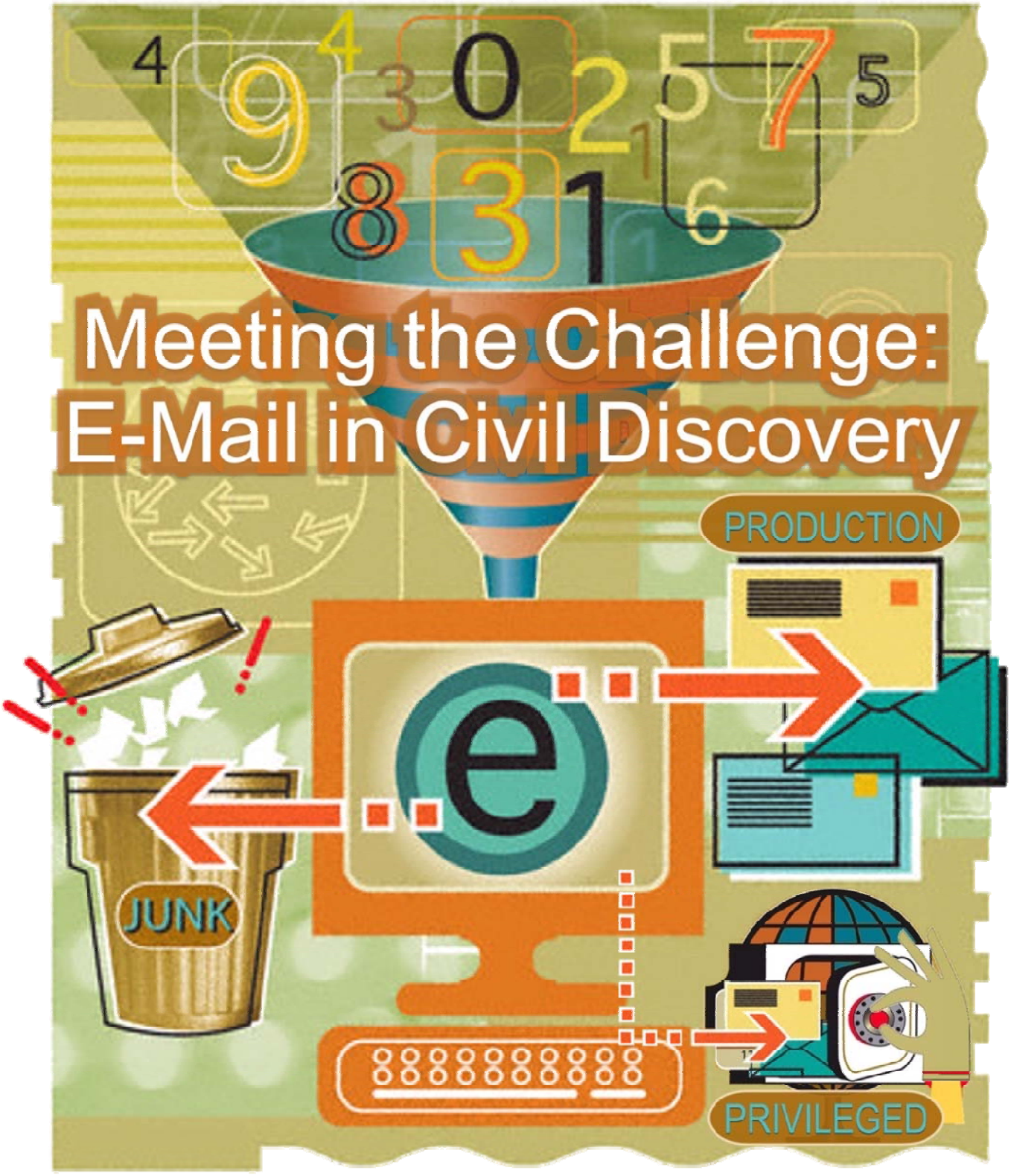

Meeting the Challenge:
E-Mail in Civil Discovery
PRODUCTION
JUNK
PRIVILEGED

Craig Ball

Technology Primer

Meeting the Challenge: E-Mail in Civil Discovery

Craig Ball

Table of Contents

Introduction

This paper looks at e-mail from the standpoint of what lawyers should know about the nuts-and-bolts of these all-important communications systems. It's technical; sometimes, *very* technical.

When you finish the paper, you'll know *a lot* more about e-mail, and along the way, you may realize that discoverable e-mail can be found in far more places than your client probably checked before the last time you said, "Yes, your Honor, we've given them the e-mail."

So, if you know what's good for you, you should probably stop reading right now.
….

Still here? Okay, you asked for it.

Get the e-mail! It's the war cry in discovery today. More than simply a feeding frenzy, it's an inevitable recognition of e-mail's importance and ubiquity. We go after e-mail because it accounts for the majority of business communications and because e-mail users tend to let their guard down and reveal plainspoken truths they'd never dare put in a memo. Or do they? A 2008 study[1] demonstrated that employees are significantly more likely to lie in e-mail messages than in traditional pen-and-paper communications. Whether replete with ugly truths or ugly lies, e-mail is telling and compelling evidence.

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 challenge 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.

This paper seeks to equip the corporate counsel or trial lawyer with some of what's needed to meet the challenge of e-mail discovery in civil litigation. It's intended to be technical because technical knowledge is what's most needed and most lacking in continuing legal education today. Even if you went to law school because you had no affinity for matters technical, it's time to dig in and learn enough to stay in the fray.

[1] http://www3.lehigh.edu/News/V2news_story.asp?iNewsID=2892 (visited 11/1/08)

Not Enough Eyeballs

Futurist Arthur C. Clarke said, "Any sufficiently advanced technology is indistinguishable from magic." E-mail, like electricity or refrigeration, is one of those magical technologies we use every day without knowing quite how it works. But, "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." You've got to make an effort to understand their systems and procedures and ask the right questions. Plus, you have to know when you aren't getting the right answers. Perhaps that's asking a lot, but well over 95% of all business documents are born digitally and only a tiny fraction are ever printed.[2] Hundreds of billions of e-mails traverse the Internet *daily,* far more than telephone and postal traffic combined,[3] and the average business person sends and receives between 50 and 150 e-mails *every business day.* E-mail contributes *500 times greater volume* to the Internet than web page content.

Think that's a lot? Then best not think about the fact that the volume is expected to nearly double by 2012,[4] and none of these numbers take into account the explosive growth in instant messaging, unified messaging or the next insanely great communication or collaboration technology that—starting next year and every year—we can hardly live without. The volume keeps increasing, and there's no end in sight. It's simply too easy, too quick and too cheap to expect anything else.

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.

Another difficulty is that this startling growth isn't happening in just one locale. E-mail lodges on servers, cell phones, laptops, home systems, thumb drives and in "the cloud," a term ethereally denoting all the places we store information online, little knowing or caring about its physical location. Within the systems, applications and devices we use to store and access e-mail, most users and even many IT professionals don't know where messages lodge or how long they hang around.

In discovery, we overlook so much that we're obliged to consider, and with respect to what we do collect, it's increasingly infeasible to put enough pairs of trained eyes in front of enough computers to review every potentially responsive electronic document.

---

[2] 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/

[3] http://www.radicati.com/?p=638 (visited 11/1/08)

[4] Id.

Instead, we must employ shortcuts that serve as proxies for lawyer judgment. Here, too, our success hinges upon our understanding of the technologies we use to extend and defend our reach.

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—and even if you hold onto backup tapes for a few significant points in time—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, Lotus Domino, Novell GroupWise 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 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 or Yahoo! Mail) for discoverable business communications?
11. Do your clients' employees use Instant Messaging on company computers or over company-owned networks?
12. How do your clients' voice messaging systems store messages, and how long are they retained?

If you are troubled that you can't answer some of these questions, you should be; but know you're not alone. Many other lawyers can't either. And don't delude yourself that these are exclusively someone else's issues, *e.g.,* your litigation support services vendor or IT expert. These are the inquiries that will soon be coming at *you* in court and when conferring with the other side. You do confer on ESI, right?

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 it is clear that 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 fifty e-mails per day, a company employing 100,000 people could find itself storing well over *1.5 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. The single file containing my Outlook e-mail is over four gigabytes in size and contains tens of thousands of 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 particular topic, I'd either need to skim/read countless 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 particular matter in litigation? Surely, I mused, someone must have found a better solution than repeating, over and over again, 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?

For this article, 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 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 de-duplication," rules-based journaling and other customizable features.

These tools, especially enterprise archival, 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.

E-Mail Systems and Files

The 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 distant third place, largely among government users.

Per a 2008 study by Ferris Research,[5] Microsoft Exchange accounts for 65% of market share among all organizations, with significantly larger shares among businesses with fewer than 49 employees and those in the health care and telecommunications sectors. Lotus Notes was found to have just 10% of overall market share, but a much higher percentage base among manufacturers with at least 5,000 employees. GroupWise's share was termed "negligible," except in niches—notably organizations in the financial services and government sectors with 100 to 999 employees—where its share reached as high as 10-15%. Blackberry servers transmit a large percentage of e-mail as well, but these messages typically find their way to or through an Exchange or Lotus mail server.

Of course, when one looks at personal and small office/home office business e-mail, it's rare to encounter server-based Exchange or Domino systems. Here, the market belongs to Internet service providers (*e.g.,* AOL, the major cable and telephone

[5] http://www.ferris.com/2008/01/31/email-products-market-shares-versions-deployed-migrations-and-software-cost/ visited 11/10/08.

companies and hundreds of smaller, local players) and web mail providers (*e.g.,* Gmail, Yahoo! Mail or Hot Mail). Users employ a variety of e-mail client applications, including Microsoft Outlook, Windows Mail (formerly Outlook Express), Eudora, Entourage (on Apple machines) and, of course, their web browser 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.

A Snippet about 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* 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. When you access your personal e-mail from your own Internet Service Provider (ISP), 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 (for Post Office Protocol, version 3) is the oldest and 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. 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 back up if disaster strikes and facilitates easy access of your e-mail, again and again, from different computers. However, few ISPs afford unlimited storage space on their servers for users' e-mail, so mailboxes quickly become "clogged" with old e-mails, and the servers start 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 back up system for the servers which transmitted, transported and delivered the messages. In short, POP is locally-stored e-mail that supports some server storage.

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 back up 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 important 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 for Windows Mail/Outlook Express. A substantially 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 an Offline Synchronization file with the extension .OST), but this is user- and client application-dependent. Though it's exceedingly rare (especially for laptops) for there to be no local e-mail stores for a MAPI machine, it's nonetheless possible, and 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, users may 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, AOL, Hotmail 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 different 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 back up system used to protect server data can yield a mother lode of server e-mail.

Depending upon the back up 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 back up tapes and the potentially high 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 back up 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 paragraph 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 very 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 Header

Now that we've waded through the alphabet soup of protocols managing the movement of an e-mail message, let's take a 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 fairly 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*.

Figure 1 is an e-mail I sent from one of my e-mail addresses to another with a small image attached. Transmitted and received in seconds using the same machine, the message was sliced-and-diced into two versions (plain text and HTML), and its image attachment was encoded into Base 64, restructured to comply with rigid Internet protocols. It then winged its way across several time zones and servers, each server prepending its own peculiar imprimatur.

Figure 1: Exemplar E-Mail as Displayed in Client Application

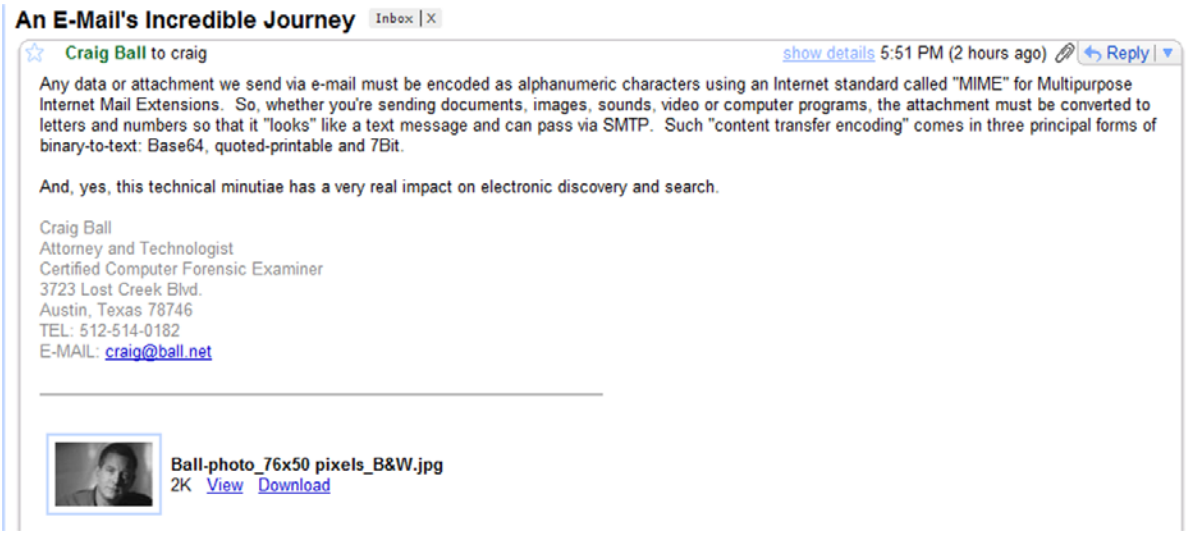


Figure 1 is just one of a variety of different ways in which an e-mail client application (in this instance the webmail application, Gmail) may display a message. When you view e-mail onscreen or print it out, you're seeing just part of the data contained in the message and attachment. Moreover, the e-mail client may be interpreting the message data according to, *e.g.,* the time zone and daylight savings time settings of your machine or its ability to read embedded formatting information. What you don't see—or see accurately—may be of little import, or it may be critical evidence. You've got to know what lies beneath to gauge its relevance.

Figure 2 (opposite) shows the source code of the Figure 1 e-mail, sent using a browser-based Gmail account. The e-mail came from the account computerforensics@gmail.com and was addressed to craig@ball.net. A small photograph in JPEG format was attached.

Before we dissect the e-mail message in Figure 2, note that any e-mail can be divided into two parts, the header and body of the message. By design, the header details the journey taken by the e-mail from origin to destination; but be cautioned that it's a fairly simple matter for a hacker to spoof (falsify) the identification of all but the final delivery server. Accordingly, where the origin or origination date of an e-mail is suspect, the actual route of the message may need to be validated at each server along its path.

In an e-mail header, each line which begins with the word "Received:" represents the transfer of the message between or within systems. 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. As the message passes through intervening hosts, each adds its own identifying information along with the date and time of transit.

E-Mail Autopsy: Tracing a Message's Incredible Journey

In this header, section **(A)** indicates the parts of the message designating the sender, addressee, recipient, date, time and subject line of the message. Importantly, the header also identifies the message as being formatted in MIME (MIME-Version: 1.0).[6] The **Content-Type: multipart/mixed** reference that follows indicates that the message holds both text and one or more attachments.

Though a message may be assigned various identification codes by the servers it transits in its journey (each enabling the administrator of the transiting e-mail server to track the message in the server logs), the message will contain one unique identifier assigned by the originating Message Transfer Agent. The unique identifier assigned to this message at **(B)** labeled "Message-ID:" is:

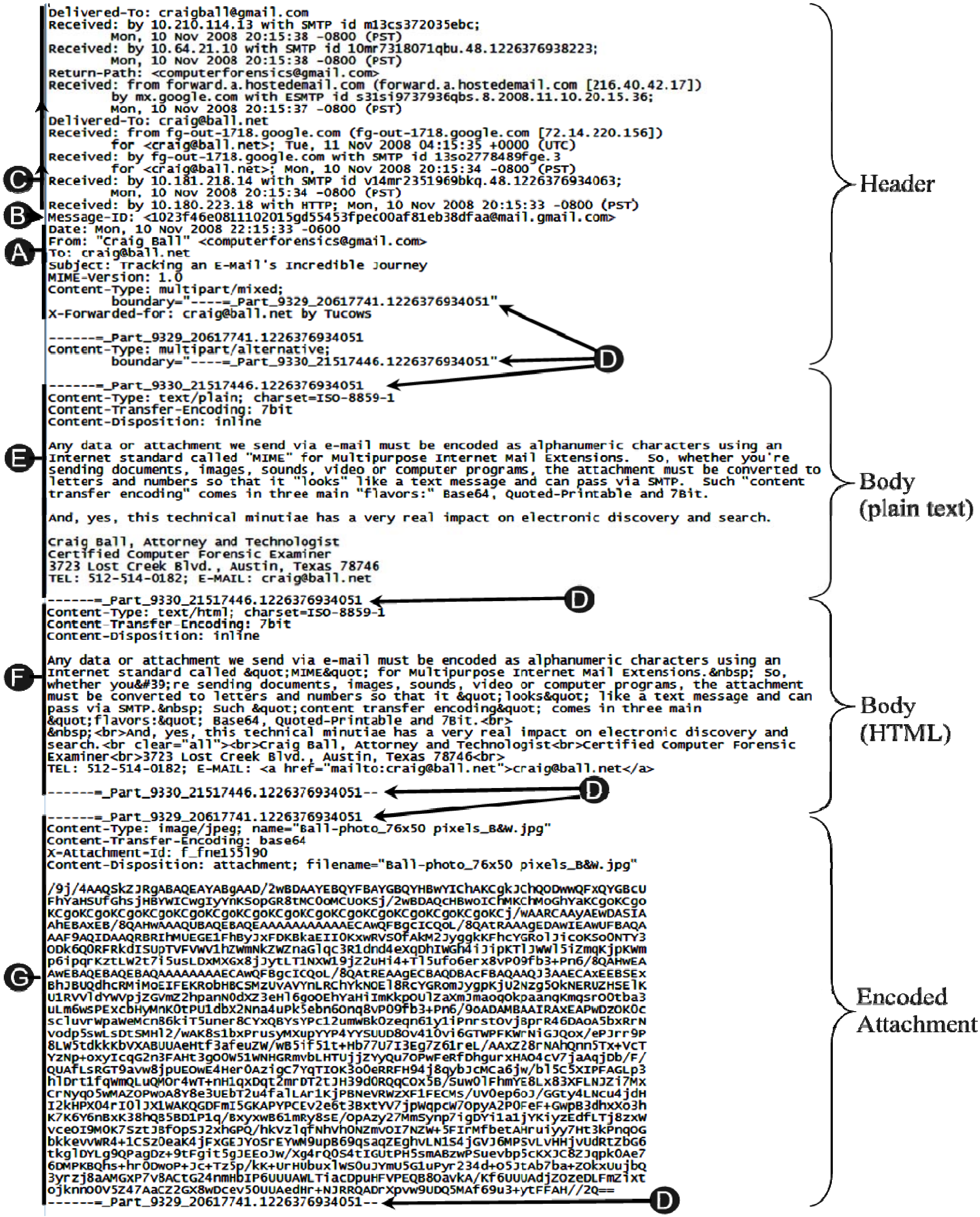
1023f46e0811102015gd55453fpec00af81eb38dfaa@mail.gmail.com.

In the line labeled "Date," both the date and time of transmittal are indicated. The time indicated is 22:15:33, and the "-0600" which follows denotes the time *difference* between the sender's local time (the system time on my computer in Austin, Texas in standard time) and Coordinated Universal Time (UTC), roughly equivalent to Greenwich Mean Time. As the offset from UTC is minus six hours on November 10, 2008, we deduce that the message was sent from a machine set to Central Standard 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.

At **(A),** we see that the message was addressed to craig@ball.net from computerforensics@gmail.com; yet, the ultimate recipient of the message is (as seen at

[6] 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.

Figure 2: Anatomy of an E-Mail



```
Delivered-To: craigball@gmail.com
Received: by 10.210.114.13 with SMTP id m13cs372035ebc;
        Mon, 10 Nov 2008 20:15:38 -0800 (PST)
Received: by 10.64.21.10 with SMTP id 10mr7318071qbu.48.1226376938223;
        Mon, 10 Nov 2008 20:15:38 -0800 (PST)
Return-Path: <computerforensics@gmail.com>
Received: from forward.a.hostedemail.com (forward.a.hostedemail.com [216.40.42.17])
        by mx.google.com with ESMTP id s31si9737936qbs.8.2008.11.10.20.15.36;
        Mon, 10 Nov 2008 20:15:37 -0800 (PST)
Delivered-To: craig@ball.net
Received: from fg-out-1718.google.com (fg-out-1718.google.com [72.14.220.156])
        for <craig@ball.net>; Tue, 11 Nov 2008 04:15:35 +0000 (UTC)
Received: by fg-out-1718.google.com with SMTP id 13so2778489fge.3
        for <craig@ball.net>; Mon, 10 Nov 2008 20:15:34 -0800 (PST)
Received: by 10.181.218.14 with SMTP id v14mr2351969bkq.48.1226376934063;
        Mon, 10 Nov 2008 20:15:34 -0800 (PST)
Received: by 10.180.223.18 with HTTP; Mon, 10 Nov 2008 20:15:33 -0800 (PST)
Message-ID: <1023f46e0811102015gd55453fpec00af81eb38dfaa@mail.gmail.com>
Date: Mon, 10 Nov 2008 22:15:33 -0600
From: "Craig Ball" <computerforensics@gmail.com>
To: craig@ball.net
Subject: Tracking an E-Mail's Incredible Journey
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="----=_Part_9329_20617741.1226376934051"
X-Forwarded-for: craig@ball.net by Tucows

------=_Part_9329_20617741.1226376934051
Content-Type: multipart/alternative;
        boundary="----=_Part_9330_21517446.1226376934051"

------=_Part_9330_21517446.1226376934051
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 7bit
Content-Disposition: inline

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 main "flavors:" Base64, Quoted-Printable and 7Bit.

And, yes, this technical minutiae has a very real impact on electronic discovery and search.

Craig Ball, Attorney and Technologist
Certified Computer Forensic Examiner
3723 Lost Creek Blvd., Austin, Texas 78746
TEL: 512-514-0182; E-MAIL: craig@ball.net
------=_Part_9330_21517446.1226376934051
Content-Type: text/html; charset=ISO-8859-1
Content-Transfer-Encoding: 7bit
Content-Disposition: inline

Any data or attachment we send via e-mail must be encoded as alphanumeric characters using an
Internet standard called &quot;MIME&quot; for Multipurpose Internet Mail Extensions.  So,
whether you&#39;re sending documents, images, sounds, video or computer programs, the attachment
must be converted to letters and numbers so that it &quot;looks&quot; like a text message and can
pass via SMTP.  Such &quot;content transfer encoding&quot; comes in three main
&quot;flavors:&quot; Base64, Quoted-Printable and 7Bit.<br>
 <br>And, yes, this technical minutiae has a very real impact on electronic discovery and
search.<br clear="all"><br>Craig Ball, Attorney and Technologist<br>Certified Computer Forensic
Examiner<br>3723 Lost Creek Blvd., Austin, Texas 78746<br>
TEL: 512-514-0182; E-MAIL: <a href="mailto:craig@ball.net">craig@ball.net</a>

------=_Part_9330_21517446.1226376934051--

------=_Part_9329_20617741.1226376934051
Content-Type: image/jpeg; name="Ball-photo_76x50 pixels_B&W.jpg"
Content-Transfer-Encoding: base64
X-Attachment-Id: f_fne155l90
Content-Disposition: attachment; filename="Ball-photo_76x50 pixels_B&W.jpg"

/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAYEBQYFBAYGBQYHBwYIChAKCgkJChQODwwQFxQYGBcU
FhYaHSUfGhsjHBYWICwgIyYnKSopGR8tMC0oMCUoKSj/2wBDAQcHBwoIChMKChMoGhYaKCgoKCgo
KCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCj/wAARCAAyAEwDASIA
AhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQA
AAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3
ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWm
p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEA
AwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSEx
BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYKNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElK
U1RVV1dYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3
uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDZOK0c
scluvrWpaWeMcn86kiT5uner8CYxQBYsYPc12umwBk0zeqn61y1iPnrstOvjBprR46DAoA5bxRrN
vodp5swLsDtSMH12/wAK8s1bxPrusyMXupYYP4YYY4YYSUUD8Ov410vi6GTWPFKWrNiGJQox/ePJrr9P
8LW5tdkkKbVXABUUAeHtf3afzeuZW/wB5if51t+Hb77U7I3Eg7Z61reL/AAxZ28rNAhQnn5Tx+VcT
YZNp+oxyIcqG2n3FAHt3gOO0W51WNHGRmvbLHTUjjzQu7OPwFeRfDhgurxHAO4cV7jaAqjDb/F/
QUAfLSRGT9avw8jpUEOwE4Her0AzigC7YqbTjJH39d0RQqCOx5B/Suw0fhmYE8Lx83XFLNJZi7Mx
CrNyqO5wMAZOPwoA8Y8e3uUEbT2u4fJ3PA/1r1KjPBNeVRWzXF1FEcs/UV0ep6oJ/GGty4LNcu4jdH
I2kHPX04rIOlJX1WAKQGDFmI5GKAPYPCEv2e6t3e6t3BxtYV7jpwQ8dhxXo3h
K7K6Y6nBxK38hQB5BD1P1q/BxywV7sztJ8dPqCEmZjuz8sE/OpAzy27MmsynP7igDYi1a1jYKiyzEdfLTj8zxw
vceOI9MOK7SztJBfOpSJ2xhGQ/hkvzEYPEE68Rue2vbyE4xjbw0EmqEdQMcQFDMvzzXm5zzFP5+fi2dhnvSFzvUrUrR+tDXJ
bkkevvWR4+1CSZ0eakJ4FXGEJYoSrEYWN9upB69qsaqZEghvLN1S4jGV6MPSvLvHHjvUdRtZbG6
tkglDYLg9QPagDz+9tFgit5gGJEEoJw/Xg4rQOS4tIGUtPH5SmABzwPmFkFVlMzg8QnKJC8ZJqpk0Ae7
6DMPKBQhs+hr0DwoP+Jc+AP4ZWrAR0AAAcK6Jpk5UtDGdtn6NpcT2G5qrBNqAAPdF7ub+yt0Bgga7
3yrzj8aAMGxP7V8ACtG24nmHbIP6UUUAAWLTiaeDpuHFVPEQB8oavkA/Kf6UUUAdjZOzeDLFmZ21xt
ojknn0OV5Z47AaCZ2GX8wDcev5OUUAedHr+NJRQADrxpvw9UDQ5MAf69u3+ytFFH/2Q==
------=_Part_9329_20617741.1226376934051--
```

the very top of the page) craigball@gmail.com. How this transpired can be deciphered from the header data, read from the bottom up.

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 10.180.223.18 at 20:15:33 -0800 (PST). Note that the server marks time in Pacific Standard Time, suggesting it may be located on the West Coast. The message is immediately handed off to another IP address 10.181.218.14 using Simple Mail Transfer Protocol, denoted by the initials SMTP. Next, we see another SMTP hand off to Google's server named "fg-out-1718.google.com" (IP address 72.14.220.156), which immediately transmits the message to a server with the IP address 216.40.42.17 and keeping time in UTC. A check of that IP address reveals that it's registered to Tucows International in Toronto, Canada.

Tucows is the host of my craig@ball.net address, which is configured to forward incoming messages to my other Gmail address, craigball@gmail.com. The forwarding is handled by a server called *forward.a.hostedemail.com*, and we then see the message received by server *MX.google.com*, transferred via SMTP to a server at IP address 10.64.21.10, then finally come to rest, delivered via SMTP to my craigball@gmail.com address via a server at 10.210.114.

As we examine the structure of the e-mail, we see that it contains content boundaries separating its constituent parts **(D).** These content boundary designators serve as delimiters; that is, sequences of one or more characters used to specify the boundary between text or data streams.[7] In order to avoid confusion of the boundary designator with message text, a complex sequence of characters is generated to serve as the two boundary designators used in this message. The first, called "_Part_9329_20617741.1226376934051," serves to separate the message header from the message body and signal the end of the message. The second delimiter, called "----=_Part_9330_21517446.1226376934051," denotes the boundaries between the segments of the message body: here, plain text content **(E)**, HTML content **(F)** and the encoded attachment **(G)**.

I didn't draft the message in *both* plain text and HTML formats, but my e-mail client thoughtfully did so to insure that my message won't confuse recipients using e-mail clients unable to display the richer formatting supported by HTML. For these recipients, there is a plain text version, too (albeit without the bolding, italics, hyperlinks and other embellishments of HTML). That the message carries alternative versions of the text is flagged by the designation at the break between header and message body stating: "**Content-Type: multipart/alternative**."

Looking more closely at the message boundaries, we see that each boundary delimiter is followed by Content-Type and Content-Transfer-Encoding designations. The plain

[7] The use of delimiters should be a familiar concept to those accustomed to specifying load file formats to accompany document image productions employed in e-discovery, where commas typically serve as field delimiters. Hence, these load files are sometimes referred to as CSV files (for comma-separated values).

text version of the message **(E)** begins: "**Content-Type: text/plain; charset=ISO-8859-1**," followed by "**Content-Transfer-Encoding: 7bit.**" The first obviously denotes plain text content using the very common ISO-8859-1 character encoding more commonly called "Latin 1."[8] The second signals that the content that follows consists of standard ASCII characters which historically employ 7 bits to encode 128 characters.

Not surprisingly, the boundary for the HTML version uses the Content-Type designator "text/html."

The most interesting and complex part of the message **(F)** starts after the second to last boundary delimiter with the specifications:
Content-Type: image/jpeg; name="Ball-photo_76x50 pixels_B&W.jpg"
Content-Transfer-Encoding: base64

The content type is self explanatory: an image in the JPEG format common to digital photography. The "name" segment obviously carries the name to be re-assigned to the attached photograph when decoded at its destination. But where, exactly, is the photograph?

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.[9]

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 gibberish comprising nearly 2,000 plain text characters **(G) and Figure 3.** 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

=

[8] In simplest terms, a character set or encoding pairs a sequence of characters (like the Latin alphabet) with numbers, byte values or other signals in much the same way as Morse code substitutes particular sequences of dots and dashes for letters. It's the digital equivalent of the Magic Decoder Rings once found in boxes of Cracker Jacks.

[9] 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 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.

Clearly, e-mail clients don't display all the information contained in a message's source but instead parse the contents into the elements we most want to see: To, From, Subject, body, and attachment. If you decide to try a little digital detective work on your own e-mail, you'll find that some e-mail client software doesn't make it easy to see complete header information. Microsoft's Outlook mail client makes it difficult to see the complete message source; however, you can see message headers for individual e-mails by opening the e-mail, then selecting "View" followed by "Options" until you see the "Internet headers" window on the Message Option menu. In Microsoft Outlook Express (now Windows Mail), highlight the e-mail item you want to analyze and then select "File" from the Menu bar, then "Properties," then click the "Details" tab followed by the "Message Source" button. For Gmail, select "Show Original" from the Reply button pull-down menu.

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, in reality, encoded into the message body. The time assigned to 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 is configured. What you see is not always what you get (or got).

Hashing and Deduplication

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."

The ability to "fingerprint" data 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. 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.

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 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.

For example, a deduplication application might hash only segments **(A)**, **(E)** and **(G)** of Figure 2. If the hash values of these segments match the hash values of the same segments of another message, can we say they are the same message? Probably, but it could also be important to evaluate emphasis added by HTML formatting (*e.g.,* text in red or underlined) or information about blind carbon copy recipients. The time values or routing information in the headers may also be important to reliably establishing authenticity, reliability or sequence.

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 whole 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 PDA 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:** Online e-mail residing in active files on enterprise servers: MS Exchange e.g., (.edb, .stm, .log files), Lotus Notes (.nsf files), Novell GroupWise (.db 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*.

• Nearline e-mail: Optical "juke box" devices, backups of user e-mail folders.

• Archived or journaled e-mail: *e.g.,* Autonomy Zantaz Enterprise Archive Solution, EMC EmailXtender, Mimosa NearPoint, Symantec Enterprise Vault.

Accessible, but Often Overlooked:

• E-mail residing on non-party servers: ISPs (IMAP, POP, HTTP servers), 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 a large number of 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 is something of an unfolding 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. 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.

***Caveat**: Before you or anyone on your behalf "poke around" on a computer system seeking a file or folder, recognize that absent the skilled use of specialized tools and techniques, such activity will result in changing data on the drive. Some of the changed data may be forensically significant (such as file access dates) and could constitute spoliation of evidence. If, under the circumstances of the case or matter, your legal or ethical obligation is to preserve the integrity of electronic evidence, then you and your client may be obliged to entrust the search only to qualified persons*

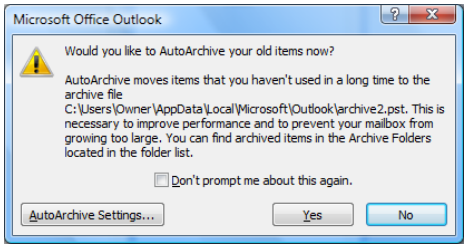
Finding Outlook E-Mail

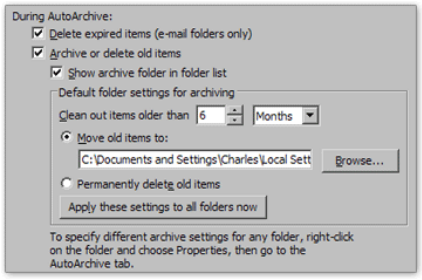
PST: Microsoft Outlook is by far the most widely used e-mail client in the business environment. Despite the confusing similarity of their names, Outlook is a much different and substantially more sophisticated application than Outlook Express (now called Windows Mail). One of many important differences is that where Outlook Express stores messages in plain text, Outlook encrypts and compresses messages. But the most significant challenge Outlook poses in discovery is the fact that 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. The Outlook PST file format is proprietary and its structure poorly documented, limiting your options when trying to view or process its contents to Outlook itself or one of a handful of PST file reader programs available for purchase and download via the Internet.

OST: While awareness of the Outlook PST file has grown, 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.

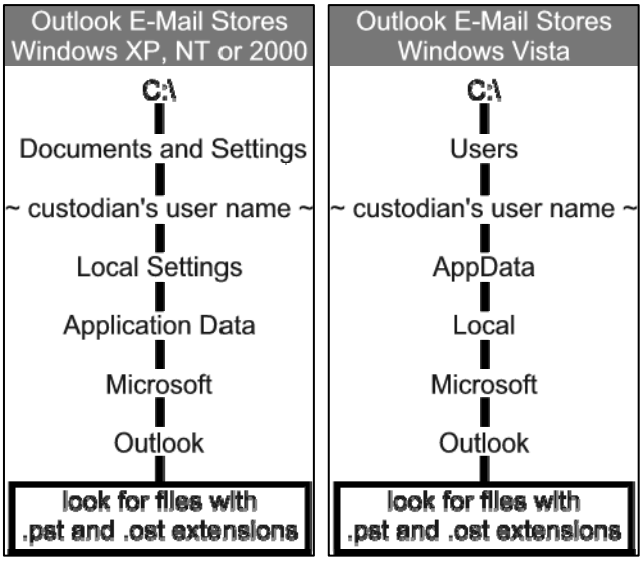
Archive.pst: Another file to consider is one customarily called, "archive.pst." As its name suggests, the archive.pst file holds older messages, either stored automatically or by user-initiated action. If you've used Outlook without manually configuring its archive settings, chances are the system periodically asks whether you'd like to auto archive older items. Every other week (by default), Outlook 2003 seeks to auto archive any Outlook items older than six months (or for Deleted and Sent items older than two months for Outlook 2007). 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 machines running Windows XP/NT/2000 or Vista, drill down from the root directory (C:\ for most users) according to the path diagram on the right for the applicable operating system. 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.

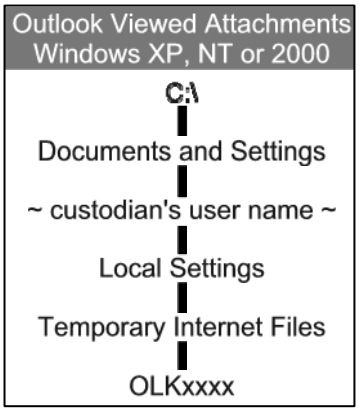


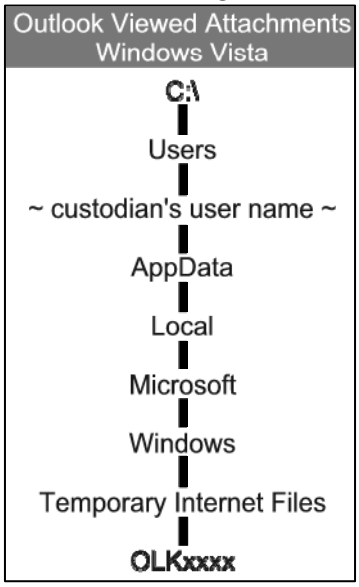
"Temporary" OLK Folders

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.[10]

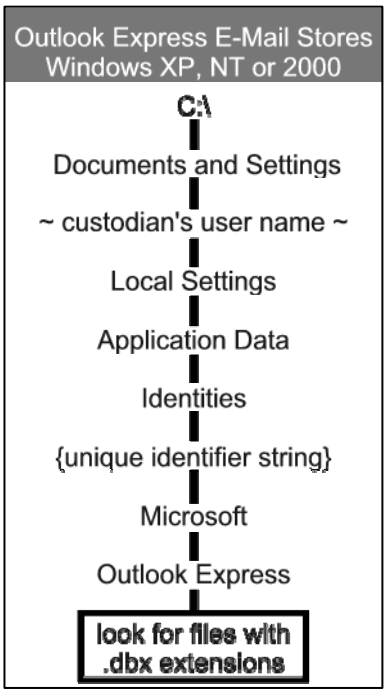


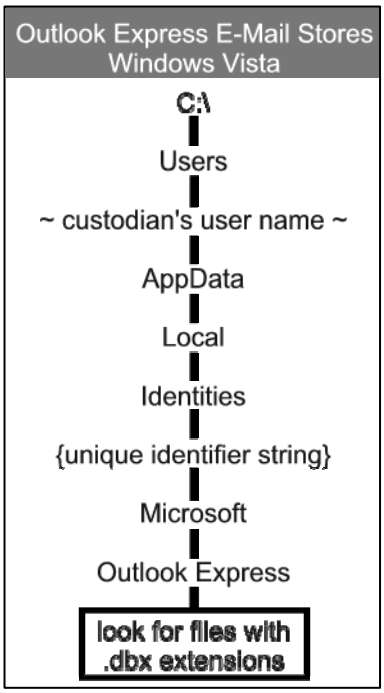


Finding Outlook Express E-Mail

Outlook Express has been bundled with every Windows operating system for about fifteen years, so you are sure to find at least the framework of an e-mail cache created by the program. Beginning with the release of Microsoft Vista, the Outlook Express application was renamed Windows Mail and the method of message storage was changed from a database format to storage as individual messages. More recently, Microsoft has sought to replace both Outlook Express on Windows XP and Windows Mail on Windows Vista with a freeware application called Windows Live Mail.

Outlook Express places e-mail in database files with the extension .dbx. The program creates a storage file for each e-mail storage folder that it displays, so expect to find at least Inbox.dbx, Outbox.dbx, Sent Items.dbx and Deleted Items.dbx. If the user has created other folders to hold e-mail, the contents of those folders will reside in a file with the structure *foldername*.dbx. Typically on a Windows XP/NT/2K system, you will find Outlook Express .dbx files in the path shown in the diagram at near right. Though less frequently encountered on a



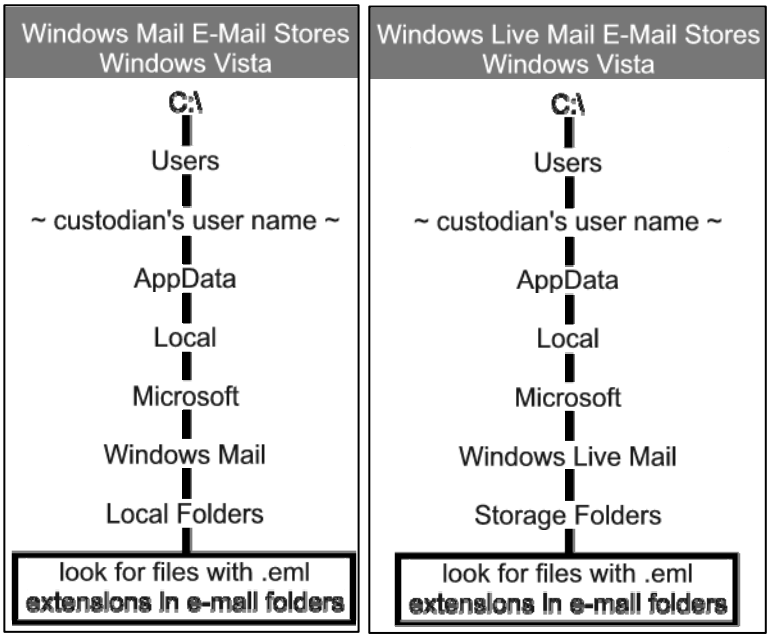


[10] By default, Windows hides system folders from users, so you may have to first make them visible. This is accomplished by starting Windows Explorer, then selecting 'Folder Options' from the Tools menu in Windows XP or 'Organize>Folder and Search Options' in Vista. Under the 'View' tab, scroll to '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.'

Windows Vista machine, the .dbx files would be found in the default location path shown at far right on preceding page. Multiple identifier strings (Globally Unique Identifiers) string listed in the Identities subfolder may be an indication of multiple e-mail stores and/or multiple users of the computer. You will need to check each Identity's path. Another approach is to use the Windows Search function (if under windows XP) to find all files ending .dbx, but be very careful to enable all three of the following Advanced Search options before running a search: Search System Folders, Search Hidden Files and Folders, and Search Subfolders. If you don't, you won't find any—or at least not all—Outlook Express e-mail stores. Be certain to check the paths of the files turned up by your search as it can be revealing to know whether those files turned up under a particular user identity, in Recent Files or even in the Recycle Bin.

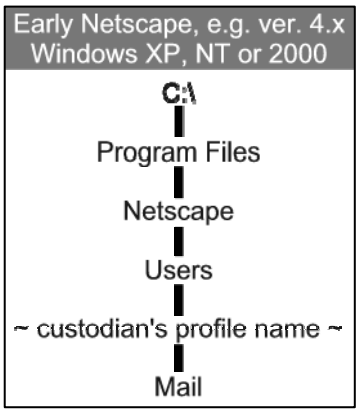
Finding Windows Mail and Windows Live Mail E-Mail Stores

You'll encounter Windows Mail on a machine running Windows Vista. By default, Windows Mail messages will be stored in oddly named individual files with the extension .eml and these housed in standard (*i.e.,* Inbox, Outbox, Sent Items, deleted Items, etc.) and user-created folders under the path diagrammed at near right.



Similarly, Windows Live Mail running on Vista will store messages as oddly named individual files with the extension .eml, within standard and user-created folders under the path seen at far right.

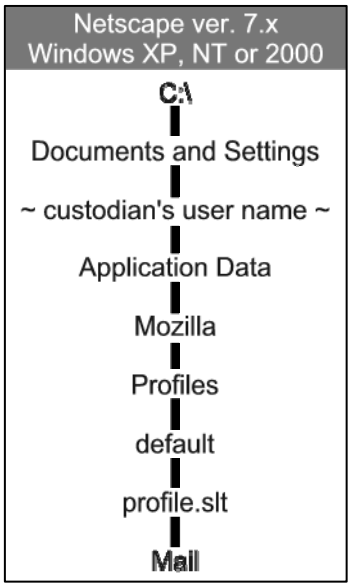
When collecting mail from these mail stores, it's important to capture both the message and the folder structure because, unlike the structured container seen in, *e.g.,* Outlook PST or OST files, the user's folder structure is not an integral part of the message storage scheme in Windows Mail or Live Mail.



Finding Netscape E-Mail

Though infrequently seen today, Netscape and its Mozilla e-mail client ruled the Internet before the browser wars left it crippled and largely forgotten. If you come across a Netscape e-mail client installation, keep in mind that the location of its e-mail stores will vary depending upon the version of the program installed. If it is an older version of the

program, such as Netscape 4.x and a default installation, you will find the e-mail stores by drilling down the path depicted at right. Expect to find two files for each mailbox folder, one containing the message text with no extension (*e.g.,* Inbox) and another which serves as an index file with a .snm extension (*e.g.,* Inbox.snm).



In the last version of Netscape to include an e-mail client (Netscape 7.x), both the location and the file structures/names were changed. Drill down using the default path shown at right and locate the folder for the e-mail account of interest, usually the name of the e-mail server from which messages are retrieved. If you don't see the Application Data folder, go to the Tools Menu, pull down to Folder Options, click on the View tab, and select "Show Hidden Files and Folders," then click "OK." You should find two files for each mailbox folder, one containing the message text with no extension (*e.g.,* Sent) and another which serves as an index file with a .msf extension (*e.g.,* Sent.msf). If you can't seem to find the e-mail stores, you can either launch a Windows search for files with the .snm and .msf extensions (*e.g.* *.msf) or, if you have access to the e-mail client program, you can check its configuration settings to identify the path and name of the folder in which e-mail is stored.

Microsoft Exchange Server

About 200 million people get their work e-mail via a Microsoft product called Exchange Server. It's been sold for about a dozen years and its latest version was introduced in 2007; although, most users continue to rely on the 2003 version 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 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!

But to underscore the potential for staggering complexity, appreciate that the latest Enterprise release of Exchange Server 2007 supports up to 50 storage groups per

server of up to 50 message stores per group, for a database size limit of 16 terabytes. If there is an upper limit on how many users can share a single message store, I couldn't ascertain what it might be!

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 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.

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.[11] Absent a change

[11] 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.

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.

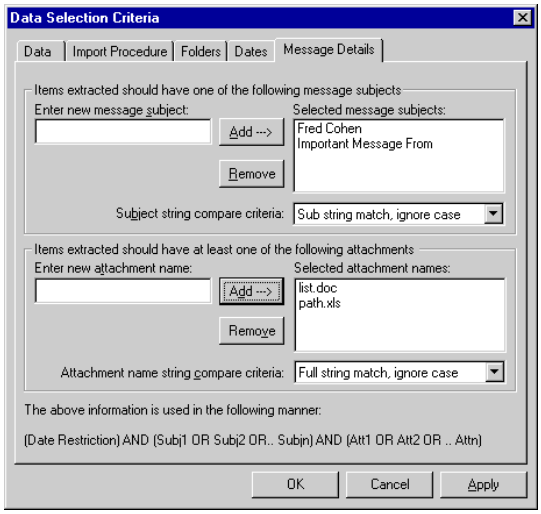
The 2003 version of Exchange Server stores data in a Storage Group containing a Mailbox Store and a Public Folder Store, each composed of two files: an .edb file and a .stm file. Mailbox Store, Priv1.edb, is a rich-text database file containing user’s email messages, text attachments and headers. Priv1.stm 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. Exchange Server 2007 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.

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.



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.

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 has the ability to 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.

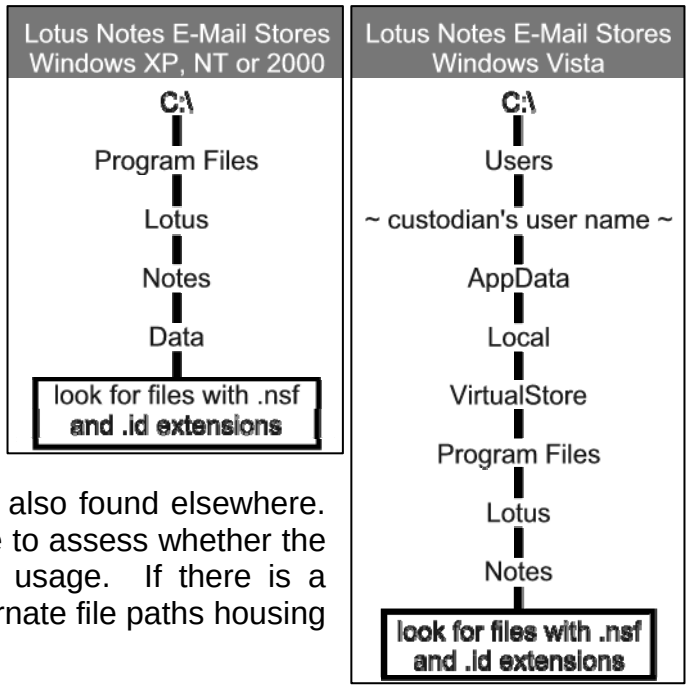
Lotus Domino Server and Notes Client

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 140 million Notes licenses sold to data worldwide.

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.[12] 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.

[12] Self-anointed "Technical Evangelist," Jeff Atwood describes 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://blogs.vertigosoftware.com/jatwood/archive/2005/08/11/1366.aspx. In fairness, Lotus Notes has been extensively overhauled since he made that observation.

Novell GroupWise

Experienced lawyers—that sound better than "older"--probably remember GroupWise. It originated as a WordPerfect virtual desktop product for messaging and calendaring called "WordPerfect Library," then became "WordPerfect Office." It changed to GroupWise when WordPerfect was acquired in 1993 by another deposed tech titan, Novell. GroupWise is alive (some might say "alive and well") in a handful of niche sectors, particularly government; but GroupWise's market share been so utterly eclipsed by its rivals as to make it seem almost extinct.

GroupWise is another tool thought of as "just an e-mail application" when it's really a Swiss army knife of data management features that happens to do e-mail, too. Because it's not a standalone e-mail server and client and because few vendors and experts have much recent experience with GroupWise, it's presents greater challenges and costs in e-discovery.

GroupWise is built on a family of databases which collectively present data comprising messages to users. That's an important distinction. Messages are less like discrete communications than reports *about* the communication, queried from a series of databases and presented *in the form of* an e-mail. User information is pulled from one database (ofuser), message content emerges from a second (ofmsg) and attachments are managed by a third database (offiles). When a user sends a GroupWise e-mail, the message is created in the user's message database and pointers to that message go to the user's Sent Items folder and the Recipients' Inboxes. Attachments go to the offiles database and pointers to attachments go out. Naturally, a more traditional method must be employed when message are sent beyond the GroupWise environment.

The prevailing practice in dealing with GroupWise e-mail is to convert messages to Outlook PST formats. The sole rationale for this seems to be that most e-discovery service providers are equipped to deal with PSTs and not native GroupWise data. Thus, the decision is driven by ignorance not evidence. Accordingly, a cottage industry has emerged dedicated to converting GroupWise ESI to other formats, but a few vendors tout their ability to work natively with GroupWise data. As often as not, conversion is a costly but harmless hurdle; but recognize that some data won't survive the leap between formats and, in choosing whether to deal with GroupWise data by conversion, you must assess whether the data sacrificed to the conversion process may be relevant and material.

Webmail

An estimated 1.2 billion people use webmail worldwide.[13] Ferris Research puts the number of business e-mail users in 2007 at around 780 million, accounting for some 6 *trillion* non-spam e-mails in sent in 2006. In April 2008, *USA Today*[14] reported the leading webmail providers' market share as:

[13] October 2007 report by technology market research firm The Radicati Group, expected to rise to 1.6 billion by 2011.

[14] http://www.usatoday.com/tech/products/2008-04-15-google-gmail-webmail_N.htm

Microsoft webmail properties:	256.2 million users
Yahoo:	254.6 million users
Google:	91.6 million users
AOL webmail properties:	48.9 million users

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.

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 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.

Keep in mind that webmail content can be fragile compared to server content. Users rarely employ a mechanism to back up webmail messages (other than the POP3 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.

Computer Forensics

Virtually any information that traverses a personal computer or other device has the potential to leave behind content that can be recovered in an examination of the machine or device by a skilled computer forensic examiner. Even container files like Outlook PST or OST files have a propensity to hold a considerable volume of recoverable information long after the user believes such data has been deleted.

Though the scope and methodology of a thorough computer forensic examination for hidden or deleted e-mail is beyond the scope of this paper,[15] readers should be mindful that a computer's operating system or **OS** (e.g., Windows or Vista, Mac or Linux) and installed software (**applications**) generate and store much more information than users realize. Some of this unseen information is **active data** readily accessible to users, but requiring skilled interpretation to be of value in illuminating human behavior. Examples include the data *about* data or **metadata** tracked by the OS and applications, but not displayed onscreen. For example, Microsoft Outlook records the date a Contact is created, but few of us customize the program to display that "date created" information.

Other active data reside in obscure locations or in coded formats less readily accessible to users, but enlightening when interpreted and correlated. Log files, hidden system files and information recorded in non-text formats are examples of **encoded data** that may reveal information about user behavior. As discussed, e-mail attachments and the contents of OST, PST and NSF files are all encoded data.

Finally, there are vast regions of hard drives and other data storage devices that hold **forensic data** even the operating systems and applications can't access. These "data landfills," 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 these vast, unstructured troves demands specialized tools, techniques and skill.

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., e-mail, phone records and voice mail, credit card transactions, keycard access data, documents and instant message communications).

Most cases require no forensic-level computer examination, so courts and litigants should closely probe whether a request for access to an opponent's machines to recover e-mail is grounded on a genuine need or is simply a fishing expedition. Except in cases involving, e.g., data theft, forgery or spoliation, computer forensics will usually be an effort of last resort for identification and production of e-mail.

The Internet has so broken down barriers between business and personal communications that workplace computers are routinely peppered with personal, privileged and confidential communications, even intimate and sexual content, and home computers normally contain some business content. 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 about others.

[15] For further reading on computer forensics, see Ball, *Five on Forensics*, http://www.craigball.com/cf.pdf and Ball, *What Judges Should Know About Computer Forensics*, published by the Federal Judicial Center and available at http://www.craigball.com/What_Judges_Computer_Forensics-200807.pdf

Accordingly, computer forensic examination should be governed by an agreed or court-ordered protocol to protect unwarranted disclosure of privileged and confidential information. Increasingly, 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 is first tendered to counsel for the party proffering the machines for examination, which party generates a privilege log and produces non-privileged, responsive data.[16]

Whether an expert or court-appointed neutral conducts the examination, the order or agreed protocol granting forensic examination of ESI should provide for 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 consulting an expert to determine feasibility, complexity or cost.

There is no more a "standard" protocol for forensic examination than there is a "standard" set of deposition questions. In either case, a good 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 or agreed protocol to afford the examiner discretion to adapt to the evidence and apply their expertise.

Why Deleted Doesn't Mean Gone

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 (often called "**file tables**") to track files and fragmented portions of files.

When a user hits "Delete," nothing happens to the actual file targeted for deletion. Instead, a change is made to the 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…our file is a needle in a haystack, lost among millions of other unallocated clusters.

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

[16] For further discussion of forensic examination protocols, see Ball in Your Court, *Problematic Protocols*, November 2008, Law Technology News;
http://www.lawtechnews.com/r5/showkiosk.asp?listing_id=2756144&pub_id=5173&category_id=27902

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 particular 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 the parties or a court direct that an examiner "recover all deleted files," that's an exercise that could take weeks, followed by countless hours spent culling corrupted files. Instead, the protocol should, as feasible, specify the *particular* file types of interest (i.e., e-mail and attachments) based upon how the machine's was used and the facts and issues in the case.

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.

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 is necessitated in most cases by the enormous volume of ESI. When keywords are not unique or less than about 6 letters long, many false positives ("**noise hits**") are encountered. 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.

Keyword searching for e-mail generally involves looking for strings invariably associated with messages (e.g., e-mail addresses) or words or phrases known or expected to be seen in deleted messages (e.g., subject lines, signatures or header data).

Because e-mail is commonly encoded, encrypted and/or compressed, and because it customarily resides in container files structured more like databases than discrete messages, computer forensic analysis for e-mail recovery is particularly challenging. On the other hand, e-mail tends to lodge in so many places and formats; it's the rare case where at least some responsive e-mail cannot be found.

As relevant, a forensic protocol geared to e-mail should include a thorough search for orphaned message collections, looking for any of the varied formats in which e-mail is stored (e.g., PST, OST, NSF, MSG, EML, MHT, DBX, IDX) and of unallocated clusters for binary signatures of deleted container files. Container files themselves should be subjected to processes that allow for recovery of double deleted messages that remain lodged within uncompacted containers.[17]

Webmail can often be found in the Internet cache (Temporary Internet Files) as well as within unallocated clusters and swap files. Desktop search and indexing programs (like Google Desktop) may also hold the full text of deleted e-mail. Moreover, devices like smart phones and PDAs employ synchronization files to store and transfer e-mail. Finally, e-mail clients like Outlook can themselves hold messages (e.g., corrupted drafts and failed transmissions) along with metadata unseen by users.

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.

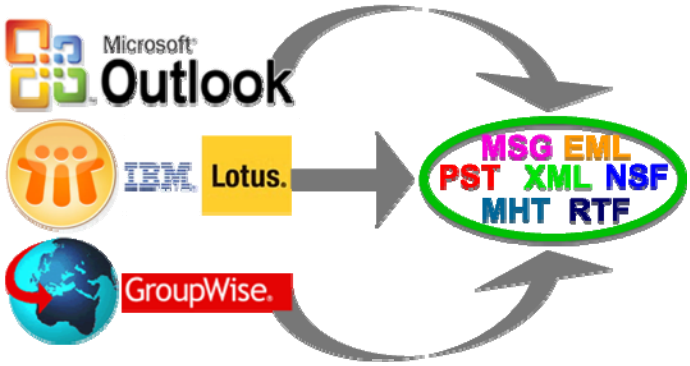
Requesting parties often demand "native production" of e-mail; but, electronic mail is rarely produced natively in the sense of supplying a duplicate of the source container file. That is, few litigants produce the entire Exchange database EDB file to the other side. Even those that produce mail in the format employed natively by the application (e.g., as a PST file) aren't likely to produce the source file but will fashion a

[17] A common technique used on PST containers is to corrupt the file header on a copy of the container file and use Microsoft's free Scanpst utility to repair it. This process sometimes recovers double deleted messages as these remain in the container until periodically compacted by Outlook. Scanpst can also be run against chunks of the unallocated clusters to ferret out deleted PSTs.

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 an 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).[18]

Similarly, producing parties employ imaged production and supply TIFF image files of messages, but in order to approximate the usability of the source 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., Summation, Concordance) 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.

Other possible format choices include XML[19] and MHT,[20] as well as Rich Text Format (RTF)--essentially plain text with improved formatting—and, for small collections, paper printouts.

---

[18] 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.

[19] XML is eXtensible Markup Language, an unfamiliar name for a familiar technology. Markup languages are coded identifiers paired with text and other information. They can define the appearance of content, like the Reveal Codes screen of Corel Inc.'s WordPerfect documents. They also serve to tag content to distinguish whether 09011957 is a birth date (09/01/1957), a phone number (0-901-1957) or a Bates number. Plus, markup languages allow machines to talk to each other in ways humans understand. For

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:

1. 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 and other non-textual ways we communicate and accentuate information in e-mail messages.
2. Support accurate electronic searchability of the message text and header data;
3. 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;
4. Preserve family relationships between messages and attachments;
5. Convey the folder structure/path of the source message;
6. Include message metadata responsive to the requester’s legitimate needs;
7. Facilitate redaction of privileged and confidential content and, as feasible, identification and sequencing akin to Bates numbering; and
8. Enable reliable date and time normalization across the messages produced.[21]

---

further information about the prospects for XML in e-discovery, see Ball in Your Court, *Trying to Love XML*, March 2008, Law Technology News;
http://www.lawtechnews.com/r5/showkiosk.asp?listing_id=1929884

[20] 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.

[21] 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 according to 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. For further discussion of date/time normalization to UTC, see Ball in Your Court, *SNAFU*, September 2008, Law Technology News;
http://www.lawtechnews.com/r5/showkiosk.asp?listing_id=2217760

Conclusion

By now, you're wishing you'd taken my advice on page one and not begun. It's too late. You know too much about e-mail to ever again trot out the "I dunno" defense.

As I look back over the preceding discussion of the nerdy things that lawyers need to know about e-mail, I'm struck by how much *more* there is to cover. We've barely touched on e-mail backup systems, review platforms, visual analytics, e-mail archival, cloud computing, search and sampling, message conversion tools, unified messaging and a host of other exciting topics.

I hope you've gleaned something useful from this paper. I invite and appreciate your suggestions for corrections and improvements. Please e-mail them to craig@ball.net.

About the Author

Craig Ball, of Austin is a Board Certified Texas trial lawyer and an accredited computer forensics expert, who's dedicated his career to teaching the bench and bar about forensic technology and trial tactics. Craig hung up his trial lawyer spurs to till the soils of justice as a court-appointed special master and consultant in electronic evidence, as well as publishing and lecturing on computer forensics, emerging technologies, digital persuasion and electronic discovery. Fortunate to supervise, consult on or serve as Special Master in connection with some of the world's largest electronic discovery projects and most prominent cases, Craig also greatly values his role as an instructor in computer forensics and electronic evidence to the Department of Justice and other law enforcement and security agencies.

Craig Ball is a prolific contributor to continuing legal and professional education programs throughout the United States, having delivered over 500 presentations and papers. Craig's articles on forensic technology and electronic discovery frequently appear in the national media, including in American Bar Association, ATLA and American Lawyer Media print and online publications. He also writes a multi-award winning monthly column on computer forensics and e-discovery for Law Technology News and Law.com called "Ball in your Court." Rated AV by Martindale Hubbell and named as one of the Best Lawyers in America and a Texas Superlawyer, Craig is a recipient of the Presidents' Award, the State Bar of Texas' most esteemed recognition of service to the profession.

Craig's been married to a trial lawyer for 21 years. He and Diana have two delightful teenagers and share a passion for world travel, cruising and computing.

Undergraduate Education: Rice University, triple major, 1979
Law School: University of Texas, 1982 with honors