but more from the fact that operating systems are designed to support multiple users and so must assign unique identities and set aside separate storage areas for different users. Even if only one person has used a Windows computer, the operating system will be structured at the time of installation so as to make way for others. Thus, finding e-mail stores will hinge on your knowledge of the User's Account Name or Globally Unique Identifier (GUID) string assigned by the operating system. This may be as simple as the user's name or as obscure as the 128-bit hexadecimal value {721A17DA-B7DD-4191-BA79-42CF68763786}. Customarily, it's both.

Finding Outlook E-Mail



PST: Microsoft Outlook has long been the most widely used e-mail client in the business environment. Outlook encrypts and compresses messages, and all of its message data and folder structure, along with all other information managed by the program (except the user's Contact data), is stored within a single, often massive, database file with the file extension .pst.

OST: While awareness of the Outlook PST file is widespread, even many lawyers steeped in e-discovery fail to consider a user's Outlook .ost file. The OST or offline synchronization file is commonly encountered on laptops configured for Exchange Server environments. It exists for the purpose of affording access to messages when the user has no active network connection. Designed to allow work to continue on, e.g., airplane flights, local OST files often hold messages purged from the server—at least until re-synchronization. It's not unusual for an OST file to hold e-mail unavailable from any other comparably-accessible source.

Archive.pst: Another file to consider is one customarily called, "archive.pst." As its name suggests, the archive.pst file holds older messages, either stored automatically or by user-initiated action. If you've used Outlook without manually configuring its archive settings, chances are the system periodically asks whether you'd like to auto archive older items. Every other week (by default), Outlook seeks to auto archive any Outlook items older than six months (or for Deleted and Sent items older than two months). Users can customize these intervals, turn archiving off or instruct the application to permanently delete old items.

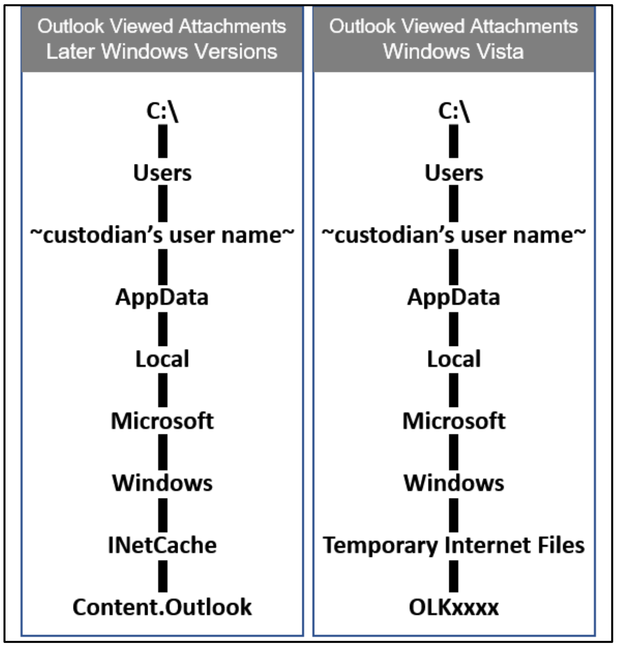
Outlook Mail Stores Paths

To find the Outlook message stores on Windows machines, drill down from the root directory (C:\ for most users) according to the path diagram shown for the applicable version of Outlook. The default filename of Outlook.pst/ost may vary if a user has opted to select a different designation or maintains multiple e-mail stores; however, it’s rare to see users depart from the default settings. Since the location of the PST and OST files can be changed by the user, it’s a good idea to do a search of all files and folders to identify any files ending with the .pst and .ost extensions.

“Temporary” OLK Folders

Note that by default, when a user opens an attachment to a message from within Outlook (as opposed to saving the attachment to disk and then opening it), Outlook stores a copy of the attachment in a “temporary” folder. But don’t be misled by the word “temporary.” In fact, the folder isn’t going anywhere and its contents—sometimes voluminous--tend to long outlast the messages that transported the attachments. Thus, litigants should be cautious about representing that Outlook e-mail is “gone” if the e-mail’s attachments are not.

The Outlook viewed attachment folder will have a varying name for every user and on every machine, but it will always begin with the letters “OLK” followed by several randomly generated numbers and uppercase letters (e.g., OLK943B, OLK7AE, OLK167, etc.). To find the OLKxxxx viewed attachments folder on machines running Windows XP/NT/2000 or Vista, drill down from the root directory according to the path diagrams on the right for the applicable operating system.[72]



Microsoft Exchange Server

Hundreds of millions of people get their work e-mail via a Microsoft product called Exchange Server. It’s been sold for twenty years and its latest version is Exchange Server 2016; although, many users continue to rely on the older versions of the product.

The key fact to understand about an e-mail server is that it’s a *database* holding the messages (and calendars, contacts, to-do lists, journals and other datasets) of multiple users. E-mail servers are configured to maximize performance, stability and disaster recovery, with little consideration given to compliance and discovery obligations. If anyone anticipated the role e-mail would play in

[72] By default, Windows hides system folders from users, so you may have to first make them visible. This is accomplished by starting Windows Explorer, then selecting ‘Folder Options’ from the Tools menu in Windows XP or ‘Organize>Folder and Search Options’ in Vista. Under the 'View' tab, scroll to ‘Files and Folders' and check 'Show hidden files and folders' and uncheck 'Hide extensions for known file types' and 'Hide protected operating system files. Finally, click ‘OK.’

virtually every aspect of business today, their prescience never influenced the design of e-mail systems. E-mail evolved largely by accident, absent the characteristics of competent records management, and only lately are tools emerging that are designed to catch up to legal and compliance duties.

The other key thing to understand about enterprise e-mail systems is that, unless you administer the system, it probably doesn't work the way you imagine. The exception to that rule is if you can distinguish between Local Continuous Replication (LCR), Clustered Continuous Replication (CCR), Single Copy Cluster (SCC) and Standby Continuous Replication (SCR). In that event, I should be reading *your* paper!

Though the preceding pages dealt with finding e-mail stores on local hard drives, in disputes involving medium- to large-sized enterprises, the e-mail server (or its cloud-based counterpart) is likely to be the initial nexus of electronic discovery efforts. The server is a productive venue in electronic discovery for many reasons, among them:

The periodic backup procedures which are a routine part of prudent server management tend to shield e-mail stores from those who, by error or guile, might delete or falsify data on local hard drives.

The ability to recover deleted mail from archival server backups may obviate the need for costly and unpredictable forensic efforts to restore deleted messages.

Data stored on a server is often less prone to tampering by virtue of the additional physical and system security measures typically dedicated to centralized computer facilities as well as the inability of the uninitiated to manipulate data in the more-complex server environment.

The centralized nature of an e-mail server affords access to many users' e-mail and may lessen the need for access to workstations at multiple business locations or to laptops and home computers.

Unlike e-mail client applications, which store e-mail in varying formats and folders, e-mail stored on a server can usually be located with relative ease and adhere to common file formats.

The server is the crossroads of corporate electronic communications and the most effective chokepoint to grab the biggest "slice" of relevant information in the shortest time, for the least cost.

The latest versions of Exchange Server and the cloud tool, Office 365, feature robust e-discovery capabilities simplifying initiation and managements of legal holds and account exports.

Of course, the big advantage of focusing discovery efforts on the mail server (*i.e.,* it affords access to thousands or millions of messages) is also its biggest disadvantage (someone has to *collect and* *review* thousands or millions of messages). Absent a carefully-crafted and, ideally, agreed-upon plan for discovery of server e-mail, both requesting and responding parties run the risk of runaway costs, missed data and wasted time.

E-mail originating on servers is generally going to fall into two realms, being online "live" data, which is deemed reasonably accessible, and offline "archival" data, routinely deemed inaccessible based on considerations of cost and burden.[73] Absent a change in procedure, "chunks" of data routinely migrate from accessible storage to less accessible realms—on a daily, weekly or monthly basis—as selected information on the server is replicated to backup media and deleted from the server's hard drives.

The ABCs of Exchange

Because it's unlikely most readers will be personally responsible for collecting e-mail from an Exchange Server and mail server configurations can vary widely, the descriptions of system architecture here are offered only to convey a rudimentary understanding of common Exchange architecture.

Older versions of Exchange Server stored data in a Storage Group containing a Mailbox Store and a Public Folder Store, each composed of two files: an .edb file and a .stm file. Mailbox Store, Priv1.edb, is a rich-text database file containing user's e-mail messages, text attachments and headers. Priv1.stm is a streaming file holding SMTP messages and containing multimedia data formatted as MIME data. Public Folder Store, Pub1.edb, is a rich-text database file containing messages, text attachments and headers for files stored in the Public Folder tree. Pub1.stm is a streaming file holding SMTP messages and containing multimedia data formatted as MIME data. Later versions of Exchange Server did away with STM files altogether, shifting their content into the EDB database files.

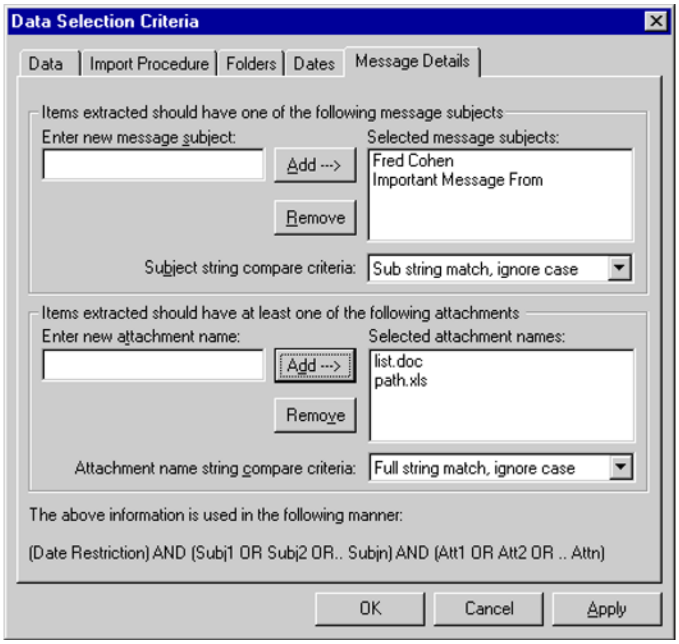
Storage Groups also contain system files and transaction logs. Transaction logs serve as a disaster recovery mechanism that helps restore an Exchange after a crash. Before data is written to an EDB file, it is first written to a transaction log. The data in the logs can thus be used to reconcile transactions after a crash.

By default, Exchange data files are located in the path X:\Program files\Exchsrvr\MDBDATA, where X: is the server's volume root. But, it's common for Exchange administrators to move the mail stores to other file paths.

[73] Lawyers and judges intent on distilling the complexity of electronic discovery to rules of thumb are prone to pigeonhole particular ESI as "accessible' or 'inaccessible" based on the media on which it resides. In fact, ESI's storage medium is just one of several considerations that bear on the cost and burden to access, search and produce same. Increasingly, backup tapes are less troublesome to search and access while active data on servers or strewn across many "accessible" systems and devices is a growing challenge.

Recovery Storage Groups and ExMerge

Two key things to understand about Microsoft Exchange are that, since 2003, an Exchange feature called Recovery Storage Group supports collection of e-mail from the server without any need to interrupt its operation or restore data to a separate recovery computer. The second key thing is that Exchange includes a simple utility for exporting the server-stored e-mail of individual custodians to separate PST container files. This utility, officially the Exchange Server Mailbox Merge Wizard but universally called ExMerge allows for rudimentary filtering of messages for export, including by message dates, folders, attachments and subject line content.



Items extracted should have one of the following message subjects

Enter new message subject:

Selected message subjects:
Fred Cohen
Important Message From

Subject string compare criteria: Sub string match, ignore case

Items extracted should have at least one of the following attachments

Enter new attachment name:

Selected attachment names:
list.doc
path.xls

Attachment name string compare criteria: Full string match, ignore case

The above information is used in the following manner:

(Date Restriction) AND (Subj1 OR Subj2 OR.. Subjn) AND (Att1 OR Att2 OR .. Attn)

ExMerge also plays a crucial role in recovering e-mails “double deleted” by users if the Exchange server has been configured to support a “dumpster retention period.” When a user deletes an e-mail, it’s automatically relegated to a “dumpster” on the Exchange Server. The dumpster holds the message for 30 days by default or until a full backup of your Exchange database is run, whichever comes first. The retention interval can be customized for a longer or shorter interval.

Later versions of Exchange Server and certain implementations of Exchange Online [Office 365] have done away with the dumpster feature and take an entirely different (and superior) approach to retention of double-deleted messages. As noted, these tools also offer purpose-built e-discovery preservation features that are much easier to implement and manage than earlier Exchange Server versions.

Journaling, Archiving and Transport Rules

Journaling is the practice of copying all e-mail to and from all users or particular users to one or more repositories inaccessible to most users. Journaling serves to preempt ultimate reliance on individual users for litigation preservation and regulatory compliance. Properly implemented, it should be entirely transparent to users and secured in a manner that eliminates the ability to alter the journaled collection.

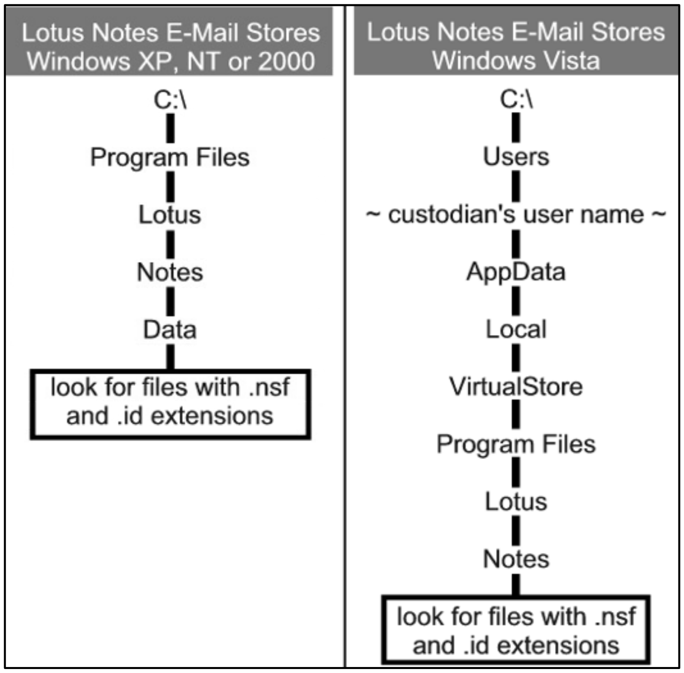
Exchange Server supports three types of journaling: Message-only journaling which does not account for blind carbon copy recipients, recipients from transport forwarding rules, or recipients from distribution group expansions; Bcc journaling, which is identical to Message-only journaling except that it captures Bcc addressee data; and Envelope Journaling which captures all data about the message, including information about those who received it. Envelope journaling is the mechanism best suited to e-discovery preservation and regulatory compliance.

Journaling should be distinguished from e-mail archiving, which may implement only selective, rules-based retention and customarily entails removal of archived items from the server for offline or near-line storage, to minimize strain on IT resources and/or implement electronic records management. However, Exchange journaling also can implement rules-based storage, so each can conceivably be implemented to play the role of the other.

A related concept is the use of Transport Rules in Exchange, which serve, *inter alia*, to implement "Chinese Walls" between users or departments within an enterprise who are ethically or legally obligated not to share information, as well as to guard against dissemination of confidential information. In simplest terms, software called *transport rules agents* "listen" to e-mail traffic, compare the content or distribution to a set of rules (conditions, exceptions and actions) and if particular characteristics are present, intercedes to block, route, flag or alter suspect communications.

Lotus Domino Server and Notes Client

Though Microsoft's Exchange and Outlook e-mail products have a greater overall market share, IBM's Lotus Domino and Notes products hold powerful sway within the world's largest corporations, especially giant manufacturing concerns and multinationals. IBM boasts of over 300 million Notes mailboxes worldwide.


Lotus Notes E-Mail Stores Windows XP, NT or 2000
C:\
Program Files
Lotus
Notes
Data
look for files with .nsf and .id extensions
Lotus Notes E-Mail Stores Windows Vista
C:\
Users
~ custodian's user name ~
AppData
Local
VirtualStore
Program Files
Lotus
Notes
look for files with .nsf and .id extensions

Lotus Notes can be unhelpfully described as a "cross-platform, secure, distributed document-oriented database and messaging framework and rapid application development environment." The main takeaway with Notes is that, unlike Microsoft Exchange, which is a purpose-built application designed for messaging and calendaring, Lotus Notes is more like a toolkit for *building* whatever capabilities you need to deal with documents—mail documents, calendaring documents and any other type of document used in business. Notes wasn't *designed* for e-mail—e-mail just happened to be one of the things it was tasked to do.[74] Notes is database driven and distinguished by its replication and security.

Lotus Notes is all about copies. Notes content, stored in Notes Storage facility or NSF files, are constantly being replicated (synchronized) here and there across the network. This guards against data loss and enables data access when the network is unavailable, but it also means that there can be many versions of Notes data stashed in various places within an enterprise. Thus, discoverable Notes mail may not be gone, but lurks within a laptop that hasn't connected to the network since the last business trip.

By default, local iterations of users' NSF and ID files will be found on desktops and laptops in the paths shown in the diagrams at right. It's imperative to collect the user's .id file along with the .nsf message container or you may find yourself locked out of encrypted content. It's also important to secure each custodian's Note's password. It's common for Notes to be installed in ways other than the default configuration, so search by extension to insure that .nsf and .id files are not also found elsewhere. Also, check the files' last modified date to assess whether the date is consistent with expected last usage. If there is a notable disparity, look carefully for alternate file paths housing later replications.

Local replications play a significant role in e-discovery of Lotus Notes mail because, built on a database and geared to synchronization of data stores, deletion of an e-mail within Lotus "broadcasts" the deletion of the same message system wide. Thus, it's less common to find undeleted iterations of messages in a Lotus environment unless you resort to backup media or find a local iteration that hasn't been synchronized after deletion.

Webmail

More than 25% of the people on the planet use webmail; so any way you slice it, webmail can't be ignored in e-discovery. Webmail holding discoverable ESI presents legal, technical and practical challenges, but the literature is nearly silent about how to address them.

The first hurdle posed by webmail is the fact that it's stored "in the cloud" and off the company grid. Short of a subpoena or court order, the only legitimate way to access and search employee web mail is with the employee's cooperation, and that's not always forthcoming. Courts nonetheless expect employers to exercise control over employees and insure that relevant, non-privileged webmail isn't lost or forgotten.

[74] Self-anointed "Technical Evangelist" Jeff Atwood described Lotus Notes this way: "It is death by a thousand tiny annoyances— the digital equivalent of being kicked in the groin upon arrival at work every day." http://www.codinghorror.com/blog/2006/02/12/ (visited 5/18/2013) In fairness, Lotus Notes has been extensively overhauled since he made that observation.

One way to assess the potential relevance of webmail is to search server e-mail for webmail traffic. If a custodian's Exchange e-mail reveals that it was the custodian's practice to e-mail business documents to or from personal webmail accounts, the webmail accounts may need to be addressed in legal hold directives and vetted for responsive material.

A second hurdle stems from the difficulty in collecting responsive webmail. How do you integrate webmail content into your review and production system? Where a few pages might be "printed" to searchable Adobe Acrobat PDF formats or paper, larger volumes require a means to dovetail online content and local collections. The most common approach is to employ a POP3 or IMAP client application to download messages from the webmail account. All of the leading webmail providers support POP3 transfer, and with the user's cooperation, it's simple to configure a clean installation of any of the client applications already discussed to capture online message stores. Before proceeding, the process should be tested against accounts that don't evidence to determine what metadata values may be changed, lost or introduced by POP3 collection.

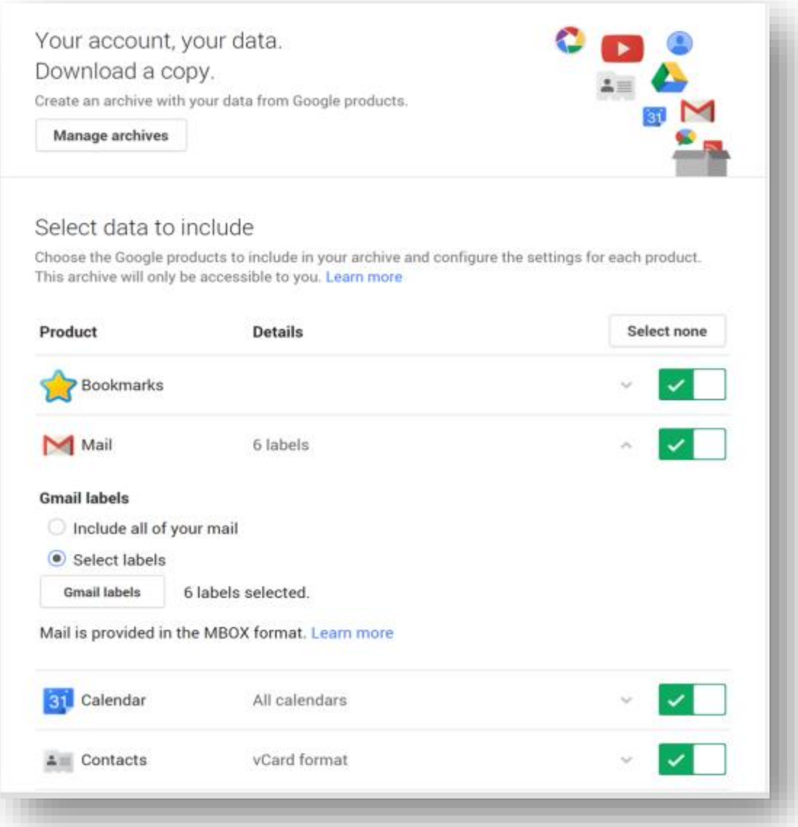
Webmail content can be fragile compared to server content. Users rarely employ a mechanism to back up webmail messages (other than the POP3 or IMAP retrieval just discussed) and webmail accounts may purge content automatically after periods of inactivity or when storage limits are exceeded. Further, users tend to delete embarrassing or incriminating content more aggressively on webmail, perhaps because they regard webmail content as personal property or the evanescent nature of account emboldens them to believe spoliation will be harder to detect and prove.

Happily, some webmail providers—notably Google Gmail—have begun to offer effective "take out" mechanisms for user cloud content, including webmail. Google does the Gmail collection *gratis* and puts it in a standard MBOX container format that can be downloaded and sequestered. Google even incorporates custom metadata values that reflect labeling and threading. You won't see these unique metadata tags if you pull the messages into an e-mail client; but, good e-discovery software will pick them up.

MBOX might not be everyone's choice for a Gmail container file; but, it's inspired. MBOX stores the messages in their original Internet message format called RFC 2822 (now RFC 5322), a superior form for e-discovery preservation and production.

Google Data Tools: Takeout

The only hard part of archiving Gmail is navigating to the right page. You get there from the Google Account Setting page by selecting "Data Tools" and looking for the "Download your Data" option on the lower right. When you click on "Create New Archive," you'll see a menu like that below where you choose whether to download all mail or just items bearing the labels you select.



Your account, your data.
Download a copy.

Create an archive with your data from Google products.

Manage archives

Select data to include

Choose the Google products to include in your archive and configure the settings for each product. This archive will only be accessible to you. Learn more

Product	Details	Select none
Bookmarks		✓
Mail	6 labels	✓

Gmail labels

○ Include all of your mail

● Select labels

Gmail labels 6 labels selected.

Mail is provided in the MBOX format. Learn more

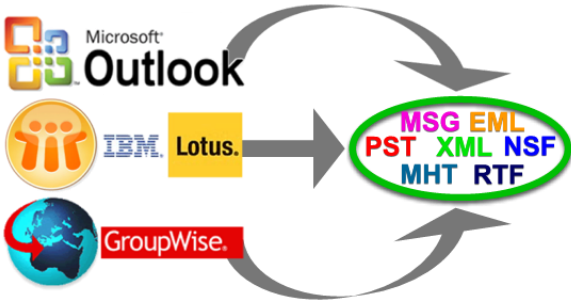
Product	Details	
Calendar	All calendars	✓
Contacts	vCard format	✓

The ability to label content within Gmail and archive only messages bearing those labels means that Gmail's powerful search capabilities can be used to identify and label potentially responsive messages, obviating the need to archive everything. It's not a workflow suited to every case; yet, it's a promising capability for keeping costs down in cases involving just a handful of custodians with Gmail.

Forms of Production

As discussed above, what users see presented onscreen as e-mail is a selective presentation of information from the header, body and attachments of the source message, determined by the capabilities and configuration of their e-mail client and engrafted with metadata supplied by that client. Meeting the obligation to produce comparable data of similar utility to the other side in discovery is no mean feat, and one that hinges on choosing suitable forms of production.

Requesting parties often demand "native production" of e-mail; but, electronic mail is rarely produced natively in the sense of supplying a duplicate of the source container file. That is, few litigants produce the entire Exchange database EDB file to the other side. Even those that produce mail in the format employed natively by the application (e.g., as a PST file) aren't likely to produce the source file but will fashion a reconstituted PST file composed of selected messages deemed responsive and non-privileged.



As applied to e-mail, "native production" instead signifies production in a form or forms that most closely approximate the contents and usability of the source. Often, this will be a form of production identical to the original (e.g., PST or NSF) or a form (like MSG or EML) that shares many of the characteristics of the source and can deliver comparable usability when paired with additional information (e.g., information about folder structures).[75] For further discussion of native forms of e-mail, see the following article, *What is Native Production of E-Mail?*

Similarly, producing parties employ imaged production and supply TIFF image files of messages, but to approximate the usability of the source, producing parties must also create and produce accompanying load files carrying the metadata and full text of the source message keyed to its images. Collectively, the load files and image data permit recipients with compatible software (*e.g.*, Relativity, Summation, Catalyst Insight) to view and search the messages. Selection of Adobe PDF documents as the form of production allows producing parties to dispense with the load files because much of the same data can be embedded in the PDF. PDF also has the added benefit of not requiring the purchase of review software.

Some producing parties favor imaged production formats in a mistaken belief that they are more secure than native production and out of a desire to emboss Bates numbers or other text (i.e., protective order language) to the face of each image. Imaged productions are more expensive than native or quasi-native productions, but, as they hew closest to the document review mechanisms long employed by law firms, they require little adaption. It remains to be seen if clients will continue to absorb higher costs solely to insulate their counsel from embracing more modern and efficient tools and techniques.

[75] When e-mail is produced as individual messages, the folder structure may be lost and with it, important context. Additionally, different container formats support different complements of metadata applicable to the message. For example, a PST container may carry information about whether a message was opened, flagged or linked to a calendar entry.

Other possible format choices include XML and MHT,[76] as well as Rich Text Format (RTF)--essentially plain text with improved formatting—and, for small collections, paper printouts.

There is no single, "perfect" form of production for e-mail, though the "best" format to use is the one on which the parties agree. Note also that there's likely not a single production format that lends itself to *all* forms of ESI. Instead, *hybrid productions* match the form of production to the characteristics of the data being produced. In a hybrid production, images are used where they are most utile or cost-effective and native formats are employed when they offer the best fit or value.

As a rule of thumb to maximize usability of data, hew closest to the format of the source data (i.e., PST for Outlook mail and NSF for Lotus Notes), but keep in mind that whatever form is chosen should be one that the requesting party has the tools and expertise to use.

Though there is no ideal form of production, we can be guided by certain ideals in selecting the forms to employ. Absent agreement between the parties or an order of the Court, the forms of production employed for electronic mail should be either the mail's native format or a form that will:

- Enable the complete and faithful reproduction of all information available to the sender and recipients of the message, including layout, bulleting, tabular formats, colors, italics, bolding, underlining, hyperlinks, highlighting, embedded images, emoticons and other non-textual ways we communicate and accentuate information in e-mail messages.
- Support accurate electronic searchability of the message text and header data;
- Maintain the integrity of the header data (To, From, Cc, Bcc, Subject and Date/Time) as discrete fields to support sorting and searching by these data;
- Preserve family relationships between messages and attachments;
- Convey the folder structure/path of the source message;
- Include message metadata responsive to the requester's legitimate needs;
- Facilitate redaction of privileged and confidential content and, as feasible, identification and sequencing akin to Bates numbering; and
- Enable reliable date and time normalization across the messages produced.[77]

[76] MHT is a shorthand reference for MHTML or MIME Hypertext markup Language. HTML is the markup language used to create web pages and rich text e-mails. MHT formats mix HTML and encoded MIME data (see prior discussion of MIME at page to represent the header, message body and attachments of an e-mail.

[77] E-mails carry multiple time values depending upon, e.g., whether the message was obtained from the sender or recipient. Moreover, the times seen in an e-mail may be offset per the time zone settings of the originating or receiving machine as well as for daylight savings time. When e-mail is produced as TIFF images or as text embedded in threads, these offsets may produce hopelessly confusing sequences.

What is Native Production of E-Mail?

Recently, I've weighed in on disputes where the parties were fighting over whether the e-mail production was sufficiently "native" to comply with the court's orders to produce natively. In one matter, the question was whether Gmail could be produced in a native format, and in another, the parties were at odds about what forms are native to Microsoft Exchange e-mail. In each instance, I saw two answers; the technically correct one and the helpful one.

I am a vocal proponent of native production for e-discovery. Native is complete. Native is functional. Native is inherently searchable. Native costs less. We will explore these advantages in later chapters. When I speak of "native" production in the context of databases, I am using a generic catchall term to describe electronic forms with superior functionality and completeness, notwithstanding the common need in e-discovery to produce less than all of a collection of ESI.

It's a Database

When we deal with e-mail in e-discovery, we are usually dealing with database content. Microsoft Exchange and Office 365 are *e-mail server applications*, and databases. Microsoft Outlook, an *e-mail client application*, is a database. Gmail, a *SaaS webmail application*, is a database. Lotus Domino, Lotus Notes, Yahoo! Mail, Hotmail and Novell GroupWise—they're all *databases*. It's important to understand this at the outset because if you think of e-mail as a collection of discrete objects (like paper letters in a manila folder), you're going to have trouble understanding why defining the "native" form of production for e-mail isn't as simple as many imagine.

Native in Transit: Text per a Protocol

E-mail is one of the oldest computer networking applications. Before people were sharing printers, and long before the internet was a household word, people were sending e-mail across networks. That early e-mail was plain text, also called ASCII text or 7-bit (because you need just seven bits of data, one less than a byte, to represent each ASCII character). In those days, there were no attachments, no pictures, not even simple enhancements like **bold**, *italic* or <u>underline</u>.

Early e-mail was something of a free-for-all, implemented differently by different systems. So the fledgling internet community circulated proposals seeking a standard. They stuck with plain text in order that older messaging systems could talk to newer systems. These proposals were called **Requests for Comment** or **RFC**s, and they came into widespread use as much by convention as by adoption (the internet being a largely anarchic realm). The RFCs lay out the form an e-mail should adhere to in order to be compatible with e-mail systems.

The RFCs concerning e-mail have gone through several major revisions since the first one circulated in 1973. The latest protocol revision is called RFC 5322 (2008), which made obsolete RFC 2822 (2001) and its predecessor, RFC 822 (1982). Another series of RFCs (RFC 2045-47, RFC 4288-89 and RFC 2049), collectively called Multipurpose Internet Mail Extensions or MIME, address ways to graft text enhancements, foreign language character sets and multimedia content onto plain text emails. These RFCs establish the form of the billions upon billions of e-mail messages that cross the internet.

So, if you asked me to state the native form of an e-mail *as it traversed the Internet between mail servers*, I'd likely answer, "plain text (7-bit ASCII) adhering to RFC 5322 and MIME." In my experience, this is the same as saying ".EML format;" and, it can be functionally the same as the MHT format, but only if the content of each message adheres strictly to the RFC and MIME protocols listed above. You can even change the file extension of a properly formatted message from EML to MHT and back to open the file in a browser or in a mail client like Outlook 2010. Try it. If you want to see what the native "plain text in transit" format looks like, change the extension from .EML to .TXT and open the file in Windows Notepad.

The appealing feature of producing e-mail in exactly the same format in which the message traversed the internet is that it's a form that holds the entire content of the message (header, message bodies and encoded attachments), and it's a form that's about as compatible as it gets in the e-mail universe. [78]

Unfortunately, the form of an e-mail *in transit* is often incomplete in terms of metadata it acquires upon receipt that may have probative or practical value; and the format in transit isn't native to the most commonly-used e-mail server and client applications, like Microsoft Exchange and Outlook. It's from these applications--*these databases*--that e-mail is collected in e-discovery.

[78] There's even an established format for storing multiple RFC 5322 messages in a container format called mbox. The mbox format was described in 2005 in RFC 4155, and though it reflects a simple, reliable way to group e-mails in a sequence for storage, it lacks the innate ability to memorialize mail features we now take for granted, like message foldering. A common workaround is to create a single mbox file named to correspond to each folder whose contents it holds (e.g., Inbox.mbox)

Outlook and Exchange

Microsoft Outlook and Microsoft Exchange are database applications that talk to each other using a protocol (machine language) called MAPI, for *Messaging Application Programming Interface*. Microsoft Exchange is an e-mail server application that supports functions like contact management, calendaring, to do lists and other productivity tools. Microsoft Outlook is an e-mail client application that accesses the contents of a user's account on the Exchange Server and may synchronize such content with local (i.e., retained by the user) container files supporting offline operation. If you can read your Outlook e-mail without a network connection, you have a local storage file.

Practice Tip (and Pet Peeve): *When your client or company runs Exchange Server and someone asks what kind of e-mail system your client or company uses, please don't say "Outlook." That's like saying "iPhone" when asked what cell carrier you use. Outlook can serve as a front-end client to Microsoft Exchange, Lotus Domino and most webmail services; so saying "Outlook" just makes you appear out of your depth (assuming you are someone who's supposed to know something about the evidence in the case).*

Outlook: The native format for data stored locally by Outlook is a file or files with the extension PST or OST. Henceforth, I'm going to speak only of PSTs, but know that either variant may be seen. PSTs are container files. They hold collections of e-mail—typically stored in multiple folders—as well as content supporting other Outlook features. The native PST found locally on the hard drive of a custodian's machine will hold all the Outlook content that the custodian can see when not connected to the e-mail server.

Because Outlook is a database application designed for managing messaging, it goes well beyond simply receiving messages and displaying their content. Outlook begins by taking messages apart and using the constituent information to populate various fields in a database. What we see as an e-mail message using Outlook is a report queried from a database. The native form of Outlook e-mail carries these fields and adds metadata not present in the transiting message. The added metadata fields include such information as the name of the folder in which the e-mail resides, whether the e-mail was read or flagged and its date and time of receipt. Moreover, because Outlook is designed to "speak" directly to Exchange using their own MAPI protocol, messages between Exchange and Outlook carry MAPI metadata not present in the "generic" RFC 5322 messaging. Whether this MAPI metadata is superfluous or invaluable depends upon what questions may arise concerning the provenance and integrity of the message. Most of the time, you won't miss it. Now and then, you'll be lost without it.

Because Microsoft Outlook is so widely used, its PST file format is widely supported by applications designed to view, process and search e-mail. Moreover, the complex structure of a PST is so well understood that many commercial applications can parse PSTs into single message formats or assemble single messages into PSTs. Accordingly, it's feasible to produce responsive messaging in a PST format while excluding messages that are non-responsive or privileged. It's also feasible to construct a production PST without calendar content, contacts, to do lists and the like. You'd be hard pressed to find a better form of production for Exchange/Outlook messaging. Here, I'm defining "better" in terms of completeness and functionality, not compatibility with your ESI review tools.

MSGs: There's little room for debate that the PST or OST container files are the native forms of data storage and interchange for a *collection* of messages (and other content) from Microsoft Outlook. But is there a native format for *individual* messages from Outlook, like the RFC 5322 format discussed above? The answer isn't clear cut. On the one hand, if you were to drag a single message from Outlook to your Windows desktop, Outlook would create that message in its proprietary MSG format. The MSG format holds the complete content of its RFC 5322 cousin plus additional metadata; but it lacks information (like foldering data) that's contained within a PST. It's not "native" in the sense that it's not a format that Outlook uses day-to-day; but it's an export format that holds more message metadata unique to Outlook. All we can say is that the MSG file is a highly compatible *near-native* format for individual Outlook messages--more complete than the transiting e-mail and less complete than the native PST. Though it's encoded in a proprietary Microsoft format (i.e., it's *not* plain text), the MSG format is so ubiquitous that, like PSTs, many applications support it as a standard format for moving messages between applications.

Exchange: The native format for data housed in an Exchange server is its database file, prosaically called the Exchange Database and sporting the file extension .EDB. The EDB holds the account content for everyone in the mail domain; so unless the case is the exceedingly rare one that warrants production of <u>all</u> the e-mail, attachments, contacts and calendars for every user, no litigant hands over their EDB.

It may be possible to create an EDB that contains only messaging from selected custodians (and excludes privileged and non-responsive content) such that you could really, truly produce in a native form. But I've never seen it done that way, and I can't think of anything to commend it over simpler approaches.

So, if you're not going to produce in the "true" native format of EDB, the desirable alternatives left to you are properly called "near-native," meaning that they preserve the requisite content and essential functionality of the native form but aren't the native form. If an alternate form doesn't preserve content and functionality, you can call it whatever you want. I lean toward "garbage," but to each his own.

E-mail is a species of ESI that doesn't suffer as mightily as, say, Word documents or Excel spreadsheets when produced in non-native forms. If one were meticulous in their text extraction, exacting in their metadata collection and careful in their load file construction, one could produce Exchange content in a way that's sufficiently complete and utile as to make a departure from the native less problematic—assuming, of course, that one produces the attachments in their native forms. That's a lot of "ifs," and what will emerge is sure to be incompatible with e-mail client applications and native review tools.

Litmus Test: Perhaps we have the makings of a litmus test to distinguish functional near-native forms from dysfunctional forms like TIFF images and load files: ***Can the form produced be imported into common e-mail client or server applications?***

You must admire the simplicity of such a test. If the e-mail produced is so distorted that not even e-mail programs can recognize it as e-mail, that's a fair and objective indication that the form of production has strayed too far from its native origins.

Gmail

The question whether it's feasible to produce Gmail in its native form triggered an order by U.S. Magistrate Judge Mark J. Dinsmore in a case styled, *Keaton v. Hannum,* 2013 U.S. Dist. LEXIS 60519 (S.D. Ind. Apr. 29, 2013). It's a seamy, sad suit brought *pro se* by an attorney named Keaton against both his ex-girlfriend, Christine Zook, and the cops who arrested Keaton for stalking Zook. It got my attention because the court cited a blog post I made some years ago. The Court wrote:

> Zook has argued that she cannot produce her Gmail files in a .pst format because no native format exists for Gmail (i.e., Google) email accounts. The Court finds this to be incorrect based on Exhibit 2 provided by Zook in her Opposition Brief. [Dkt. 92 at Ex. 2 (Ball, Craig: Latin: To Bring With You Under Penalty of Punishment, EDD Update (Apr. 17, 2010)).] Exhibit 2 explains that, although Gmail does not support a "Save As" feature to generate a single message format or PST, the messages can be downloaded to Outlook and saved as .eml or.msg files, or, as the author did, generate a PDF Portfolio – "a collection of multiple files in varying format that are housed in a single, viewable and searchable container." [Id.] In fact, Zook has already compiled most of her archived Gmail emails between her and Keaton in a .pst format when Victim.pst was created. It is not impossible to create a "native" file for Gmail emails.

Id. at 3.

I'm gratified when a court cites my work, and here, I'm especially pleased that the Court took an enlightened approach to "native" forms in the context of e-mail discovery. Of course, one strictly defining "native" to exclude near-native forms might be aghast at the loose lingo; but the more important takeaway from the decision is the need to strive for the most functional and complete forms when true native is out-of-reach or impractical.

Gmail is a giant database in a Google data center someplace (or in many places). I'm sure I don't know what the native file format for cloud-based Gmail might be. Mere mortals don't get to peek at the guts of Google. But, I'm also sure that it doesn't matter, because even if I *could* name the native file format, I couldn't obtain that format, nor could I faithfully replicate its functionality locally.[79]

Since I can't get "true" native, how can I otherwise mirror the completeness and functionality of native Gmail? After all, a litigant doesn't seek native forms for grins. A litigant seeks native forms to secure the unique benefits native brings, principally functionality and completeness.

[79] It was once possible to create complete, offline replications of Gmail using a technology called Gears; however, Google discontinued support of Gears some time ago. Gears' successor, called "Gmail Offline for Chrome," limits its offline collection to just a month's worth of Gmail, making it a complete non-starter for e-discovery. Moreover, neither of these approaches employs true native forms as each was designed to support a different computing environment.

There are a range of options for preserving a substantial measure of the functionality and completeness of Gmail. One would be to produce in Gmail.

HUH?!?!

Yes, you could conceivably open a fresh Gmail account for production, populate it with responsive messages and turn over the access credentials for same to the requesting party. That's probably as close to true native as you can get (though some metadata will change), and it flawlessly mirrors the functionality of the source. Still, it's not what most people expect or want. It's certainly not a form they can pull into their favorite e-discovery review tool.

Alternatively, as the Court noted in *Keaton v. Hannum,* an IMAP[80] capture to a PST format (using Microsoft Outlook or a collection tool) is a practical alternative. The resultant PST won't look or work exactly like Gmail (i.e., messages won't thread in the same way and flagging will be different); but it will supply a large measure of the functionality and completeness of the Gmail source. Plus, it's a form that lends itself to many downstream processing options.

So, What's the native form of that e-mail?

Which answer do you want; the technically correct one or the helpful one? No one is a bigger proponent of native production than I am; but I'm finding that litigants can get so caught up in the quest for native that they lose sight of what truly matters.

Where e-mail is concerned, we should be less captivated by the term "native" and more concerned with specifying the actual form or forms that are best suited to supporting what we need and want to do with the data. That means understanding the differences between the forms (e.g., what information they convey and their compatibility with review tools), not just demanding native like it's a brand name.

[80] IMAP (for Internet Message Access Protocol) is another way that e-mail client and server applications can talk to one another. The latest version of IMAP is described in RFC 3501. IMAP is not a form of e-mail storage; it is a means by which the structure (i.e., foldering) of webmail collections can be replicated in local mail client applications like Microsoft Outlook. Another way that mail clients communicate with mail servers is the Post Office Protocol or POP; however, POP is limited in important ways, including in its inability to collect messages stored outside a user's Inbox. Further, POP does not replicate foldering. Outlook "talks" to Exchange servers using MAPI and to other servers and webmail services using MAPI (or via POP, if MAPI is not supported).

When I seek “native” for a Word document or an Excel spreadsheet, it’s because I recognize that the entire native file—and *only* the native file—supports the level of completeness and functionality I need, a level that can’t be fairly replicated in any other form. But when I seek native production of e-mail, I don’t expect to receive the entire “true” native file. I understand that responsive and privileged messages must be segregated from the broader collection and that there are a variety of near native forms in which the responsive subset can be produced so as to closely mirror the completeness and functionality of the source.

When it comes to e-mail, what matters most is getting all the valuable information within and about the message in a fielded form that doesn’t completely destroy its character as an e-mail message.

So, let’s not get *too* literal about native forms when it comes to e-mail. Don’t seek native to prove a point. Seek native to prove your case.

____

Postscript: When I publish an article extolling the virtues of native production, I usually get a comment or two saying, “TIFF and load files are good enough.” I can’t always tell if the commentator means “good enough to fairly serve the legitimate needs of the case” or “good enough for those sleazy bastards on the other side.” I suspect they mean both. Either way, it might surprise readers to know that, when it comes to e-mail, I agree with the first assessment…with a few provisos.

First, TIFF and load file productions can be good enough for production of e-mail if no one minds paying more than necessary. It generally costs more to extract text and convert messages to images than it does to leave it in a native or near-native form. But that’s only part of the extra expense. TIFF images of messages are MUCH larger files than their native or near native counterparts. With so many service providers charging for ingestion, processing, hosting and storage of ESI on a per-gigabyte basis, those bigger files continue to chew away at both side's bottom lines, month-after-month.

Second, TIFF and load file productions are good enough for those who only have tools to review TIFF and load file productions. There’s no point in giving light bulbs to those without electricity. On the other hand, just because you don't pay your light bill, must I sit in the dark?

Third, because e-mails and attachments have the unique ability to be encoded entirely in plain text, a load file can carry the complete contents of a message and its contents as RFC 5322-compliant text accompanied by MAPI metadata fields. It's one of the few instances where it's possible to furnish a load file that simply and genuinely compensates for most of the shortcomings of TIFF productions. Yet, it's not done.

Finally, TIFF and load file productions are good enough for requesting parties who just don't care. A lot of requesting parties fall into that category, and they're not looking to change. They just want to get the e-mail, and they don't give a flip about cost, completeness, utility, metadata, efficiency, authentication or any of the rest. If both sides and the court are content not to care, TIFF and load files really are good enough.

❁ Exercise 13: E-Mail Anatomy

GOALS: The goals of this exercise are for the student to:

1. Delve into the anatomy of an e-mail message, identifying its essential components.

OUTLINE: Students will create and transmit an e-mail message and explore its structure.

Background

In addition to being the most sought-after ESI in electronic discovery, e-mail is one of the oldest computer networking applications. Before people were sharing printers, and long before the internet was a household word, people were sending e-mail across networks. That early e-mail was plain text, also called ASCII text or 7-bit (because you needed just seven bits of data, one less than a byte, to represent each ASCII character). In those days, there were no attachments, no pictures, not even simple enhancements like **bold**, *italic* or underline.

As previously discussed, early e-mail was something of a chaotic situation, implemented differently by different systems. So the fledgling internet community circulated proposals seeking a standard. They stuck with plain text in order that older messaging systems could talk to newer systems. These proposals were called **Requests for Comment** or **RFC**s, and they came into widespread use as much by convention as by adoption (the internet being a largely anarchic realm). The RFCs lay out the form an e-mail should adhere to in order to be compatible with e-mail systems.

The RFCs concerning e-mail have gone through several major revisions since the first one circulated in 1973. The latest protocol revision is called RFC 5322 (2008), which made obsolete RFC 2822 (2001) and its predecessor, RFC 822 (1982). Another series of RFCs (RFC 2045-47, RFC 4288-89 and RFC 2049), collectively called **Multipurpose Internet Mail Extensions** or **MIME**, address ways to graft text enhancements, foreign language character sets and multimedia content onto plain text emails. These RFCs establish the form of the billions upon billions of e-mail messages that cross the internet.

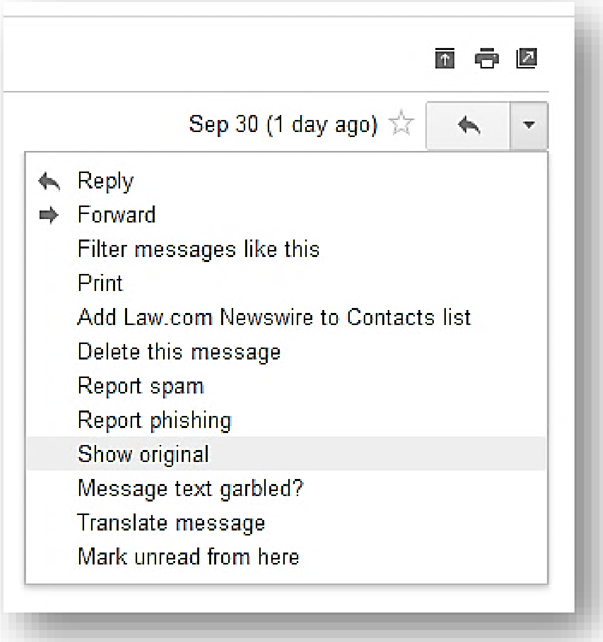
In this exercise, we will get examine the structure of e-mail as dictated by the RFC and MIME standards. This exercise should take no more than about 15 minutes to complete.

Step 1: Draft and transmit a message with a tiny attachment

Using your e-mail account of choice, draft an e-mail message to yourself. ***Optimally, send the message to an alternate e-mail address than the one you use to transmit the file.***[81] Keep the body of the body of the e-mail short and impersonal (as you will be including same in your submission of your work for grading). Attach any very small (<5kb) gif or jpg image file to the e-mail.[82] Be sure to use a small image file because you're going to paste the entire message into a text document and you don't want that document to run to pages and pages of base64 encoding. Send the e-mail you drafted.

Step 2: Copy the Message Source and Save It

Now, find the *received* e-mail and access the message source. The method to do so varies according to the webmail service or mail client application used. For example, Gmail allows you to "Show original" by pulling down a menu near the time received at the upper right of each message (see illustration at right). If you use Apple's Mail client, go to View>Message>Raw Source. If you have trouble finding the message source in your mail client or account, run a Google search for "view message source in *X*," where *X* is the name of your mail client (e.g., Outlook) or service (e.g., Hotmail).



When you get to the source, be sure it includes the message header, message body and the attachment in base64, then select the entire message source and paste it into a blank document (use Notepad in Windows or TextEdit on a Mac). Avoid using a Microsoft Word document; but, if you must use MS Word, change the font to Lucinda Console and use narrow margins so the Base64 content has a straight right edge.

Now, save the text document you've just created as ***Your_Surname_Exercise 19.txt***.

Step 3: Dissect the First Message

Open another blank document or e-mail to use as a place to paste information. You can handwrite the information in the blanks below, but it's much easier to copy and paste the data electronically into a file or e-mail you submit.

[81] When you send a message to yourself at the same mail account, the message need not traverse the Internet. Consequently, some of the features we seek to explore in this exercise will be absent. For example, if you send from Gmail, send it to your business or school account. If you have no alternate account, try sending it to a friend's or family member's account and have them forward the message back to you.

[82] If you can't find a sufficiently small image, use this one of Judge John Facciola in high school: http://craigball.com/fatch.jpg

Question 1:
Your e-mail should be in MIME. From the message source, what is the MIME-Version? __________

Boundaries: The various parts of a MIME multipart message are defined by boundaries, usually long character strings required to be unique in each message. Your e-mail message should have at least two different boundary values, each preceded by the statement, "**boundary=**." When used as separators, each boundary will be preceded by two hyphens, and the last usage of each boundary will be followed by two hyphens. The information above the first boundary definition is the "**Message Header**." Note that the message header contains the important To, From, Subject and Date information for the message.

Question 2:
Identify the first two unique boundaries in your message and fill them in below (better yet, copy and paste them into an electronic document):

First Boundary: ____

Second Boundary: ____

Note how the first boundary value serves to separate the three main sections of the message (Header, Message Body and Attachment) and the second boundary value separates the alternate message body types (i.e., Text/Plain and Text/HTML).

Message IDs: According to the RFC mail specifications, each message transmitted via e-mail should incorporate a unique message identifier value called "Message-ID."

Question 3:
Find the Message-ID value in your message and record it below: (or copy and paste, etc.):

Message-ID: ____

Evidence Tip: *Many forged e-mail messages are contrived by altering the message bodies of genuine messages. Forgers often overlook the Message-ID value, making it possible to detect the forgery and identify the genuine message that was altered.*

Attachments: Drop down to the last section of your message source containing the Base64 encoded image (look for the next to last usage of the boundary and "Content-Type: image/*type of image you attached;"* followed by the name of the image file you attached in quotes).

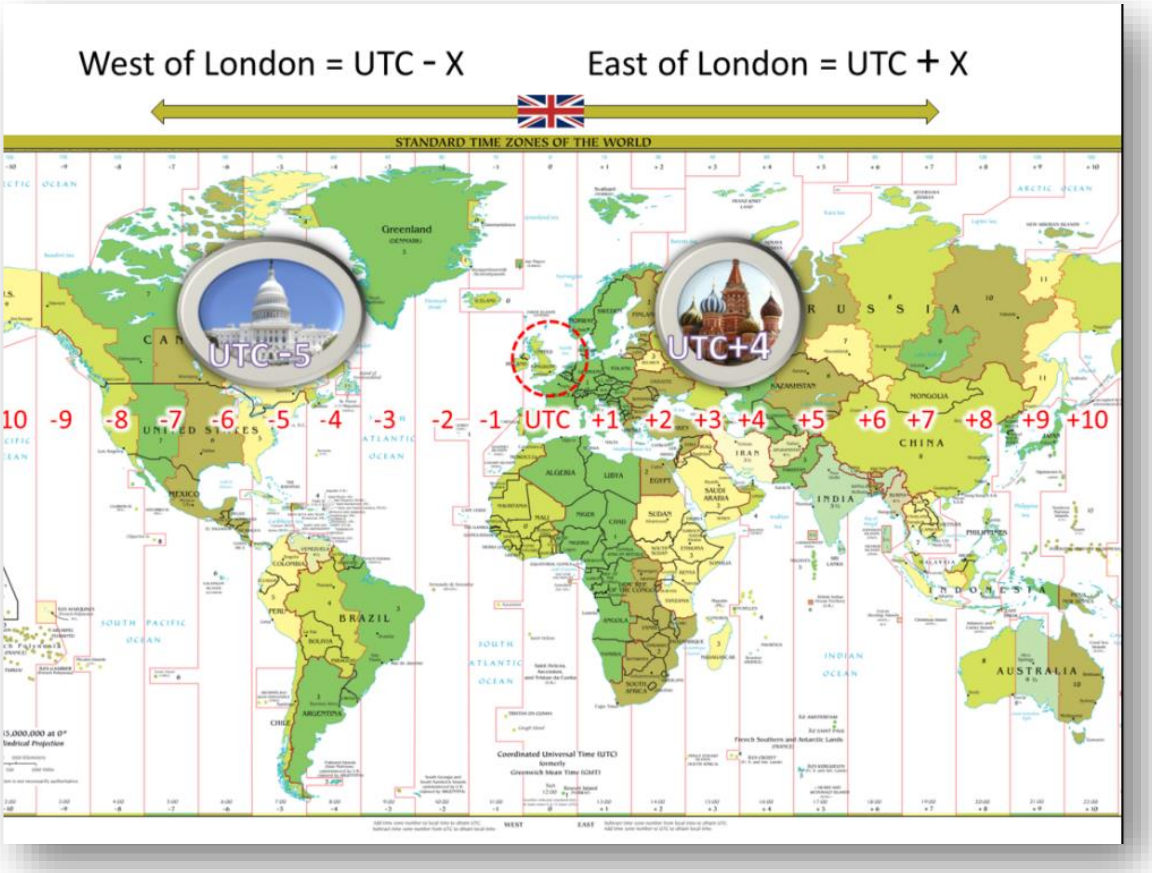
Question 4:
Apart from the name of the attached image file, do you see any other system metadata values *for the image*, such as Date Modified or Date Created? (yes or no): ____________________
If yes, record them here: ____

Coordinated Universal Time (UTC): Time values in e-mail message headers are often expressed with reference to Coordinated Universal Time (*Temps Universel Coordonné* in French), the primary world time standard. UTC is often called Zulu time by the military and aviators and is essentially the same as Greenwich Mean Time (GMT), the time on the official clock located at the Royal Observatory in Greenwich, England. A **UTC offset** value expresses the difference between the stated local time and UTC, allowing messages to be **normalized** on a consistent time line, notwithstanding differing time zones and daylight savings time settings.

Question 5:

Look in the header of your message source and identify all UTC offset values present. These will be expressed as negative values (e.g., -0400 for a message sent from a machine set to EDT):

SENT UTC OFFSET: _______ RECEIVED UTC OFFSET: _______ OTHER UTC OFFSET: _____



⚛ Exercise 14: Encoded Time Values in Message Boundaries

GOALS: The goals of this exercise are for the students to:

1. Apply their knowledge of encoding, e-mail boundaries, decimal and hexadecimal to determine whether an e-mail is legitimate or forged.
2.

OUTLINE: Students will decode e-mail boundaries to assess the integrity of an attachment.

"Time heals all wounds." "Time is money." "Time flies."

To these memorable mots, I add one more: "*Time is truth*."

A defining feature of electronic evidence is its connection to temporal metadata or *timestamps*. Electronically stored information is frequently described by time metadata denoting when ESI was created, modified, accessed, transmitted, or received. *Clues to time are clues to truth* because temporal metadata helps establish and refute authenticity, accuracy, and relevancy.

But in the realms of electronic evidence and digital forensics, time is tricky. It hides in peculiar places, takes freakish forms, and doesn't always mean what we imagine. Because time is truth, it's valuable to know *where* to find temporal clues and *how* to interpret them correctly.

Everyone who works with electronic evidence understands that files stored in a Windows (NTFS) environment are paired with so-called "MAC times," which have nothing to do with Apple Mac computers or even the *MAC address* identifying a machine on a network. In the context of time, MAC is an initialization for **M**odified, **A**ccessed and **C**reated times.

That doesn't *sound* tricky. Modified means changed, accessed means opened and created means authored, right? *Wrong.* A file's modified time can change due to actions neither discernible to a user nor reflective of user-contributed edits. Accessed times change from events (like a virus scan) that most wouldn't regard as accesses. Moreover, Windows stopped reliably updating file access times way back in 2007 when it introduced the Windows Vista operating system. Created *may* coincide with the date a file is authored, but it's as likely to flow from the copying of the file to new locations and storage media ("created" meaning *created in that location*). Copying a file in Windows produces an object that appears to have been created *after* it's been modified!

it's crucial to protect the integrity of metadata in e-discovery, so changing file creation times by copying is a big no-no. Accordingly, e-discovery collection and processing tools perform the nifty trick of changing MAC times on copies to match times on the files copied. Thus, targeted collection alters every file collected, but done correctly, original metadata values are restored, and hash values don't change. *Remember:* system metadata values aren't stored within the file they describe so system metadata values aren't included in the calculation of a file's hash value. The upshot is that **changing a file's system metadata values—including its filename and MAC times—doesn't affect the file's hash value.**

Conversely and ironically, opening a Microsoft Word document without making a change to the file's contents *can* change the file's hash value when the application updates internal metadata like the editing clock. Yes, there's even a timekeeping feature in Office applications!

Other tricky aspects of MAC times arise from the fact that time means nothing without place. When we raise our glasses with the justification, "It's five o'clock somewhere," we are acknowledging that time is a ground truth. "Time" means time *in a time zone*, adjusted for daylight savings and expressed as a *UTC Offset* stating the number of time zones ahead of or behind GMT, time at the Royal Observatory in Greenwich, England atop the Prime or "zero" Meridian.

Time values of computer files are typically stored in UTC, for *Coordinated Universal Time*, essentially Greenwich Mean Time (GMT) and sometimes called Zulu or "Z" time, military shorthand for zero meridian time. When stored times are displayed, they are adjusted by the computer's operating system to conform to the user's local time zone and daylight savings time rules. So, in e-discovery and computer forensics, it's essential to know if a time value is a local time value adjusted for the location and settings of the system or if it's a UTC value. The latter is preferred in e-discovery because it enables time normalization of data and communications, supporting the ability to order data from different locales and sources across a uniform timeline.

Time values are especially important to the reliable ordering of email communications. Most e-mails are conversational threads, often a mishmash of "live" messages (with their rich complement of header data, encoded attachments and metadata) and embedded text strings of older messages. If the senders and receivers occupy different time zones, the timeline suffers: replies precede messages that prompted them, and embedded text strings make it child's play to alter times and text. It's just one more reason I always seek production of e-mail evidence in native and near-native forms, not as static images. Mail headers hold data that support authenticity and integrity—data you'll never see produced in a load file.

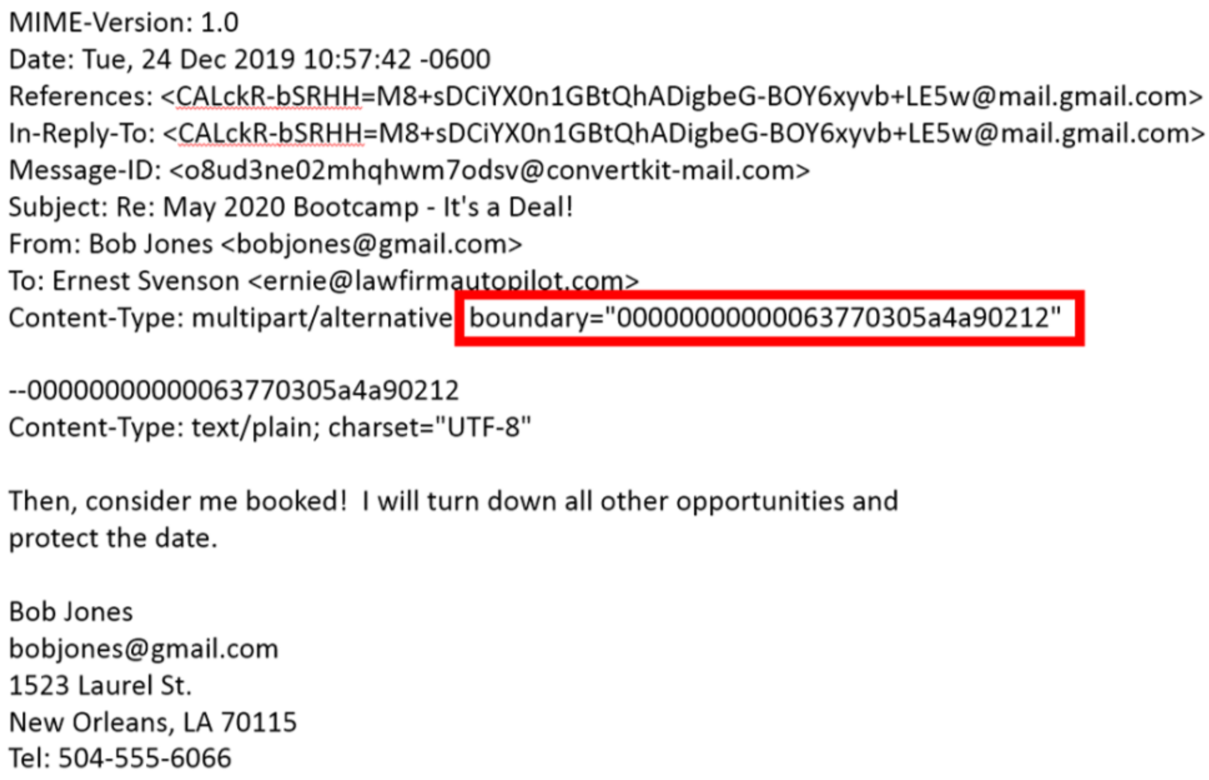
Underscoring that last point, the next exercise explores *time values embedded in mail boundaries.*

Time Values Embedded in Mail Boundaries

If you know where to look in digital evidence, you'll find time values hidden like Easter eggs. When time values are absent or untrustworthy, forensic examiners draw on hidden time values—or, more accurately, *encoded* time values—to construct timelines or reveal forgeries.

E-mail must adhere to structural conventions to traverse the internet and be understood by different e-mail programs. One of these conventions is the use of a Content-Type declaration and setting of content *boundaries*, enabling systems to distinguish the message header region from the message body and attachment regions.

The next illustration is a snippet of simplified code from a forged Gmail message. To see the underlying code of a Gmail message, users can select “Show original” from the message options drop-down menu (*i.e.*, the ‘three dots’).

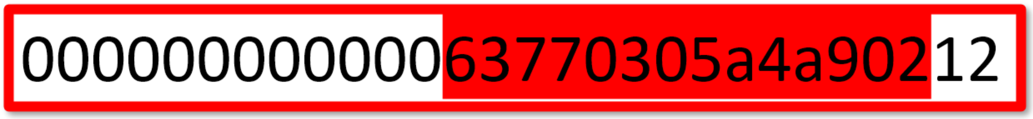


```
MIME-Version: 1.0
Date: Tue, 24 Dec 2019 10:57:42 -0600
References: <CALckR-bSRHH=M8+sDCiYX0n1GBtQhADigbeG-BOY6xyvb+LE5w@mail.gmail.com>
In-Reply-To: <CALckR-bSRHH=M8+sDCiYX0n1GBtQhADigbeG-BOY6xyvb+LE5w@mail.gmail.com>
Message-ID: <o8ud3ne02mhqhwm7odsv@convertkit-mail.com>
Subject: Re: May 2020 Bootcamp - It's a Deal!
From: Bob Jones <bobjones@gmail.com>
To: Ernest Svenson <ernie@lawfirmautopilot.com>
Content-Type: multipart/alternative boundary="000000000000063770305a4a90212"

--000000000000063770305a4a90212
Content-Type: text/plain; charset="UTF-8"

Then, consider me booked!  I will turn down all other opportunities and
protect the date.

Bob Jones
bobjones@gmail.com
1523 Laurel St.
New Orleans, LA 70115
Tel: 504-555-6066
```

The line partly outlined in red advises that the message will be “multipart/alternative,” indicating that there will be multiple versions of the content supplied; commonly a plain text version followed by an HTML version. To prevent confusion of the boundary designator with message text, a complex sequence of characters is generated to serve as the content boundary. The boundary is declared to be “000000000000063770305a4a90212” and delineates a transition from the header to the plain text version (shown) to the HTML version that follows (not shown).

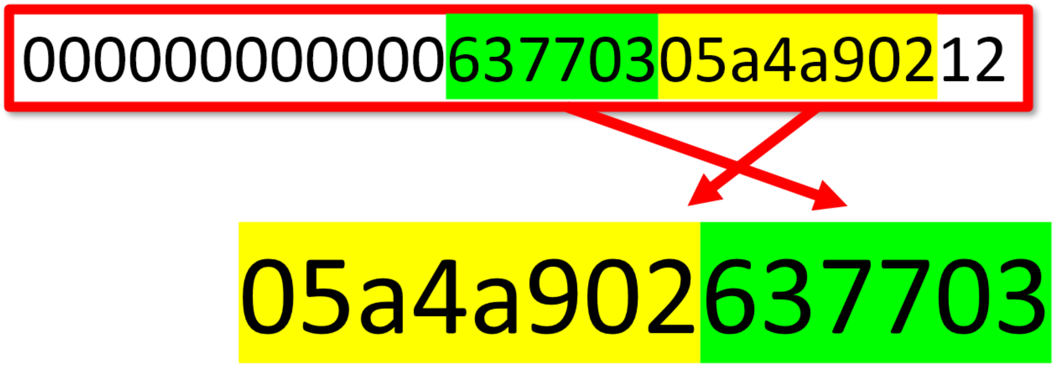
Thus, a boundary’s sole *raison d’être* is to separate parts of an e-mail; but because a boundary must be unique to serve its purpose, programmers insure against collision with message text (and other boundaries) by integrating time data into the boundary text. Now, watch how we decode that time data.

Here’s our boundary, and I’ve highlighted fourteen hexadecimal characters in red:



000000000000063770305a4a90212

Next, I've parsed the highlighted text into six- and eight-character strings, reversed their order and concatenated the strings to create a new hexadecimal number:



A decimal number is Base 10. A *hexa*decimal number is Base 16. They are merely different ways of notating numeric values. So, 05a4a902637703 is just a really big number. If we convert it to its decimal value, it becomes: 1,588,420,680,054,531. That's 1 quadrillion, 588 trillion, 420 billion, 680 million, 54 thousand, 531. Like I said, a BIG number.

But a big number…of *what*?

Here's where it gets amazing (or borderline insane, depending on your point of view).

It's the number of microseconds that have elapsed since January 1, 1970 (midnight UTC), not counting leap seconds. A microsecond is a millionth of a second, and 1/1/1970 is the "Epoch Date" for the Unix operating system. An Epoch Date is the date from which a computer measures system time. Some systems resolve the Unix timestamp to seconds (10-digits), milliseconds (13-digits) or microseconds (16-digits).

When you make that curious calculation, the resulting date proves to be Saturday, May 2, 2020 6:58:00.054 AM UTC-05:00 DST. That's the genuine date and time the forged message was sent. It's not magic; it's just math.

Had the timestamp been created by the Windows operating system, the number would signify *the number of 100 nanosecond intervals between midnight (UTC) on January 1, 1601 and the precise time the message was sent.*

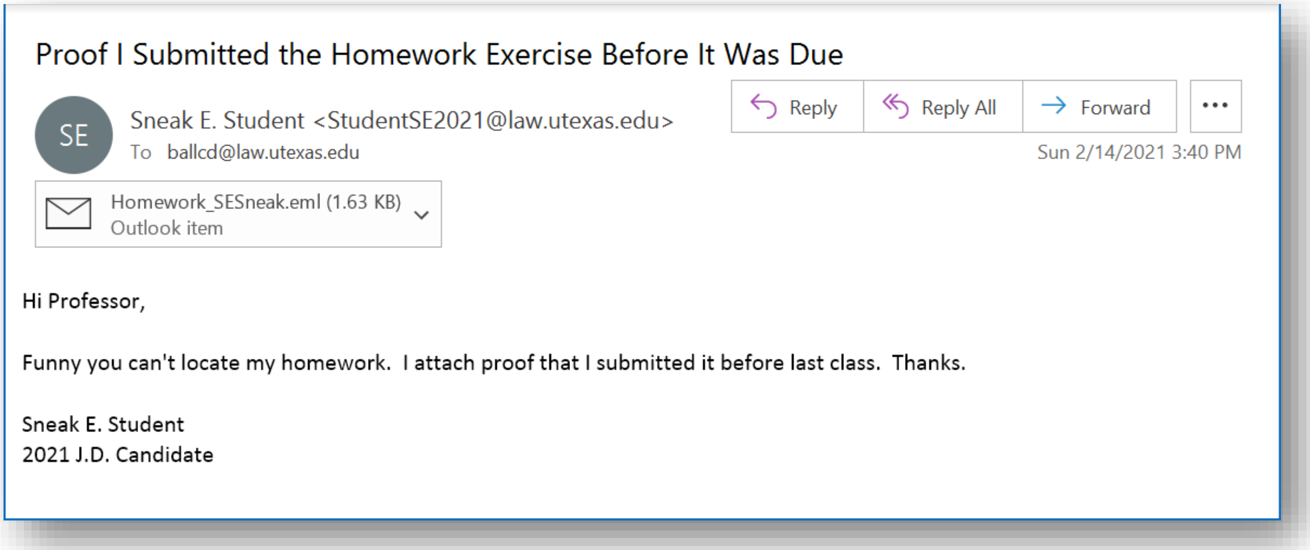
Why January 1, 1601? Because that's the "Epoch Date" for Microsoft Windows. Again, an Epoch Date is the date from which a computer measures system time. Unix and POSIX elected to measure time in seconds from January 1, 1970. Apple used one second intervals since January 1, 1904, and MS-DOS used seconds since January 1, 1980. Windows went with 1/1/1601 because, when the Windows operating system was being designed, we were in the first 400-year cycle of the Gregorian calendar (implemented in 1582 to replace the Julian calendar). Rounding up to the start of the first full century of the 400-year cycle made the math cleaner.

Timestamps are everywhere in e-mail, hiding in plain sight. You'll find them in boundaries, message IDs, DKIM stamps and SMTP IDs. Each server handoff adds its own timestamp. It's the rare e-mail forger who will find *every* embedded timestamp and correctly modify them all to conceal the forgery.

Exercise 14: Background

It's Sunday morning, February 14, 2021 and, Sneak E. Student, didn't turn in the *Find the Secret Word* exercise due before class on February 10, 2021. When asked about it, Sneak swears he e-mailed his answer right before class and promises to forward a copy of the original transmittal to prove it.

That afternoon, Sneak E. Student, emails the following:



Proof I Submitted the Homework Exercise Before It Was Due

Sneak E. Student <StudentSE2021@law.utexas.edu>
To ballcd@law.utexas.edu

Sun 2/14/2021 3:40 PM

Homework_SESneak.eml (1.63 KB)
Outlook item

Hi Professor,

Funny you can't locate my homework. I attach proof that I submitted it before last class. Thanks.

Sneak E. Student
2021 J.D. Candidate

Here's the message source:

[Server path data intentionally omitted]

```
MIME-Version: 1.0
Date: Sun, 14 Feb 2021 15:40:02 -0600
Message-ID: <CALckR-aWbcreMVY0-
A0E2r4LVkOUy3QU42qn7OgnjLSEBLqi2Q@mail.gmail.com>
Subject: Proof I Submitted the Homework Exercise Before It Was Due
From: Craig Ball <craig@ball.net>
To: ballcd@utexas.edu
Content-Type: multipart/mixed; boundary="000000000000762e5205bb52b5dd"

--000000000000762e5205bb52b5dd
Content-Type: multipart/alternative; boundary="000000000000762e4f05bb52b5db"

--000000000000762e4f05bb52b5db
Content-Type: text/plain; charset="UTF-8"

Hi Professor,
Funny you can't locate my homework.  I attach proof that I submitted it
before last class.  Thanks.

Sneak E. Student
2021 J.D. Candidate

--000000000000762e4f05bb52b5db
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

<div dir=3D"ltr">Hi Professor,<div dir=3D"ltr" class=3D"gmail_signature" da=
ta-smartmail=3D"gmail_signature"><div dir=3D"ltr"><div dir=3D"ltr"><div dir=
=3D"ltr"><div dir=3D"ltr"><div dir=3D"ltr"><div dir=3D"ltr"><div dir=3D"ltr=
"><div dir=3D"ltr"></div></div></div></div></div></div></div></div></div><d=
iv><br></div><div>Funny you can&#39;t locate my homework.=C2=A0 I attach pr=
oof that I submitted it before last class.=C2=A0 Thanks.</div><div><br></di=
v><div>Sneak E. Student</div><div>2021 J.D. Candidate</div></div>

--000000000000762e4f05bb52b5db--
--000000000000762e5205bb52b5dd
Content-Type: message/rfc822; name="Homework_SESneak.eml"
Content-Disposition: attachment; filename="Homework_SESneak.eml"
X-Attachment-Id: f_kl5o9k6v0
Content-ID: <f_kl5o9k6v0>

MIME-Version: 1.0
Date: Wed, 10 Feb 2021 14:39:26 -0600
References: <CAJVHkuuwoSEgdJcq9i0BxVEks0txdqRNDB_OkPe_D-
G35aGzmA@mail.gmail.com>
In-Reply-To: <CAJVHkuuwoSEgdJcq9i0BxVEks0txdqRNDB_OkPe_D-
G35aGzmA@mail.gmail.com>
Message-ID: <CALckR-Y6XoDJZ-2Mo-nL_z-yKTP_ySM7G2pJQGL0MNfv-
vZm9Q@mail.gmail.com>
Subject: Re: Here are my Answers to the Homework
From: Sneak E. Student <studentS2021@law.ut.edu>
To: Craig Ball <craig@ball.net>
Content-Type: multipart/mixed;
      boundary="A_0000000000003376a205bb4c5824"

--A_0000000000003376a205bb4c5824
Content-Type: multipart/alternative;
      boundary="B_0000000000003376a205bb4c5824"


--B_0000000000003376a205bb4c5824
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

Here you go, Professor.  Just under the wire!

Sneak E. Student
J.D Candidate 2021

--B_0000000000003376a205bb4c5824
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<html>
<head>
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Diso-8859-=
1">
<style type=3D"text/css" style=3D"display:none;"> P {margin-top:0;margin-bo=
ttom:0;} </style>
```

```
</head>
<body dir=3D"ltr">
<div style=3D"font-family: Calibri, Arial, Helvetica, sans-serif; font-size=
: 12pt; color: rgb(0, 0, 0);">
Here you go, Professor. &nbsp;Just under the wire!
<div><br>
</div>
<div>Sneak E. Student</div>
J.D &nbsp;Candidate 2021<br>
</div>
</body>
</html>

--B_000000000003376a205bb4c5824--

--A_000000000003376a205bb4c5824
Content-Type: application/msword; name="SStudent_secret word answer.rtf"
Content-Description: SStudent_secret word answer.rtf
Content-Disposition: attachment; filename="SStudent_secret word answer.rtf";
      size=214; creation-date="Wed, 10 Feb 2021 08:39:26 GMT";
      modification-date="Wed, 10 Feb 2021 08:39:26 GMT"
Content-Transfer-Encoding: base64

e1xydGYxXGFuc2lcYW5zaWNwZzEyNTJcZGVmZjBcbm91aWNvbXBhdFxkZWZsYW5nMTAzM3tcZm9u
dHRibHtcZjBcZm5pbFxmY2hhcnNldDAgQ2FsaWJyaTt9fQ0Ke1wqXGdlbmVyYXRvciBSaWNoZWQy
MCA2LjMuOTYwMH1cdmlld2tpbmQ0XHVjMSANClxwYXJkXHNhMjAwXHNsMjc2XHNsbXVsdDFcZjBc
ZnMyMlxsYW5nOSBUaGUgc2VjcmV0IHdvcmQgaXMgB6YXJmXHBhcg0KfQ0KAA==

--A_000000000003376a205bb4c5824--
--000000000000762e5205bb52b5dd–
```

Exercise 14: Investigate the Integrity of the Digital Evidence

Using timestamps embedded in message boundaries, can you determine if Sneak E. Student submitted his answer to the Find the Secret Word assignment before 2:40pm CST on February 10, 2021?

Steps to Solve This Exercise:

1. Select the correct message boundary value to analyze for its embedded timestamp.
2. Extract and compose the hexadecimal timestamp in the boundary.
3. Convert the hexadecimal timestamp to a decimal timestamp value.
4. Determine the date and time the timestamp was created.

Step 1: Select the correct message boundary value.

Putting our knowledge of email anatomy to work, we see from the message source that Sneak sent an e-mail dated **Wed, 10 Feb 2021 14:39:26 -0600** as an attachment to an e-mail dated **Sun, 14 Feb 2021 15:40:02 -0600.** The earlier e-mail holds its own attachment: a rich text format (RTF) text file named **SStudent_secret word answer.rtf.**

Both the attached e-mail message and its own embedded attachment reflect February 10 dates and times which, if accurate, suggest a prior timely submission of the homework assignment. By design, e-mails are made up entirely of plain text so they can traverse even the oldest legacy e-mail systems. Accordingly, the date and time values are plain text and can be easily altered with a text editor. If that's all we had to go on, we would be forced to conclude that the assignment was timely.

Still, we know that e-mails may contain timestamps encoded in message boundaries revealing the genuine date of a forged communication. If we look at the timestamps, we may be able to find evidence corroborating or refuting the plain text dates. However, there are four distinct boundaries declared in the message source.

```
boundary="000000000000762e5205bb52b5dd"
boundary="000000000000762e4f05bb52b5db"
boundary="A_0000000000003376a205bb4c5824"
boundary="B_0000000000003376a205bb4c5824"
```

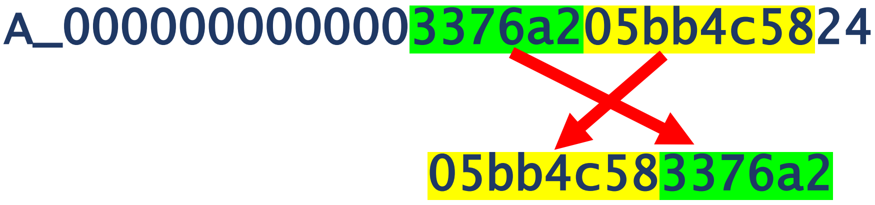
There's no cause to question the timing of the February 14 transmittal, so we would expect those boundaries to resolve to February 14 timestamps. Instead, we want to look at the boundaries in the e-mail attachment purporting to be created on February 10, 2021. Because the purportedly older e-mail message is an attachment to a subsequent transmittal, we should not be surprised to find that the boundaries of the attachment are flanked by the boundaries of the transmittal.

The boundaries declared in the attachment are:

```
boundary="A_0000000000003376a205bb4c5824"
boundary="B_0000000000003376a205bb4c5824"
```

Step 2: Extract and compose the hexadecimal timestamp in the boundary.

We want to parse the boundary string in the manner demonstrated above, dropping the two final characters and creating a concatenated string of the preceding eight and six characters in the boundary, like so:



Our hexadecimal time stamp is **05bb4c583376a2**.

Step 3: Convert the hexadecimal timestamp to a decimal timestamp value.

Now, you must determine the decimal equivalent of the hexadecimal value **05bb4c583376a2.** You *could* do this manually, of course, but it's must faster to use a conversion tool. You could download a conversion app for the task, or you can locate an online converter. If you Google *Hex to Decimal Converter*, many suitable options come up. I used the calculator at https://www.rapidtables.com/convert/number/hex-to-decimal.html, but again, any converter will do.

Exercise 14A: What is the decimal value of the hex value 05bb4c583376a2?

Answer: ____

Step 4: Determine the date and time the timestamp was created.

With luck, the decimal number you've just determined reflects the number of microseconds that elapsed between the Epoch date of January 1, 1970 and the time the timestamp was created. You could make such a calculation manually knowing there are 86,400,000,000 microseconds in a day, but that's the sort of task where computers excel. So, again, we can turn to the web to find a suitable conversion tool. If you Google, say, "convert decimal to epoch date," you'll find a range of online tools that will complete the conversion from a decimal value to a date value. I used https://www.epochconverter.com/ but any should do.

Exercise 14B: What is the date and time of the decimal value in your answer to 14A, above? Make sure the date you supply is expressed as U.S. Central Standard Time, not GMT or UTC.

Answer: ____

⚛ Exercise 15: Forensic Imaging

GOALS: The goals of this exercise are for the student to:

1. Address distinctions between forensically sound imaging versus copies, clones and targeted collection; and

2. Use a forensic imaging utility to create and validate a forensically sound image of evidence media.

CAVEAT: The techniques and tools employed in this exercise should not be employed against real evidence in connection with pending or contemplated litigation because crucial steps (like write blocking) required to protect the integrity of the evidence have been omitted for expediency.

When users empty deleted files from Windows recycle bins, they aren't gone. The operating system simply ceases to track them, freeing the clusters the deleted data occupies for reallocation to new files. Eventually, these unallocated clusters may be reused, and their contents overwritten, but until that happens, Windows turns a blind eye to them. Because Windows only sees active data, it only copies active data. Forensically sound preservation safeguards the entire drive—data and metadata in allocated clusters PLUS artifacts in slack space and unallocated clusters--including the deleted data they hold.

Accordingly, think of the Three Commandments of forensically sound preservation as:

1. Don't alter the evidence;
2. Accurately and thoroughly replicate the contents; and
3. Prove the preceding objectives were met.

These standards cannot be met in every situation—notably, in the logical acquisition of a live server or physical acquisition of a phone or tablet device—but parties deviating from a "change nothing" standard should weigh, disclose and defend any deviation.

Distinguishing "Clones" and "Collections" from "Images"

Even lawyers steeped in electronic data discovery may confuse active file imaging (i.e., Ghosting or targeted collection), cloning and forensically sound imaging. You shouldn't. If someone suggests an active data duplicate is forensically sound, set them straight and reserve "forensically sound" to describe only processes preserving all the information on the media.

The terms “clone” and “image” are often used interchangeably, along with others like “bit stream copy” and “mirror.” So long as the duplicate is created in a forensically sound way and can be reliably verified to be so, the name attached to the duplicate doesn’t make much difference.

A “targeted collection” is merely the copying of some or all of *the active data* on an evidence drive. It omits the contents of unallocated clusters and file slack space, and typically leaves behind operating system artifacts of importance to forensic analysis. Accordingly, targeted collection may be sufficient for the limited purposes of e-discovery; but it’s almost always insufficient as a method to preserve data for forensic analysis.

The term “drive image” is most closely associated with a method of forensic duplication whereby all of the data and metadata on the source drive are stored in a file or series of files which, though structurally different from the source drive, can be reconstituted (“restored”) in such a way as to be a forensically-sound duplicate of the source drive. A drive image may be used with compression algorithms to store of the source drive data in a more compact fashion. Though a drive image is capable of being restored to create a clone drive, modern drive analysis software is designed to “virtually restore” the drive, reading directly from the image file and “seeing” the forensically sound duplicate drive without the necessity for restoration.

How do you make a “forensically-sound” duplicate of a drive?

Although forensic examiners use similar techniques and equipment, there is no one “recognized” or “approved” way to create a forensically-sound duplicate of a drive. There are several hardware and software tools well-suited to the task, each with strengths and weaknesses, but all are capable of creating a forensically-sound duplicate of a hard drive when used correctly. Keep in mind that there are many different types of digital media out there, and a tool well-suited to one may be incapable of duplicating another. You simply need to know what you are doing and match the tool to the task

Don’t Alter the Evidence

Forensic examiners are careful to prevent the preservation process from effecting changes to the source evidence in a process called “write blocking.” Write blocking is achieved in three principal ways:

1. By the use of write blocking hardware (devices) interposed between the evidence drive and any computer. The hardware write blocker intercepts any “writes” or changes that the computer might otherwise effect upon the evidence;
2. By the use of write blocking software tools designed to achieve the same prophylactic end; or
3. By the use of operating systems (like Linux) that can be reliably configured so as not to write to the evidence media.

We will forego the use of write blocking techniques in this exercise; however, a competent forensic examiner would not do so absent compelling justification. There is simply too much risk posed to the integrity of the evidence.

Accurately and Thoroughly Replicate the Contents

Approaches to forensically sound duplication range from generic software capable of producing a bit stream duplicate to custom-built applications exclusively for forensic drive duplication to handheld devices that automate nearly the entire process.

Duplication methodologies fall into two camps: those which create a drive image (a file or collection of files which can be virtually or physically restored to match the source evidence) and those which create a clone drive (a fully operational one-to-one copy of the evidence drive). Cloning was once common but is now an outmoded approach that has almost entirely given way to drive imaging. Again, done right, either approach works, but drive imaging enjoys significant advantages deriving from encapsulating the evidence data into authenticable image files unaffected by the file system for the target drive. A clone, being fully operational, is more prone to corruption of the evidence data it holds stemming from improper handling. Simply attaching a clone to a computer without the use of write blocking procedures may serve to destroy the integrity of the evidence.

Prove the Image is forensically sound

Considering the size of modern hard drives, one way you *can't* prove the validity of your duplicate is by manually comparing the data. It's just impossible. So, the process of verification must be automated and foolproof. To appreciate the solution, take a moment to ponder the problem: how can you examine billions or trillions of sectors on a duplicate drive and be certain that every one of them has precisely the same value and is in the exact same relative location as on the source drive? Not just be *certain* but be more reliably certain than fingerprints or DNA evidence. This is where we say "thanks" to the mathematical geniuses who gave up normal human interaction to dedicate their lives to algorithms, arrays and one-way computations. These are the brainiacs who thought up "hash functions" and "message digests."

Recall that a hash function accepts a value of any size as its input, performs a complex calculation on that input and returns a value of fixed length as its output. The output value functions as a unique representation of the input. Put in a complex "message" and out pops a long, fixed string of letters and number bearing no discernable relationship to the message but which can only be generated by that one input. Accordingly, the output is called a "message digest." The truly marvelous feature of this is that the computation only works in one direction ("one way"). It's considered "computationally infeasible" to decode the input from the output. Since the input message can be anything, someone had the very bright idea to use the entire contents of a hard drive or thumb drive as the input and—voila!—the output becomes a fingerprint of that drive's contents and layout. Change so much as one single bit somewhere on the drive and the message digest changes dramatically. Since the fingerprint is unique to the inputted message (here, the data on the drive) only a forensically-sound duplicate of the drive could generate the same message digest.

Most software and hardware tools sold for the purpose of forensically sound drive duplication integrate hashing for immediate authentication. These will typically create a file record or ancillary file memorializing the hash values of the source and target as well as other data identifying the evidence and demonstrating the integrity of the process.

Exercise 15: Creating a Forensic Image of a Thumb Drive

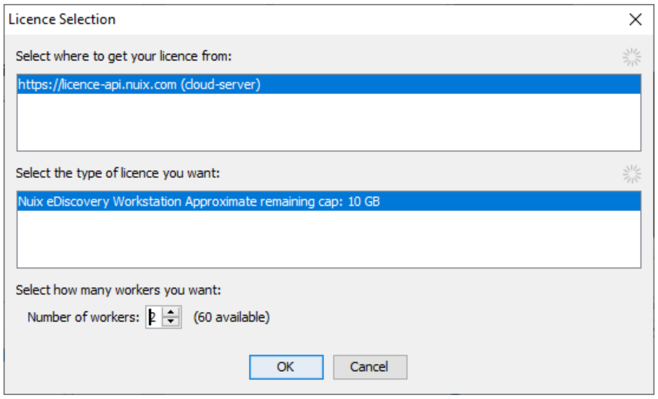
Step 1: Download and Install Nuix Imager

We will use two software tools from Nuix for two exercises this semester. These two applications, Nuix Imager and Nuix Workstation, can be found on your EVIDENCE thumb drive and in Canvas in **Files> 4_SOFTWARE INSTALLER FILES**. I've supplied versions for Windows and Mac. For this exercise, you must install the version of NUIX IMAGER suited to your system:

Windows:	nuix-imager-amd64-9.4.0.105.msi
Mac:	Nuix Imager 9.4.0.105.dmg

Step 2A: Attach the Evidence Thumb Drive and Run Nuix Imager

Plug the furnished "evidence" thumb drive into your machine.[83] Note the drive letter assigned to it by the computer at insertion. Launch Nuix Imager. When you get the screen at right, click OK. **Don't select more than two workers.**



Licence Selection

Select where to get your licence from:

https://licence-api.nuix.com (cloud-server)

Select the type of licence you want:

Nuix eDiscovery Workstation Approximate remaining cap: 10 GB

Select how many workers you want:

Number of workers: 2 (60 available)

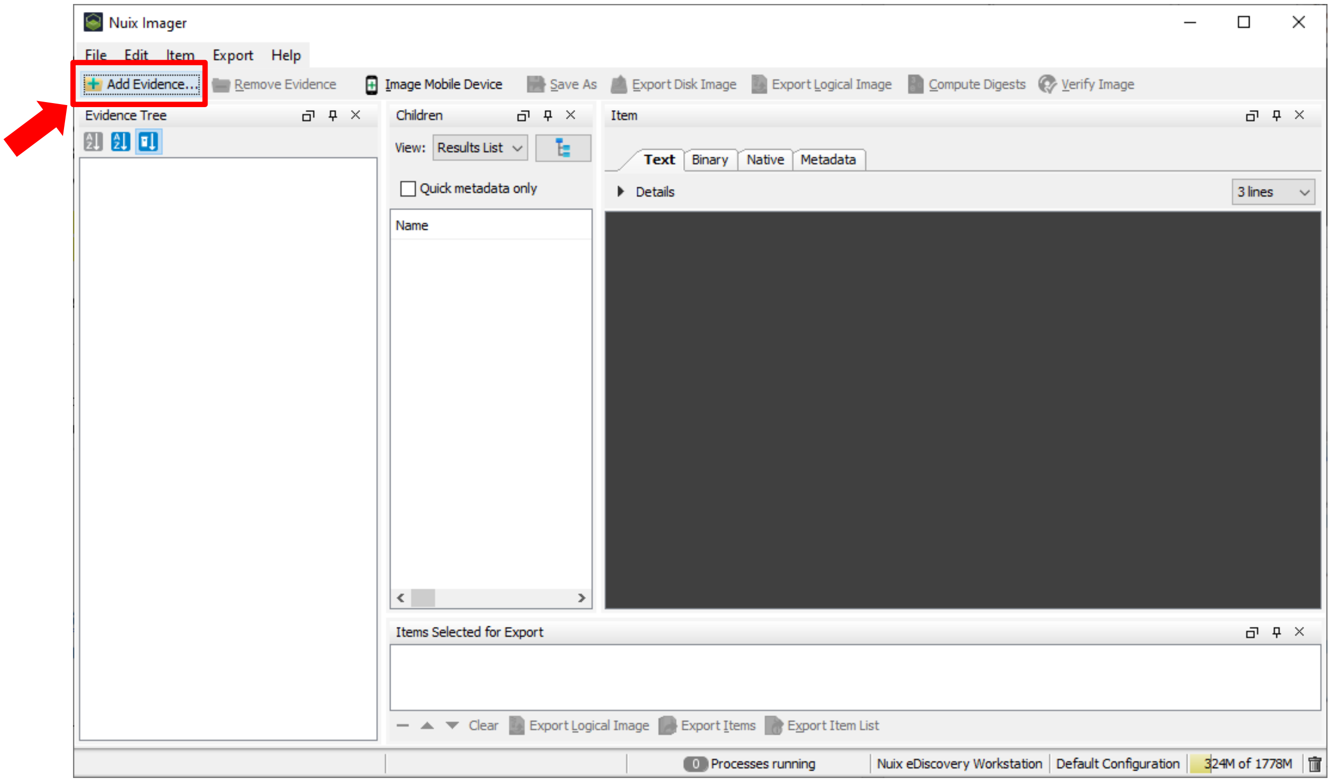
Use these credentials when requested:

User name: Craig.Ball
Temporary password: GoNuix1!

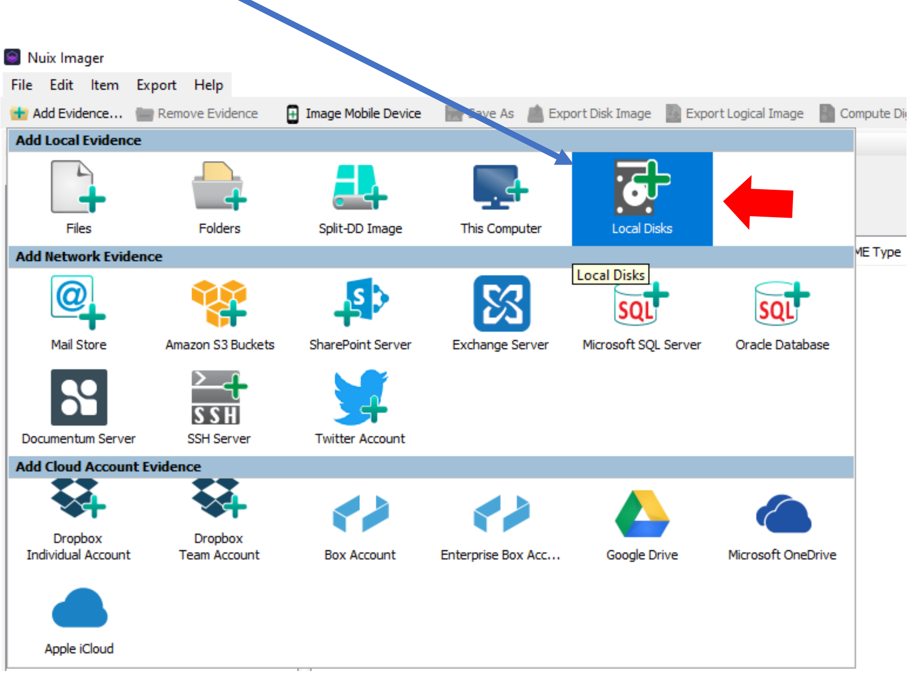
[83] In the event COVID prevents you from obtaining your evidence thumb drive, you may need to use a 6GB or larger thumb drive of your own as your "evidence drive." In that event, let me know a few days before the assignment is due and I'll provide a means for you to make your drive resemble the evidence drive. Be aware that the size, sector count, contents and hash values of your evidence drive will not match those in the illustrations below.

Step 2B: Add Evidence

Click the Add “Evidence Item” button on the menu bar.

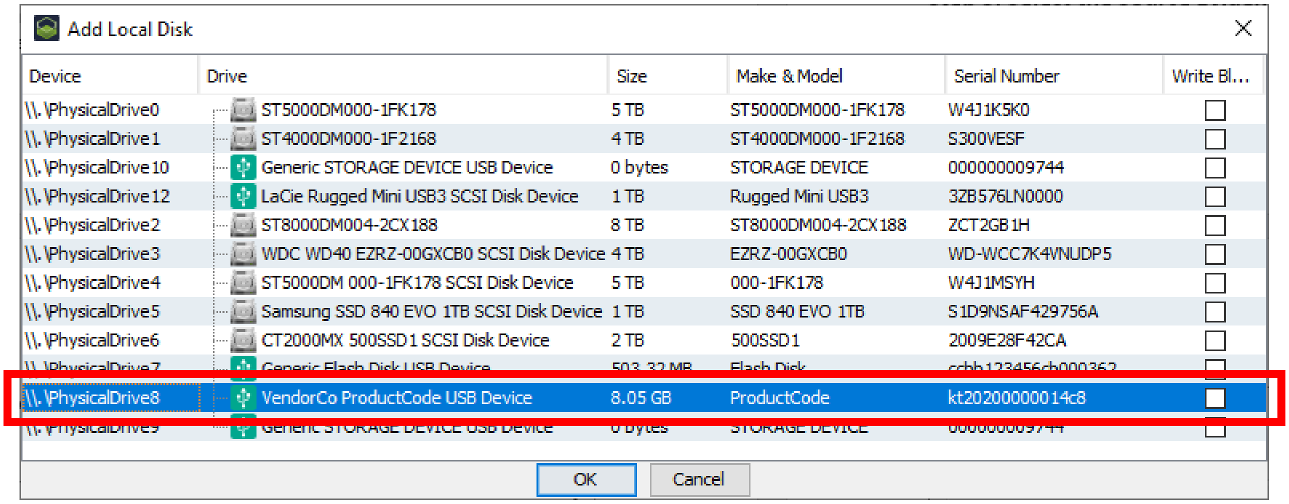


Click “**Local Disks**”



Step 3: Select the Source Evidence

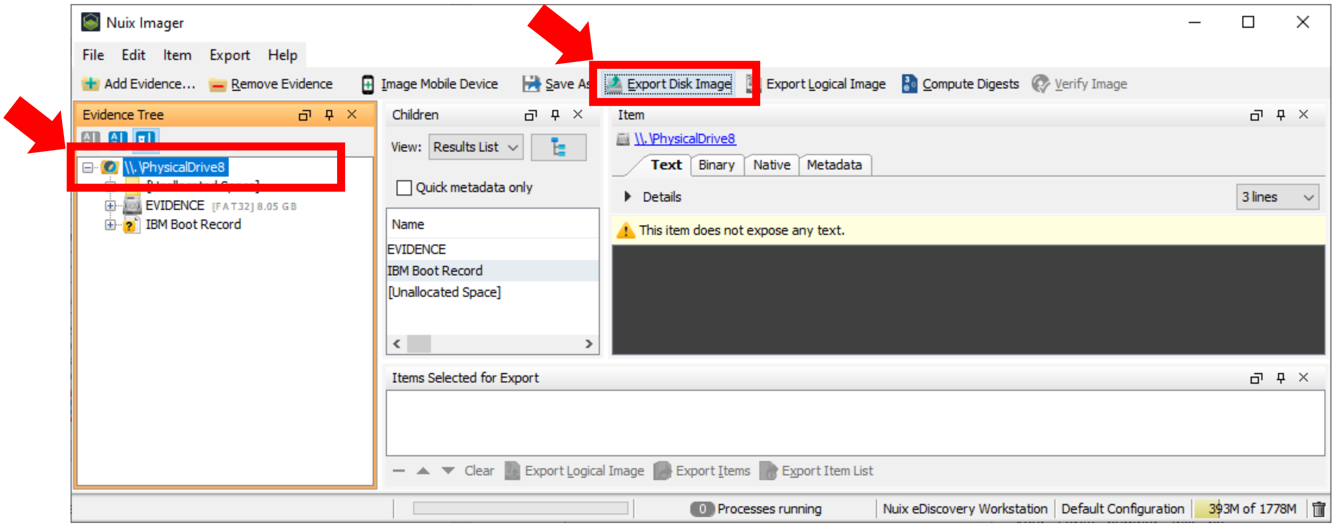
From the “Add Local Disk menu, select the evidence drive, a 8.05GB Flash Disk. *DO NOT select any larger drive or your boot drive.* Click OK.



Add Local Disk

Device	Drive	Size	Make & Model	Serial Number	Write Bl...
\\.\PhysicalDrive0	ST5000DM000-1FK178	5 TB	ST5000DM000-1FK178	W4J1K5K0	□
\\.\PhysicalDrive1	ST4000DM000-1F2168	4 TB	ST4000DM000-1F2168	S300VESF	□
\\.\PhysicalDrive10	Generic STORAGE DEVICE USB Device	0 bytes	STORAGE DEVICE	000000009744	□
\\.\PhysicalDrive12	LaCie Rugged Mini USB3 SCSI Disk Device	1 TB	Rugged Mini USB3	3ZB576LN0000	□
\\.\PhysicalDrive2	ST8000DM004-2CX188	8 TB	ST8000DM004-2CX188	ZCT2GB1H	□
\\.\PhysicalDrive3	WDC WD40 EZRZ-00GXCB0 SCSI Disk Device	4 TB	EZRZ-00GXCB0	WD-WCC7K4VNUDP5	□
\\.\PhysicalDrive4	ST5000DM 000-1FK178 SCSI Disk Device	5 TB	000-1FK178	W4J1MSYH	□
\\.\PhysicalDrive5	Samsung SSD 840 EVO 1TB SCSI Disk Device	1 TB	SSD 840 EVO 1TB	S1D9NSAF429756A	□
\\.\PhysicalDrive6	CT2000MX 500SSD1 SCSI Disk Device	2 TB	500SSD1	2009E28F42CA	□
\\.\PhysicalDrive7	Generic Flash Disk USB Device	503.32 MB	Flash Disk	ccbb123456cb000362	□
\\.\PhysicalDrive8	VendorCo ProductCode USB Device	8.05 GB	ProductCode	kt20200000014c8	□
\\.\PhysicalDrive9	Generic STORAGE DEVICE USB Device	0 bytes	STORAGE DEVICE	000000009744	□

Select the drive you will image by clicking on its \\Physical Drive number (your number will likely be different than that shown here), the select **Export Disk Image**.

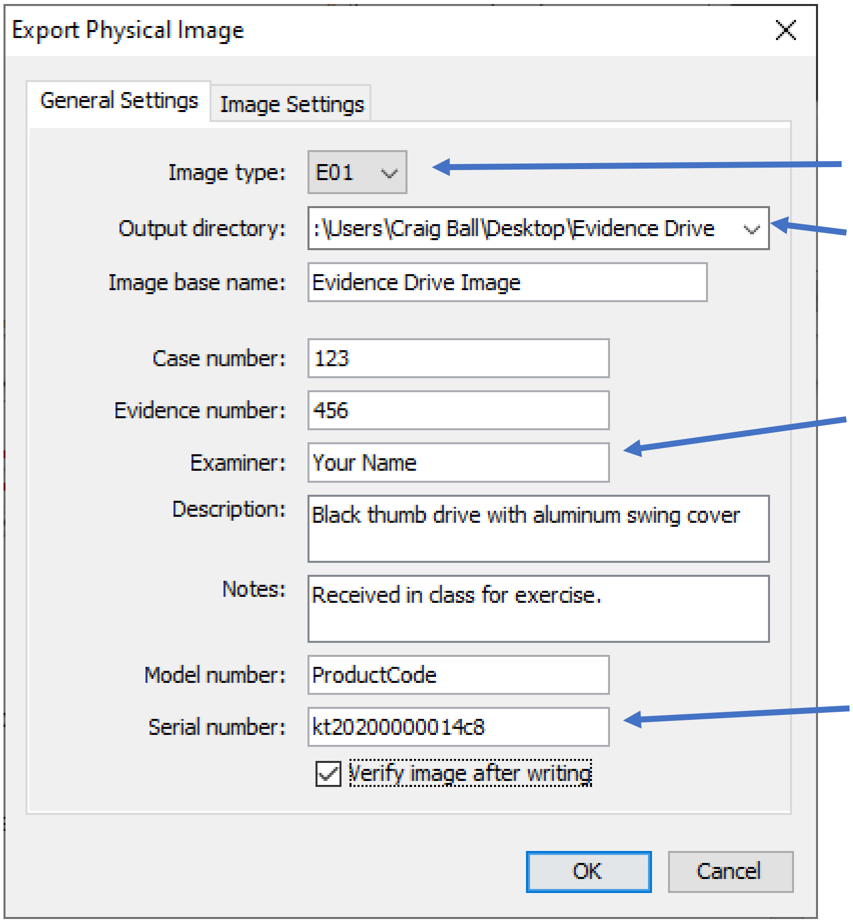


Nuix Imager
File Edit Item Export Help
Add Evidence... Remove Evidence Image Mobile Device Save As Export Disk Image Export Logical Image Compute Digests Verify Image
Evidence Tree
\\.\PhysicalDrive8
[Unallocated Space]
EVIDENCE [FAT32] 8.05 GB
IBM Boot Record
Children
View: Results List
Quick metadata only
Name
EVIDENCE
IBM Boot Record
[Unallocated Space]
Item
\\.\PhysicalDrive8
Text Binary Native Metadata
Details
3 lines
Items Selected for Export
Clear Export Logical Image Export Items Export Item List
0 Processes running
Nuix eDiscovery Workstation
Default Configuration
393M of 1778M

This item does not expose any text.

Step 4: Create a folder on your desktop called **Evidence Drive**. Be sure you have 8GB of free space available on your drive.

Complete the **Export Physical Image** menu as shown below, selecting *your* desktop Evidence Drive folder as the Output Directory. Click OK.



Export Physical Image

Image type: E01

Output directory: :\Users\Craig Ball\Desktop\Evidence Drive

Image base name: Evidence Drive Image

Case number: 123

Evidence number: 456

Examiner: Your Name

Description: Black thumb drive with aluminum swing cover

Notes: Received in class for exercise.

Model number: ProductCode

Serial number: kt20200000014c8

☑ Verify image after writing

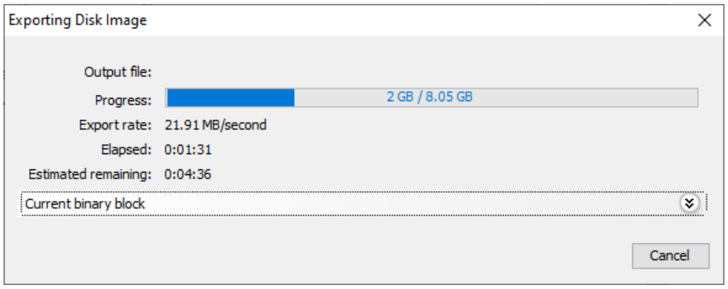
Leave the Image Type “ E01”

Be sure to select YOUR Evidence Drive folder here.

Put your name here

Your serial number will be different than the one shown

The image process begins. You can monitor progess in the Exporting Disk Image progress bar. It should take no more than about ten minutes. It varies.



Exporting Disk Image

Output file:

Progress: 2 GB / 8.05 GB

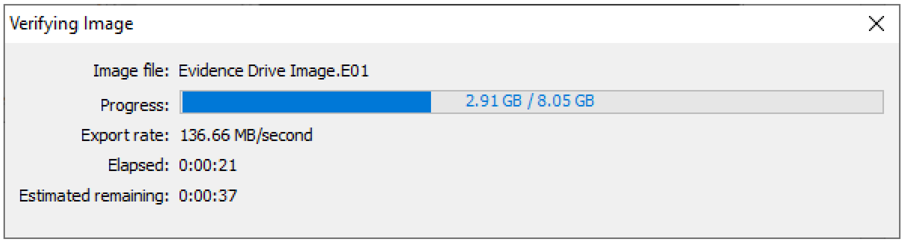
Export rate: 21.91 MB/second

Elapsed: 0:01:31

Estimated remaining: 0:04:36

Current binary block

Upon completion of the image, verification begins.



Verifying Image

Image file: Evidence Drive Image.E01

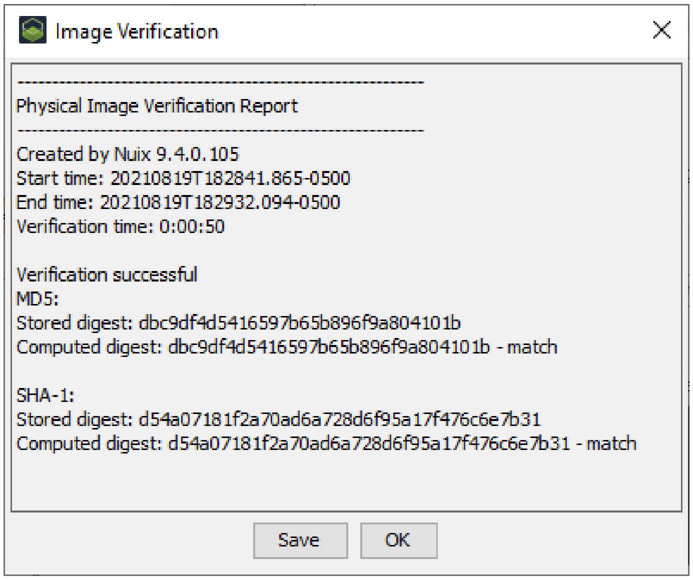
Progress: 2.91 GB / 8.05 GB

Export rate: 136.66 MB/second

Elapsed: 0:00:21

Estimated remaining: 0:00:37

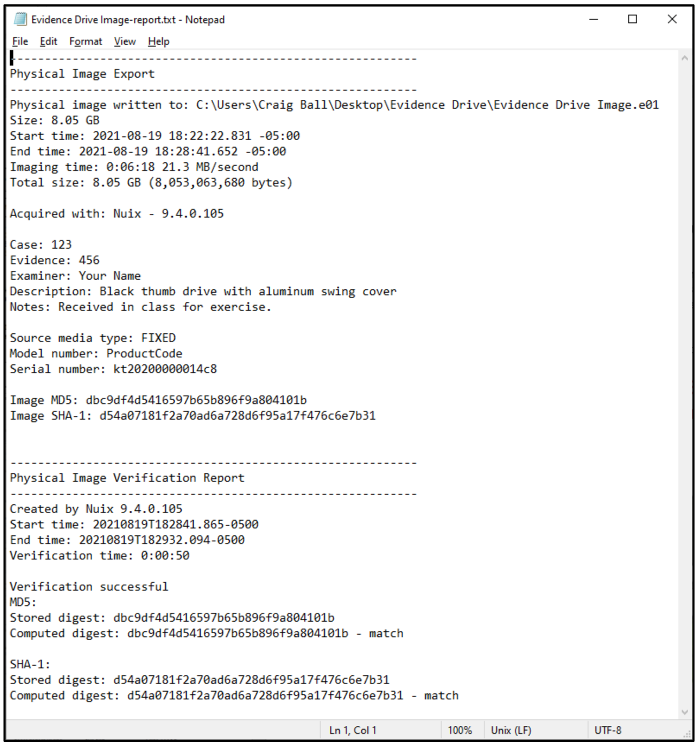
A Physical Image Verification report (right) should appear (your values will be different):



```
-----------------------------------------------------------
Physical Image Verification Report
-----------------------------------------------------------
Created by Nuix 9.4.0.105
Start time: 20210819T182841.865-0500
End time: 20210819T182932.094-0500
Verification time: 0:00:50

Verification successful
MD5:
Stored digest: dbc9df4d5416597b65b896f9a804101b
Computed digest: dbc9df4d5416597b65b896f9a804101b - match

SHA-1:
Stored digest: d54a07181f2a70ad6a728d6f95a17f476c6e7b31
Computed digest: d54a07181f2a70ad6a728d6f95a17f476c6e7b31 - match
```

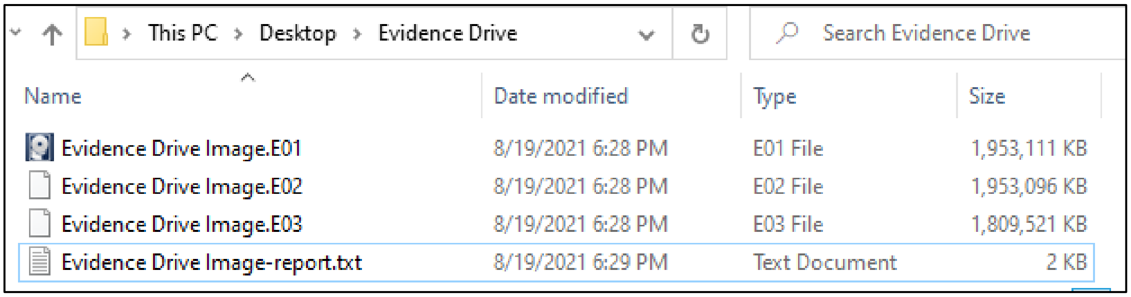
You're done! You can remove the evidence thumb drive.

An Image Export Report is stored in your Evidence Drive folder.



```
-----------------------------------------------------------
Physical Image Export
-----------------------------------------------------------
Physical image written to: C:\Users\Craig Ball\Desktop\Evidence Drive\Evidence Drive Image.e01
Size: 8.05 GB
Start time: 2021-08-19 18:22:22.831 -05:00
End time: 2021-08-19 18:28:41.652 -05:00
Imaging time: 0:06:18 21.3 MB/second
Total size: 8.05 GB (8,053,063,680 bytes)

Acquired with: Nuix - 9.4.0.105

Case: 123
Evidence: 456
Examiner: Your Name
Description: Black thumb drive with aluminum swing cover
Notes: Received in class for exercise.

Source media type: FIXED
Model number: ProductCode
Serial number: kt20200000014c8

Image MD5: dbc9df4d5416597b65b896f9a804101b
Image SHA-1: d54a07181f2a70ad6a728d6f95a17f476c6e7b31


-----------------------------------------------------------
Physical Image Verification Report
-----------------------------------------------------------
Created by Nuix 9.4.0.105
Start time: 20210819T182841.865-0500
End time: 20210819T182932.094-0500
Verification time: 0:00:50

Verification successful
MD5:
Stored digest: dbc9df4d5416597b65b896f9a804101b
Computed digest: dbc9df4d5416597b65b896f9a804101b - match

SHA-1:
Stored digest: d54a07181f2a70ad6a728d6f95a17f476c6e7b31
Computed digest: d54a07181f2a70ad6a728d6f95a17f476c6e7b31 - match
```

Your Evidence Drive folder (below) contains four files comprising the image set. Collectively, the three files with E0x extensions hold the entirety of the data on the evidence thumb drive.



Name	Date modified	Type	Size
Evidence Drive Image.E01	8/19/2021 6:28 PM	E01 File	1,953,111 KB
Evidence Drive Image.E02	8/19/2021 6:28 PM	E02 File	1,953,096 KB
Evidence Drive Image.E03	8/19/2021 6:28 PM	E03 File	1,809,521 KB
Evidence Drive Image-report.txt	8/19/2021 6:29 PM	Text Document	2 KB

Please keep track of the image set you created as you will use it again in a later exercise!

Opportunities and Obstacles: E-Discovery from Mobile Devices

Do you live two lives, one online and the other off? Millions lead lives divided between their physical presence in the real world and a deeply felt presence in virtual worlds, where they chat, post, friend, like and lurk. They are constantly checking themselves in and checking others out in cyberspace. In both worlds, they leave evidence behind. They generate evidence in the real world that comes to court as testimony, records and tangible items. Likewise, they generate vast volumes of digital evidence in cyberspace, strewn across modern electronic systems, sites, devices and applications.

Trial lawyers who know how to marshal and manage evidence from the real world are often lost when confronted with cyber evidence. Here, we take an introductory look at discovery from mobile devices.

The Blessing and Curse of ESI

Even if you don't know that data volume is growing at a compound annual rate of 42 percent, you probably sense it. This exponential growth suggests there's little point feeling overwhelmed by data volumes *today* because we are facing volumes *ten times as great in five years*, and *fifty times* as great in ten years.[84] Today is tomorrow's "good old days."

There's going to be a lot more electronic evidence; but, there's still time to *choose* how you deal with it.

A lawyer can curse electronic evidence and imagine he or she is preserving, collecting and requesting all they need without cell phones, the Cloud and all that other 'e-stuff.'

Or, the lawyer can see that electronic evidence is powerful, probative and downright amazing, and embrace it as the best thing to happen to the law since pen and ink. Never in human history have we enjoyed more or more persuasive ways to prove our cases.

Mobile Miracle

According to the U.S. Center for Disease Control, more than 41% of American households have no landline phone. They rely on wireless service alone. For those between the ages of 25 and 29, *two-thirds* are wireless-only. Per an IDC report sponsored by Facebook, four out of five people start using their smartphones within 15 minutes of waking up and for most, it's the very first thing they do, ahead of brushing their teeth or answering nature's call. For those in the lowest economic stratum, mobile phones are the principal and often sole source of Internet connectivity.

[84] Market research firm IDC predicted that digital data would grow at a compound annual growth rate of 42 percent through 2020, attributable to the proliferation of smart phones, tablets, Cloud applications, digital entertainment and the Internet of Things.

Apple has sold more than one billion iPhones worldwide from 2007 to 2016. These hold apps drawn from the more than 2.3 million apps offered in the iOS App Store, compounding the more than 25 billion times these apps have been downloaded and installed.

Worldwide, phones running the competing Android operating system account for three times as many activations as Apple phones. The United States Supreme Court summed it up handily: "*Today many of the more than 90% of American adults who own cell phones keep on their person a digital record of nearly every aspect of their lives*."[85]

Within this comprehensive digital record lies a cornucopia of probative evidence gathered using a variety of sensors and capabilities. The latest smart phones contain a microphone, fingerprint reader, barometer, accelerometer, compass, gyroscope, three radio systems, near field communications capability, proximity, touch, light and moisture sensors, a high-resolution still and video camera and a global positioning system.[86] As well, users contribute countless texts, email messages, social networking interactions and requests calls for web and app data.

Smart phones serve as a source of the following data:

- SIM card data
- Files
- Wi-Fi history
- Call logs
- Photographs and video
- Contacts
- Geolocation data
- E-mail
- Voicemail
- Chat
- SMS and MMS
- Application data
- Web history
- Calendar
- Bookmarks
- Task lists
- Notes
- Music and rich media

Mustering Mobile

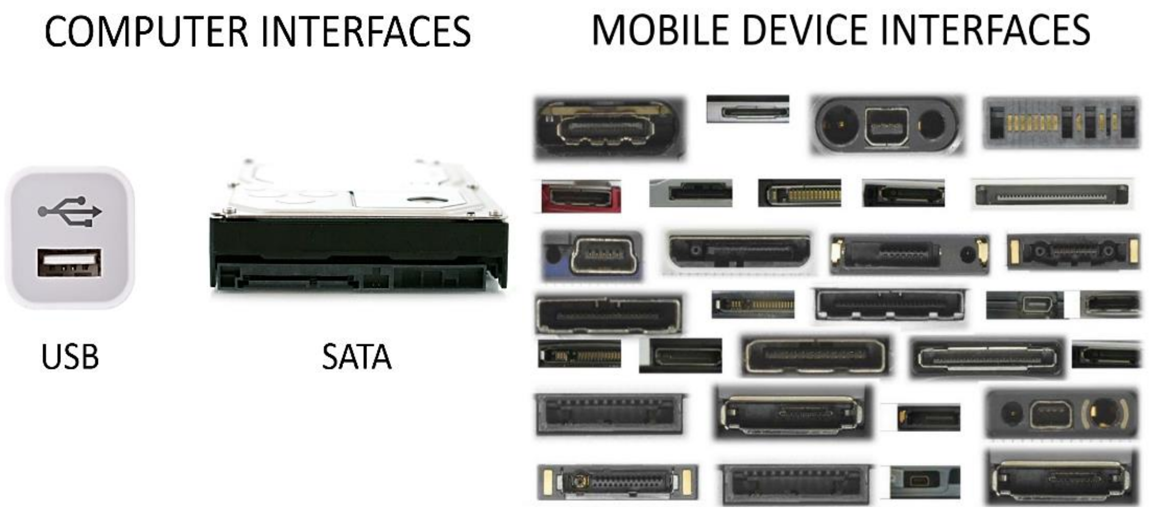
For the last decade, lawyers have been learning to cope with electronic evidence. We *know* how to acquire the contents of hard drives. We *know* about imaging and targeted collection. We've gotten better at culling, filtering and processing PC and server data. After all, most corporate data lives within identical file and messaging systems, and even those scary databases tend to be built on just a handful of well-known platforms. Too, we've got good tools and lots of skilled personnel to call on.

Now, let's talk mobile.

[85] *Riley v. California*, 573 U.S. ___ (2014).

[86] In support of 911 emergency services, U.S. law requires the GPS locator function when the phone is on.

Let's talk interfaces. We've been acquiring from hard drives for thirty years, using two principal interfaces: PATA and SATA. We've been grabbing data over USB for 17 years, and the USB 1, 2 and 3 interfaces all connect the same way with full backward compatibility. But phones and tablets? The plugs change almost annually (30-pin dock? Lightning? Thunderbolt?). The internal protocols change faster still: try <u>seven</u> generations of iOS in <u>five</u> years.

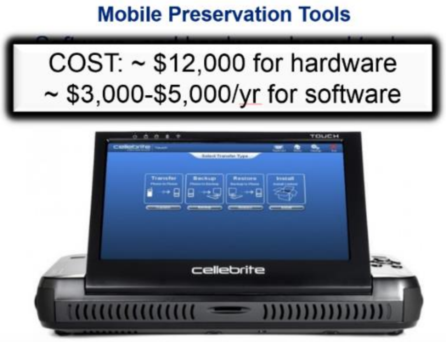


Let's talk operating systems. Two principal operating systems have ruled the roost in P.C. operating systems for decades: Windows and MacOS. Although the Android and iOS operating systems command huge market shares, there are still dozens of competing proprietary mobile operating systems in the world marketplace.

COMPUTERS
MOBILE DEVICES
MAC
WINDOWS
TIZEN
MOTOROLA
Firefox OS
DCT4
EXTx
Sailfish
QCP
Yaffs
symbian
I855
MCU
htc
P2K
OUIDOO
NOKIA
HUAWEI
JFFS2
ANDROID
BlackBerry
ubuntu
OSE
Windows Phone
QUALCOMM
bada
EFS2
XSR
iOS
FAT

Let's talk encryption. There is content on phones and tablets (*e.g.,* e-mail messaging) that we cannot acquire at all as a consequence of unavoidable encryption. Apple lately claims that it has so woven encryption into its latest products that it couldn't gain access to some content on its products if it tried. The law enforcement community depends on the hacker community to come up with ways to get evidence from iPhones and iPads. What's wrong with THAT picture?

Let's talk tools. Anyone can move information off a PC. Forensic disk imaging software is free and easy to use. You can buy a write blocker suitable for forensically-sound acquisition for as little as $25.00. But, what have you got that will preserve the contents of an iPhone or iPad? Are you going to synch it with iTunes? Does iTunes grab all you're obliged to preserve? If it did (and it doesn't), what now? How are you going to get that iTunes data into an e-discovery review platform? *There's no app for that.*



Mobile Preservation Tools

COST: ~ $12,000 for hardware
~ $3,000-$5,000/yr for software

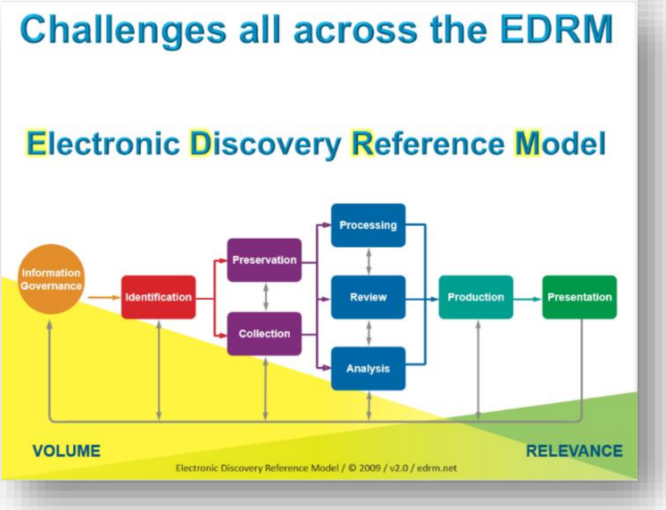
Let's talk time. It takes longer to acquire a 64Gb iPhone than it does to acquire a 640Gb hard drive. A fully-loaded iPad may take 48 hours. Moreover, you can acquire several hard drives simultaneously; but, most who own tools to acquire phones and tablets can process just one at a time. It's about as non-scalable a workflow as your worst e-discovery nightmare.

iPad+128 GB
with Retina display

A full up 128GB iPad?
~ 48 hours
to image!

Challenges All Across the EDRM

The Electronic Discovery Reference Model or EDRM is an iconic workflow schematic that depicts the end-to-end e-discovery process. It's a handy context in which to address the ways that mobile devices pose challenges in e-discovery.



Challenges all across the EDRM

Electronic Discovery Reference Model

Electronic Discovery Reference Model / © 2009 / v2.0 / edrm.net

Information Governance:

Businesses adopt a BYOD (Bring Your Own Device) model when they allow employees to connect their personal phones and tablets to the corporate network. Securing the ability to access these devices for e-discovery requires employers obtaining consents in employment agreements.

Identification:

Mobile devices tend to be replaced and upgraded more frequently than laptop and desktop computers; accordingly, it's harder to maintain an up-to-date data map for mobile devices. Mobile devices also do not support remote collection software of the sort that makes it feasible to search other network-connected computer systems. Too, the variety of apps and difficulty navigating the file systems of mobile devices complicates the ability to catalog contents.

Preservation:
It's common for companies and individuals to own mobile devices yet lack any means by which the contents of the phone or tablet can be duplicated and preserved when the need to do so arises in anticipation of litigation. Even the seemingly simple task of preserving text messages can be daunting to the user who realizes that, *e.g.*, the iPhone offers no easy means to download or print text messages.

Collection: As there are few, if any, secure ways to preserve mobile data *in situ,* preservation of mobile generally entails collection from the device, by a computer forensic expert, and tends to be harder, slower and costlier than collection from PC/server environments.

Processing: The unpacking, ingestion, indexing and volume reduction of electronically stored information on mobile devices is referred to as "Processing," and it's complicated by the fact that so many devices have their own unique operating systems. Moreover, each tends to secure data in unique, effective ways, such that encrypted data cannot be processed at all if it is not first decrypted.

Review:
Review of electronic evidence tends to occur in so-called "review platforms," including those with well-known names like Concordance and Relativity. For the most part, these (and message archival and retrieval systems) are not equipped to support ingestion and review of all the types and forms of electronic evidence that can be elicited from modern mobile devices and applications.

Analysis:
Much mobile data--particularly the shorthand messaging data that accounts for so much mobile usage—tend not to be good candidates for advanced analytics tools like Predictive Coding.

Geolocation

Cell phones have always been trackable by virtue of their essential communication with cell tower sites. Moreover, and by law, any phone sold in the U.S. must be capable of precise GPS-style geolocation in order to support 9-1-1 emergency response services. Your phone broadcasts its location all the time with a precision better than ten meters. Phones are also pinging for Internet service by polling nearby routers for open IP connections and identifying themselves and the routers. You can forget about turning off all this profligate pinging and polling. Anytime your phone is capable of communicating by voice, text or data, you are generating and collecting geolocation data. Anytime. Every time. And when you interrupt that capability that, too, leaves a telling record.

Phones are just the tip of the iceberg. The burgeoning Internet of Things (IoT) is a cornucopia of geolocation data. My Nest thermostat knows if I'm home or away and senses my presence as I walk by. The cameras in my home store my comings and goings in the Cloud for a week at a time. When someone enters, I get a text. My cell phone controls door locks and lighting, all by conversing across the Web. I can instruct Alexa, my Amazon Echo virtual assistant to turn on and off lights, and thanks to a free service called *If This Then That* (IFTTT), I can ask iPhone's Siri to turn lights on and off by *texting* them, at the cost of leaving an indelible record of even that innocuous act. Plus, Siri is now listening *all the time* while my Phone charges, not just when I push the home button and summon her. *"Hey Siri, can you be my alibi?"*

Production:
Finally, how will you produce data that's unique to a particular app in such a way that the data can be viewed by those who lack both the device and the app? Much work remains with respect to forms of production best suited to mobile data and how to preserve the integrity, completeness and utility of the data as it moves out of the proprietary phone/app environment and into the realm of more conventional e-discovery tools.

So, What Do I Do?
Though mobile is unlike anything we've faced in e-discovery and there are few affordable tools extant geared to preserving and processing mobile evidence, we are not relieved of the duty to preserve it in anticipation of litigation and produce it when discoverable.

You first hurdle may be persuading the phone's user to part with it intact. Mobile devices are unique in terms of intimacy and dependency. Unlike computers, mobile devices are constant companions, often on our person. The attachment many feel to their mobile phone cannot be overstated. It is simply inconceivable to them to part with their phones for an hour or two, let alone overnight or indefinitely. Many would be unable to contact even their spouse, children or closest friends without access to the data stored on their phones. Their mobile phone number may be the only way they can be contacted in the event of an emergency. Their phones wake them up in the morning, summon their ride to work, buy their morning bagel and serve as an essential link to almost every aspect of their social and business lives. Smart phones have become the other half of their brains.

So, when you advise a mobile user that you must take their devices away from them in order to collect information in discovery, you may be shocked at the level of resistance--even panic or duplicity--that request prompts. You need a plan and a reliable projection as to when the device will be returned. Ideally, you can furnish a substitute device that can be immediately configured to mirror the one taken without unduly altering evidence. Don't forget to obtain the credentials required to access the device (*e.g.*, PIN code or other passwords). Further, be wary of affording users the opportunity to delete contents or wipe the device by resetting to factory settings.[87] Perhaps due to the intimate relationship users have with their devices, mobile users tend to adopt an even more proprietary and protective mien than computer users.

Four Options for Mobile Preservation
In civil cases, before you do anything with a mobile device, it's good practice to back it up using the native application (e.g., iTunes for iPhones and iPads and preserve the backup). This gives you a path back to the data and a means to provision a substitute device, if needed. Then, you have four options when it comes to preserving data on mobile devices:

1. ***Prove You Don't Have to Do It:*** If you can demonstrate that there is no information on the mobile device that won't be obtained and preserved from another more-accessible source then

[87] Contents can often be erased by users entering the wrong password repeatedly, and it's not uncommon to see users making this "mistake" on the eve of being required to surrender their phones.

you may be relieved of the obligation to collect from the device. This was easier in the day when many companies employed Blackberry Enterprise Servers to redirect data to then-ubiquitous Blackberry phones. Today, it's much harder to posit that a mobile device has no unique content. But, if that's your justification to skip retention of mobile data, you should be prepared to prove that anything you'd have grabbed from the phone was obtained from another source.

It's an uphill battle to argue that a mobile device meets the definition of a "not reasonably accessible" source of discoverable data. The contents of mobile devices are readily accessible to users of the devices even if they are hard for others to access and collect.

2. ***Sequester the Device:*** From the standpoint of overall cost of preservation, it may be cheaper and easier to replace the device, put the original in airplane mode (to prevent changes to contents and remote wipes) and sequester it. Be sure to obtain and test credentials permitting access to the contents before sequestration.

3. ***Search for Software Solutions:*** Depending upon the nature of the information that must be preserved, it may be feasible to obtain applications designed to pull and preserve specific contents. For example, if you only need to preserve messaging, there are applications geared to that purpose, such as iMazing, Decipher TextMessage or Ecamm PhoneView. Before using unknown software, assess what it's limitations may be in terms of the potential for altering metadata values or leaving information behind.

4. ***Get the credentials, Hire a Pro and Image It:*** Though technicians with the training and experience to forensically image phones are scarce and may be pricey, it remains the most defensible approach to preservation. Forensic examiners expert in mobile acquisition will have invested in specialized tools like Cellebrite UFED, Micro Systemation XRY, Lantern or Oxygen Forensic Suite. Forensic imaging exploits three levels of access to the contents of mobile devices referred to as Physical, Logical and File System access. Though a physical level image is the most complete, it is also the slowest and hardest to obtain in that the device may need to be "rooted" or "jailbroken" in order to secure access to data stored on the physical media. Talk with the examiner about the approaches best suited to the device and matter and, again, be sure to get the user's credentials (i.e., PIN and passwords) and supply them to the examiner. Encryption schemes employed by the devices increasingly serve to frustrate use of the most complete imaging techniques. In those case, some data is simply unobtainable by any current forensic imaging methodology.

Simple, Scalable Solutions to iOS-Device Preservation

Can anyone doubt the changes wrought by the modern “smart” cellphone? All day, we see drivers looking at their phones, some so engrossed they fail to move when the light turns green. Phones have altered the progress of traffic in every community. We've turned into distracted digital zombies behind the wheels of our cars.

Distracted driving has eclipsed speeding and drunken driving as the leading cause of motor vehicle collisions. Walking into fixed objects while texting is reportedly the most common reason young people visit emergency rooms today. Instances of “distracted walking” injury have doubled every year since 2006. Doing the math, 250 ER visits in 2006 are over half a million ER visits today, *because we walk into poles, doors and parked cars while texting!*
Look around you (if you can pry your eyes from your screen). How many are using their phones? At a concert, how many are experiencing it through the lens of their cell phone cameras? How many selfies? How many texts? How many apps?

Lately I’ve begun asking audiences how many are never more than an arm’s length from their phones. A majority raise their hands. These are tech-wary lawyers. Most are Boomers, not Millennials.

Smart phones have changed us. Litigants are at a turning point in meeting e-discovery duties, and lawyers ignore this sea change at peril.

Today, if you fail to advise clients to preserve relevant and unique mobile data when under a preservation duty, you’re committing malpractice.

Yes, I used the “M” word, and not lightly.

Two things have changed such that we can’t hide our heads in the sand anymore when it comes to mobile evidence. First, the data on phones and tablets is *not* a copy of information held elsewhere. It’s unique, and often relevant, probative evidence. Second, the locking down of phone content has driven the preservation of mobile content from the esoteric realm of computer forensics to the readily accessible world of apps and backups. These developments mean that, notwithstanding the outdated rationales lawyers trot out for ignoring mobile, the time has come to accept that mobile is routinely within the scope of preservation obligations.

It was convenient to ignore mobile in e-discovery. Mobile was a black hole. You had to hire technical experts to use expensive tools to preserve the contents of phones, and it was like pulling teeth to get users to surrender their devices for the time it took to image them. Users protested, *"My mobile phone is the only way the kids' school can reach me in an emergency, and I can't use another phone because everyone texts now, and anyway, WHO REMEMBERS PHONE NUMBERS ANYMORE?"*

A few years ago, mobile phones shared some of the characteristics of personal computers in that they held latent data that could be recovered using specialized tools sold for princely sums by a couple of shadowy tech companies. So, the preservation of mobile devices slipped into the shadows, too. Phones and tablets were *forensic* evidence, and only forensic examiners could collect their contents.

Although everyone uses mobile devices all day, the contents of mobile devices were deemed "not reasonably accessible" because everyone agreed it was too costly and burdensome to preserve phones.

But, now there are easy, low-cost ways to preserve relevant mobile content *without taking phones away from users.* Because it's become quick, easy and cheap to preserve, mobile content is readily accessible, and its preservation (when potentially relevant) is likely proportionate under the Federal Rules of Civil Procedure.

In e-discovery today, the forensic-level preservation of phones—the sort geared to deleted content and forensic artifacts—is a fool's errand. As the public learned from the FBI's tussle with Apple over unlocking the iPhones of the San Bernardino terrorists, modern smart phones are locked down hard. Content is encrypted and even the keys to access the encrypted content are themselves encrypted. Phone forensics isn't what it used to be. More and more, we can't get to that cornucopia of recoverable forensically-significant data.

Yet, it's quick, easy and free for a user to generate a full, unencrypted backup of a phone without surrendering possession. The user can even place the backup in a designated location for safekeeping by counsel or IT. Will this be a "forensic image" of the contents? Strictly speaking, no. But as the phone manufacturers tighten their security, "forensic imaging" becomes less and less likely to yield up content of the sort encompassed by a routine e-discovery preservation obligation. Not every case is a job for C.S.I.

I grant that a full unencrypted backup of an iPhone isn't going to encompass all the data that might be gleaned by a pull-out-all-stops forensic preservation of the phone. But so what? As my corporate colleagues love to say, "*the standard for ESI preservation isn't perfect.*" I always agree adding, "*but it isn't lousy either.*" Preserving by backup isn't perfect; but, it isn't lousy. It's good *enough*, and far superior to what is currently being done to preserve mobile evidence, *i.e., absolutely nothing*.

Preservation of mobile device content must become a standard component of a competent preservation effort except where the mobile content can be shown to be beyond scope. Mobile content has become so relevant and unique, and the ability to preserve it so undemanding, that the standard must be preservation.

The Need

Some of you are likely reading this on your phone or tablet. If not, it's a virtual certainty that your phone or tablet are nearby. Few of us separate from our mobile devices for more than minutes a day. On average, cell users spend four hours a day looking at that little screen. On a*verage.* If your usage is much less, someone else's is much more.

It took 30 years for e-mail to displace paper as our primary target in discovery. It's taken barely 10 for mobile data, especially texts, to unseat e-mail as the Holy Grail of probative electronic evidence. *Mobile is where evidence lives now*; yet in most cases, mobile data remains "off the table" in discovery. It's infrequently preserved, searched or produced.

> ***No one can say that mobile data isn't likely to be relevant, unique and material.*** Today, the most candid communications aren't e-mail, they're text messages. Mobile devices are our principal conduit to online information, eclipsing use of laptops and desktops. Texts and app data reside primarily and *exclusively* on mobile devices.
>
> ***No one can say that mobile data isn't reasonably accessible.*** We use phones continuously, for everything from games to gossip to geolocation. Texts are durable (the default setting on an iPhone is to keep texts "Forever"). Mobile content easily replicates as data backed up and synched to laptops, desktops and online repositories like iCloud. The mobile preservation burden pales compared to that we take for granted in the preservation of potentially-relevant ESI on servers and personal computers.
>
> ***Modest Burden.*** That's what this article is about. My goal is that you see for yourself that the preservation burden is minimal when it comes to preserving the most common and relevant mobile data. I'll go so far as to say that *the burden of preserving mobile device content, even at an enterprise scale, is less than that of preserving a comparable volume of data on laptop or desktop computers.* Too, the workflows are as *defensible and auditable as any we accept as reasonable in meeting other ESI preservation duties.*

Three Principles

The following three principles underscore the need for efficient, defensible preservation of relevant mobile content:

- When mobile data may be unique and relevant, it should be preserved in anticipation of litigation. This principle is especially compelling when the preservation burden is trivial (as by use of the backup technique described below). You can demonstrate the absence of relevant data by, *e.g.*, sampling the contents of devices; but standing alone, a policy barring the use of a device to store relevant data is *not* sufficient proof that such device has not, in fact, been used to store data. Too often, practice belies policy, particularly for messaging
- Mobile preservation should be a customary feature of a defensible litigation hold; but absent issues of spoliation, few matters warrant the added cost of mobile preservation by forensics experts or the burden and disruption of separating users from mobile devices.
- Legitimate concerns respecting personal privacy and privilege do *not* justify a failure to preserve relevant mobile data, although they *will* dictate *how* data is protected, processed, searched, reviewed and produced.

Three Provisos:

As you undertake the exemplar workflow in the exercises and ponder how you might adapt it to your needs, consider the following three provisos:

- *The method demonstrated here is but one simple, scalable and defensible method* to preserve iPhone content. It's not necessarily the only way or the optimum way.
- *Preservation isn't production.* Lawyers' abilities to search, review and produce mobile content in utile and complete forms hasn't kept pace with the obligation to do so, or on a par with other responsive sources of ESI. This article and these exercises are about routine *preservation*; they don't address downstream processes and production except insofar as ensuring that the information preserved remains readily amenable to all methods of search, review and production in e-discovery.
- *Please challenge, but don't dismiss.* The duty to preserve is real and immediate; but there's room for honest debate about what depth and exactitude of mobile preservation is warranted case to case. In weighing any method, compare it to the alternative. *If you reject a preservation method because you deem it flawed, is the alternative a superior method or nothing at all?* "None" is rarely the proper choice when it comes to mobile evidence. Preserving "most" is better than "none," but, considerations of risk may dictate that one preserve "all" over "most." In turn, considerations of proportionality may elevate "most" over "all." It's sensible to ask, "*Is the incremental cost of forensic-level preservation by experts justified by relevant and unique content? If not, might 'good' be good enough?"*

Defensibility

Ignoring mobile evidence isn't the path taken by competent, ethical attorneys. We must employ methods of preservation that aren't unduly costly or burdensome yet pose little risk that a judge will find the methods unreasonable. The essence of defensibility is the ability to show that an action was prudent per a good faith assessment of what was known, or in the exercise of diligence should have been known, when the action occurred. If mobile content required to be preserved is lost, the Court will ask: "*Was the preservation method employed reasonably calculated to guard against loss or corruption of potentially-relevant mobile data?*" This will entail consideration of the method, its deployment and its oversight. These considerations are addressed below in Audit and Verification.

Custodian-Directed Preservation

The predominant approach to preservation in e-discovery entails use of a legal hold directive instructing custodians to act to preserve potentially-relevant ESI. This is custodian-directed preservation, and it's been justifiably criticized for its many flaws, among them that:

- It requires custodians to make judgments concerning relevance, materiality and privilege;
- It obliges custodians to complete tasks, like lexical search, without proper tools or training;
- It demands effort without affording custodians the time, resources and guidance to succeed; and
- It doesn't deter custodians who seek to destroy or change inculpatory or embarassing data.

Custodian-directed preservation is key to a defensible legal hold process; however, it's just part of a proper process and is best paired with other efforts, like IT-initiated holds, that defray its shortcomings.

So, if custodian-directed preservation is problematic, why put custodians in charge of preserving their own devices instead of handing the devices over to digital forensics experts for imaging? Isn't that inviting the fox to guard the henhouse?

The signal challenge to preserving mobile devices is persuding custodians to part with them. By empowering custodians to preserve the data themselves, custodians need never surrender custody of their devices. Accordingly, users are less threatended by the process and less inclined to fight or subvert it. Backing up an iPhone is simple and quick; and crucially, the process affords the custodian neither the need nor the practical ability to select or omit content. Compare that to tasking a custodian to collect e-mail or documents, where it's easy to overlook or deliberately omit material with little chance of detection.

The advantages of custodian-directed preservation of mobile devices by backup are:

- Custodians need not make judgments concerning relevance, materiality and privilege;
- Custodians need not run searches or require no special tools or training;
- The backup process is speedy, easy to autheticate and lets custodians retain their phone;
- It's difficult to omit content from a backup and, once created, backups are hard to alter.

Scalability and Proportionality

Scalability describes the ability of a system or process to handle a growing number of tasks or a larger volume of data. It's a crucial consideration in all phases of e-discovery, but particularly challenging when dealing with mobile data. Historically, preserving mobile data was a one-off task: seldom undertaken and typically for only a handful of devices. Preserving the contents of a single phone by engaging a digital forensics specialist to image the device was the norm, and though costly, the obligation rarely had to scale to dozens or hundreds of far-flung devices. For one or two phones, you could do it in a day or two for, say, a thousand dollars.

Now, imagine you must preserve the texts and call data from the mobile devices of sales reps, one each in all fifty United States, the District of Columbia, Puerto Rico and Guam. Fifty-three iPhones. What are your options? Let's compare:

1. **Instruct all custodians to overnight courier their phones to your trusty forensic examiner.** In turn, the examiner will image each device and overnight each back when the work is complete.
    - Cost: Under $30,000.00 without rush or overtime fees.
    - Timing: Assuming no glitches, most users will have their phones back within about four to five business days, as few labs possess the equipment permitting them to image more than a couple of phones simultaneously. As well, 53 packages must be correctly processed, logged as evidence, re-packaged and returned to the correct custodian.
        - How many businesses can idle their national sales staff for four to five days?
        - How many reps will be willing to hand over their phones for four to five days?

2. **Send your trusty forensic examiner to 53 locations to image each phone.**
    - Cost: $50-$60,000.00 in professional time; add a comparable sum for travel costs.
    - Timing: A month or more. It's a 19-hour flight to Guam, 11 hours to Hawaii and nine to Alaska. Equipment must travel, and each custodian must part with their phone for the better part of a day.
        - Caveat: Some states license forensic examiners. It may not be legal for an unlicensed examiner to come into the jurisdiction to acquire the image.

3. **Engage 53 local, licensed (as required) examiners to image each device.**
    - Cost: $35-$50,000.00 in examiner fees, plus the professional time required to locate, vet and contract with each examiner. There will also be travel time assessed, albeit with little airfare and hotel expense.
    - Timing: Weeks, at best. Fifty-three data sets from as many senders must be correctly packaged and returned to you, and each custodian must still part with their phone.

All three options implicate proportionality concerns. All are expensive, disruptive and time-consuming. Accordingly, many litigants opt not to preserve the content of mobile devices, claiming phones don't hold relevant data in the face of compelling contrary evidence and a dearth of supportive metrics.

Let's compare the custodian-directed option:

4. **Direct and instruct 53 custodians to back up their devices, collecting the data as desired.**
    - Cost: None, insofar as discrete expenditures. Of course, discovery is never "free" because time costs money. The expense to notify the custodians and follow up on compliance is attendant to all methods, and administrative costs don't count against any. Expenses, if any, for the custodian-directed method hinge on whether you preserve backup data *in situ,* collect it via network transfer or ship it on physical media. Each method demands *some* effort of each custodian, whether that entails coordinating with an examiner to tender and retrieve a device or connecting the device to a computer for an iTunes backup. The latter is far easier and least disruptive.
    - Timing: A day or two. Sure, some custodians may be on vacation, and some may miss or ignore the request; however, such risks afflict every method. Only the custodian-directed method makes it possible to preserve the many, widespread devices in hours, not days or weeks. The custodian need only get to a computer with the device, whereas a forensic examiner must get to the device or the device must get to the examiner.

The custodian-directed method scales easily for phones and tablets. Custodians need never part with their devices, so there is no business interruption. It's speedy. It requires no special tools, cabling or software and no technical expertise. Moreover, the process poses almost no risk of loss or alteration of the relevant data and is unlikely to prompt custodians to game the process. There are no operating system compatibility issues. Remote screen-sharing handily facilitates any desired oversight and audit. In short, cost and burden are so trivial that relevance alone should be the pole star in deciding whether to preserve mobile content.

Audit and Verification

Recently, my friend and fellow forensic examiner, Scott Moulton, visited New Orleans. Over beignets and café au lait in the French Quarter, I made the case for the preservation methodology described here. Scott's a brilliant examiner and hard-eyed skeptic. I wanted him to kick the tires and find flaws.

At first, Scott wouldn't take off his forensic examiner hat and don an e-discovery thinking cap. He extolled the benefits of hiring a qualified forensic examiner and the specialized forensics tools we use to dig for esoteric artifacts. "Hire me. Hire *you*!" I liked the sound of that, and Scott liked the idea of motorcycling through the lower 48 and D.C. gathering digital evidence like some two-wheeled remake of Cannonball Run meets Revenge of the Nerds.

Still, Scott conceded that in the context of e-discovery, there really isn't much iPhone data preserved using a costly forensics tool versus preservation using iTunes. Our training and tool sets don't add much when preserving mobile data for discovery.

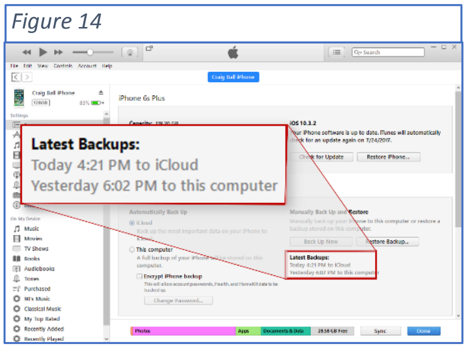
Once Scott warmed to the methodology for its speed and low cost, he questioned how the process could be quality checked for integrity. "What if the backup was interrupted or failed," he asked, "How would we know?"

It's a good point. Most experienced forensic examiners have found an image acquired in the field to be incomplete or unusable back in the lab. Thankfully, it's rare; but, sooner or later, it happens. *There are always gremlins*. Custodial-initiated preservation benefits from oversight and audit, if only because the risk of gremlins *feels* greater when custodians are in charge.

If iTunes successfully completes a backup, the backup event can be verified several ways:

1. In iTunes (with the device connected), by looking at the device summary for the attached device and noting the latest backups. ***Fig. 1, right top.***
2. In iTunes (with or without the device connected), under ***Edit>Preferences>Devices***. **Fig.2, right.** This lists the backed-up devices by name with time of backup. Hovering the mouse pointer over a listing will bring up further details about the device backed up (model, software version and build, serial number, phone number, IMEI and MEID). ***Fig. 3 right bottom.***
3. By confirming the date and time values for the folder containing the latest backup (stored by default in:

Figure 14



iPhone 6s Plus

iOS 10.3.2
Your iPhone software is up to date. iTunes will automatically check for an update again on 7/24/2017.

Automatically Back Up
○ iCloud
Back up the most important data on your iPhone to iCloud.
○ This computer
A full backup of your iPhone will be stored on this computer.
□ Encrypt iPhone backup
This will allow account passwords, Health, and HomeKit data to be backed up.

Manually Back Up and Restore
Manually back up your iPhone to this computer or restore a backup stored on this computer.

Latest Backups:
Today 4:21 PM to iCloud
Yesterday 6:02 PM to this computer

Figure 3

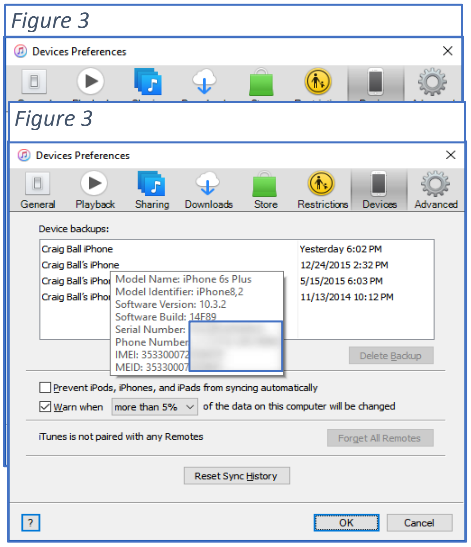

Devices Preferences

Figure 3

Devices Preferences

Device backups:

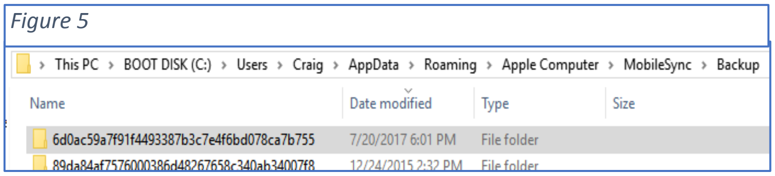
Craig Ball iPhone	Yesterday 6:02 PM
Craig Ball's iPhone	12/24/2015 2:32 PM
Craig Ball's iPho	5/15/2015 6:03 PM
Craig Ball's iPho	11/13/2014 10:12 PM

Model Name: iPhone 6s Plus
Model Identifier: iPhone8,2
Software Version: 10.3.2
Software Build: 14F89
Serial Number:
Phone Number
IMEI: 353300072
MEID: 35330007
[redacted]

□ Prevent iPods, iPhones, and iPads from syncing automatically
☑ Warn when more than 5% of the data on this computer will be changed

iTunes is not paired with any Remotes

Figure 5



This PC › BOOT DISK (C:) › Users › Craig › AppData › Roaming › Apple Computer › MobileSync › Backup

Name	Date modified	Type	Size
6d0ac59a7f91f4493387b3c7e4f6bd078ca7b755	7/20/2017 6:01 PM	File folder	
89da84af7576000386d48267658c340ab34007f8	12/24/2015 2:32 PM	File folder	

C:\Users\user's account name\AppData\Roaming\Apple Computer\MobileSync\Backup\). ***Fig. 4 below.***

There are several sensible ways to verify and audit a custodian-directed preservation effort. Tailor the method to the potential for failure and the willingness of a sponsoring witness to vouch for the integrity of the process if challenged. A proper audit trail could be as simple as the custodian supplying a screenshot (ALT-Print Screen) of the details panel for the latest backup (as seen when one hovers over backups in Devices Preferences, as described above and seen in **Fig. 3**). A second approach is the use of cryptographic hashing, and a third, the use of remote screen-sharing and -recording software to permit step-by-step oversight of the work by the sponsoring witness or designee. Also, device backup sets may be sampled and tested for accuracy and completeness. It's important to do *something* to audit and verify the effort; but proportionality suggests you needn't do *everything*.

What You Won't Get with a Backup

An iPhone backup won't preserve e-mail stored on the iPhone. This is by design. Per Apple, an unencrypted iTunes backup also won't include:

- Content from the iTunes and App Stores, or PDFs downloaded directly to iBooks
- Content synced from iTunes, like imported MP3s or CDs, videos, books, and photos
- Photos already stored in the cloud, like My Photo Stream, and iCloud Photo Library
- Touch ID settings
- Apple Pay information and settings
- Activity, Health and Keychain data

Why not use iCloud?

At some point, you *will* use iCloud for preservation; but currently, an iCloud backup is not equal to an iTunes backup. It preserves less data, and byte-for-byte, it takes more time to create than an iTunes backup. Additionally, iCloud encrypts all backups, making them a future challenge for processing and search should a user's credentials be unavailable.

Why an Unencrypted Backup?

This is a compromise. On the one hand, an encrypted iTunes backup preserves more information than an unencrypted backup. Apple won't store passwords, website history, Health data and Wi-Fi settings in an unencrypted backup. On the other hand, many tools can't process the contents of an encrypted backup, even with user credentials, and no tool can process an encrypted backup without credentials. Accordingly, we collect the data as an unencrypted backup, obviating the need for user credentials. To protect the data and add efficiency, we compress and optionally encrypt the backup set using credentials chosen for the legal hold project, not each user's credentials.

Encryption

Encryption is a crucial security tool to protect client data collected in e-discovery, but it's better to manage credentials systematically for the e-discovery project instead of according to each custodian's preference. However, because mobile devices employ layers of encryption, obtaining an unencrypted backup won't serve to unlock encrypted application data. You must obtain and preserve the user's access credentials for that data.

Many users employ the same password for multiple sources, so requiring a user to disclose credentials serves to compromise the security of sources not collected. Assuage concerns by detailing steps taken to protect users' credentials. An unlocked spreadsheet with each custodian's password(s) may be a convenience for the legal team, but it's a cybersecurity nightmare. Keep that in mind when furnishing credentials to service providers, and be sure your vendors are handling passwords securely.

Why Compress the Backup Data?

One reason we compress the data to a Zip file is to make it easier to copy to new media. Smaller data volumes move faster. However, depending upon the composition of the data backed up, the compressed Zip file may be much smaller or hardly smaller at all. My backup set compressed by just 2%. Much of the data on my iPhone consists of JPEG photos already in a compressed format, and it's hard to compress data that's already compressed as there's little 'space' to squeeze out by further compression.

So why bother compressing the backup files?

Two reasons. First, placing the preserved data in a Zip file guards against overwriting the data by a subsequent backup of the device. Second, depending upon the Zip tool employed to compress the file, the Zip process affords a means to securely encrypt the data without having to install an encryption tool. Any Windows or Mac machine can create compressed and encrypted Zip files.

❁ Exercise 16: Preserving a Mobile Device

GOALS: The goals of this exercise are for the student to:

1. Assess the cost and burden of preserving a mobile device like a smartphone or pad;
2. Gain experience useful to help guide litigants in meeting preservation duties; and
3. Assess issues of data integrity and defensibility when custodians manage preservation.

OUTLINE: Students will preserve the contents of their personal phones, applying the methodology and software suited to their device using the following Exemplar Phone Backup Instructions.

Assignment: Complete all the tasks outlined in the **Exemplar Phone Backup Instruction for Custodian-Directed Backup** that follow for either your iPhone or Android phone. Via e-mail, submit the data described in the last step of the Backup Instructions to craig@ball.net. You are NOT expected to submit any of your personal data. Submit *only* the **name, date/time and size of the zip file you create**.

NOTE: If you don't have enough space on your computer to hold the Phone image, look at the article, "Redirecting iPhone Backup Files to External Media" that follows the exemplar backup instructions or, for Android phones, change the location where CoolMuster lands the backup data to an external storage device or site.

Exemplar iPhone Backup Instruction for Custodian-Directed Backup

[[NOTE: This draft directive is offered to assist counsel in formulating language suited to the needs of the case and controlling law. It is not a form to be deployed without counsel. This example omits optional steps to encrypt the data set and transfer same to a distal repository for preservation, as such steps are frequently unnecessary to meet preservation duties].

Dear [Custodian]:

You recently acknowledged your obligation to preserve information relevant to a dispute between our company and ______________. Please see the _____________ hold notice for further details.

Within *48 hours of your receipt of this notice, y*ou must preserve the contents of your company-issued iPhone. If you cannot comply, please advise me at once by e-mail or phone. *Time is of the essence.*

You must make an unencrypted backup using iTunes and compress the backup folder per the instructions below. *Do not assume that you have been automatically making an unencrypted backup or preserving what's required using iCloud. You must carefully follow the procedures set out below.*

What you will need:

- Your company-issued iPhone and its USB charge/sync cable;
- Your company-issued desktop or laptop computer with the iTunes program installed. The computer must have available (unused) storage space on its boot (C:) drive exceeding *twice* the storage capacity of the iPhone. That is, if you have a 128GB capacity iPhone, use a computer with at least 256GB of unused storage space on its C: drive. You can find the capacity of the iPhone in Settings>General>About>Capacity. You can find the available storage on your computer's boot (C:) drive using File Explorer on a Windows machine or Finder on a Mac.

Time Required: One to two hours (most of it unattended "machine" time)

It will take about 10-15 minutes to follow these instructions, update iTunes, if needed, and begin the backup. The backup will complete in under 30 minutes, and you can continue to use the phone during the backup process (*but don't disconnect the charge/sync cable*). Then, it should take less than an hour to compress the data and 10 minutes or so to confirm successful compression and report on results. So long as the computer is secure and powered up throughout the process, you do not need to supervise, or leave the iPhone connected once backup completes.

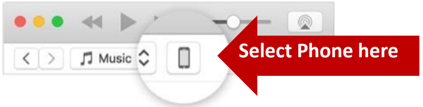
Follow These Steps:

1. Open iTunes and check for updates (Help>Check for Updates). Install the latest version of iTunes if not installed.

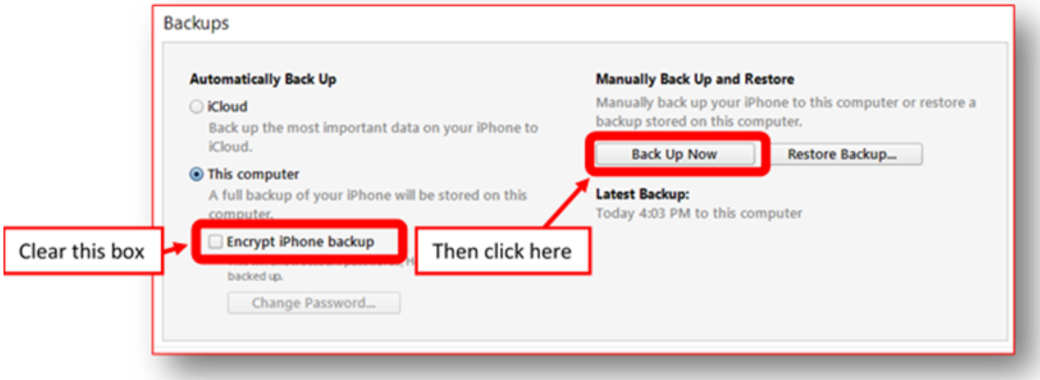
2. Connect your iPhone to a USB 2.0 or 3.0 port on the computer using a USB charge/sync cable.

3. If a message asks for your device passcode or to Trust This Computer, follow the onscreen steps.

4. Select your iPhone when it appears in iTunes. Click Summary in the sidebar.



5. In the Summary pane, be sure to uncheck "Encrypt iPhone Backup," then click "Back Up Now." You need not otherwise modify your Backups settings.


Backups

Automatically Back Up

○ iCloud
Back up the most important data on your iPhone to iCloud.

● This computer
A full backup of your iPhone will be stored on this computer.

□ Encrypt iPhone backup
backed up.

Change Password...

Manually Back Up and Restore

Manually back up your iPhone to this computer or restore a backup stored on this computer.

Back Up Now | Restore Backup...

Latest Backup:
Today 4:03 PM to this computer

6. Monitor the progress of the backup at the top center of the iTunes window. After the process ends, see if your backup finished successfully. If you're using iTunes for Windows, choose Edit>Preferences>Devices from the menu bar at the top of the iTunes window. If you're using iTunes for Mac, go to iTunes Preferences>Devices. You should see the name of your device with the date and time that iTunes created the backup. If you see 🔒 beside the name of your device, you need to be certain you unchecked "Encrypt iPhone Backup" and repeat the process until you do not see 🔒 beside the name of your device.
7. You can now disconnect your phone from the computer.

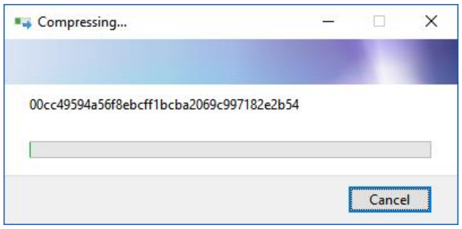
8. Locate the backup folder:

- **Windows:** Using File Explore, navigate to:
C:\Users\*your account name*\AppData\Roaming\Apple Computer\MobileSync\Backup\ where "*your account name*" is the name of your Window's User ID on the machine.

- **Mac:** Using Finder, select Go>Go to Folder on the Finder menu and enter:
~/Library/Application Support/MobileSync/Backup/

In both Windows and Mac, the Backup folder will contain one or more subfolders with 40-character names like *12da34bf5678900386c48267658d340eb34007f8*. **If there are multiple subfolders, i*dentify the subfolder that has the last modified date and time that matches the time you started this backup.***

9. **Compress the contents of the subfolder**: In Windows, right click on the subfolder just identified and select "***Send to>Compressed (zipped) folder***." A progress panel like the one at right should appear. On a Mac, right click on the subfolder and select "Compress." Do not turn off your computer or reboot. Allow the compression process to complete. It could take less than an hour to finish depending upon the type and volume of data backed up.



00cc49594a56f8ebcff1bcba2069c997182e2b54

10. Once compression has completed, Windows users should again navigate to the backup folder (see step 8 above) to confirm the presence of a file with the same name as the subfolder you identified but with the file extension .zip. Record the name, date/time and size of the zip file. *[If you cannot see file extensions on your Windows machine, open "My Computer," click "Tools" and click "Folder Options" or click "View" and then "Options" depending on your version of Windows. In the Folder Options window, click the "View" tab. Uncheck the box that says, "Hide file extensions for known file types." This should make file extensions visible.]*

11. By reply e-mail, send the **name, date/time and size of the zip file you just created**. *Do not delete or open this file. It must be preserved without alteration until further notice.*

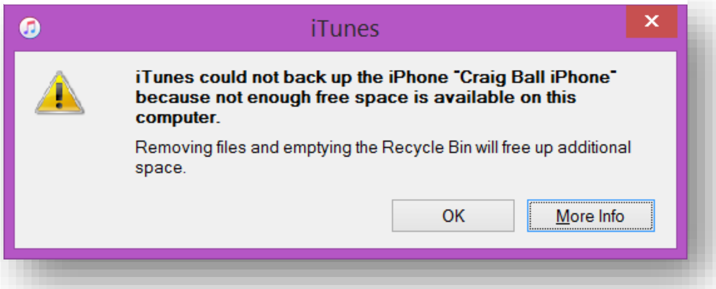
Your supervisor is copied here to insure you are afforded the time, oversight and support needed to comply in a timely way. Thank you for your cooperation. Call me at ____________ with any questions.

Redirecting iPhone Backup Files to External Media

Q. What if I don't have enough space on my Windows C: drive to hold the backup?

A. Smart phones have evolved to capture a *lot* of data. Ten years ago, you couldn't store more than 8GB of data on an iPhone. Today, they store up to 256GB, 32 times as much . So, an iTunes backup may fail to complete because not enough free space is available on the computer performing the backup. You may be able to resolve this by, *e.g.,* emptying the Recycle Bin; but, if you simply can't garner enough space on the boot drive where Apple stores the backup by default, you may need to "trick" your Windows machine into storing the backup on a sufficiently-sized alternate or external storage medium.

Figure 6



iTunes

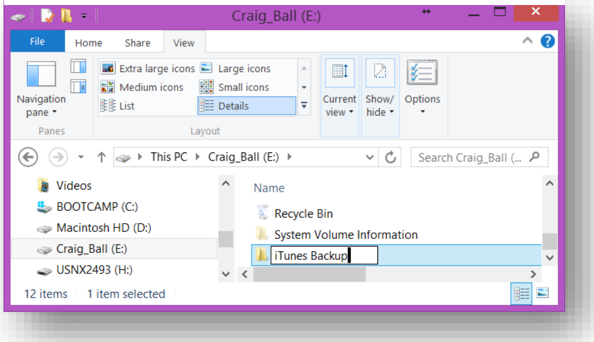
iTunes could not back up the iPhone "Craig Ball iPhone" because not enough free space is available on this computer.

Removing files and emptying the Recycle Bin will free up additional space.

How to Redirect an iTunes Backup Location in Windows

Step 1. Create a new backup folder on a disk with enough space to create your backup (roughly. twice the capacity of your iPhone is ample). In Figure 6, I've created the new iTunes backup location on my E: drive (a 250GB thumb drive) and named it "iTunes_Backup:" You can name yours anything you'd like.

Figure 7



Step 2. Rename the current iTunes backup folder Using Windows File Explorer, navigate to your current iTunes "Backup" folder. By default, it's:
C:\Users\*your account name*\AppData\Roaming\Apple Computer\MobileSync\
where "*your account name*" is the name of your Window's User ID on the machine.

Right click on the "Backup" folder and rename it. I called mine "Old_Backup;" but here again, call it whatever you like.

3. Redirect the Old Backup Folder Address to the New One

Here, it gets a tad tricky because you must use a Windows Command line interface. Make it easier on yourself by writing down the full paths to the old and new backup folders. *You must get both right for the redirection to work.*

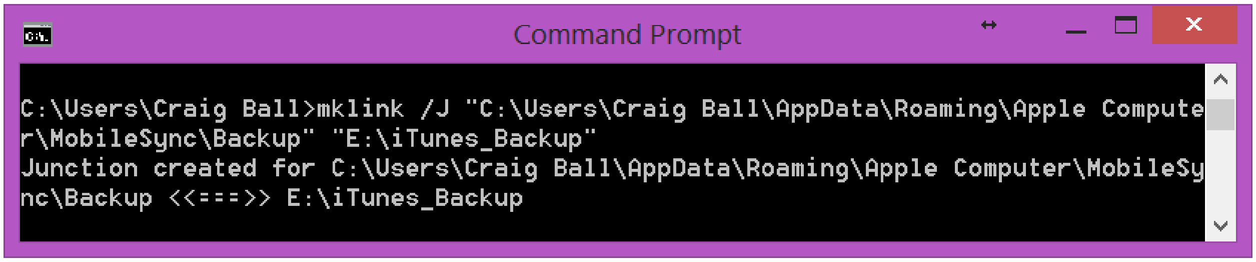
The old one *should* be:
C:\Users\*your account name*\AppData\Roaming\Apple Computer\MobileSync\Backup

The new path is on whatever storage medium you chose, using whatever path and folder name you gave it in step 1, above (mine was "**E:\iTunes_Backup**").

Open a command prompt window by pressing the Windows key on your keyboard, then typing CMD or by pressing the Shift key on your keyboard while right clicking in an open area of any folder, then selecting "Y and selecting "*Open command window here*" from the menu.

At the command line, carefully type the following command:
mklink /J "*path to old backup location*" "*path to new backup location*"
where you substitute the old and new paths you've written down. *Be sure to enclose each path in quotation makes, as shown.*

On my machine, the command and response looked like Figure 7:
Figure 8



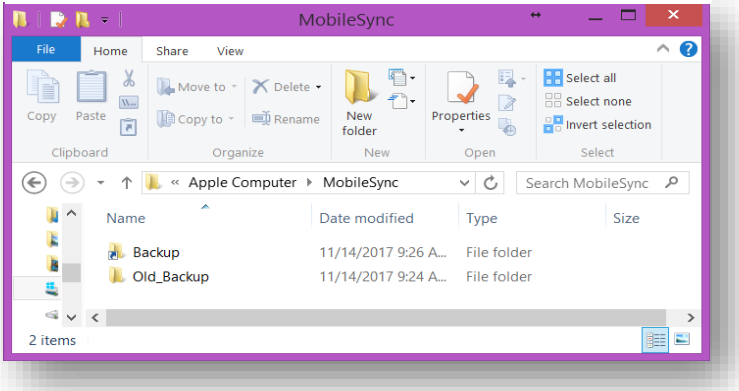
```
C:\Users\Craig Ball>mklink /J "C:\Users\Craig Ball\AppData\Roaming\Apple Compute
r\MobileSync\Backup" "E:\iTunes_Backup"
Junction created for C:\Users\Craig Ball\AppData\Roaming\Apple Computer\MobileSy
nc\Backup <<===>> E:\iTunes_Backup
```

The "junction created" refers to a Windows symbolic link, a **Directory Junction,** that will serve to redirect any actions that would have been performed on the old backup folder to be redirected to the new one.

What Note: The **mklink /J** command creates a symbolic link to the new folder from the old one. It's like creating a shortcut of D:\Backup from the original MobileSync\Backup folder. You can test the effect by double-clicking on the Backup folder in MobileSync. It will take you to the new Backup folder.

Now, if you look in your MobileSync folder:
(C:\Users\*your account name*\AppData\Roaming\Apple Computer\MobileSync)
you will see a folder shortcut named “Backup” alongside your renamed former backup folder as mine appears in Figure 8.

Figure 9



Name	Date modified	Type	Size
Backup	11/14/2017 9:26 A...	File folder	
Old_Backup	11/14/2017 9:24 A...	File folder	

4. Move your Old Backups

If desired, you can move your old iTunes backup files from your old renamed Backup folder to your new backup folder and delete them from the old location.

5. Run your iTunes Backup

Be sure the media you selected to hold the relocated backup is attached. Now, run your iTunes backup as usual and, if all is working, the backup will be created where you created the new backup folder.

Preserve the Contents of your Android Phone Using CoolMuster Android Assistant

CoolMuster is a commercial tool enabling the backup and collection of content (media, contacts, SMS, call logs, apps, etc.) from phones and other devices using the Android operating system.

If AT&T is your cell carrier, you should temporarily disable In Advanced Messaging before backup in order to collect all your messages. See instructions here.

Step 1: Download and install the CoolMuster Android Assistant software suited to your computer (PC or Mac) from the appropriate link below:
For Windows: https://www.coolmuster.com/downloads/cool-android-assistant.exe
For Mac: https://www.coolmuster.com/downloads/cool-android-assistant-for-mac.dmg

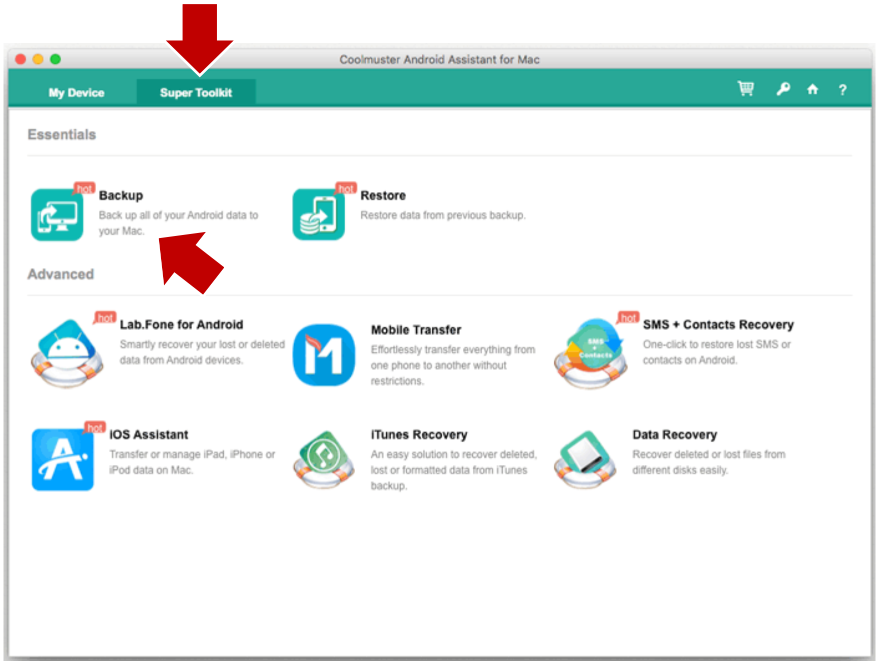
E-mail me at craig@ball.net to obtain the license needed to register the software. You DO NOT have to pay to use the software. I will purchase and supply a license for you.

For Windows installation assistance, click here
For Mac help, click here.

Step 2: Start Android Assistant and Connect to your Phone via USB cable. If asked to activate USB debugging mode on your Android device, follow the onscreen instructions to do so. If your phone appears in the main interface screen of the program, great. If not, go here to troubleshoot your connection.

Step 3: Go to "Super Toolkit" section where you can see the option to "Backup" or "Restore" (right):

Step 4: Click the "Backup" button on Super Toolkit, and select all contents to backup EXCEPT Music. Click "Browse" to choose an output location on your system to save your backup, then click "Back Up" to begin the backup process. **Make note of the target folder you specified.**



Back up all of your Android data to your Mac.

Restore data from previous backup.

Smartly recover your lost or deleted data from Android devices.

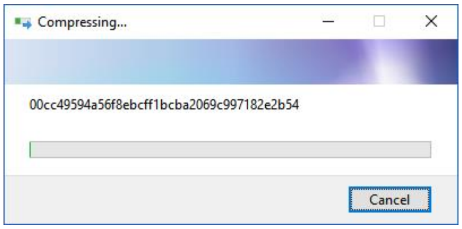
Effortlessly transfer everything from one phone to another without restrictions.

One-click to restore lost SMS or contacts on Android.

Transfer or manage iPad, iPhone or iPod data on Mac.

An easy solution to recover deleted, lost or formatted data from iTunes backup.

Recover deleted or lost files from different disks easily.



00cc49594a56f8ebcff1bcba2069c997182e2b54

Step 5: Navigate to the target folder holding your backup data and create a compressed file holding the target folder. of the backup files. In Windows, right click on the subfolder just identified and select "***Send to>Compressed (zipped) folder***." A progress panel like the one at right should appear. On a Mac, right click on the subfolder and select "Compress." Do not turn off your computer or reboot. Allow the compression process to complete. It could take less than an hour to finish depending upon the type and volume of data backed up.

Step 6: Once compression has completed, Windows users should again navigate to the backup folder to confirm the presence of a file with the same name as the subfolder you identified but with the file extension .zip. **Record the name, date/time and size of the zip file.** *[If you cannot see file extensions on your Windows machine, open "My Computer," click "Tools" and click "Folder Options" or click "View" and then "Options" depending on your version of Windows. In the Folder Options window, click the "View" tab. Uncheck the box that says, "Hide file extensions for known file types." This should make file extensions visible.]*

By reply e-mail, send me the **name, date/time and size of the zip file you just created**. ***DO NOT send me the file! It holds your private data***. *Do not delete or open the compressed file. It should be preserved without alteration until the end of the semester please.*

Note: These tools generally work without a hitch; but sometimes they push back. ***I'm here to help you.*** Don't suffer in frustration for hours. Get in touch with me and let's get to the bottom of the problem. Glitches are endemic to e-discovery and computer forensics; so, working through technical issues goes with the territory. E-mail me at craig@ball.net or text me at 713-320-6066. You can voice call me anytime between 8:30am and midnight.

Obtaining and Preserving Social Media Content as Evidence

Social Media Content (SMC) is a rich source of evidence. Photos and posts shed light on claims of disability and damages, establish malicious intent and support challenges to parental fitness--to say nothing of criminals who post selfies at crime scenes or holding stolen goods, drugs and weapons. SMC may expose mental instability, propensity to violence, hate speech, racial animus or misogyny. SMC is increasingly a medium for business messaging and the primary channel for cross-border communications. In short, SMC and messaging are heirs-apparent to e-mail in their importance to e-discovery.

Competence demands swift identification and preservation of SMC.

Static page captures or "screenshots" of SMC grabbed from a web browser or phone are notoriously unreliable, tedious to collect and inherently unsearchable. Applications like X1 Social Discovery and service providers like Hanzo can help with SMC preservation; but the task demands little technical savvy and no specialized tools. Major SMC sites offer straightforward ways users can access and download their content. Armed with a client's login credentials, lawyers, too, can undertake the ministerial task of preserving SMC without greater risk of becoming a witness than if they'd photocopied paper records.

Collecting your Client's SMC

Collecting SMC is a two-step process of requesting the data followed by downloading. Minutes to hours or longer may elapse between a request and download availability. Having your client handle collection weakens the chain of custody; so, instruct the client to forward download links to you or your designee for collection. Better yet, do it all yourself.

Obtain your client's user ID, password for each account and written consent to collect. Consider instructing your client to change account passwords for your use, re-enabling customary passwords following collection. Clients may need to temporarily disable two-factor account security. Download data promptly as downloads are available briefly (often just for a few days).

Collection Steps for Seven Social Media Sites

Facebook: After login, go *to Settings>Your Facebook Information>Download Your Information*. Select the data and date ranges to collect (*e.g.*, Posts, Messages, Photos, Comments, Friends, etc.). Facebook will e-mail the account holder when the data is ready for download (from the *Available Copies* tab on the user's *Download Your Information* page). Facebook also offers an *Access Your Information* link for review before download.

Twitter: After login, go to *Settings and Privacy>Your Twitter Data>Download Your Twitter Data.* Re-enter the password and choose *Request Archive*. Twitter will e-mail the account holder when a compressed file holding the data is ready for download. *Twitter permits one archive retrieval a month.*

Google: Go to *https://accounts.google.com*, select *Use Another Account* and login to client's account. Choose *Data and Personalization>Download Your Data*. Select data to include (be sure your client has expressly authorized collection) and the archival format (*e.g.,* zip). Google will e-mail the account holder when a compressed file holding the data is ready for download.

Instagram: Login and go to the user's profile. Select the gear icon (*Settings*), then *Privacy and Security>Request Download.* The data will be in JSON format inside a compressed file. Once decompressed, it can be viewed using any free online JSON parser.

LinkedIn: Login and select *Me>Settings and Privacy.* Under the Privacy tab, choose *Getting a Copy of Your Data* and the specific data sought. If uncertain, choose *Download Larger Data Archive*. Click *Request Archive*.

Snapchat: Login at https://accounts.snapchat.com and select *My Data>Submit Request.*

Tumblr: Login and select *Account>Settings>Privacy>Request Privacy Data.* The downloaded data will be in a compressed file in JSON format.

Review and Authentication

SMC is often voluminous and encoded in unfamiliar formats like JSON (JavaScript Object Notation), an open format for data interchange. So, as with other information collected in e-discovery, the competent way to index, search, review and tag electronic evidence is by use of e-discovery review tools, *e.g.*, Relativity, iConect, Logikcull, Everlaw, etc.

Though not essential, it's prudent to calculate a hash value for preserved SMC to demonstrate its integrity. *See*, *e.g.*, FRE 902(13) and (14). A hash value is a digital fingerprint of data. If the hash value obtained when the data was collected matches the hash value when used, the data is demonstrably unchanged. Many hashing tools can be downloaded online at no cost.

Caveat: There are no "guest passes" to social media accounts. When you log in as the account holder, you stand in the account holder's shoes. Keep good records of access and note what you did while logged in. Likewise, never seek or consent to access an opponent's social media account using opponent's credentials or you open yourself up to claims that you added or altered content.

The “E-Discovery Rules” (1,16,26,34 & 45) of the Federal Rules of Civil Procedure with Committee Notes accompanying 2006 and 2015 Amendments

[Note: Some provisions highlighted to emphasize their importance]

Rule 1. Scope and Purpose

These rules govern the procedure in all civil actions and proceedings in the United States district courts, except as stated in Rule 81. They should be construed, administered, and employed by the court and the parties to secure the just, speedy, and inexpensive determination of every action and proceeding.

Notes

(As amended Dec. 29, 1948, eff. Oct. 20, 1949; Feb. 28, 1966, eff. July 1, 1966; Apr. 22, 1993, eff. Dec. 1, 1993; Apr. 30, 2007, eff. Dec. 1, 2007; Apr. 29, 2015, eff. Dec. 1, 2015.)

Committee Notes on Rules—2015 Amendment

Rule 1 is amended to emphasize that just as the court should construe and administer these rules to secure the just, speedy, and inexpensive determination of every action, so the parties share the responsibility to employ the rules in the same way. Most lawyers and parties cooperate to achieve these ends. But discussions of ways to improve the administration of civil justice regularly include pleas to discourage over-use, misuse, and abuse of procedural tools that increase cost and result in delay. Effective advocacy is consistent with — and indeed depends upon — cooperative and proportional use of procedure.

This amendment does not create a new or independent source of sanctions. Neither does it abridge the scope of any other of these rules.

***

Rule 16. Pretrial Conferences; Scheduling; Management

(a) Purposes of a Pretrial Conference. In any action, the court may order the attorneys and any unrepresented parties to appear for one or more pretrial conferences for such purposes as:

(1) expediting disposition of the action;

(2) establishing early and continuing control so that the case will not be protracted because of lack of management;

(3) discouraging wasteful pretrial activities;

(4) improving the quality of the trial through more thorough preparation; and

(5) facilitating settlement.

(b) Scheduling.

(1) *Scheduling Order.* Except in categories of actions exempted by local rule, the district judge—or a magistrate judge when authorized by local rule—must issue a scheduling order:

(A) after receiving the parties' report under Rule 26(f); or

(B) after consulting with the parties' attorneys and any unrepresented parties at a scheduling conference.

(2) *Time to Issue.* The judge must issue the scheduling order as soon as practicable, but unless the judge finds good cause for delay, the judge must issue it within the earlier of 90 days after any defendant has been served with the complaint or 60 days after any defendant has appeared.

(3) *Contents of the Order.*

(A) *Required Contents.* The scheduling order must limit the time to join other parties, amend the pleadings, complete discovery, and file motions.

(B) *Permitted Contents.* The scheduling order may:

(i) modify the timing of disclosures under Rules 26(a) and 26(e)(1);

(ii) modify the extent of discovery;

(iii) provide for disclosure, discovery, or preservation of electronically stored information;

(iv) include any agreements the parties reach for asserting claims of privilege or of protection as trial-preparation material after information is produced, including agreements reached under Federal Rule of Evidence 502;

(v) direct that before moving for an order relating to discovery, the movant must request a conference with the court;

(vi) set dates for pretrial conferences and for trial; and

(vii) include other appropriate matters.

(4) *Modifying a Schedule.* A schedule may be modified only for good cause and with the judge's consent.

(c) Attendance and Matters for Consideration at a Pretrial Conference.

(1) *Attendance.* A represented party must authorize at least one of its attorneys to make stipulations and admissions about all matters that can reasonably be anticipated for discussion at a pretrial conference. If appropriate, the court may require that a party or its representative be present or reasonably available by other means to consider possible settlement.

(2) *Matters for Consideration.* At any pretrial conference, the court may consider and take appropriate action on the following matters:

(A) formulating and simplifying the issues, and eliminating frivolous claims or defenses;

(B) amending the pleadings if necessary or desirable;

(C) obtaining admissions and stipulations about facts and documents to avoid unnecessary proof, and ruling in advance on the admissibility of evidence;

(D) avoiding unnecessary proof and cumulative evidence, and limiting the use of testimony under Federal Rule of Evidence 702;

(E) determining the appropriateness and timing of summary adjudication under Rule 56;

(F) controlling and scheduling discovery, including orders affecting disclosures and discovery under Rule 26 and Rules 29 through 37;

(G) identifying witnesses and documents, scheduling the filing and exchange of any pretrial briefs, and setting dates for further conferences and for trial;

(H) referring matters to a magistrate judge or a master;

(I) settling the case and using special procedures to assist in resolving the dispute when authorized by statute or local rule;

(J) determining the form and content of the pretrial order;

(K) disposing of pending motions;

(L) adopting special procedures for managing potentially difficult or protracted actions that may involve complex issues, multiple parties, difficult legal questions, or unusual proof problems;

(M) ordering a separate trial under Rule 42(b) of a claim, counterclaim, crossclaim, third-party claim, or particular issue;

(N) ordering the presentation of evidence early in the trial on a manageable issue that might, on the evidence, be the basis for a judgment as a matter of law under Rule 50(a) or a judgment on partial findings under Rule 52(c);

(O) establishing a reasonable limit on the time allowed to present evidence; and

(P) facilitating in other ways the just, speedy, and inexpensive disposition of the action.

(d) Pretrial Orders. After any conference under this rule, the court should issue an order reciting the action taken. This order controls the course of the action unless the court modifies it.

(e) Final Pretrial Conference and Orders. The court may hold a final pretrial conference to formulate a trial plan, including a plan to facilitate the admission of evidence. The conference must be held as close to the start of trial as is reasonable, and must be attended by at least one attorney who will conduct the trial for each party and by any unrepresented party. The court may modify the order issued after a final pretrial conference only to prevent manifest injustice.

(f) Sanctions.

(1) *In General.* On motion or on its own, the court may issue any just orders, including those authorized by Rule 37(b)(2)(A)(ii)–(vii), if a party or its attorney:

(A) fails to appear at a scheduling or other pretrial conference;

(B) is substantially unprepared to participate—or does not participate in good faith—in the conference; or

(C) fails to obey a scheduling or other pretrial order.

(2) *Imposing Fees and Costs.* Instead of or in addition to any other sanction, the court must order the party, its attorney, or both to pay the reasonable expenses—including attorney's fees—incurred because of any noncompliance with this rule, unless the noncompliance was substantially justified or other circumstances make an award of expenses unjust.

Notes

(As amended Apr. 28, 1983, eff. Aug. 1, 1983; Mar. 2, 1987, eff. Aug. 1, 1987; Apr. 22, 1993, eff. Dec. 1, 1993; Apr. 12, 2006, eff. Dec. 1, 2006; Apr. 30, 2007, eff. Dec. 1, 2007; Apr. 29, 2015, eff. Dec. 1, 2015.)

Committee Notes on Rules—2006 Amendment

The amendment to Rule 16(b) is designed to alert the court to the possible need to address the handling of discovery of electronically stored information early in the litigation if such discovery is expected to occur. Rule 26(f) is amended to direct the parties to discuss discovery of electronically stored information if such discovery is contemplated in the action. Form 35 is amended to call for a report to the court about the results of this discussion. In many instances, the court's involvement early in the litigation will help avoid difficulties that might otherwise arise.

Rule 16(b) is also amended to include among the topics that may be addressed in the scheduling order any agreements that the parties reach to facilitate discovery by minimizing the risk of waiver of privilege or work-product protection. Rule 26(f) is amended to add to the discovery plan the parties' proposal for the court to enter a case-management or other order adopting such an agreement. The parties may agree to various arrangements. For example, they may agree to initial provision of requested materials without waiver of privilege or protection to enable the party seeking production to designate the materials desired or protection for actual production, with the privilege review of only those materials to follow. Alternatively, they may agree that if privileged or protected information is inadvertently produced, the producing party may by timely notice assert the privilege or protection and obtain return of the materials without waiver. Other arrangements are possible. In most circumstances, a party who receives information under such an arrangement cannot assert that production of the information waived a claim of privilege or of protection as trial-preparation material.

An order that includes the parties' agreement may be helpful in avoiding delay and excessive cost in discovery. *See Manual for Complex Litigation*(4th) §11.446. Rule 16(b)(6) recognizes the propriety of including such agreements in the court's order. The rule does not provide the court with authority to enter such a case-management or other order without party agreement, or limit the court's authority to act on motion.

Committee Notes on Rules—2015 Amendment

The provision for consulting at a scheduling conference by "telephone, mail, or other means" is deleted. A scheduling conference is more effective if the court and parties engage in direct simultaneous communication. The conference may be held in person, by telephone, or by more sophisticated electronic means.

The time to issue the scheduling order is reduced to the earlier of 90 days (not 120 days) after any defendant has been served, or 60 days (not 90 days) after any defendant has appeared. This change, together with the shortened time for making service under Rule 4(m), will reduce delay at the beginning of litigation. At the same time, a new provision recognizes that the court may find good cause to extend the time to issue the scheduling order. In some cases it may be that the parties cannot prepare adequately for a meaningful Rule 26(f) conference and then a scheduling conference in the time allowed. Litigation involving complex issues, multiple parties, and large organizations, public or private, may be more likely to need extra time to establish meaningful collaboration between counsel and the people who can supply the information needed to participate in a useful way. Because the time for the Rule 26(f) conference is geared to the time for the scheduling conference or order, an order extending the time for the scheduling conference will also extend the time for the Rule 26(f) conference. But in most cases it will be desirable to hold at least a first scheduling conference in the time set by the rule.

Three items are added to the list of permitted contents in Rule 16(b)(3)(B).

The order may provide for preservation of electronically stored information, a topic also added to the provisions of a discovery plan under Rule 26(f)(3)(C). Parallel amendments of Rule 37(e) recognize that a duty to preserve discoverable information may arise before an action is filed.

The order also may include agreements incorporated in a court order under Evidence Rule 502 controlling the effects of disclosure of information covered by attorney-client privilege or work-product protection, a topic also added to the provisions of a discovery plan under Rule 26(f)(3)(D).

Finally, the order may direct that before filing a motion for an order relating to discovery the movant must request a conference with the court. Many judges who hold such conferences find them an efficient way to resolve most discovery disputes without the delay and burdens attending a formal motion, but the decision whether to require such conferences is left to the discretion of the judge in each case.

***

Rule 26. Duty to Disclose; General Provisions Governing Discovery

(a) Required Disclosures.

(1) *Initial Disclosure.*

(A) *In General.* Except as exempted by Rule 26(a)(1)(B) or as otherwise stipulated or ordered by the court, a party must, without awaiting a discovery request, provide to the other parties:

(i) the name and, if known, the address and telephone number of each individual likely to have discoverable information—along with the subjects of that information—that the disclosing party may use to support its claims or defenses, unless the use would be solely for impeachment;

(ii) a copy—or a description by category and location—of all documents, electronically stored information, and tangible things that the disclosing party has in its possession, custody, or control and may use to support its claims or defenses, unless the use would be solely for impeachment;

(iii) a computation of each category of damages claimed by the disclosing party—who must also make available for inspection and copying as under Rule 34 the documents or other evidentiary material, unless privileged or protected from disclosure, on which each computation is based, including materials bearing on the nature and extent of injuries suffered; and

(iv) for inspection and copying as under Rule 34, any insurance agreement under which an insurance business may be liable to satisfy all or part of a possible judgment in the action or to indemnify or reimburse for payments made to satisfy the judgment.

(B) *Proceedings Exempt from Initial Disclosure.* The following proceedings are exempt from initial disclosure:

(i) an action for review on an administrative record;

(ii) a forfeiture action in rem arising from a federal statute;

(iii) a petition for habeas corpus or any other proceeding to challenge a criminal conviction or sentence;

(iv) an action brought without an attorney by a person in the custody of the United States, a state, or a state subdivision;

(v) an action to enforce or quash an administrative summons or subpoena;

(vi) an action by the United States to recover benefit payments;

(vii) an action by the United States to collect on a student loan guaranteed by the United States;

(viii) a proceeding ancillary to a proceeding in another court; and

(ix) an action to enforce an arbitration award.

(C) *Time for Initial Disclosures—In General.* A party must make the initial disclosures at or within 14 days after the parties' Rule 26(f)conference unless a different time is set by stipulation or court order, or unless a party objects during the conference that initial disclosures are not appropriate in this action and states the objection in the proposed discovery plan. In ruling on the objection, the court must determine what disclosures, if any, are to be made and must set the time for disclosure.

(D) *Time for Initial Disclosures—For Parties Served or Joined Later.* A party that is first served or otherwise joined after the Rule 26(f)conference must make the initial disclosures within 30 days after being served or joined, unless a different time is set by stipulation or court order.

(E) *Basis for Initial Disclosure; Unacceptable Excuses.* A party must make its initial disclosures based on the information then reasonably available to it. A party is not excused from making its disclosures because it has not fully investigated the case or because it challenges the sufficiency of another party's disclosures or because another party has not made its disclosures.

(2) *Disclosure of Expert Testimony.*

(A) *In General.* In addition to the disclosures required by Rule 26(a)(1), a party must disclose to the other parties the identity of any witness it may use at trial to present evidence under Federal Rule of Evidence 702, 703, or 705.

(B) *Witnesses Who Must Provide a Written Report.* Unless otherwise stipulated or ordered by the court, this disclosure must be accompanied by a written report—prepared and signed by the witness—if the witness is one retained or specially employed to provide expert testimony in the case or one whose duties as the party's employee regularly involve giving expert testimony. The report must contain:

(i) a complete statement of all opinions the witness will express and the basis and reasons for them;

(ii) the facts or data considered by the witness in forming them;

(iii) any exhibits that will be used to summarize or support them;

(iv) the witness's qualifications, including a list of all publications authored in the previous 10 years;

(v) a list of all other cases in which, during the previous 4 years, the witness testified as an expert at trial or by deposition; and

(vi) a statement of the compensation to be paid for the study and testimony in the case.

(C) *Witnesses Who Do Not Provide a Written Report.* Unless otherwise stipulated or ordered by the court, if the witness is not required to provide a written report, this disclosure must state:

(i) the subject matter on which the witness is expected to present evidence under Federal Rule of Evidence 702, 703, or 705; and

(ii) a summary of the facts and opinions to which the witness is expected to testify.

(D) Time to Disclose Expert Testimony. A party must make these disclosures at the times and in the sequence that the court orders. Absent a stipulation or a court order, the disclosures must be made:

(i) at least 90 days before the date set for trial or for the case to be ready for trial; or

(ii) if the evidence is intended solely to contradict or rebut evidence on the same subject matter identified by another party under Rule 26(a)(2)(B) or (C), within 30 days after the other party's disclosure.

(E) *Supplementing the Disclosure.* The parties must supplement these disclosures when required under Rule 26(e).

(3) *Pretrial Disclosures.*

(A) *In General.* In addition to the disclosures required by Rule 26(a)(1) and (2), a party must provide to the other parties and promptly file the following information about the evidence that it may present at trial other than solely for impeachment:

(i) the name and, if not previously provided, the address and telephone number of each witness—separately identifying those the party expects to present and those it may call if the need arises;

(ii) the designation of those witnesses whose testimony the party expects to present by deposition and, if not taken stenographically, a transcript of the pertinent parts of the deposition; and

(iii) an identification of each document or other exhibit, including summaries of other evidence—separately identifying those items the party expects to offer and those it may offer if the need arises.

(B) *Time for Pretrial Disclosures; Objections.* Unless the court orders otherwise, these disclosures must be made at least 30 days before trial. Within 14 days after they are made, unless the court sets a different time, a party may serve and promptly file a list of the following objections: any objections to the use under Rule 32(a) of a deposition designated by another party under Rule 26(a)(3)(A)(ii); and any objection, together with the grounds for it, that may be made to the admissibility of materials identified under Rule 26(a)(3)(A)(iii). An objection not so made—except for one under Federal Rule of Evidence 402 or 403—is waived unless excused by the court for good cause.

(4) *Form of Disclosures.* Unless the court orders otherwise, all disclosures under Rule 26(a) must be in writing, signed, and served.

(b) Discovery Scope and Limits.

(1) *Scope in General.* Unless otherwise limited by court order, the scope of discovery is as follows: Parties may obtain discovery regarding any nonprivileged matter that is relevant to any party's claim or defense and proportional to the needs of the case, considering the importance of the issues

at stake in the action, the amount in controversy, the parties’ relative access to relevant information, the parties’ resources, the importance of the discovery in resolving the issues, and whether the burden or expense of the proposed discovery outweighs its likely benefit. Information within this scope of discovery need not be admissible in evidence to be discoverable.

(2) *Limitations on Frequency and Extent.*

(A) *When Permitted.* By order, the court may alter the limits in these rules on the number of depositions and interrogatories or on the length of depositions under Rule 30. By order or local rule, the court may also limit the number of requests under Rule 36.

(B) *Specific Limitations on Electronically Stored Information.* A party need not provide discovery of electronically stored information from sources that the party identifies as not reasonably accessible because of undue burden or cost. On motion to compel discovery or for a protective order, the party from whom discovery is sought must show that the information is not reasonably accessible because of undue burden or cost. If that showing is made, the court may nonetheless order discovery from such sources if the requesting party shows good cause, considering the limitations of Rule 26(b)(2)(C). The court may specify conditions for the discovery.

(C) *When Required.* On motion or on its own, the court must limit the frequency or extent of discovery otherwise allowed by these rules or by local rule if it determines that:

(i) the discovery sought is unreasonably cumulative or duplicative, or can be obtained from some other source that is more convenient, less burdensome, or less expensive;

(ii) the party seeking discovery has had ample opportunity to obtain the information by discovery in the action; or

(iii) the proposed discovery is outside the scope permitted by Rule 26(b)(1).

(3) *Trial Preparation: Materials.*

(A) *Documents and Tangible Things.* Ordinarily, a party may not discover documents and tangible things that are prepared in anticipation of litigation or for trial by or for another party or its representative (including the other party's attorney, consultant, surety, indemnitor, insurer, or agent). But, subject to Rule 26(b)(4), those materials may be discovered if:

(i) they are otherwise discoverable under Rule 26(b)(1); and

(ii) the party shows that it has substantial need for the materials to prepare its case and cannot, without undue hardship, obtain their substantial equivalent by other means.

(B) *Protection Against Disclosure.* If the court orders discovery of those materials, it must protect against disclosure of the mental impressions, conclusions, opinions, or legal theories of a party's attorney or other representative concerning the litigation.

(C) *Previous Statement.* Any party or other person may, on request and without the required showing, obtain the person's own previous statement about the action or its subject matter. If the request is refused, the person may move for a court order, and Rule 37(a)(5) applies to the award of expenses. A previous statement is either:

(i) a written statement that the person has signed or otherwise adopted or approved; or

(ii) a contemporaneous stenographic, mechanical, electrical, or other recording—or a transcription of it—that recites substantially verbatim the person's oral statement.

(4) *Trial Preparation: Experts.*

(A) *Deposition of an Expert Who May Testify.* A party may depose any person who has been identified as an expert whose opinions may be presented at trial. If Rule 26(a)(2)(B) requires a report from the expert, the deposition may be conducted only after the report is provided.

(B) *Trial-Preparation Protection for Draft Reports or Disclosures.* Rules 26(b)(3)(A) and (B) protect drafts of any report or disclosure required under Rule 26(a)(2), regardless of the form in which the draft is recorded.

(C) *Trial-Preparation Protection for Communications Between a Party's Attorney and Expert Witnesses.* Rules 26(b)(3)(A) and (B) protect communications between the party's attorney and any witness required to provide a report under Rule 26(a)(2)(B), regardless of the form of the communications, except to the extent that the communications:

(i) relate to compensation for the expert's study or testimony;

(ii) identify facts or data that the party's attorney provided and that the expert considered in forming the opinions to be expressed; or

(iii) identify assumptions that the party's attorney provided and that the expert relied on in forming the opinions to be expressed.

(D) *Expert Employed Only for Trial Preparation.* Ordinarily, a party may not, by interrogatories or deposition, discover facts known or opinions held by an expert who has been retained or specially employed by another party in anticipation of litigation or to prepare for trial and who is not expected to be called as a witness at trial. But a party may do so only:

(i) as provided in Rule 35(b); or

(ii) on showing exceptional circumstances under which it is impracticable for the party to obtain facts or opinions on the same subject by other means.

(E) *Payment.* Unless manifest injustice would result, the court must require that the party seeking discovery:

(i) pay the expert a reasonable fee for time spent in responding to discovery under Rule 26(b)(4)(A) or (D); and

(ii) for discovery under (D), also pay the other party a fair portion of the fees and expenses it reasonably incurred in obtaining the expert's facts and opinions.

(5) *Claiming Privilege or Protecting Trial-Preparation Materials.*

(A) *Information Withheld.* When a party withholds information otherwise discoverable by claiming that the information is privileged or subject to protection as trial-preparation material, the party must:

(i) expressly make the claim; and

(ii) describe the nature of the documents, communications, or tangible things not produced or disclosed—and do so in a manner that, without revealing information itself privileged or protected, will enable other parties to assess the claim.

(B) *Information Produced.* If information produced in discovery is subject to a claim of privilege or of protection as trial-preparation material, the party making the claim may notify any party that received the information of the claim and the basis for it. After being notified, a party must promptly return, sequester, or destroy the specified information and any copies it has; must not use or disclose the information until the claim is resolved; must take reasonable steps to retrieve the information if the party disclosed it before being notified; and may promptly present the information to the court under seal for a determination of the claim. The producing party must preserve the information until the claim is resolved.

(c) Protective Orders.

(1) *In General.* A party or any person from whom discovery is sought may move for a protective order in the court where the action is pending—or as an alternative on matters relating to a deposition, in the court for the district where the deposition will be taken. The motion must include a certification that the movant has in good faith conferred or attempted to confer with other affected parties in an effort to resolve the dispute without court action. The court may, for good cause, issue an order to protect a party or person from annoyance, embarrassment, oppression, or undue burden or expense, including one or more of the following:

(A) forbidding the disclosure or discovery;

(B) specifying terms, including time and place or the allocation of expenses, for the disclosure or discovery;

(C) prescribing a discovery method other than the one selected by the party seeking discovery;

(D) forbidding inquiry into certain matters, or limiting the scope of disclosure or discovery to certain matters;

(E) designating the persons who may be present while the discovery is conducted;

(F) requiring that a deposition be sealed and opened only on court order;

(G) requiring that a trade secret or other confidential research, development, or commercial information not be revealed or be revealed only in a specified way; and

(H) requiring that the parties simultaneously file specified documents or information in sealed envelopes, to be opened as the court directs.

(2) *Ordering Discovery.* If a motion for a protective order is wholly or partly denied, the court may, on just terms, order that any party or person provide or permit discovery.

(3) *Awarding Expenses.* Rule 37(a)(5) applies to the award of expenses.

(d) Timing and Sequence of Discovery.

(1) *Timing.* A party may not seek discovery from any source before the parties have conferred as required by Rule 26(f), except in a proceeding exempted from initial disclosure under Rule 26(a)(1)(B), or when authorized by these rules, by stipulation, or by court order.

(2) Early Rule 34 Requests.

Time to Deliver. More than 21 days after the summons and complaint are served on a party, a request under Rule 34 may be delivered:

(i) to that party by any other party, and

(ii) by that party to any plaintiff or to any other party that has been served.

(B) When Considered Served. The request is considered to have been served at the first Rule 26(f) conference.

(3) *Sequence.* Unless the parties stipulate or the court orders otherwise for the parties' and witnesses' convenience and in the interests of justice:

(A) methods of discovery may be used in any sequence; and

(B) discovery by one party does not require any other party to delay its discovery.

(e) Supplementing Disclosures and Responses.

(1) *In General.* A party who has made a disclosure under Rule 26(a)—or who has responded to an interrogatory, request for production, or request for admission—must supplement or correct its disclosure or response:

(A) in a timely manner if the party learns that in some material respect the disclosure or response is incomplete or incorrect, and if the additional or corrective information has not otherwise been made known to the other parties during the discovery process or in writing; or

(B) as ordered by the court.

(2) *Expert Witness.* For an expert whose report must be disclosed under Rule 26(a)(2)(B), the party's duty to supplement extends both to information included in the report and to information given during the expert's deposition. Any additions or changes to this information must be disclosed by the time the party's pretrial disclosures under Rule 26(a)(3) are due.

(f) Conference of the Parties; Planning for Discovery.

(1) *Conference Timing.* Except in a proceeding exempted from initial disclosure under Rule 26(a)(1)(B) or when the court orders otherwise, the parties must confer as soon as practicable—and in any event at least 21 days before a scheduling conference is to be held or a scheduling order is due under Rule 16(b).

(2) *Conference Content; Parties' Responsibilities.* In conferring, the parties must consider the nature and basis of their claims and defenses and the possibilities for promptly settling or resolving the case; make or arrange for the disclosures required by Rule 26(a)(1); discuss any issues about preserving discoverable information; and develop a proposed discovery plan. The attorneys of record and all unrepresented parties that have appeared in the case are jointly responsible for arranging the conference, for attempting in good faith to agree on the proposed discovery plan, and for submitting to the court within 14 days after the conference a written report outlining the plan. The court may order the parties or attorneys to attend the conference in person.

(3) *Discovery Plan.* A discovery plan must state the parties' views and proposals on:

(A) what changes should be made in the timing, form, or requirement for disclosures under Rule 26(a), including a statement of when initial disclosures were made or will be made;

(B) the subjects on which discovery may be needed, when discovery should be completed, and whether discovery should be conducted in phases or be limited to or focused on particular issues;

(C) any issues about disclosure, discovery, or preservation of electronically stored information, including the form or forms in which it should be produced;

(D) any issues about claims of privilege or of protection as trial-preparation materials, including—if the parties agree on a procedure to assert these claims after production—whether to ask the court to include their agreement in an order under Federal Rule of Evidence 502;

(E) what changes should be made in the limitations on discovery imposed under these rules or by local rule, and what other limitations should be imposed; and

(F) any other orders that the court should issue under Rule 26(c) or under Rule 16(b) and (c).

(4) *Expedited Schedule.* If necessary to comply with its expedited schedule for Rule 16(b) conferences, a court may by local rule:

(A) require the parties' conference to occur less than 21 days before the scheduling conference is held or a scheduling order is due under Rule 16(b); and

(B) require the written report outlining the discovery plan to be filed less than 14 days after the parties' conference, or excuse the parties from submitting a written report and permit them to report orally on their discovery plan at the Rule 16(b) conference.

(g) Signing Disclosures and Discovery Requests, Responses, and Objections.

(1) *Signature Required; Effect of Signature.* Every disclosure under Rule 26(a)(1) or (a)(3) and every discovery request, response, or objection must be signed by at least one attorney of record in the attorney's own name—or by the party personally, if unrepresented—and must state the signer's address, e-mail address, and telephone number. By signing, an attorney or party certifies that to the best of the person's knowledge, information, and belief formed after a reasonable inquiry:

(A) with respect to a disclosure, it is complete and correct as of the time it is made; and

(B) with respect to a discovery request, response, or objection, it is:

(i) consistent with these rules and warranted by existing law or by a nonfrivolous argument for extending, modifying, or reversing existing law, or for establishing new law;

(ii) not interposed for any improper purpose, such as to harass, cause unnecessary delay, or needlessly increase the cost of litigation; and

(iii) neither unreasonable nor unduly burdensome or expensive, considering the needs of the case, prior discovery in the case, the amount in controversy, and the importance of the issues at stake in the action.

(2) *Failure to Sign.* Other parties have no duty to act on an unsigned disclosure, request, response, or objection until it is signed, and the court must strike it unless a signature is promptly supplied after the omission is called to the attorney's or party's attention.

(3) *Sanction for Improper Certification.* If a certification violates this rule without substantial justification, the court, on motion or on its own, must impose an appropriate sanction on the signer, the party on whose behalf the signer was acting, or both. The sanction may include an order to pay the reasonable expenses, including attorney's fees, caused by the violation.

Notes

(As amended Dec. 27, 1946, eff. Mar. 19, 1948; Jan. 21, 1963, eff. July 1, 1963; Feb. 28, 1966, eff. July 1, 1966; Mar. 30, 1970, eff. July 1, 1970; Apr. 29, 1980, eff. Aug. 1, 1980; Apr. 28, 1983, eff. Aug. 1, 1983; Mar. 2, 1987, eff. Aug. 1, 1987; Apr. 22, 1993, eff. Dec. 1, 1993; Apr. 17, 2000, eff. Dec. 1, 2000; Apr. 12, 2006, eff. Dec. 1, 2006; Apr. 30, 2007, eff. Dec. 1, 2007; Apr. 28, 2010, eff. Dec. 1, 2010; Apr. 29, 2015, eff. Dec. 1, 2015.)

Committee Notes on Rules—2006 Amendment

Subdivision (a). Rule 26(a)(1)(B) is amended to parallel Rule 34(a) by recognizing that a party must disclose electronically stored information as well as documents that it may use to support its claims or defenses. The term “electronically stored information” has the same broad meaning in Rule 26(a)(1) as in Rule 34(a). This amendment is consistent with the 1993 addition of Rule 26(a)(1)(B). The term “data compilations” is deleted as unnecessary because it is a subset of both documents and electronically stored information.

Changes Made After Publication and Comment. As noted in the introduction [omitted], this provision was not included in the published rule. It is included as a conforming amendment, to make Rule 26(a)(1) consistent with the changes that were included in the published proposals.

[*Subdivision (a)(1)(E)*.] Civil forfeiture actions are added to the list of exemptions from Rule 26(a)(1) disclosure requirements. These actions are governed by new Supplemental Rule G. Disclosure is not likely to be useful.

Subdivision (b)(2). The amendment to Rule 26(b)(2) is designed to address issues raised by difficulties in locating, retrieving, and providing discovery of some electronically stored information. Electronic storage systems often make it easier to locate and retrieve information. These advantages are properly taken into account in determining the reasonable scope of discovery in a particular case. But some sources of electronically stored information can be accessed only with substantial burden and cost. In a particular case, these burdens and costs may make the information on such sources not reasonably accessible.

It is not possible to define in a rule the different types of technological features that may affect the burdens and costs of accessing electronically stored information. Information systems are designed to provide ready access to information used in regular ongoing activities. They also may be designed so as to provide ready access to information that is not regularly used. But a system may retain information on sources that are accessible only by incurring substantial burdens or costs. Subparagraph (B) is added to regulate discovery from such sources.

Under this rule, a responding party should produce electronically stored information that is relevant, not privileged, and reasonably accessible, subject to the (b)(2)(C) limitations that apply to all discovery. The responding party must also identify, by category or type, the sources containing potentially responsive information that it is neither searching nor producing. The identification should, to the extent possible, provide enough detail to enable the requesting party to evaluate the burdens and costs of providing the discovery and the likelihood of finding responsive information on the identified sources.

A party's identification of sources of electronically stored information as not reasonably accessible does not relieve the party of its common-law or statutory duties to preserve evidence. Whether a responding party is required to preserve unsearched sources of potentially responsive information that it believes are not reasonably accessible depends on the circumstances of each case. It is often useful for the parties to discuss this issue early in discovery.

The volume of—and the ability to search—much electronically stored information means that in many cases the responding party will be able to produce information from reasonably accessible sources that will fully satisfy the parties' discovery needs. In many circumstances the requesting party should obtain and evaluate the information from such sources before insisting that the responding party search and produce information contained on sources that are not reasonably accessible. If the requesting party continues to seek discovery of information from sources identified as not reasonably accessible, the parties should discuss the burdens and costs of accessing and retrieving the information, the needs that may establish good cause for requiring all or part of the requested discovery even if the information sought is not reasonably accessible, and conditions on obtaining and producing the information that may be appropriate.

If the parties cannot agree whether, or on what terms, sources identified as not reasonably accessible should be searched and discoverable information produced, the issue may be raised either by a motion to compel discovery or by a motion for a protective order. The parties must confer before bringing either motion. If the parties do not resolve the issue and the court must decide, the responding party must show that the identified sources of information are not reasonably accessible because of undue burden or cost. The requesting party may need discovery to test this assertion. Such discovery might take the form of requiring the responding party to conduct a sampling of information contained on the sources identified as not reasonably accessible; allowing some form of inspection of such sources; or taking depositions of witnesses knowledgeable about the responding party's information systems.

Once it is shown that a source of electronically stored information is not reasonably accessible, the requesting party may still obtain discovery by showing good cause, considering the limitations of Rule 26(b)(2)(C) that balance the costs and potential benefits of discovery. The decision whether to require a responding party to search for and produce information that is not reasonably accessible depends not only on the burdens and costs of doing so, but also on whether those burdens and costs can be justified in the circumstances of the case. Appropriate considerations may include: (1) the specificity of the discovery request; (2) the quantity of information available from other and more easily accessed sources; (3) the failure to produce relevant information that seems likely to have existed but is no longer available on more easily accessed sources; (4) the likelihood of finding relevant, responsive information that cannot be obtained from other, more easily accessed sources; (5) predictions as to the importance and usefulness of the further information; (6) the importance of the issues at stake in the litigation; and (7) the parties' resources.

The responding party has the burden as to one aspect of the inquiry—whether the identified sources are not reasonably accessible in light of the burdens and costs required to search for, retrieve, and produce whatever responsive information may be found. The requesting party has the burden of showing that its need for the discovery outweighs the burdens and costs of locating, retrieving, and producing the information. In some cases, the court will be able to determine whether the identified sources are not reasonably accessible and whether the requesting party has shown good cause for some or all of the discovery, consistent with the limitations of Rule 26(b)(2)(C), through a single proceeding or presentation. The good-cause determination, however, may be complicated because the court and parties may know little about what information the sources identified as not reasonably accessible might contain, whether it is relevant, or how valuable it may be to the litigation. In such cases, the parties may need some focused discovery, which may include sampling of the sources, to learn more about what burdens and costs are involved in accessing the information, what the information consists of, and how valuable it is for the litigation in light of information that can be obtained by exhausting other opportunities for discovery.

The good-cause inquiry and consideration of the Rule 26(b)(2)(C) limitations are coupled with the authority to set conditions for discovery. The conditions may take the form of limits on the amount, type, or sources of information required to be accessed and produced. The conditions may also include payment by the requesting party of part or all of the reasonable costs of obtaining information from sources that are not reasonably accessible. A requesting party's willingness to share or bear the access costs may be weighed by the court in determining whether there is good cause. But the producing party's burdens in reviewing the information for relevance and privilege may weigh against permitting the requested discovery.

The limitations of Rule 26(b)(2)(C) continue to apply to all discovery of electronically stored information, including that stored on reasonably accessible electronic sources.

Changes Made after Publication and Comment. This recommendation modifies the version of the proposed rule amendment as published. Responding to comments that the published proposal seemed to require identification of information that cannot be identified because it is not reasonably accessible, the rule text was clarified by requiring identification of sources that are not reasonably accessible. The test of reasonable accessibility was clarified by adding "because of undue burden or cost."

The published proposal referred only to a motion by the requesting party to compel discovery. The rule text has been changed to recognize that the responding party may wish to determine its search and potential preservation obligations by moving for a protective order.

The provision that the court may for good cause order discovery from sources that are not reasonably accessible is expanded in two ways. It now states specifically that the requesting party is the one who must show good cause, and it refers to consideration of the limitations on discovery set out in present Rule 26(b)(2)(i), (ii), and (iii).

The published proposal was added at the end of present Rule 26(b)(2). It has been relocated to become a new subparagraph (B), allocating present Rule 26(b)(2) to new subparagraphs (A) and (C). The Committee Note was changed to reflect the rule text revisions. It also was shortened. The shortening was accomplished in part by deleting references to problems that are likely to become antique as technology continues to evolve, and in part by deleting passages that were at a level of detail better suited for a practice manual than a Committee Note.

The changes from the published proposed amendment to Rule 26(b)(2) are set out below. [Omitted]

Subdivision (b)(5). The Committee has repeatedly been advised that the risk of privilege waiver, and the work necessary to avoid it, add to the costs and delay of discovery. When the review is of electronically stored information, the risk of waiver, and the time and effort required to avoid it, can increase substantially because of the volume of electronically stored information and the difficulty in ensuring that all information to be produced has in fact been reviewed. Rule 26(b)(5)(A) provides a procedure for a party that has withheld information on the basis of privilege or protection as trial-preparation material to make the claim so that the requesting party can decide whether to contest the claim and the court can resolve the dispute. Rule 26(b)(5)(B) is added to provide a procedure for a party to assert a claim of privilege or trial-preparation material protection after information is produced in discovery in the action and, if the claim is contested, permit any party that received the information to present the matter to the court for resolution.

Rule 26(b)(5)(B) does not address whether the privilege or protection that is asserted after production was waived by the production. The courts have developed principles to determine whether, and under what circumstances, waiver results from inadvertent production of privileged or protected information. Rule 26(b)(5)(B) provides a procedure for presenting and addressing these issues. Rule 26(b)(5)(B) works in tandem with Rule 26(f), which is amended to direct the parties to discuss privilege issues in preparing their discovery plan, and which, with amended Rule 16(b), allows the parties to ask the court to include in an order any agreements the parties reach regarding issues of privilege or trial-preparation material protection. Agreements reached under Rule 26(f)(4) and orders including such agreements entered under Rule 16(b)(6) may be considered when a court determines whether a waiver has occurred. Such agreements and orders ordinarily control if they adopt procedures different from those in Rule 26(b)(5)(B).

A party asserting a claim of privilege or protection after production must give notice to the receiving party. That notice should be in writing unless the circumstances preclude it. Such circumstances could include the assertion of the claim during a deposition. The notice should be as specific as possible in identifying the information and stating the basis for the claim. Because the receiving party must decide whether to challenge the claim and may sequester the information and submit it to the court for a ruling on whether the claimed privilege or protection applies and whether it has been waived, the notice should be sufficiently detailed so as to enable the receiving party and the court to understand the basis for the claim and to determine whether waiver has occurred. Courts will continue to examine whether a claim of privilege or protection was made at a reasonable time when delay is part of the waiver determination under the governing law.

After receiving notice, each party that received the information must promptly return, sequester, or destroy the information and any copies it has. The option of sequestering or destroying the information is included in part because the receiving party may have incorporated the information in protected trial-preparation materials. No receiving party may use or disclose the information pending resolution of the privilege claim. The receiving party may present to the court the questions whether the information is privileged or protected as trial-preparation material, and whether the privilege or protection has been waived. If it does so, it must provide the court with the grounds for the privilege or protection specified in the producing party's notice, and serve all parties. In presenting the question, the party may use the content of the information only to the extent permitted by the applicable law of privilege, protection for trial-preparation material, and professional responsibility.

If a party disclosed the information to nonparties before receiving notice of a claim of privilege or protection as trial-preparation material, it must take reasonable steps to retrieve the information and to return it, sequester it until the claim is resolved, or destroy it.

Whether the information is returned or not, the producing party must preserve the information pending the court's ruling on whether the claim of privilege or of protection is properly asserted and whether it was waived. As with claims made under Rule 26(b)(5)(A), there may be no ruling if the other parties do not contest the claim.

Changes Made After Publication and Comment. The rule recommended for approval is modified from the published proposal. The rule is expanded to include trial-preparation protection claims in addition to privilege claims.

The published proposal referred to production "without intending to waive a claim of privilege." This reference to intent was deleted because many courts include intent in the factors that determine whether production waives privilege.

The published proposal required that the producing party give notice "within a reasonable time." The time requirement was deleted because it seemed to implicate the question whether production effected a waiver, a question not addressed by the rule, and also because a receiving party cannot practicably ignore a notice that it believes was unreasonably delayed. The notice procedure was further changed to require that the producing party state the basis for the claim.

Two statements in the published Note have been brought into the rule text. The first provides that the receiving party may not use or disclose the information until the claim is resolved. The second provides that if the receiving party disclosed the information before being notified, it must take reasonable steps to retrieve it. [1]

The rule text was expanded by adding a provision that the receiving party may promptly present the information to the court under seal for a determination of the claim.

The published proposal provided that the producing party must comply with Rule 26(b)(5)(A) after making the claim. This provision was deleted as unnecessary.

Changes are made in the Committee Note to reflect the changes in the rule text.

The changes from the published rule are shown below. [Omitted]

Subdivision (f). Rule 26(f) is amended to direct the parties to discuss discovery of electronically stored information during their discovery-planning conference. The rule focuses on "issues relating to disclosure or discovery of electronically stored information"; the discussion is not required in cases not involving electronic discovery, and the amendment imposes no additional requirements in those cases. When the parties do anticipate disclosure or discovery of electronically stored information, discussion at the outset may avoid later difficulties or ease their resolution.

When a case involves discovery of electronically stored information, the issues to be addressed during the Rule 26(f) conference depend on the nature and extent of the contemplated discovery and of the parties' information systems. It may be important for the parties to discuss those systems, and accordingly important for counsel to become familiar with those systems before the conference. With that information, the parties can develop a discovery plan that takes into account the capabilities of their computer systems. In appropriate cases identification of, and early discovery from, individuals with special knowledge of a party's computer systems may be helpful.

The particular issues regarding electronically stored information that deserve attention during the discovery planning stage depend on the specifics of the given case. *See Manual for Complex Litigation* (4th) §40.25(2) (listing topics for discussion in a proposed order regarding meet-and-confer sessions). For example, the parties may specify the topics for such discovery and the time period for which discovery will be sought. They may identify the various sources of such information within a party's control that should be searched for electronically stored information. They may discuss whether the information is reasonably accessible to the party that has it, including the burden or cost of retrieving and reviewing the information. *See* Rule 26(b)(2)(B). Rule 26(f)(3) explicitly directs the parties to discuss the form or forms in which electronically stored information might be produced. The parties may be able to reach agreement on the forms of production, making discovery more efficient. Rule 34(b) is amended to permit a requesting party to specify the form or forms in which it wants electronically stored information produced. If the requesting party does not specify a form, Rule 34(b) directs the responding party to state the forms it intends to use in the production. Early discussion of the forms of production may facilitate the application of Rule 34(b) by allowing the parties to determine what forms of production will meet both parties' needs. Early identification of disputes over the forms of production may help avoid the expense and delay of searches or productions using inappropriate forms.

Rule 26(f) is also amended to direct the parties to discuss any issues regarding preservation of discoverable information during their conference as they develop a discovery plan. This provision applies to all sorts of discoverable information, but can be particularly important with regard to electronically stored information. The volume and dynamic nature of electronically stored information may complicate preservation obligations. The ordinary operation of computers involves both the automatic creation and the automatic deletion or overwriting of certain information. Failure to address preservation issues early in the litigation increases uncertainty and raises a risk of disputes.

The parties' discussion should pay particular attention to the balance between the competing needs to preserve relevant evidence and to continue routine operations critical to ongoing activities. Complete or broad cessation of a party's routine computer operations could paralyze the party's activities. *Cf. Manual for Complex Litigation* (4th) §11.422 ("A blanket preservation order may be prohibitively expensive and unduly burdensome for parties dependent on computer systems for their day-to-day operations.") The parties should take account of these considerations in their discussions, with the goal of agreeing on reasonable preservation steps.

The requirement that the parties discuss preservation does not imply that courts should routinely enter preservation orders. A preservation order entered over objections should be narrowly tailored. Ex parte preservation orders should issue only in exceptional circumstances.

Rule 26(f) is also amended to provide that the parties should discuss any issues relating to assertions of privilege or of protection as trial-preparation materials, including whether the parties can facilitate discovery by agreeing on procedures for asserting claims of privilege or protection after production and whether to ask the court to enter an order that includes any agreement the parties reach. The Committee has repeatedly been advised about the discovery difficulties that can result from efforts to guard against waiver of privilege and work-product protection. Frequently parties find it necessary to spend large amounts of time reviewing materials requested through discovery to avoid waiving privilege. These efforts are necessary because materials subject to a claim of privilege or protection are often difficult to identify. A failure to withhold even one such item may result in an argument that there has been a waiver of privilege as to all other privileged materials on that subject matter. Efforts to avoid the risk of waiver can impose substantial costs on the party producing the material and the time required for the privilege review can substantially delay access for the party seeking discovery.

These problems often become more acute when discovery of electronically stored information is sought. The volume of such data, and the informality that attends use of e-mail and some other types of electronically stored information, may make privilege determinations more difficult, and privilege review correspondingly more expensive and time consuming. Other aspects of electronically stored information pose particular difficulties for privilege review. For example, production may be sought of information automatically included in electronic files but not apparent to the creator or to readers. Computer programs may retain draft language, editorial comments, and other deleted matter (sometimes referred to as "embedded data" or "embedded edits") in an electronic file but not make them apparent to the reader. Information describing the history, tracking, or management of an electronic file (sometimes called "metadata") is usually not apparent to the reader viewing a hard copy or a screen image. Whether this information should be produced may be among the topics discussed in the Rule 26(f) conference. If it is, it may need to be reviewed to ensure that no privileged information is included, further complicating the task of privilege review.

Parties may attempt to minimize these costs and delays by agreeing to protocols that minimize the risk of waiver. They may agree that the responding party will provide certain requested materials for initial examination without waiving any privilege or protection—sometimes known as a "quick peek." The requesting party then designates the documents it wishes to have actually produced. This designation is the Rule 34 request. The responding party then responds in the usual course, screening only those documents actually requested for formal production and asserting privilege claims as provided in Rule 26(b)(5)(A). On other occasions, parties enter agreements—sometimes called "clawback agreements"—that production without intent to waive privilege or protection should not be a waiver so long as the responding party identifies the documents mistakenly produced, and that the documents should be returned under those circumstances. Other voluntary arrangements may be appropriate depending on the circumstances of each litigation. In most circumstances, a party who receives information under such an arrangement cannot assert that production of the information waived a claim of privilege or of protection as trial-preparation material.

Although these agreements may not be appropriate for all cases, in certain cases they can facilitate prompt and economical discovery by reducing delay before the discovering party obtains access to documents, and by reducing the cost and burden of review by the producing party. A case-management or other order including such agreements may further facilitate the discovery process. Form 35 is amended to include a report to the court about any agreement regarding protections against inadvertent forfeiture or waiver of privilege or protection that the parties have reached, and Rule 16(b) is amended to recognize that the court may include such an agreement in a case-management or other order. If the parties agree to entry of such an order, their proposal should be included in the report to the court.

Rule 26(b)(5)(B) is added to establish a parallel procedure to assert privilege or protection as trial-preparation material after production, leaving the question of waiver to later determination by the court.

Changes Made After Publication and Comment. The Committee recommends a modified version of what was published. Rule 26(f)(3) was expanded to refer to the form "or forms" of production, in parallel with the like change in Rule 34. Different forms may be suitable for different sources of electronically stored information.

The published Rule 26(f)(4) proposal described the parties' views and proposals concerning whether, on their agreement, the court should enter an order protecting the right to assert privilege after production. This has been revised to refer to the parties' views and proposals concerning any issues relating to claims of privilege, including—if the parties agree on a procedure to assert such claims after production—whether to ask the court to include their agreement in an order. As with Rule 16(b)(6), this change was made to avoid any implications as to the scope of the protection that may be afforded by court adoption of the parties' agreement.

Rule 26(f)(4) also was expanded to include trial-preparation materials.

The Committee Note was revised to reflect the changes in the rule text.

The changes from the published rule are shown below. [Omitted]

Committee Notes on Rules—2015 Amendment

Rule 26(b)(1) is changed in several ways.

Information is discoverable under revised Rule 26(b)(1) if it is relevant to any party's claim or defense and is proportional to the needs of the case. The considerations that bear on proportionality are moved from present Rule 26(b)(2)(C)(iii), slightly rearranged and with one addition.

Most of what now appears in Rule 26(b)(2)(C)(iii) was first adopted in 1983. The 1983 provision was explicitly adopted as part of the scope of discovery defined by Rule 26(b)(1). Rule 26(b)(1) directed the court to limit the frequency or extent of use of discovery if it determined that "the discovery is unduly burdensome or expensive, taking into account the needs of the case, the amount in controversy, limitations on the parties' resources, and the importance of the issues at stake in the litigation." At the same time, Rule 26(g) was added. Rule 26(g) provided that signing a discovery request, response, or objection certified that the request, response, or objection was "not unreasonable or unduly burdensome or expensive, given the needs of the case, the discovery already had in the case, the amount in controversy, and the importance of the issues at stake in the litigation." The parties thus shared the responsibility to honor these limits on the scope of discovery.

The 1983 Committee Note stated that the new provisions were added "to deal with the problem of overdiscovery. The objective is to guard against redundant or disproportionate discovery by giving the court authority to reduce the amount of discovery that may be directed to matters that are otherwise proper subjects of inquiry. The new sentence is intended to encourage judges to be more aggressive in identifying and discouraging discovery overuse. The grounds mentioned in the amended rule for limiting discovery reflect the existing practice of many courts in issuing protective orders under Rule 26(c).... On the whole, however, district judges have been reluctant to limit the use of the discovery devices."

The clear focus of the 1983 provisions may have been softened, although inadvertently, by the amendments made in 1993. The 1993 Committee Note explained: "[F]ormer paragraph (b)(1) [was] subdivided into two paragraphs for ease of reference and to avoid renumbering of paragraphs (3) and (4)." Subdividing the paragraphs, however, was done in a way that could be read to separate the proportionality provisions as "limitations," no longer an integral part of the (b)(1) scope provisions. That appearance was immediately offset by the next statement in the Note: "Textual changes are then made in new paragraph (2) to enable the court to keep tighter rein on the extent of discovery."

The 1993 amendments added two factors to the considerations that bear on limiting discovery: whether "the burden or expense of the proposed discovery outweighs its likely benefit," and "the importance of the proposed discovery in resolving the issues." Addressing these and other limitations added by the 1993 discovery amendments, the Committee Note stated that "[t]he revisions in Rule 26(b)(2) are intended to provide the court with broader discretion to impose additional restrictions on the scope and extent of discovery . . . ."

The relationship between Rule 26(b)(1) and (2) was further addressed by an amendment made in 2000 that added a new sentence at the end of (b)(1): "All discovery is subject to the limitations imposed by Rule 26(b)(2)(i), (ii), and (iii)[now Rule 26(b)(2)(C)]." The Committee Note recognized that "[t]hese limitations apply to discovery that is otherwise within the scope of subdivision (b)(1)." It explained that the Committee had been told repeatedly that courts were not using these limitations as originally intended. "This otherwise redundant cross-reference has been added to emphasize the need for active judicial use of subdivision (b)(2) to control excessive discovery."

The present amendment restores the proportionality factors to their original place in defining the scope of discovery. This change reinforces the Rule 26(g) obligation of the parties to consider these factors in making discovery requests, responses, or objections.

Restoring the proportionality calculation to Rule 26(b)(1) does not change the existing responsibilities of the court and the parties to consider proportionality, and the change does not place on the party seeking discovery the burden of addressing all proportionality considerations.

Nor is the change intended to permit the opposing party to refuse discovery simply by making a boilerplate objection that it is not proportional. The parties and the court have a collective responsibility to consider the proportionality of all discovery and consider it in resolving discovery disputes.

The parties may begin discovery without a full appreciation of the factors that bear on proportionality. A party requesting discovery, for example, may have little information about the burden or expense of responding. A party requested to provide discovery may have little information about the importance of the discovery in resolving the issues as understood by the requesting party. Many of these uncertainties should be addressed and reduced in the parties' Rule 26(f) conference and in scheduling and pretrial conferences with the court. But if the parties continue to disagree, the discovery dispute could be brought before the court and the parties' responsibilities would remain as they have been since 1983. A party claiming undue burden or expense ordinarily has far better information — perhaps the only information — with respect to that part of the determination. A party claiming that a request is important to resolve the issues should be able to explain the ways in which the underlying information bears on the issues as that party understands them. The court's responsibility, using all the information provided by the parties, is to consider these and all the other factors in reaching a case-specific determination of the appropriate scope of discovery.

The direction to consider the parties' relative access to relevant information adds new text to provide explicit focus on considerations already implicit in present Rule 26(b)(2)(C)(iii). Some cases involve what often is called "information asymmetry." One party — often an individual plaintiff — may have very little discoverable information. The other party may have vast amounts of information, including information that can be readily retrieved and information that is more difficult to retrieve. In practice these circumstances often mean that the burden of responding to discovery lies heavier on the party who has more information, and properly so.

Restoring proportionality as an express component of the scope of discovery warrants repetition of parts of the 1983 and 1993 Committee Notes that must not be lost from sight. The 1983 Committee Note explained that "[t]he rule contemplates greater judicial involvement in the discovery process and thus acknowledges the reality that it cannot always operate on a self-regulating basis." The 1993 Committee Note further observed that "[t]he information explosion of recent decades has greatly increased both the potential cost of wide-ranging discovery and the potential for discovery to be used as an instrument for delay or oppression." What seemed an explosion in 1993 has been exacerbated by the advent of e-discovery. The present amendment again reflects the need for continuing and close judicial involvement in the cases that do not yield readily to the ideal of effective party management. It is expected that discovery will be effectively managed by the parties in many cases. But there will be important occasions for judicial management, both when the parties are legitimately unable to resolve important differences and when the parties fall short of effective, cooperative management on their own.

It also is important to repeat the caution that the monetary stakes are only one factor, to be balanced against other factors. The 1983 Committee Note recognized "the significance of the substantive issues, as measured in philosophic, social, or institutional terms. Thus the rule recognizes that many cases in public policy spheres, such as employment practices, free speech, and other matters, may have importance far beyond the monetary amount involved." Many other substantive areas also may involve litigation that seeks relatively small amounts of money, or no money at all, but that seeks to vindicate vitally important personal or public values.

So too, consideration of the parties' resources does not foreclose discovery requests addressed to an impecunious party, nor justify unlimited discovery requests addressed to a wealthy party. The 1983 Committee Note cautioned that "[t]he court must apply the standards in an even-handed manner that will prevent use of discovery to wage a war of attrition or as a device to coerce a party, whether financially weak or affluent."

The burden or expense of proposed discovery should be determined in a realistic way. This includes the burden or expense of producing electronically stored information. Computer-based methods of searching such information continue to develop, particularly for cases involving large volumes of electronically stored information. Courts and parties should be willing to consider the opportunities for reducing the burden or expense of discovery as reliable means of searching electronically stored information become available.

A portion of present Rule 26(b)(1) is omitted from the proposed revision. After allowing discovery of any matter relevant to any party's claim or defense, the present rule adds: "including the existence, description, nature, custody, condition, and location of any documents or other tangible things and the identity and location of persons who know of any discoverable matter." Discovery of such matters is so deeply entrenched in practice that it is no longer necessary to clutter the long text of Rule 26 with these examples. The discovery identified in these examples should still be permitted under the revised rule when relevant and proportional to the needs of the case. Framing intelligent requests for electronically stored information, for example, may require detailed information about another party's information systems and other information resources.

The amendment deletes the former provision authorizing the court, for good cause, to order discovery of any matter relevant to the subject matter involved in the action. The Committee has been informed that this language is rarely invoked. Proportional discovery relevant to any party's claim or defense suffices, given a proper understanding of what is relevant to a claim or defense. The distinction between matter relevant to a claim or defense and matter relevant to the subject matter was introduced in 2000. The 2000 Note offered three examples of information that, suitably focused, would be relevant to the parties' claims or defenses. The examples were "other incidents of the same type, or involving the same product"; "information about organizational arrangements or filing systems"; and "information that could be used to impeach a likely witness." Such discovery is not foreclosed by the amendments. Discovery that is relevant to the parties' claims or defenses may also support amendment of the pleadings to add a new claim or defense that affects the scope of discovery.

The former provision for discovery of relevant but inadmissible information that appears "reasonably calculated to lead to the discovery of admissible evidence" is also deleted. The phrase has been used by some, incorrectly, to define the scope of discovery. As the Committee Note to the 2000 amendments observed, use of the "reasonably calculated" phrase to define the scope of discovery "might swallow any other limitation on the scope of discovery." The 2000 amendments sought to prevent such misuse by adding the word "Relevant" at the beginning of the sentence, making clear that "'relevant' means within the scope of discovery as defined in this subdivision . . . ." The "reasonably calculated" phrase has continued to create problems, however, and is removed by these amendments. It is replaced by the direct statement that "Information within this scope of discovery need not be admissible in evidence to be discoverable." Discovery of nonprivileged information not admissible in evidence remains available so long as it is otherwise within the scope of discovery.

Rule 26(b)(2)(C)(iii) is amended to reflect the transfer of the considerations that bear on proportionality to Rule 26(b)(1). The court still must limit the frequency or extent of proposed discovery, on motion or on its own, if it is outside the scope permitted by Rule 26(b)(1).

Rule 26(c)(1)(B) is amended to include an express recognition of protective orders that allocate expenses for disclosure or discovery. Authority to enter such orders is included in the present rule, and courts already exercise this authority. Explicit recognition will forestall the temptation some parties may feel to contest this authority. Recognizing the authority does not imply that cost-shifting should become a common practice. Courts and parties should continue to assume that a responding party ordinarily bears the costs of responding.

Rule 26(d)(2) is added to allow a party to deliver Rule 34 requests to another party more than 21 days after that party has been served even though the parties have not yet had a required Rule 26(f) conference. Delivery may be made by any party to the party that has been served, and by that party to any plaintiff and any other party that has been served. Delivery does not count as service; the requests are considered to be served at the first Rule 26(f) conference. Under Rule 34(b)(2)(A) the time to respond runs from service. This relaxation of the discovery moratorium is designed to facilitate focused discussion during the Rule 26(f) conference. Discussion at the conference may produce changes in the requests. The opportunity for advance scrutiny of requests delivered before the Rule 26(f) conference should not affect a decision whether to allow additional time to respond.

Rule 26(d)(3) is renumbered and amended to recognize that the parties may stipulate to case-specific sequences of discovery.

Rule 26(f)(3) is amended in parallel with Rule 16(b)(3) to add two items to the discovery plan — issues about preserving electronically stored information and court orders under Evidence Rule 502.

***

Rule 34. Producing Documents, Electronically Stored Information, and Tangible Things, or Entering onto Land, for Inspection and Other Purposes

(a) In General. A party may serve on any other party a request within the scope of Rule 26(b):

(1) to produce and permit the requesting party or its representative to inspect, copy, test, or sample the following items in the responding party's possession, custody, or control:

(A) any designated documents or electronically stored information—including writings, drawings, graphs, charts, photographs, sound recordings, images, and other data or data compilations—stored in any medium from which information can be obtained either directly or, if necessary, after translation by the responding party into a reasonably usable form; or

(B) any designated tangible things; or

(2) to permit entry onto designated land or other property possessed or controlled by the responding party, so that the requesting party may inspect, measure, survey, photograph, test, or sample the property or any designated object or operation on it.

(b) Procedure.

(1) *Contents of the Request.* The request:

(A) must describe with reasonable particularity each item or category of items to be inspected;

(B) must specify a reasonable time, place, and manner for the inspection and for performing the related acts; and

(C) may specify the form or forms in which electronically stored information is to be produced.

(2) *Responses and Objections.*

(A) *Time to Respond.* The party to whom the request is directed must respond in writing within 30 days after being served or — if the request was delivered under Rule 26(d)(2) — within 30 days after the parties' first Rule 26(f) conference. A shorter or longer time may be stipulated to under Rule 29 or be ordered by the court.

(B) *Responding to Each Item.* For each item or category, the response must either state that inspection and related activities will be permitted as requested or state with specificity the grounds for objecting to the request, including the reasons. The responding party may state that it will produce copies of documents or of electronically stored information instead of permitting inspection. The production must then be completed no later than the time for inspection specified in the request or another reasonable time specified in the response.

(C) *Objections.* An objection must state whether any responsive materials are being withheld on the basis of that objection. An objection to part of a request must specify the part and permit inspection of the rest.

(D) *Responding to a Request for Production of Electronically Stored Information.* The response may state an objection to a requested form for producing electronically stored information. If the responding party objects to a requested form—or if no form was specified in the request—the party must state the form or forms it intends to use.

(E) *Producing the Documents or Electronically Stored Information.* Unless otherwise stipulated or ordered by the court, these procedures apply to producing documents or electronically stored information:

(i) A party must produce documents as they are kept in the usual course of business or must organize and label them to correspond to the categories in the request;

(ii) If a request does not specify a form for producing electronically stored information, a party must produce it in a form or forms in which it is ordinarily maintained or in a reasonably usable form or forms; and

(iii) A party need not produce the same electronically stored information in more than one form.

(c) Nonparties. As provided in Rule 45, a nonparty may be compelled to produce documents and tangible things or to permit an inspection.

Notes

(As amended Dec. 27, 1946, eff. Mar. 19, 1948; Mar. 30, 1970, eff. July 1, 1970; Apr. 29, 1980, eff. Aug. 1, 1980; Mar. 2, 1987, eff. Aug. 1, 1987; Apr. 30, 1991, eff. Dec. 1, 1991; Apr. 22, 1993, eff. Dec. 1, 1993; Apr. 12, 2006, eff. Dec. 1, 2006; Apr. 30, 2007, eff. Dec. 1, 2007; Apr. 29, 2015, eff. Dec. 1, 2015.)

Committee Notes on Rules—2006 Amendment

Subdivision (a). As originally adopted, Rule 34 focused on discovery of “documents” and “things.” In 1970, Rule 34(a) was amended to include discovery of data compilations, anticipating that the use of computerized information would increase. Since then, the growth in electronically stored information and in the variety of systems for creating and storing such information has been dramatic. Lawyers and judges interpreted the term “documents” to include electronically stored information because it was obviously improper to allow a party to evade discovery obligations on the basis that the label had not kept pace with changes in information technology. But it has become increasingly difficult to say that all forms of electronically stored information, many dynamic in nature, fit within the traditional concept of a "document." Electronically stored information may exist in dynamic databases and other forms far different from fixed expression on paper. Rule 34(a) is amended to confirm that discovery of electronically stored information stands on equal footing with discovery of paper documents. The change clarifies that Rule 34 applies to information that is fixed in a tangible form and to information that is stored in a medium from which it can be retrieved and examined. At the same time, a Rule 34 request for production of "documents" should be understood to encompass, and the response should include, electronically stored information unless discovery in the action has clearly distinguished between electronically stored information and "documents."

Discoverable information often exists in both paper and electronic form, and the same or similar information might exist in both. The items listed in Rule 34(a) show different ways in which information may be recorded or stored. Images, for example, might be hard-copy documents or electronically stored information. The wide variety of computer systems currently in use, and the rapidity of technological change, counsel against a limiting or precise definition of electronically stored information. Rule 34(a)(1) is expansive and includes any type of information that is stored electronically. A common example often sought in discovery is electronic communications, such as e-mail. The rule covers—either as documents or as electronically stored information—information "stored in any medium," to encompass future developments in computer technology. Rule 34(a)(1) is intended to be broad enough to cover all current types of computer-based information, and flexible enough to encompass future changes and developments.

References elsewhere in the rules to "electronically stored information" should be understood to invoke this expansive approach. A companion change is made to Rule 33(d), making it explicit that parties choosing to respond to an interrogatory by permitting access to responsive records may do so by providing access to electronically stored information. More generally, the term used in Rule 34(a)(1) appears in a number of other amendments, such as those to Rules 26(a)(1), 26(b)(2), 26(b)(5)(B), 26(f), 34(b), 37(f), and 45. In each of these rules, electronically stored information has the same broad meaning it has under Rule 34(a)(1). References to "documents" appear in discovery rules that are not amended, including Rules 30(f), 36(a), and 37(c)(2). These references should be interpreted to include electronically stored information as circumstances warrant.

The term "electronically stored information" is broad, but whether material that falls within this term should be produced, and in what form, are separate questions that must be addressed under Rules 26(b), 26(c), and 34(b).

The Rule 34(a) requirement that, if necessary, a party producing electronically stored information translate it into reasonably usable form does not address the issue of translating from one human language to another. *See In re Puerto Rico Elect. Power Auth.*, 687 F.2d 501, 504–510 (1st Cir. 1989).

Rule 34(a)(1) is also amended to make clear that parties may request an opportunity to test or sample materials sought under the rule in addition to inspecting and copying them. That opportunity may be important for both electronically stored information and hard-copy materials. The current rule is not clear that such testing or sampling is authorized; the amendment expressly permits it. As with any other form of discovery, issues of burden and intrusiveness raised by requests to test or sample can be addressed under Rules 26(b)(2) and 26(c). Inspection or testing of certain types of electronically stored information or of a responding party's electronic information system may raise issues of confidentiality or privacy. The addition of testing and sampling to Rule 34(a) with regard to documents and electronically stored information is not meant to create a routine right of direct access to a party's electronic information system, although such access might be justified in some circumstances. Courts should guard against undue intrusiveness resulting from inspecting or testing such systems.

Rule 34(a)(1) is further amended to make clear that tangible things must—like documents and land sought to be examined—be designated in the request.

Subdivision (b). Rule 34(b) provides that a party must produce documents as they are kept in the usual course of business or must organize and label them to correspond with the categories in the discovery request. The production of electronically stored information should be subject to comparable requirements to protect against deliberate or inadvertent production in ways that raise unnecessary obstacles for the requesting party. Rule 34(b) is amended to ensure similar protection for electronically stored information.

The amendment to Rule 34(b) permits the requesting party to designate the form or forms in which it wants electronically stored information produced. The form of production is more important to the exchange of electronically stored information than of hard-copy materials, although a party might specify hard copy as the requested form. Specification of the desired form or forms may facilitate the orderly, efficient, and cost-effective discovery of electronically stored information. The rule recognizes that different forms of production may be appropriate for different types of electronically stored information. Using current technology, for example, a party might be called upon to produce word processing documents, e-mail messages, electronic spreadsheets, different image or sound files, and material from databases. Requiring that such diverse types of electronically stored information all be produced in the same form could prove impossible, and even if possible could increase the cost and burdens of producing and using the information. The rule therefore provides that the requesting party may ask for different forms of production for different types of electronically stored information.

The rule does not require that the requesting party choose a form or forms of production. The requesting party may not have a preference. In some cases, the requesting party may not know what form the producing party uses to maintain its electronically stored information, although Rule 26(f)(3) is amended to call for discussion of the form of production in the parties' prediscovery conference.

The responding party also is involved in determining the form of production. In the written response to the production request that Rule 34 requires, the responding party must state the form it intends to use for producing electronically stored information if the requesting party does not specify a form or if the responding party objects to a form that the requesting party specifies. Stating the intended form before the production occurs may permit the parties to identify and seek to resolve disputes before the expense and work of the production occurs. A party that responds to a discovery request by simply producing electronically stored information in a form of its choice, without identifying that form in advance of the production in the response required by Rule 34(b), runs a risk that the requesting party can show that the produced form is not reasonably usable and that it is entitled to production of some or all of the information in an additional form. Additional time might be required to permit a responding party to assess the appropriate form or forms of production.

If the requesting party is not satisfied with the form stated by the responding party, or if the responding party has objected to the form specified by the requesting party, the parties must meet and confer under Rule 37(a)(2)(B) in an effort to resolve the matter before the requesting party can file a motion to compel. If they cannot agree and the court resolves the dispute, the court is not limited to the forms initially chosen by the requesting party, stated by the responding party, or specified in this rule for situations in which there is no court order or party agreement.

If the form of production is not specified by party agreement or court order, the responding party must produce electronically stored information either in a form or forms in which it is ordinarily maintained or in a form or forms that are reasonably usable. Rule 34(a) requires that, if necessary, a responding party "translate" information it produces into a "reasonably usable" form. Under some circumstances, the responding party may need to provide some reasonable amount of technical support, information on application software, or other reasonable assistance to enable the requesting party to use the information. The rule does not require a party to produce electronically stored information in the form it [sic] which it is ordinarily maintained, as long as it is produced in a reasonably usable form. But the option to produce in a reasonably usable form does not mean that a responding party is free to convert electronically stored information from the form in which it is ordinarily maintained to a different form that makes it more difficult or burdensome for the requesting party to use the information efficiently in the litigation. If the responding party ordinarily maintains the information it is producing in a way that makes it searchable by electronic means, the information should not be produced in a form that removes or significantly degrades this feature.

Some electronically stored information may be ordinarily maintained in a form that is not reasonably usable by any party. One example is "legacy" data that can be used only by superseded systems. The questions whether a producing party should be required to convert such information to a more usable form, or should be required to produce it at all, should be addressed under Rule 26(b)(2)(B).

Whether or not the requesting party specified the form of production, Rule 34(b) provides that the same electronically stored information ordinarily be produced in only one form.

Changes Made after Publication and Comment. The proposed amendment recommended for approval has been modified from the published version. The sequence of "documents or electronically stored information" is changed to emphasize that the parenthetical exemplifications apply equally to illustrate "documents" and "electronically stored information." The reference to "detection devices" is deleted as redundant with "translated" and as archaic.

The references to the form of production are changed in the rule and Committee Note to refer also to "forms." Different forms may be appropriate or necessary for different sources of information.

The published proposal allowed the requesting party to specify a form for production and recognized that the responding party could object to the requested form. This procedure is now amplified by directing that the responding party state the form or forms it intends to use for production if the request does not specify a form or if the responding party objects to the requested form.

The default forms of production to be used when the parties do not agree on a form and there is no court order are changed in part. As in the published proposal, one default form is "a form or forms in which [electronically stored information] is ordinarily maintained." The alternative default form, however, is changed from "an electronically searchable form" to "a form or forms that are reasonably usable." "[A]n electronically searchable form" proved to have several defects. Some electronically stored information cannot be searched electronically. In addition, there often are many different levels of electronic searchability—the published default would authorize production in a minimally searchable form even though more easily searched forms might be available at equal or less cost to the responding party.

The provision that absent court order a party need not produce the same electronically stored information in more than one form was moved to become a separate item for the sake of emphasis.

The Committee Note was changed to reflect these changes in rule text, and also to clarify many aspects of the published Note. In addition, the Note was expanded to add a caveat to the published amendment that establishes the rule that documents—and now electronically stored information—may be tested and sampled as well as inspected and copied. Fears were expressed that testing and sampling might imply routine direct access to a party's information system. The Note states that direct access is not a routine right, "although such access might be justified in some circumstances."

The changes in the rule text since publication are set out below. [Omitted]

Committee Notes on Rules—2015 Amendment

Several amendments are made in Rule 34, aimed at reducing the potential to impose unreasonable burdens by objections to requests to produce.

Rule 34(b)(2)(A) is amended to fit with new Rule 26(d)(2). The time to respond to a Rule 34 request delivered before the parties' Rule 26(f) conference is 30 days after the first Rule 26(f) conference.

Rule 34(b)(2)(B) is amended to require that objections to Rule 34 requests be stated with specificity. This provision adopts the language of Rule 33(b)(4), eliminating any doubt that less specific objections might be suitable under Rule 34. The specificity of the objection ties to the new provision in Rule 34(b)(2)(C) directing that an objection must state whether any responsive materials are being withheld on the basis of that objection. An objection may state that a request is overbroad, but if the objection recognizes that some part of the request is appropriate the objection should state the scope that is not overbroad. Examples would be a statement that the responding party will limit the search to documents or electronically stored information created within a given period of time prior to the events in suit, or to specified sources. When there is such an objection, the statement of what has been withheld can properly identify as matters "withheld" anything beyond the scope of the search specified in the objection.

Rule 34(b)(2)(B) is further amended to reflect the common practice of producing copies of documents or electronically stored information rather than simply permitting inspection. The response to the request must state that copies will be produced. The production must be completed either by the time for inspection specified in the request or by another reasonable time specifically identified in the response. When it is necessary to make the production in stages the response should specify the beginning and end dates of the production.

Rule 34(b)(2)(C) is amended to provide that an objection to a Rule 34 request must state whether anything is being withheld on the basis of the objection. This amendment should end the confusion that frequently arises when a producing party states several objections and still produces information, leaving the requesting party uncertain whether any relevant and responsive information has been withheld on the basis of the objections. The producing party does not need to provide a detailed description or log of all documents withheld, but does need to alert other parties to the fact that documents have been withheld and thereby facilitate an informed discussion of the objection. An objection that states the limits that have controlled the search for responsive and relevant materials qualifies as a statement that the materials have been "withheld."

***

Rule 45. Subpoena

(a) In General.

(1) *Form and Contents.*

(A) *Requirements—In General.* Every subpoena must:

(i) state the court from which it issued;

(ii) state the title of the action and its civil-action number;

(iii) command each person to whom it is directed to do the following at a specified time and place: attend and testify; produce designated documents, electronically stored information, or tangible things in that person's possession, custody, or control; or permit the inspection of premises; and

(iv) set out the text of Rule 45(d) and (e).

(B) *Command to Attend a Deposition—Notice of the Recording Method.* A subpoena commanding attendance at a deposition must state the method for recording the testimony.

(C) *Combining or Separating a Command to Produce or to Permit Inspection; Specifying the Form for Electronically Stored Information.* A command to produce documents, electronically stored information, or tangible things or to permit the inspection of premises may be included in a subpoena commanding attendance at a deposition, hearing, or trial, or may be set out in a separate subpoena. A subpoena may specify the form or forms in which electronically stored information is to be produced.

(D) *Command to Produce; Included Obligations.* A command in a subpoena to produce documents, electronically stored information, or tangible things requires the responding person to permit inspection, copying, testing, or sampling of the materials.

(2) *Issuing Court.* A subpoena must issue from the court where the action is pending.

(3) *Issued by Whom.* The clerk must issue a subpoena, signed but otherwise in blank, to a party who requests it. That party must complete it before service. An attorney also may issue and sign a subpoena if the attorney is authorized to practice in the issuing court.

(4) *Notice to Other Parties Before Service.* If the subpoena commands the production of documents, electronically stored information, or tangible things or the inspection of premises before trial, then before it is served on the person to whom it is directed, a notice and a copy of the subpoena must be served on each party.

(b) Service.

(1) *By Whom and How; Tendering Fees.* Any person who is at least 18 years old and not a party may serve a subpoena. Serving a subpoena requires delivering a copy to the named person and, if the subpoena requires that person's attendance, tendering the fees for 1 day's attendance and the mileage allowed by law. Fees and mileage need not be tendered when the subpoena issues on behalf of the United States or any of its officers or agencies.

(2) *Service in the United States.* A subpoena may be served at any place within the United States.

(3) *Service in a Foreign Country.* 28 U.S.C. §1783 governs issuing and serving a subpoena directed to a United States national or resident who is in a foreign country.

(4) *Proof of Service.* Proving service, when necessary, requires filing with the issuing court a statement showing the date and manner of service and the names of the persons served. The statement must be certified by the server.

(c) Place of Compliance.

(1) *For a Trial, Hearing, or Deposition.* A subpoena may command a person to attend a trial, hearing, or deposition only as follows:

(A) within 100 miles of where the person resides, is employed, or regularly transacts business in person; or

(B) within the state where the person resides, is employed, or regularly transacts business in person, if the person

(i) is a party or a party's officer; or

(ii) is commanded to attend a trial and would not incur substantial expense.

(2) *For Other Discovery.* A subpoena may command:

(A) production of documents, electronically stored information, or tangible things at a place within 100 miles of where the person resides, is employed, or regularly transacts business in person; and

(B) inspection of premises at the premises to be inspected.

(d) Protecting a Person Subject to a Subpoena; Enforcement.

(1) *Avoiding Undue Burden or Expense; Sanctions.* A party or attorney responsible for issuing and serving a subpoena must take reasonable steps to avoid imposing undue burden or expense on a person subject to the subpoena. The court for the district where compliance is required must enforce this duty and impose an appropriate sanction—which may include lost earnings and reasonable attorney's fees—on a party or attorney who fails to comply.

(2) *Command to Produce Materials or Permit Inspection.*

(A) *Appearance Not Required.* A person commanded to produce documents, electronically stored information, or tangible things, or to permit the inspection of premises, need not appear in person at the place of production or inspection unless also commanded to appear for a deposition, hearing, or trial.

(B) *Objections.* A person commanded to produce documents or tangible things or to permit inspection may serve on the party or attorney designated in the subpoena a written objection to inspecting, copying, testing or sampling any or all of the materials or to inspecting the premises—or to producing electronically stored information in the form or forms requested. The objection must be served before the earlier of the time specified for compliance or 14 days after the subpoena is served. If an objection is made, the following rules apply:

(i) At any time, on notice to the commanded person, the serving party may move the court for the district where compliance is required for an order compelling production or inspection.

(ii) These acts may be required only as directed in the order, and the order must protect a person who is neither a party nor a party's officer from significant expense resulting from compliance.

(3) *Quashing or Modifying a Subpoena.*

(A) *When Required.* On timely motion, the court for the district where compliance is required must quash or modify a subpoena that:

(i) fails to allow a reasonable time to comply;

(ii) requires a person to comply beyond the geographical limits specified in Rule 45(c);

(iii) requires disclosure of privileged or other protected matter, if no exception or waiver applies; or

(iv) subjects a person to undue burden.

(B) *When Permitted.* To protect a person subject to or affected by a subpoena, the court for the district where compliance is required may, on motion, quash or modify the subpoena if it requires:

(i) disclosing a trade secret or other confidential research, development, or commercial information; or

(ii) disclosing an unretained expert's opinion or information that does not describe specific occurrences in dispute and results from the expert's study that was not requested by a party.

(C) *Specifying Conditions as an Alternative.* In the circumstances described in Rule 45(d)(3)(B), the court may, instead of quashing or modifying a subpoena, order appearance or production under specified conditions if the serving party:

(i) shows a substantial need for the testimony or material that cannot be otherwise met without undue hardship; and

(ii) ensures that the subpoenaed person will be reasonably compensated.

(e) Duties in Responding to a Subpoena.

(1) *Producing Documents or Electronically Stored Information.* These procedures apply to producing documents or electronically stored information:

(A) *Documents.* A person responding to a subpoena to produce documents must produce them as they are kept in the ordinary course of business or must organize and label them to correspond to the categories in the demand.

(B) *Form for Producing Electronically Stored Information Not Specified.* If a subpoena does not specify a form for producing electronically stored information, the person responding must produce it in a form or forms in which it is ordinarily maintained or in a reasonably usable form or forms.

(C) *Electronically Stored Information Produced in Only One Form.* The person responding need not produce the same electronically stored information in more than one form.

(D) *Inaccessible Electronically Stored Information.* The person responding need not provide discovery of electronically stored information from sources that the person identifies as not reasonably accessible because of undue burden or cost. On motion to compel discovery or for a protective order, the person responding must show that the information is not reasonably accessible because of undue burden or cost. If that showing is made, the court may nonetheless order discovery from such sources if the requesting party shows good cause, considering the limitations of Rule 26(b)(2)(C). The court may specify conditions for the discovery.

(2) *Claiming Privilege or Protection.*

(A) *Information Withheld.* A person withholding subpoenaed information under a claim that it is privileged or subject to protection as trial-preparation material must:

(i) expressly make the claim; and

(ii) describe the nature of the withheld documents, communications, or tangible things in a manner that, without revealing information itself privileged or protected, will enable the parties to assess the claim.

(B) *Information Produced.* If information produced in response to a subpoena is subject to a claim of privilege or of protection as trial-preparation material, the person making the claim may notify any party that received the information of the claim and the basis for it. After being notified, a party must promptly return, sequester, or destroy the specified information and any copies it has; must not use or disclose the information until the claim is resolved; must take reasonable steps to retrieve the information if the party disclosed it before being notified; and may promptly present the information under seal to the court for the district where compliance is required for a determination of the claim. The person who produced the information must preserve the information until the claim is resolved.

(f) Transferring a Subpoena-Related Motion. When the court where compliance is required did not issue the subpoena, it may transfer a motion under this rule to the issuing court if the person subject to the subpoena consents or if the court finds exceptional circumstances. Then, if the attorney for a person subject to a subpoena is authorized to practice in the court where the motion was made, the attorney may file papers and appear on the motion as an officer of the issuing court. To enforce its order, the issuing court may transfer the order to the court where the motion was made.

(g) Contempt. The court for the district where compliance is required — and also, after a motion is transferred, the issuing court — may hold in contempt a person who, having been served, fails without adequate excuse to obey the subpoena or an order related to it.

Notes

(As amended Dec. 27, 1946, eff. Mar. 19, 1948; Dec. 29, 1948, eff. Oct. 20, 1949; Mar. 30, 1970, eff. July 1, 1970; Apr. 29, 1980, eff. Aug. 1, 1980; Apr. 29, 1985, eff. Aug. 1, 1985; Mar. 2, 1987, eff. Aug.

1, 1987; Apr. 30, 1991, eff. Dec. 1, 1991; Apr. 25, 2005, eff. Dec. 1, 2005; Apr. 12, 2006, eff. Dec. 1, 2006; Apr. 30, 2007, eff. Dec. 1, 2007; Apr. 16, 2013, eff. Dec. 1, 2013.)

Committee Notes on Rules—2006 Amendment

Rule 45 is amended to conform the provisions for subpoenas to changes in other discovery rules, largely related to discovery of electronically stored information. Rule 34 is amended to provide in greater detail for the production of electronically stored information. Rule 45(a)(1)(C) is amended to recognize that electronically stored information, as defined in Rule 34(a), can also be sought by subpoena. Like Rule 34(b), Rule 45(a)(1) is amended to provide that the subpoena can designate a form or forms for production of electronic data. Rule 45(c)(2) is amended, like Rule 34(b), to authorize the person served with a subpoena to object to the requested form or forms. In addition, as under Rule 34(b), Rule 45(d)(1)(B) is amended to provide that if the subpoena does not specify the form or forms for electronically stored information, the person served with the subpoena must produce electronically stored information in a form or forms in which it is usually maintained or in a form or forms that are reasonably usable. Rule 45(d)(1)(C) is added to provide that the person producing electronically stored information should not have to produce the same information in more than one form unless so ordered by the court for good cause.

As with discovery of electronically stored information from parties, complying with a subpoena for such information may impose burdens on the responding person. Rule 45(c) provides protection against undue impositions on nonparties. For example, Rule 45(c)(1) directs that a party serving a subpoena "shall take reasonable steps to avoid imposing undue burden or expense on a person subject to the subpoena," and Rule 45(c)(2)(B) permits the person served with the subpoena to object to it and directs that an order requiring compliance "shall protect a person who is neither a party nor a party's officer from significant expense resulting from" compliance. Rule 45(d)(1)(D) is added to provide that the responding person need not provide discovery of electronically stored information from sources the party identifies as not reasonably accessible, unless the court orders such discovery for good cause, considering the limitations of Rule 26(b)(2)(C), on terms that protect a nonparty against significant expense. A parallel provision is added to Rule 26(b)(2).

Rule 45(a)(1)(B) is also amended, as is Rule 34(a), to provide that a subpoena is available to permit testing and sampling as well as inspection and copying. As in Rule 34, this change recognizes that on occasion the opportunity to perform testing or sampling may be important, both for documents and for electronically stored information. Because testing or sampling may present particular issues of burden or intrusion for the person served with the subpoena, however, the protective provisions of Rule 45(c) should be enforced with vigilance when such demands are made. Inspection or testing of certain types of electronically stored information or of a person's electronic information system may raise issues of confidentiality or privacy. The addition of sampling and testing to Rule 45(a) with regard to documents and electronically stored information is not meant to create a routine right of direct access to a person's electronic information system, although such access might be justified in some circumstances. Courts should guard against undue intrusiveness resulting from inspecting or testing such systems.

Rule 45(d)(2) is amended, as is Rule 26(b)(5), to add a procedure for assertion of privilege or of protection as trial-preparation materials after production. The receiving party may submit the information to the court for resolution of the privilege claim, as under Rule 26(b)(5)(B).

Federal Rule of Evidence 502

Attorney-Client Privilege and Work Product; Limitations on Waiver

The following provisions apply, in the circumstances set out, to disclosure of a communication or information covered by the attorney-client privilege or work-product protection.

(a) Disclosure Made in a Federal Proceeding or to a Federal Office or Agency; Scope of a Waiver. When the disclosure is made in a federal proceeding or to a federal office or agency and waives the attorney-client privilege or work-product protection, the waiver extends to an undisclosed communication or information in a federal or state proceeding only if:

(1) the waiver is intentional;
(2) the disclosed and undisclosed communications or information concern the same subject matter; and
(3) they ought in fairness to be considered together.

(b) Inadvertent Disclosure. When made in a federal proceeding or to a federal office or agency, the disclosure does not operate as a waiver in a federal or state proceeding if:

(1) the disclosure is inadvertent; (2) the holder of the privilege or protection took reasonable steps to prevent disclosure; and (3) the holder promptly took reasonable steps to rectify the error, including (if applicable) following Federal Rule of Civil Procedure 26 (b)(5)(B).

(c) Disclosure Made in a State Proceeding. When the disclosure is made in a state proceeding and is not the subject of a state-court order concerning waiver, the disclosure does not operate as a waiver in a federal proceeding if the disclosure:

(1) would not be a waiver under this rule if it had been made in a federal proceeding; or (2) is not a waiver under the law of the state where the disclosure occurred.

(d) Controlling Effect of a Court Order. A federal court may order that the privilege or protection is not waived by disclosure connected with the litigation pending before the court — in which event the disclosure is also not a waiver in any other federal or state proceeding.

(e) Controlling Effect of a Party Agreement. An agreement on the effect of disclosure in a federal proceeding is binding only on the parties to the agreement, unless it is incorporated into a court order.

(f) Controlling Effect of this Rule. Notwithstanding Rules 101 and 1101, this rule applies to state proceedings and to federal court-annexed and federal court-mandated arbitration proceedings, in the circumstances set out in the rule. And notwithstanding Rule 501, this rule applies even if state law provides the rule of decision.

(g) Definitions. In this rule:

(1) “attorney-client privilege” means the protection that applicable law provides for confidential attorney-client communications; and

(2) “work-product protection” means the protection that applicable law provides for tangible material (or its intangible equivalent) prepared in anticipation of litigation or for trial.

FRE Rule 902. Evidence That Is Self-Authenticating ***[as amended effective 2017, footnote added]***

The following items of evidence are self-authenticating; they require no extrinsic evidence of authenticity in order to be admitted:

…

> ***(13) Certified Records Generated by an Electronic Process or System.*** A record generated by an electronic process or system that produces an accurate result, as shown by a certification of a qualified person that complies with the certification requirements of Rule 902(11) or (12).[88] The proponent must also meet the notice requirements of Rule 902(11).
>
> ***(14) Certified Data Copied from an Electronic Device, Storage Medium, or File.*** Data copied from an electronic device, storage medium, or file, if authenticated by a process of digital identification, as shown by a certification of a qualified person that complies with the certification requirements of Rule (902(11) or (12). The proponent also must meet the notice requirements of Rule 902 (11).

Committee Notes on Rules—2017 Amendment

Paragraph (14). The amendment sets forth a procedure by which parties can authenticate data copied from an electronic device, storage medium, or an electronic file, other than through the testimony of a foundation witness. As with the provisions on business records in Rules 902(11) and (12), the Committee has found that the expense and inconvenience of producing an authenticating witness for this evidence is often unnecessary. It is often the case that a party goes to the expense of producing an authentication witness, and then the adversary either stipulates authenticity before the witness is called or fails to challenge the authentication testimony once it is presented. The amendment provides a procedure in which the parties can determine in advance of trial whether a real challenge to authenticity will be made, and can then plan accordingly.

Today, data copied from electronic devices, storage media, and electronic files are ordinarily authenticated by "hash value". A hash value is a number that is often represented as a sequence

[88] **(11)** ***Certified Domestic Records of a Regularly Conducted Activity*****.** The original or a copy of a domestic record that meets the requirements of Rule 803(6)(A)-(C), as shown by a certification of the custodian or another qualified person that complies with a federal statute or a rule prescribed by the Supreme Court. Before the trial or hearing, the proponent must give an adverse party reasonable written notice of the intent to offer the record — and must make the record and certification available for inspection — so that the party has a fair opportunity to challenge them.
(12) ***Certified Foreign Records of a Regularly Conducted Activity*****.** In a civil case, the original or a copy of a foreign record that meets the requirements of Rule 902(11), modified as follows: the certification, rather than complying with a federal statute or Supreme Court rule, must be signed in a manner that, if falsely made, would subject the maker to a criminal penalty in the country where the certification is signed. The proponent must also meet the notice requirements of Rule 902(11).

of characters and is produced by an algorithm based upon the digital contents of a drive, medium, or file. If the hash values for the original and copy are different, then the copy is not identical to the original. If the hash values for the original and copy are the same, it is highly improbable that the original and copy are not identical. Thus, identical hash values for the original and copy reliably attest to the fact that they are exact duplicates. This amendment allows self-authentication by a certification of a qualified person that she checked the hash value of the proffered item and that it was identical to the original. The rule is flexible enough to allow certifications through processes other than comparison of hash value, including by other reliable means of identification provided by future technology.

Nothing in the amendment is intended to limit a party from establishing authenticity of electronic evidence on any ground provided in these Rules, including through judicial notice where appropriate.

A proponent establishing authenticity under this Rule must present a certification containing information that would be sufficient to establish authenticity were that information provided by a witness at trial. If the certification provides information that would be insufficient to authenticate the record of the certifying person testified, then authenticity is not established under this Rule.

The reference to the "certification requirements of Rule 902(11) or (12)" is only to the procedural requirements for a valid certification. There is no intent to require, or permit, a certification under this Rule to prove the requirements of Rule 803(6). Rule 902(14) is solely limited to authentication, and any attempt to satisfy a hearsay exception must be made independently.

A certification under this Rule can only establish that the proffered item is authentic. The opponent remains free to object to admissibility of the proffered item on other grounds—including hearsay, relevance, or in criminal cases the right to confrontation. For example, in a criminal case in which data copied from a hard drive is proffered, the defendant can still challenge hearsay found in the hard drive, and can still challenge whether the information on the hard drive was placed there by the defendant.

A challenge to the authenticity of electronic evidence may require technical information about the system or process at issue, including possibly retaining a forensic technical expert; such factors will affect whether the opponent has a fair opportunity to challenge the evidence given the notice provided.

The reference to Rule 902(12) is intended to cover certifications that are made in a foreign country.

Exemplar FRE 902(14) Certification (Craig Ball)

The Rules don't supply the language required to appear in an FRE 902 (14) certification; however, a declaration like the following would seem to suffice to prove up a duplicate, amended to fit the particular data and methods used and the declarant. *Be sure to make clear that the baseline hash value used for comparison and authentication was calculated contemporaneously with the acquisition of the copy or image being authenticated*. A hash value only describes data at the time of hashing, so may only authenticate the contents of a dataset as they existed at the time the contents were hashed. *Note: this only serves to prove identicality, not cure hearsay objections.*

My name is_______________. I am a domiciliary of_________, over the age of eighteen years and competent and qualified to make this declaration.

The supplied data was transferred to me on a [*DESCRIBE TRANSFER MEDIA*], serial number ____________________, and consisted of:

[*IDENTIFY AND DESCRIBE DATASET*]

Based on information supplied to me, [*PERSON MAKING DUPLICATE*] copied the source data using [*IDENTIFY TOOL*]. I am familiar with this tool and have used it routinely and successfully over many years to create forensically sound images of digital media. I know it to be a capable tool for forensically-sound duplication of digital media and one widely accepted as standard and reliable by digital forensic experts. Based on the reporting supplied and routinely generated by the tool, the acquisition completed successfully, imaging ____sectors (___GB) without identification of bad sectors.

The reported [*MD5/SHA1/SHA256*] hash value of the data supplied when acquired was [*HASH VALUE*]. Using [*IDENTIFY TOOL*], I verified that the data supplied to me matched the hash value obtained when the data was acquired. I am trained and experienced in the generation and use of cryptographic hash values for testing and authenticating the integrity of digital evidence.

On [*DATE*], I re-verified the integrity of the drive image used in my work in this case. The [*MD5/SHA1/SHA256*] hash value is unchanged from the hash value calculated at acquisition. Accordingly, I can attest that the data received by me is authentic in that it was and remains an exact duplicate of the contents of the [SOURCE MEDIA], serial number: __________, attributed to [IDENTIFY SOURCE DEVICE AND CUSTODIAN] and originally acquired on [*ACQUISITION DATE*]. All my processing, extraction, production and interpretation of the contents of the drive image were made using the hash-authenticated data. <SIGNED>

Texas Rules of Civil Procedure

Part II – Rules of Practice in District and County Court

TRCP Rule 196.4 Electronic or Magnetic Data (enacted 1999)

To obtain discovery of data or information that exists in electronic or magnetic form, the requesting party must specifically request production of electronic or magnetic data and specify the form in which the requesting party wants it produced. The responding party must produce the electronic or magnetic data that is responsive to the request and is reasonably available to the responding party in its ordinary course of business. If the responding party cannot – through reasonable efforts – retrieve the data or information requested or produce it in the form requested, the responding party must state an objection complying with these rules. If the court orders the responding party to comply with the request, the court must also order that the requesting party pay the reasonable expenses of any extraordinary steps required to retrieve and produce the information.

Questions to Consider:

1. What is entailed in specifically requesting electronic or magnetic data?
2. How much specificity is needed to specify the form of production sought?
3. Are the native forms of data as created, used and stored by the responding party's software and systems synonymous with the forms "reasonably available to the responding party in its ordinary course of business?"
4. Is a responding party obliged by this rule to undertake reasonable efforts to retrieve requested data? Is it ever appropriate to object without making any effort?
5. What sorts of circumstances would give rise to "extraordinary" steps required to retrieve and produce information so as to trigger mandatory cost shifting? If litigation is "extraordinary," are any steps to respond to e-discovery also extraordinary?
6. What is the role of "proportionality" in limiting the scope of discovery and appropriate forms of production?