Blown to Bits

Your Life, Liberty, and Happiness After the Digital Explosion

Hal Abelson
Ken Ledeen
Harry Lewis

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales
international@pearson.com

Visit us on the Web: www.informit.com/aw

Library of Congress Cataloging-in-Publication Data:

Abelson, Harold.
Blown to bits : your life, liberty, and happiness after the digital explosion / Hal Abelson, Ken Ledeen, Harry Lewis.
p. cm.
ISBN 0-13-713559-9 (hardback : alk. paper) 1. Computers and civilization. 2. Information technology—Technological innovations. 3. Digital media. I. Ledeen, Ken, 1946- II. Lewis, Harry R. III. Title.
QA76.9.C66A245 2008
303.48'33—dc22

2008005910

For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671 3447

ISBN-13: 978-0-13-713559-2
ISBN-10: 0-13-713559-9
Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.
Third printing December 2008

Editor in Chief
Mark Taub

Acquisitions Editor
Greg Doench

Development Editor
Michael Thurston

Managing Editor
Gina Kanouse

Senior Project Editor
Kristy Hart

Copy Editor
Water Crest Publishing, Inc.

Indexer
Erika Millen

Proofreader
Williams Woods Publishing Services

Publishing Coordinator
Michelle Housley

Interior Designer and Composition
Nonie Ratcliff

Cover Designer
Chuti Prasertsith

CHAPTER 3

Ghosts in the Machine

Secrets and Surprises of Electronic Documents

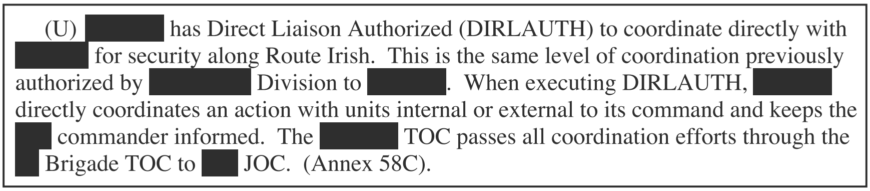
What You See Is Not What the Computer Knows

On March 4, 2005, Italian journalist Giuliana Sgrena was released from captivity in Baghdad, where she had been held hostage for a month. As the car conveying her to safety approached a checkpoint, it was struck with gunfire from American soldiers. The shots wounded Sgrena and her driver and killed an Italian intelligence agent, Nicola Calipari, who had helped engineer her release.

A fierce dispute ensued about why U.S soldiers had rained gunfire on a car carrying citizens of one of its Iraq war allies. The Americans claimed that the car was speeding and did not slow when warned. The Italians denied both claims. The issue caused diplomatic tension between the U.S. and Italy and was a significant political problem for the Italian prime minister.

The U.S. produced a 42-page report on the incident, exonerating the U.S. soldiers. The report enraged Italian officials. The Italians quickly released their own report, which differed from the U.S. report in crucial details.

Because the U.S. report included sensitive military information, it was heavily redacted before being shared outside military circles (see Figure 3.1). In another time, passages would have been blacked out with a felt marker, and the document would have been photocopied and given to reporters. But in the information age, the document was redacted and distributed electronically, not physically. The redacted report was posted on a web site the allies used to provide war information to the media. In an instant, it was visible to any of the world's hundreds of millions of Internet users.

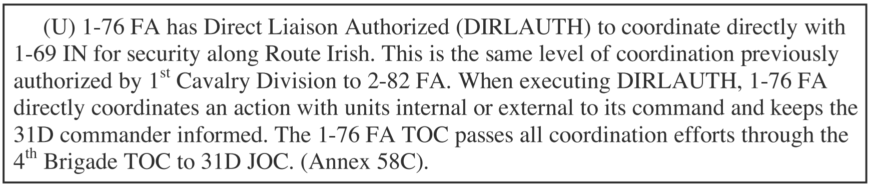


(U) [redacted] has Direct Liaison Authorized (DIRLAUTH) to coordinate directly with [redacted] for security along Route Irish. This is the same level of coordination previously authorized by [redacted] Division to [redacted]. When executing DIRLAUTH, [redacted] directly coordinates an action with units internal or external to its command and keeps the [redacted] commander informed. The [redacted] TOC passes all coordination efforts through the [redacted] Brigade TOC to [redacted] JOC. (Annex 58C).

Source: `http://www.corriere.it/Media/Documenti/Classified.pdf`, extract from page 10.

FIGURE 3.1 Section from page 10 of redacted U.S. report on the death of Italian journalist Nicola Calipari. Information that might have been useful to the enemy was blacked out.

One of those Internet users was an Italian blogger, who scrutinized the U.S. report and quickly recovered the redacted text using ordinary office software. The blogger posted the full text of the report (see Figure 3.2) on his own web site. The unredacted text disclosed positions of troops and equipment, rules of engagement, procedures followed by allied troops, and other information of interest to the enemy. The revelations were both dangerous to U.S. soldiers and acutely embarrassing to the U.S. government, at a moment when tempers were high among Italian and U.S. officials. In the middle of the most high-tech war in history, how could this fiasco have happened?



(U) 1-76 FA has Direct Liaison Authorized (DIRLAUTH) to coordinate directly with 1-69 IN for security along Route Irish. This is the same level of coordination previously authorized by 1st Cavalry Division to 2-82 FA. When executing DIRLAUTH, 1-76 FA directly coordinates an action with units internal or external to its command and keeps the 31D commander informed. The 1-76 FA TOC passes all coordination efforts through the 4th Brigade TOC to 31D JOC. (Annex 58C).

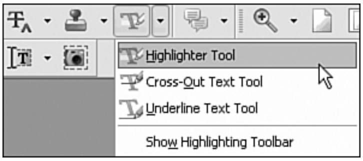
Source: `http://www.corriere.it/Media/Documenti/Unclassified.doc`.

FIGURE 3.2 The text of Figure 3.1 with the redaction bars electronically removed.

Paper documents and electronic documents are useful in many of the same ways. Both can be inspected, copied, and stored. But they are not equally useful for all purposes. Electronic documents are easier to change, but paper documents are easier to read in the bathtub. In fact, the metaphor of a series of bits as a "document" can be taken only so far. When stretched beyond its breaking point, the "document" metaphor can produce surprising and damaging results—as happened with the Calipari report.

Office workers love "WYSIWYG" interfaces—"What You See Is What You Get." They edit the electronic document on the screen, and when they print it, it looks just the same. They are deceived into thinking that what is in the computer is a sort of miniaturized duplicate of the image on the screen, instead of computer codes that produce the picture on the screen. In fact, the WYSIWYG metaphor is imperfect, and therefore risky. The report on the death of Nicola Calipari illustrates what can go wrong when users accept such a metaphor too literally. What the authors of the document saw was dramatically different from what they got.

The report had been prepared using software that creates PDF files. Such software often includes a "Highlighter Tool," meant to mimic the felt markers that leave a pale mark on ordinary paper, through which the underlying text is visible (see Figure 3.3). The software interface shows the tool's icon as a marker writing a yellow stripe, but the user can change the color of the stripe. Probably someone tried to turn the Highlighter Tool into a redaction tool by changing its color to black, unaware that what was visible on the screen was not the same as the contents of the electronic document.



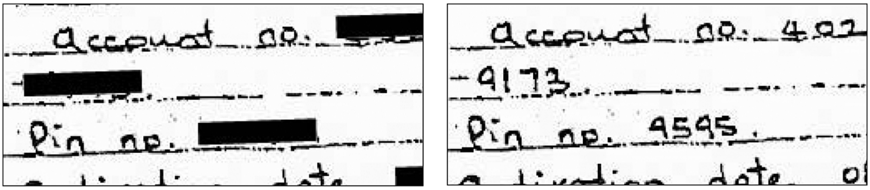
Reprinted with permission from Adobe Systems Incorporated.

FIGURE 3.3 Adobe Acrobat Highlighter Tool, just above the middle. On the screen, the "highlighter" is writing yellow ink, but with a menu command, it can be changed to any other color.

The Italian blogger guessed that the black bars were nothing more than overlays created using the Highlighter Tool, and that the ghostly traces of the invisible words were still part of the electronic document that was posted on the web. With that realization, he easily undid the black "highlighting" to reveal the text beneath.

Just as disturbing as this mistake is the fact that two major newspapers had quite publicly made the same mistake only a few years before. On April 16, 2000, the *New York Times* had detailed a secret CIA history of attempts by the U.S. to overthrow Iran's government in 1953. The newspaper reproduced sections of the CIA report, with black redaction bars to obscure the names of CIA operatives within Iran. The article was posted on the Web in mid-June, 2000, accompanied by PDFs of several pages of the CIA report. John Young, who administers a web site devoted to publishing government-restricted documents, removed the redaction bars and revealed the names of CIA agents. A controversy ensued about the ethics and legality of the disclosure, but the names are still available on the Web as of this writing.

The *Washington Post* made exactly the same mistake in 2002, when it published an article about a demand letter left by the Washington snipers, John Allen Muhammad and John Lee Malvo. As posted on the *Post*'s web site, certain information was redacted in a way that was easily reversed by an inquisitive reader of the online edition of the paper (see Figure 3.4). The paper fixed the problem quickly after its discovery, but not quickly enough to prevent copies from being saved.



account no: [redacted]

[redacted]

Pin no. [redacted]

account no: 402

-9173

Pin no. 4595.

Source: Washington Post web site, transferred to web.bham.ac.uk/forensic/news/02/sniper2.html. Actual images taken from slide 29 of http://www.ccc.de/congress/2004/fahrplan/files/316-hidden-data-slides.pdf.

FIGURE 3.4 Letter from the Washington snipers. On the left, the redacted letter as posted on the *Washington Post* web site. On the right, the letter with the redaction bars electronically removed.

What might have been done in these cases, instead of posting the PDF with the redacted text hidden but discoverable? The Adobe Acrobat software has a security feature, which uses encryption (discussed in Chapter 5, "Secret Bits") to make it impossible for documents to be altered by unauthorized persons, while still enabling anyone to view them. Probably those who created these documents did not know about this feature, or about commercially available software called Redax, which government agencies use to redact text from documents created by Adobe Acrobat.

A clumsier, but effective, option would be to scan the printed page, complete with its redaction bars. The resulting file would record only a series of black and white dots, losing all the underlying typographical structure—font names and margins, for example. Whatever letters had once been "hidden" under the redaction bars could certainly not be recovered, yet this solution has an important disadvantage.

One of the merits of formatted text documents such as PDFs is that they can be "read" by a computer. They can be searched, and the text they contain can be copied. With the document reduced to a mass of black and white dots, it could no longer be manipulated as text.

A more important capability would be lost as well. The report would be unusable by programs that vocalize documents for visually impaired readers. A blind reader could “read” the U.S. report on the Calipari incident, because software is available that “speaks” the contents of PDF documents. A blind reader would find a scanned version of the same document useless.

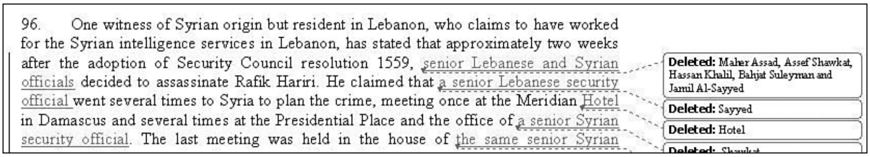
Tracking Changes—and Forgetting That They Are Remembered

In October, 2005, UN prosecutor Detlev Mehlis released to the media a report on the assassination of former Lebanese Prime Minister Rafik Hariri. Syria had been suspected of engineering the killing, but Syrian President Bashar al-Assad denied any involvement. The report was not final, Mehlis said, but there was “evidence of both Lebanese and Syrian involvement.” Deleted, and yet uncovered by the reporters who were given the document, was an incendiary claim: that Assad’s brother Maher, commander of the Republican Guard, was personally involved in the assassination.

Microsoft Word offers a “Track Changes” option. If enabled, every change made to the document is logged as part of the document itself—but ordinarily not shown. The document bears its entire creation history: who made each change, when, and what it was. Those editing the document can also add comments—which would not appear in the final document, but may help editors explain their thinking to their colleagues as the document moves around electronically within an office.

Of course, information about strategic planning is not meant for outsiders to see, and in the case of legal documents, can have catastrophic consequences if revealed. It is a simple matter to remove these notes about the document’s history—but someone has to remember to do it! The UN prosecutor neglected to remove the change history from his Microsoft Word document, and a reporter discovered the deleted text (see Figure 3.5). (Of course, in Middle Eastern affairs, one cannot be too suspicious. Some thought that Mehlis had intentionally left the text in the document, as a warning to the Syrians that he knew more than he was yet prepared to acknowledge.)

A particularly negligent example of document editing involved SCO Corporation, which claimed that several corporations violated its intellectual property rights. In early 2004, SCO filed suit in a Michigan court against Daimler Chrysler, claiming Daimler had violated terms of its Unix software agreement with SCO. But the electronic version of its complaint carried its modification history with it, revealing a great deal of information about SCO’s litigation planning. In particular, when the change history was revealed, it turned out that until exactly 11:10 a.m. on February 18, 2004, SCO had instead planned to sue a different company, Bank of America, in federal rather than state court, for copyright infringement rather than breach of contract!



96. One witness of Syrian origin but resident in Lebanon, who claims to have worked for the Syrian intelligence services in Lebanon, has stated that approximately two weeks after the adoption of Security Council resolution 1559, senior Lebanese and Syrian officials decided to assassinate Rafik Hariri. He claimed that a senior Lebanese security official went several times to Syria to plan the crime, meeting once at the Meridian Hotel in Damascus and several times at the Presidential Place and the office of a senior Syrian security official. The last meeting was held in the house of the same senior Syrian

Deleted: Maher Assad, Assef Shawkat, Hassan Khalil, Bahjat Suleyman and Jamil Al-Sayyed

Deleted: Sayyed

Deleted: Hotel

Deleted: Shawkat

Source: Section of UN report, posted on Washington Post web site, `www.washingtonpost.com/wp-srv/world/syria/mehlis.report.doc`.

FIGURE 3.5 Section from the UN report on the assassination of Rafik Hariri. An earlier draft stated that Maher Assad and others were suspected of involvement in the killing, but in the document as it was released, their names were replaced with the phrase "senior Lebanese and Syrian officials."

Saved Information About a Document

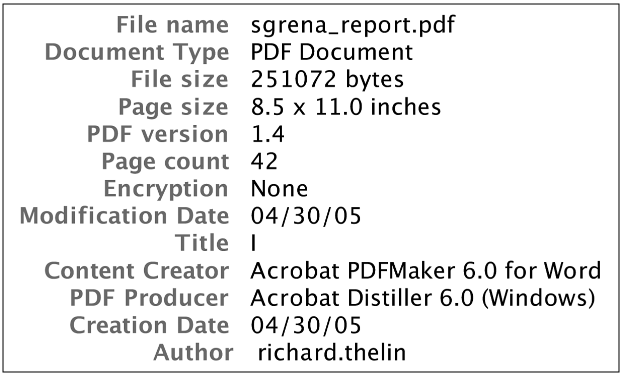
An electronic document (for example, one produced by text-processing software) often includes information that is *about* the document—so-called *metadata*. The most obvious example is the name of the file itself. File names carry few risks. For example, when we send someone a file as an email attachment, we realize that the recipient is going to see the name of the file as well as its contents.

But the file is often tagged with much more information than just its name. The metadata generally includes the name associated with the owner of the computer, and the dates the file was created and last modified—often useful information, since the recipient can tell whether she is receiving an older or newer version than the version she already

FORGING METADATA

Metadata can help prove or refute claims. Suppose Sam emails his teacher a homework paper after the due date, with a plea that the work had been completed by the deadline, but was undeliverable due to a network failure. If Sam is a cheater, he could be exposed if he doesn't realize that the "last modified" date is part of the document. However, if Sam *is* aware of this, he could "stamp" the document with the right time by re-setting the computer's clock before saving the file. The name in which the computer is registered and other metadata are also forgeable, and therefore are of limited use as evidence in court cases.

has. Some word processors include version information as well, a record of who changed what, when, and why. But the unaware can be trapped even by such innocent information, since it tends not to be visible unless the recipient asks to see it. In Figure 3.6, the metadata reveals the name of the military officer who created the redacted report on the death of Nicola Calipari.



File name	sgrena_report.pdf
Document Type	PDF Document
File size	251072 bytes
Page size	8.5 x 11.0 inches
PDF version	1.4
Page count	42
Encryption	None
Modification Date	04/30/05
Title	I
Content Creator	Acrobat PDFMaker 6.0 for Word
PDF Producer	Acrobat Distiller 6.0 (Windows)
Creation Date	04/30/05
Author	richard.thelin

Reprinted with permission from Adobe Systems Incorporated.

FIGURE 3.6 Part of the metadata of the Calipari report, as revealed by the "Properties" command of Adobe Acrobat Reader. The data shows that Richard Thelin was the author, and that he altered the file less than two minutes after creating it. Thelin was a Lieutenant Colonel in the U.S. Marine Corps at the time of the incident.

Authorship information leaked in this way can have real consequences. In 2003, the British government of Tony Blair released documentation of its case for joining the U.S. war effort in Iraq. The document had many problems—large parts of it turned out to have been plagiarized from a 13-year-old PhD thesis. Equally embarrassing was that the electronic fingerprints of four civil servants who created it were left on the document when it was released electronically on the No. 10 Downing Street web site. According to the *Evening Standard of London*, "All worked in propaganda units controlled by Alastair Campbell, Tony Blair's director of strategy and communications," although the report had supposedly been the work of the Foreign Office. The case of the "dodgy dossier" caused an uproar in Parliament.

You don't have to be a businessperson or government official to be victimized by documents bearing fingerprints. When you send someone a document as an attachment to an email, very likely the document's metadata shows who actually created it, and when. If you received it from someone else and then altered it, that may show as well. If you put the text of the document into the body of your email instead, the metadata won't be included; the message will be just the text you see on the screen. Be sure of what you are sending before you send it!

Can the Leaks Be Stopped?

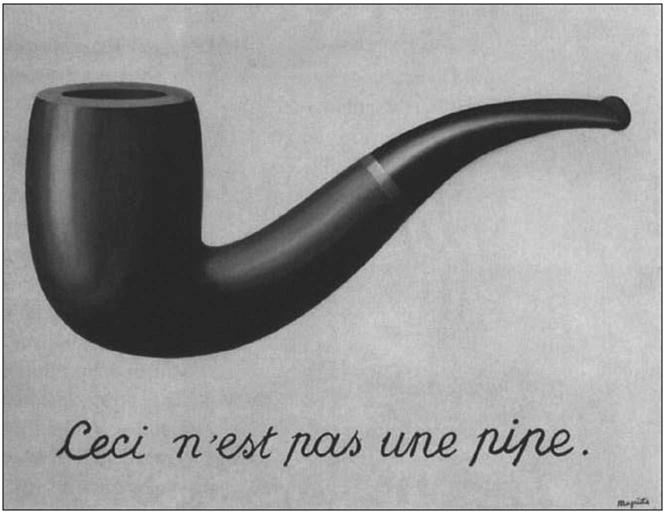
Even in the most professional organizations, and certainly in ordinary households, knowledge about technological dangers and risks does not spread instantaneously to everyone who should know it. The Calipari report was published five years after the *New York Times* had been embarrassed. How can users of modern information technology—today, almost all literate people—stay abreast of knowledge about when and how to protect their information?

It is not easy to prevent the leakage of sensitive information that is hidden in documents but forgotten by their creators, or that is captured as metadata. In principle, offices should have a check-out protocol so that documents are cleansed before release. But in a networked world, where email is a critical utility, how can offices enforce document release protocols without rendering simple tasks cumbersome? A rather harsh measure is to prohibit use of software that retains such information; that was the solution adopted by the British government in the aftermath of the "dodgy dossier" scandal. But the useful features of the software are then lost at the same time. A protocol can be established for converting "rich" document formats such as that of Microsoft Word to formats that retain less information, such as Adobe PDF. But it turns out that measures used to eradicate personally identifiable information from documents don't achieve as thorough a cleansing as is commonly assumed.

At a minimum, office workers need education. Their software has great capabilities they may find useful, but many of those useful features have risks as well. And we all just need to think about what we are doing with our documents. We all too mindlessly re-type keystrokes we have typed a hundred times in the past, not pausing to think that the hundred and first situation may be different in some critical way!

Representation, Reality, and Illusion

René Magritte, in his famous painting of a pipe, said "This isn't a pipe" (see Figure 3.7). Of course it isn't; it's a painting of a pipe. The image is made out of paint, and Magritte was making a metaphysical joke. The painting is entitled "The treachery of images," and the statement that the image isn't the reality is part of the image itself.



Los Angeles County Museum of Art. Purchased with funds provided by the Mr. and Mrs. William Preston Harrison Collection. Photograph © 2007 Museum Associates/LACMA.

FIGURE 3.7 Painting by Magritte. The legend says "This isn't a pipe." Indeed, it's only smudges of paint that make you think of a pipe, just as an electronic document is only bits representing a document.

When you take a photograph, you capture inside the camera something from which an image can be produced. In a digital camera, the bits in an electronic memory are altered according to some pattern. The image, we say, is "represented" in the camera's memory. But if you took out the memory and looked at it, you couldn't see the image. Even if you printed the pattern of 0s and 1s stored in the memory, the image wouldn't appear. You'd have to know *how* the bits represent the image in order to get at the image itself. In the world of digital photography, the format of the bits has been standardized, so that photographs taken on a variety of cameras can be displayed on a variety of computers and printed on a variety of printers.

The general process of digital photography is shown in Figure 3.8. Some external reality—a scene viewed through a camera lens, for example—is turned into a string of bits. The bits somehow capture useful information about reality, but there is nothing "natural" about the way reality is captured. The representation is a sort of ghost of the original, not identical to the original and actually quite unlike it, but containing enough of the soul of the original to be useful later on. The representation follows rules. The rules are arbitrary conventions and the product of human invention, but they have been widely accepted so photographs can be exchanged.

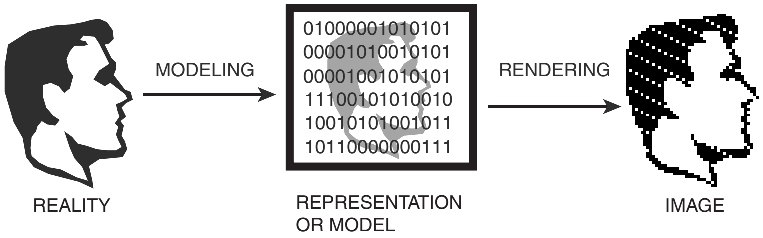


FIGURE 3.8 Reproducing an image electronically is a two-stage process. First, the scene is translated into bits, creating a digital model. Then the model is rendered as a visible image. The model can be stored indefinitely, communicated from one place to another, or computationally analyzed and enhanced to produce a different model before it is rendered. The same basic structure applies to the reproduction of video and audio.

The representation of the photograph in bits is called a *model* and the process of capturing it is called *modeling*. The model is turned into an image by *rendering* the model; this is what happens when you transfer the bits representing a digital photograph to a computer screen or printer. Rendering brings the ghost back to life. The image resembles, to the human eye, the original reality—provided that the model is good enough. Typically, a model that is not good enough—has too few bits, for example—cannot produce an image that convincingly resembles the reality it was meant to capture.

Modeling always omits information. Magritte's painting doesn't smell like a pipe; it has a different patina than a pipe; and you can't turn it around to see what the other side of the pipe looks like. Whether the omitted information is irrelevant or essential can't be judged without knowing how the model is going to be used. Whoever creates the model and renders it has the power to shape the experience of the viewer.

The process of modeling followed by rendering applies to many situations other than digital photography. For example, the same transformations happen when music is captured on a CD or as an MP3. The rendering process produces audible music from a digital representation, via stereo speakers or a headset. CDs and MP3s use quite distinct modeling methods, with CDs generally capturing music more accurately, using a larger number of bits.

Knowing that digital representations don't resemble the things they represent explains the difference between the terms "analog" and "digital." An analog telephone uses a continuously varying electric signal to represent a continuously varying sound—the voltage of the telephone signal is an "analog" of the sound it resembles—in the same way that Magritte applied paint smoothly to canvas to mimic the shape of the pipe. The shift from analog to digital technologies, in telephones, televisions, cameras, X-ray machines, and many other devices, at first seems to lose the immediacy and simplicity of the old devices. But the enormous processing power of modern computers makes the digital representation far more flexible and useful.

Indeed, the same general processes are at work in situations where *there is no "reality" because the images are of things that have never existed.* Examples are video games, animated films, and virtual walk-throughs of unbuilt architecture. In these cases, the first step of Figure 3.8 is truncated. The "model" is created not by capturing reality in an approximate way, but by pure synthesis: as the strokes of an artist's electronic pen, or the output of computer-aided design software.

Can We Be Sure a Photo Is Unretouched?

Cryptographic methods (discussed in Chapter 5) can establish that a digital photograph has not been altered. A special camera gets a digital key from the "image verification system," attaches a "digital signature" (see Chapter 5) to the image and uploads the image and the signature to the verification system. The system processes the received image with the same key and verifies that the same signature results. The system is secure because it is impossible, with any reasonable amount of computation, to produce another image that would yield the same signature with this key.

The severing of the immediate connection between representation and reality in the digital world has created opportunities, dangers, and puzzles. One of the earliest triumphs of "digital signal processing," the science of doing computations on the digital representations of reality, was to remove the scratches and noise from old recordings of the great singer Enrico Caruso. No amount of analog electronics could have cleaned up the old records and restored the clarity to Caruso's voice.

And yet the growth of digital "editing" has its dark side as well. Photo-editing software such as Photoshop can be used to alter photographic evidence presented to courts of law.

The movie *Toy Story* and its descendants are unlikely to put human actors out of work in the near future, but how should society think about synthetic child pornography? "Kiddie porn" is absolutely illegal, unlike other forms of pornography, because of the harm done to the children who are abused to produce it. But what about pornographic images of children who do not exist and never have—who are simply the creation of a skilled graphic synthesizer? Congress outlawed such virtual kiddie porn in 1996, in a law that prohibited any image that "is, or appears to be, of a minor engaging in sexually explicit conduct." The Supreme Court overturned the law on First Amendment grounds. Prohibiting images that "appear to" depict children is going too far, the court ruled—such synthetic pictures, no matter how abhorrent, are constitutionally protected free speech.

In this instance at least, reality matters, not what images appear to show. Chapter 7, "You Can't Say That on the Internet," discusses other cases in which society is struggling to control social evils that are facilitated by information technology. In the world of exploded assumptions about reality and artifice, laws that combat society's problems may also compromise rights of free expression.

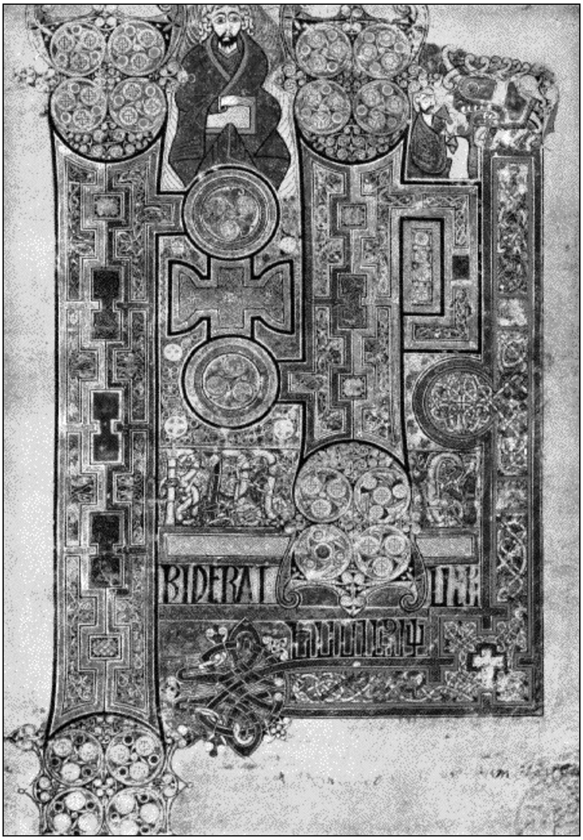
In the world of exploded assumptions about reality and artifice, laws that combat society's problems may also compromise rights of free expression.

What Is the Right Representation?

Figure 3.9 is a page from the Book of Kells, one of the masterpieces of medieval manuscript illumination, produced around A.D. 800 in an Irish monastery. The page contains a few words of Latin, portrayed in an astoundingly complex interwoven lacework of human and animal figures, whorls, and crosshatching. The book is hundreds of pages long, and in the entire work no two of the letters or decorative ornaments are drawn the same way. The elaborately ornate graphic shows just 21 letters (see Figure 3.10).

Digital Cameras and Megapixels

Megapixels—millions of pixels—are a standard figure of merit for digital cameras. If a camera captures too few pixels, it can't take good photographs. But no one should think that more pixels invariably yield a better image. If a digital camera has a low-quality lens, more pixels will simply produce a more precise representation of a blurry picture!



Copyright © Trinity College, Dublin.

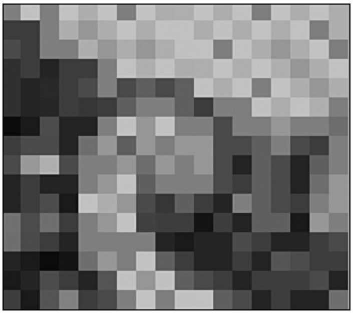
FIGURE 3.9 Opening page of the Gospel of St. John from the Book of Kells.

IN PRINCIPIO ERAT VERBUM

FIGURE 3.10 The words of the beginning of the gospel of St. John. In the book of Kells, the easiest word to spot is ERAT, just to the left of center about a quarter of the way up the page.

Do these two illustrations contain the same information? The answer depends on what information is meant to be recorded. If the only important thing were the Latin prose, then either representation might be equally good, though Figure 3.10 is easier to read. But the words themselves are far from the only important thing in the Book of Kells. It is one of the great works of Western art and craftsmanship.

A graphic image such as Figure 3.9 is represented as a rectangular grid of many rows and columns, by recording the color at each position in the grid (see Figure 3.11). To produce such a representation, the page itself is scanned, one narrow row after the next, and each row is divided horizontally into tiny square "picture elements" or *pixels*. An image representation based on a division into pixels is called a *raster* or *bitmap representation*. The representation corresponds to the structure of a computer screen (or a digital TV screen), which is also divided into a grid of individual pixels—how many pixels, and how small they are, affect the quality and price of the display.

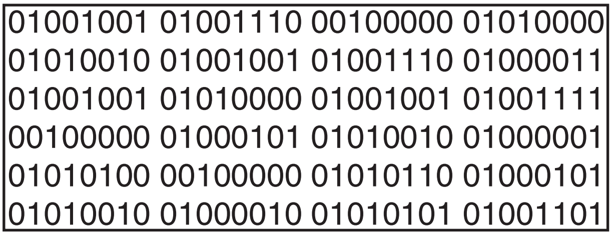


Copyright © Trinity College, Dublin.

FIGURE 3.11 A detail enlarged from the upper-right corner of the opening page of John from the Book of Kells.

What would be the computer representation of the mere Latin text, Figure 3.10? The standard code for the Roman alphabet, called ASCII for the American Standard Code for Information Interchange, assigns a different 8-bit code to each letter or symbol. ASCII uses one byte (8 bits) per character. For example, A = 01000001, a = 01100001, $ = 00100100, and 7 = 00110111.

The equation 7 = 00110111 means that the bit pattern used to represent the symbol "7" in a string of text is 00110111. The space character has its own code, 00100000. Figure 3.12 shows the ASCII representation of the characters "IN PRINCIPIO ERAT VERBUM," a string of 24 bytes or 192 bits. We've separated the long string of bits into bytes to improve readability ever so slightly! But inside the computer, it would just be one bit after the next.



```
01001001 01001110 00100000 01010000
01010010 01001001 01001110 01000011
01001001 01010000 01001001 01001111
00100000 01000101 01010010 01000001
01010100 00100000 01010110 01000101
01010010 01000010 01010101 01001101
```

FIGURE 3.12 ASCII bit string for the characters of "IN PRINCIPIO ERAT VERBUM."

So `01001001` represents the letter I. But not always! Bit strings are used to represent many things other than characters. For example, the same bit string `01001001`, if interpreted as the representation of a whole number in binary notation, represents 73. A computer cannot simply look at a bit string `01001001` and know whether it is supposed to represent the letter I or the number 73 or data of some other type, a color perhaps. A computer can interpret a bit string only if it knows the conventions that were used to create the document—the intended interpretation of the bits that make up the file.

The meaning of a bit string is a matter of convention. Such conventions are arbitrary at first. The code for the letter I could have been `11000101` or pretty much anything else. Once conventions have become accepted through a social process of agreement and economic incentive, they became nearly as inflexible as if they were physical laws. Today, millions of computers assume

FILENAME EXTENSIONS

The three letters after the dot at the end of a filename indicate how the contents are to be interpreted. Some examples are as follows:

Extension	File Type
.doc	Microsoft Word document
.odt	OpenDocument text document
.ppt	Microsoft PowerPoint document
.ods	OpenDocument Spreadsheet
.pdf	Adobe Portable Document Format
.exe	Executable program
.gif	Graphics Interchange Format (uses 256-color palette)
.jpg	JPEG graphic file (Joint Photographic Experts Group)
.mpg	MPEG movie file (Moving Picture Experts Group)

that `01001001`, if interpreted as a character, represents the letter I, and the universal acceptance of such conventions is what makes worldwide information flows possible.

The document format is the key to turning the representation into a viewable document. If a program misinterprets a document as being in a different format from the one in which it was created, only nonsense will be rendered. Computers not equipped with software matching the program that created a document generally refuse to open it.

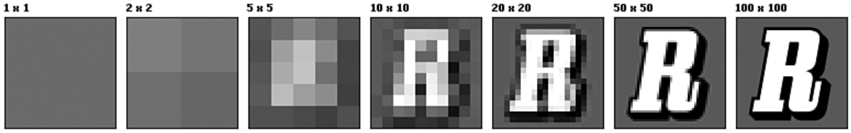
Which representation is "better," a raster image or ASCII? The answer depends on the use to which the document is to be put. For representation of freeform shapes in a great variety of shades and hues, a raster representation is unbeatable, provided the pixels are small enough and there are enough of them. But it is hard even for a trained human to find the individual letters within Figure 3.9, and it would be virtually impossible for a computer program. On the other hand, a document format based on ASCII codes for characters, such as the PDF format, can easily be searched for text strings.

The PDF format includes more than simply the ASCII codes for the text. PDF files include information about typefaces, the colors of the text and of the background, and the size and exact positions of the letters. Software that produces PDFs is used to typeset elegant documents such as this one. In other words, PDF is actually a *page description language* and describes visible features that are typographically meaningful. But for complicated pictures, a graphical format such as JPG must be used. A mixed document, such as these pages, includes graphics within PDF files.

Reducing Data, Sometimes Without Losing Information

Let's take another look at the page from the Book of Kells, Figure 3.9, and the enlargement of a small detail of that image, Figure 3.11. The computer file from which Figure 3.9 was printed is 463 pixels wide and 651 pixels tall, for a total of about 300,000 individual pixels. The pages of the Book of Kells measure about 10 by 13 inches, so the raster image has only about 50 pixels per inch of the original work. That is too few to capture the rich detail of the original—Figure 3.11 actually shows one of the animal heads in the top-right corner of the page. A great deal of detail was lost when the original page was scanned and turned into pixels. The technical term for the problem is *undersampling.* The scanning device "samples" the color value of the original document at discrete points to create the representation of the document, and in this case, the samples are too far apart to preserve detail that is visible to the naked eye in the original.

The answer to undersampling is to increase the resolution of the scan—the number of samples per inch. Figure 3.13 shows how the quality of an image improves with the resolution. In each image, each pixel is colored with the "average" color of part of the original.



Credit as in Wikipedia, `en.wikipedia.org/wiki/Image:Resolution_illustration.png`.

FIGURE 3.13 A shape shown at various resolutions, from 1 × 1 to 100 × 100 pixels. A square block consisting of many pixels of a single shade can be represented much more compactly than by repeating the code for that shade as many times as there are pixels.

But, of course, a price is paid for increased resolution. The more pixels in the representation of an image, the more memory is needed to hold the representation. Double the resolution, and the memory needed goes up by a factor of four, since the resolution doubles both vertically and horizontally.

Standard software uses a variety of representational techniques to represent raster graphics more concisely. Compression techniques are of two kinds: "lossless" and "lossy." A *lossless* representation is one that allows exactly the same image to be rendered. A *lossy* representation allows an approximation to the same image to be rendered—an image that is different from the original in ways the human eye may or may not be able to discern.

AUDIO COMPRESSION

MP3 is a lossy compression method for audio. It uses a variety of tricks to create small data files. For example, human ears are not far enough apart to hear low-frequency sounds stereophonically, so MP3s may record low frequencies in mono and play the same sound to both speakers, while recording and playing the higher frequencies in stereo! MP3s are "good enough" for many purposes, but a trained and sensitive ear can detect the loss of sound quality.

One method used for lossless image compression takes advantage of the fact that in most images, the color doesn't change from pixel to pixel—the image has *spatial coherence*, to use the official term. Looking at the middle and rightmost images in Figure 3.13, for example, makes clear that in the 100 × 100 resolution image, the 100 pixels in a

10 × 10 square in the top-left corner are all the same color; there is no need to repeat a 24-bit color value 100 times in the representation of the image.

Accordingly, graphic representations have ways of saying "all pixels in this block have the same color value." Doing so can reduce the number of bits significantly.

Depending on how an image will be used, a lossy compression method might be acceptable. What flashes on your TV is gone before you have time to scrutinize the individual pixels. But in some cases, only lossless compression is satisfactory. If you have the famous Zapruder film of the Kennedy assassination and want to preserve it in a digital archive, you want to use a lossless compression method once you have digitized it at a suitably fine resolution. But if you are just shipping off the image to a low-quality printer such as those used to print newspapers, lossy compression might be fine.

Technological Birth and Death

The digital revolution was possible because the capacity of memory chips increased, relentlessly following Moore's Law. Eventually, it became possible to store digitized images and sounds at such high resolution that their quality was higher than analog representations. Moreover, the price became low enough that the storage chips could be included in consumer goods. But more than electrical engineering is involved. At more than a megabyte per image, digital cameras and HD televisions would still be exotic rarities. A *megabyte* is about a million bytes, and that is just too much data per image. The revolution also required better algorithms—better computational methods, not just better hardware—and fast, cheap processing chips to carry out those algorithms.

For example, digital video compression utilizes *temporal coherence* as well as spatial coherence. Any portion of the image is unlikely to change much in color from frame to frame, so large parts of a picture typically do not have to be retransmitted to the home when the frame changes after a thirtieth of a second. At least, that is true in principle. If a woman in a TV image walks across a fixed landscape, only her image, and a bit of landscape that newly appears from behind her once she passes it, needs be transmitted—*if* it is computationally feasible to compare the second frame to the first before it is transmitted and determine exactly where it differs from its predecessor. To keep up with the video speed, there is only a thirtieth of a second to do that computation. And a complementary computation has to be carried out at the other end—the previously transmitted frame must be modified to reflect the newly transmitted information about what part of it should change one frame time later.

Digital movies could not have happened without an extraordinary increase in speed and drop in price in computing power. Decompression algorithms are built into desktop photo printers and cable TV boxes, cast in silicon in chips more powerful than the fastest computers of only a few years ago. Such compact representations can be sent quickly through cables and as satellite signals. The computing power in the cable boxes and television sets is today powerful enough to reconstruct the image from the representation of what has changed. Processing is power.

By contrast, part of the reason the compact disk is dying as a medium for distributing music is that it doesn't hold enough data. At the time the CD format was adopted as a standard, decompression circuitry for CD players would have been too costly for use in homes and automobiles, so music could not be recorded in compressed form. The magic of Apple's iPod is not just the huge capacity and tiny physical size of its disk—it is the power of the processing chip that renders the stored model as music.

The birth of new technologies presage the death of old technologies. Digital cameras killed the silver halide film industry; analog television sets will soon be gone; phonograph records gave way to cassette tapes, which in turn gave way to compact disks, which are themselves now dying in favor of digital music players with their highly compressed data formats.

The periods of transition between technologies, when one emerges and threatens another that is already in wide use, are often marked by the exercise of power, not always progressively. Businesses that dominate old technologies are sometimes innovators, but often their past successes make them slow to change. At their worst, they may throw up roadblocks to progress in an attempt to hold their ground in the marketplace. Those roadblocks may include efforts to scare the public about potential disruptions to familiar practices, or about the dollar costs of progress.

Data formats, the mere conventions used to intercommunicate information, can be remarkably contentious, when a change threatens the business of an incumbent party, as the Commonwealth of Massachusetts learned when it tried to change its document formats. The tale of Massachusetts and OpenDocument illustrates how hard change can be in the digital world, although it sometimes seems to change on an almost daily basis.

Data Formats as Public Property

No one owns the Internet, and everyone owns the Internet. No government controls the whole system, and in the U.S., the federal government controls only the computers of government agencies. If you download a web page to your home computer, it will reach you through the cooperation of several, perhaps dozens, of private companies between the web server and you.

This flexible and constantly changing configuration of computers and communication links developed because the Internet is in its essence not hardware, but protocols—the conventions that computers use for sending bits to each other (see the Appendix). The most basic Internet Protocol is known as IP. The Internet was a success because IP and the designs for the other protocols became public standards, available for anyone to use. Anyone could build on top of IP. Any proposed higher-level protocol could be adopted as a public standard if it met the approval of the networking community. The most important protocol exploiting IP is known as TCP. TCP is used by email and web software to ship messages reliably between computers, and the pair of protocols is known as TCP/IP. The Internet might not have developed that way had proprietary networking protocols taken hold in the early days of networking.

Uploading and Downloading

Historically, we thought of the Internet as consisting of powerful corporate "server" machines located "above" our little home computers. So when we retrieved material from a server, we were said to be "downloading," and when we transferred material from our machine to a server, we were "uploading." Many personal machines are now so powerful that the "up" and "down" metaphors are no longer descriptive, but the language is still with us. See the Appendix, and also the explanation of "peer-to-peer" in Chapter 6, "Balance Toppled."

It was not always thus. Twenty to thirty years ago, all the major computer companies—IBM, DEC, Novell, and Apple—had their own networking protocols. The machines of different companies did not intercommunicate easily, and each company hoped that the rest of the world would adopt its protocols as standards. TCP/IP emerged as a standard because agencies of the U.S. government insisted on its use in research that it sponsored—the Defense Department for the ARPANET, and the National Science Foundation for NSFnet. TCP/IP was embedded in the Berkeley Unix operating system, which was developed under federal grants and came to be widely used in universities. Small companies quickly moved to use TCP/IP for their new products. The big companies moved to adopt it more slowly. The Internet, with all of its profusion of services and manufacturers, could not have come into existence had one of the incumbent manufacturers won the argument—and they failed even though their networking products were technologically superior to the early TCP/IP implementations.

File formats stand at a similar fork in the road today. There is increasing concern about the risks of commercial products evolving into standards. Society will be better served, goes the argument, if documents are stored in formats hammered out by standards organizations, rather than disseminated as part of commercial software packages. But consensus around one *de facto* commercial standard, the `.doc` format of Microsoft Word, is already well advanced.

Word's `.doc` format is proprietary, developed by Microsoft and owned by Microsoft. Its details are now public, but Microsoft can change them at any time, without consultation. Indeed, it does so regularly, in order to enhance the capabilities of its software—and new releases create incompatibilities with legacy documents. Some documents created with Word 2007 can't be opened in Word 2003 without a software add-on, so even all-Microsoft offices risk document incompatibilities if they don't adjust to Microsoft's format changes. Microsoft does not exclude competitors from adopting its format as their own document standard—but competitors would run great risks in building on a format they do not control.

In a large organization, the cost of licensing Microsoft Office products for thousands of machines can run into the millions of dollars. In an effort to create competition and to save money, in 2004 the European Union advanced the use of an "OpenDocument Format" for exchange of documents among EU businesses and governments. Using ODF, multiple companies could enter the market, all able to read documents produced using each other's software.

In September, 2005, the Commonwealth of Massachusetts decided to follow the EU initiative. Massachusetts announced that effective 15 months later, all the state's documents would have to be stored in OpenDocument Format. About 50,000 state-owned computers would be affected. State officials estimated the cost savings at about $45 million. But Eric Kriss, the state's secretary of administration and finance, said that more than software cost was at stake. Public documents were public property; access should never require the cooperation of a single private corporation.

Microsoft did not accept the state's decision without an argument. The company rallied advocates for the disabled to its side, claiming that no available OpenDocument software had the accessibility features Microsoft offered. Microsoft, which already had state contracts that extended beyond the switchover date, also argued that adopting the ODF standard would be unfair to Microsoft and costly to Massachusetts. "Were this proposal to be adopted, the significant costs incurred by the Commonwealth, its citizens, and the private sector would be matched only by the levels of confusion and incompatibility that would result...." Kriss replied, "The question is whether a sovereign state has the obligation to ensure that its public documents remain forever free and unencumbered by patent, license, or other technical impediments. We say, yes, this is an imperative. Microsoft says they disagree and want the world to use their proprietary formats." The rhetoric quieted down, but the pressure increased. The stakes were high for Microsoft, since where Massachusetts went, other states might follow.

OpenDocument, Open Source, Free

These three distinct concepts all aim, at least in part, to slow the development of software monopolies. OpenDocument (opendocument.xml.org) is an open standard for file formats. Several major computer corporations have backed the effort, and have promised not to raise intellectual property issues that would inhibit the development of software meeting the standards. Open source (opensource.org) is a software development methodology emphasizing shared effort and peer review to improve quality. The site openoffice.org provides a full suite of open source office productivity tools, available without charge. Free software—"Free as in freedom, not free beer" (www.fsf.org, www.gnu.org)—"is a matter of the users' freedom to run, copy, distribute, study, change, and improve the software."

Three months later, neither Kriss nor Quinn was working for the state. Kriss returned to private industry as he had planned to do before joining the state government. The *Boston Globe* published an investigation of Quinn's travel expenses, but the state found him blameless. Tired of the mudslinging, under attack for his decision about open standards, and lacking Kriss's support, on December 24, Quinn announced his resignation. Quinn suspected "Microsoft money and its lobbyist machine" of being behind the *Globe* investigation and the legislature's resistance to his open standard initiative.

The deadline for Massachusetts to move to OpenDocuments has passed, and as of the fall of 2007, the state's web site still says the switchover will occur in the future. In the intervening months, the state explains, it became possible for Microsoft software to read and write OpenDocument formats, so the shift to OpenDocument would not eliminate Microsoft from the office software competition. Nonetheless, other software companies would not be allowed to compete for the state's office software business until "accessibility characteristics of the applications meet or exceed those of the currently deployed office suite"—i.e., Microsoft's. For the time being, Microsoft has the upper hand, despite the state's effort to wrest from private hands the formats of its public documents.

Which bits mean what in a document format is a multi-billion dollar business. As in any big business decisions, money and politics count, reason becomes entangled with rhetoric, and the public is only one of the stakeholders with an interest in the outcome.

Hiding Information in Images

The surprises in text documents are mostly things of which the authors were ignorant or unaware. Image documents provide unlimited opportunities for hiding things intentionally—hiding secrets from casual human observers, and obscuring open messages destined for human recipients so anti-spam software won't filter them out.

The Spam Wars

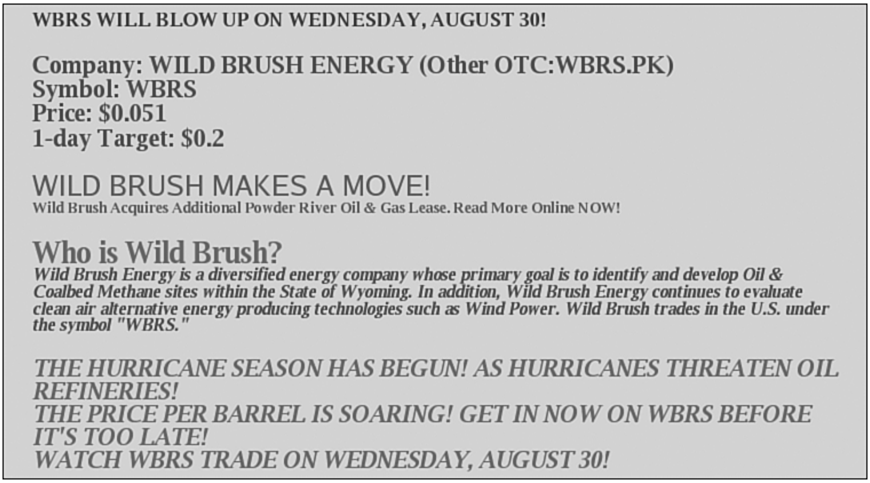
Many of us are used to receiving email pleas such as this one: *I am Miss Faatin Rahman the only child/daughter of late mrs helen rahman Address: Rue 142 Marcory Abidjan Cote d'ivoire west africa, I am 20 years old girl. I lost my parent, and I have an inheritance from my late mother, My parents were very wealthy farmers and cocoa merchant when they were alive, After the death of my father, long ago, my mother was controling his business untill she was poisoned by her business associates which she suffered and died, ... I am crying and seeking for your kind assistance in the following ways: To provide a safe bank account into where the money will be transferred for investment....*

If you get such a request, don't respond to it! Money will flow out of, not into, your bank account. Most people know not to comply. But mass emails are so cheap that getting one person out of a million to respond is enough to make the spammer financially successful.

"Spam filters" are programs that intercept email on its way into the in-box and delete messages like these before we read them. This kind of spam follows such a standard style that it is easy to spot automatically, with minimal risk that any real correspondence with banks or African friends will be filtered out by mistake.

But the spam artists have fought back. Many of us have received emails like the one in Figure 3.14. Why can't the spam filter catch things like this?

Word-processing software includes the name and size of the font in conjunction with the coded characters themselves, as well as other information, such as the color of the letters and the color of the background. Because the underlying text is represented as ASCII codes, however, it remains relatively easy to locate individual letters or substrings, to add or delete text, and to perform other such common text-processing operations. When a user positions a cursor over the letter on the screen, the program can figure out the location within the file of the character over which the cursor is positioned. Computer software can, in turn, render the character codes as images of characters.



WBRS WILL BLOW UP ON WEDNESDAY, AUGUST 30!

Company: WILD BRUSH ENERGY (Other OTC:WBRS.PK)
Symbol: WBRS
Price: $0.051
1-day Target: $0.2

WILD BRUSH MAKES A MOVE!
Wild Brush Acquires Additional Powder River Oil & Gas Lease. Read More Online NOW!

Who is Wild Brush?
Wild Brush Energy is a diversified energy company whose primary goal is to identify and develop Oil & Coalbed Methane sites within the State of Wyoming. In addition, Wild Brush Energy continues to evaluate clean air alternative energy producing technologies such as Wind Power. Wild Brush trades in the U.S. under the symbol "WBRS."

THE HURRICANE SEASON HAS BEGUN! AS HURRICANES THREATEN OIL REFINERIES!
THE PRICE PER BARREL IS SOARING! GET IN NOW ON WBRS BEFORE IT'S TOO LATE!
WATCH WBRS TRADE ON WEDNESDAY, AUGUST 30!

FIGURE 3.14 Graphic spam received by one of the authors. Although it looks like text, the computer "sees" it as just an image, like a photograph. Because it doesn't realize that the pixels are forming letters, its spam filters cannot identify it as spam.

But just because a computer screen shows a recognizable letter of the alphabet, this does not mean that the underlying representation is by means of standard character codes. A digitized photograph of text may well look identical to an image rendered from a word-processing document—that is, the two utterly different representations may give rise to exactly the same image.

And that is one reason why, in the battle between spam producers and makers of spam filters, the spam producers currently have the upper hand. The spam of Figure 3.14 was produced in graphical form, even though what is represented is just text. As the underlying representation is pixels and not ASCII, spam like this makes it through all the filters we know about!

The problem of converting raster graphics to ASCII text is called *character recognition*. The term *optical character recognition*, or OCR, is used when the original document is a printed piece of paper. The raster graphic representation is the result of scanning the document, and then some character recognition algorithm is used to convert the image into a sequence of character codes. If the original document is printed in a standard typeface and is relatively free of smudges and smears, contemporary OCR software is quite accurate, and is now incorporated into commercially available scanners commonly packaged as multipurpose devices that also print, photocopy, and fax. Because OCR algorithms are now reasonably effective and widely available, the next generation of spam filters will likely classify emails such as Figure 3.14 as spam.

OCR and spam are merely an illustration of a larger point. Representation determines what can be done with data. In principle, many representations may be equivalent. But in practice, the secrecy of formatting information and the computation required to convert one format to another may limit the usefulness of the data itself.

Hiding Information in Plain Sight

During World War I, the German Embassy in Washington, DC sent a message to Berlin that began thus: "PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE." U.S. intelligence was reading all the German telegrams, and this one might have seemed innocuous enough. But the first letters of the words spelled out "PERSHING," the name of a U.S. Navy vessel. The entire telegram had nothing to do with embargoes. It was about U.S. ship movements, and the initial letters read in full, "PERSHING SAILS FROM N.Y. JUNE 1."

Steganography is the art of sending secret messages in imperceptible ways. Steganography is different from *cryptography*, which is the art of sending messages that are indecipherable. In a cryptographic communication, it is assumed that if Alice sends a message to Bob, an adversary may well intercept the message and recognize that it holds a secret. The objective is to make the message unreadable, except to Bob, if it falls into the hands of such an eavesdropper or enemy. In the world of electronic communication, sending an encrypted message is likely to arouse suspicion of electronic monitoring software. By contrast, in a steganographic message from Alice to Bob, the communication itself arouses no suspicion. It may even be posted on a web site and seem entirely innocent. Yet hidden in plain sight, in a way known only to Alice and Bob, is a coded message.

Steganography has been in use for a long time. The *Steganographia* of Johannes Trithemius (1462–1516) is an occult text that includes long conjurations of spirits. The first letters of the words of these mystic incantations encode other hidden messages, and the book was influential for a century after it was written. Computers have created enormous opportunities for steganographic communications. As a very simple example, consider an ordinary word-processing document—a simple love letter, for example. Print it out or view it on the screen, and it seems to be about Alice's sweet nothings to Bob, and nothing more. But perhaps Alice included a paragraph at the end *in which she changed the font color to white*. The software renders the white text on the white background, which looks exactly like the white background.

But Bob, if he knows what to look for, can make it visible—for example, by printing on black paper (just as the text could be recovered from the electronically redacted Calipari report).

If an adversary has any reason to think a trick like this might be in use, the adversary can inspect Alice's electronic letter using software that looks for messages hidden using just this technique. But there are many places to look for steganographic messages, and many ways to hide the information.

Since each Roman letter has an eight-bit ASCII code, a text can be hidden within another as long as there is an agreed-upon method for encoding 0s and 1s. For example, what letter is hidden in this sentence?

```
Steganographic algorithms hide messages inside photos, text, and
other data.
```

The answer is "I," the letter whose ASCII character code is 01001001. In the first eight words of the sentence, words beginning with consonants encode 0 bits and words beginning with vowels encode 1s (see Figure 3.15).

```
Steganographic algorithms hide messages inside photos, text, and other data.
0              1          0    0        1      0       0     1
```

FIGURE 3.15 A steganographic encoding of text within text. Initial consonants encode 0, vowels encode 1, and the first eight words encode the 8-bit ASCII code for the letter "I."

A steganographic method that would seem to be all but undetectable involves varying ever so slightly the color values of individual pixels within a photograph. Red, green, and blue components of a color determine the color itself. A color is represented internally as one byte each for red, green, and blue. Each 8-bit string represents a numerical value between 0 and 255. Changing the rightmost bit from a 1 to a 0 (for example, changing 00110011 to 00110010), changes the numerical value by subtracting one—in this case, changing the color value from 51 to 50. That results in a change in color so insignificant that it would not be noticed, certainly not as a change in a single pixel. But the rightmost bits of the color values of pixels in the graphics files representing photographs can then carry quite large amounts of information, without raising any suspicions. The recipient decodes the message not by rendering the bits as visible images, but by inspecting the bits themselves, and picking out the significant 0s and 1s.

Who uses steganography today, if anyone? It is very hard to know. *USA Today* reported that terrorists were communicating using steganography in early 2001. A number of software tools are freely available that make steganography easy. Steganographic detectors—what are properly known as steganalysis tools—have also been developed, but their usefulness as yet seems to be limited. Both steganography and steganalysis software is freely available on the World Wide Web (see, for example, `www.cotse.com/tools/stega.htm` and `www.outguess.org/detection.php`).

The use of steganography to transmit secret messages is today easy, cheap, and all but undetectable. A foreign agent who wanted to communicate with parties abroad might well encode a bit string in the tonal values of an MP3 or the color values of pixels in a pornographic image on a web page. So much music and pornography flows between the U.S. and foreign countries that the uploads and downloads would arouse no suspicion!

The Scary Secrets of Old Disks

By now, you may be tempted to delete all the files on your disk drive and throw it away, rather than run the risk that the files contain unknown secrets. That isn't the solution: Even deleted files hold secrets!

A few years ago, two MIT researchers bought 158 used disk drives, mostly from eBay, and recovered what data they could. Most of those who put the disks up for sale had made some effort to scrub the data. They had dragged files into the desktop trash can. Some had gone so far as to use the Microsoft Windows FORMAT command, which warns that it will destroy all data on the disk.

Yet only 12 of the 158 disk drives had truly been sanitized. Using several methods well within the technical capabilities of today's teenagers, the researchers were able to recover user data from most of the others. From 42 of the disks, they retrieved what appeared to be credit card numbers. One of the drives seemed to have come from an Illinois automatic teller machine and contained 2,868 bank account numbers and account balances. Such data from single business computers would be a treasure trove for criminals. But most of the drives from home computers also contained information that the owners would consider extremely sensitive: love letters, pornography, complaints about a child's cancer therapy, and grievances about pay disputes, for example. Many of the disks contained enough data to identify the primary user of the computer, so that the sensitive information could be tied back to an individual whom the researchers could contact.

Cloud Computing

One way to avoid having problems with deleted disk files and expensive document-processing software is not to keep your files on your disks in the first place! In "cloud computing," the documents stay on the disks of a central service provider and are accessed through a web browser. "Google Docs" is one such service, which boasts very low software costs, but other major software companies are rumored to be exploring the market for cloud computing. If Google holds your documents, they are accessible from anywhere the Internet reaches, and you never have to worry about losing them—Google's backup procedures are better than yours could ever be. But there are potential disadvantages. Google's lawyers would decide whether to resist subpoenas. Federal investigators could inspect bits passing through the U.S., even on a trip between other countries.

The users of the computers had for the most part done what they thought they were supposed to do—they deleted their files or formatted their disks. They probably knew not to release toxic chemicals by dumping their old machines in a landfilll, but they did not realize that by dumping them on eBay, they might be releasing personal information into the digital environment. Anyone in the world could have bought the old disks for a few dollars, and all the data they contained. What is going on here, and is there anything to do about it?

Disks are divided into blocks, which are like the pages of a book—each has an identifying address, like a page number, and is able to hold a few hundred bytes of data, about the same amount as a page of text in a book. If a document is larger than one disk block, however, the document is typically not stored in consecutive disk blocks. Instead, each block includes a piece of the document, and the address of the block where the document is continued. So the entire document may be physically scattered about the disk, although logically it is held together as a chain of references of one block to another. Logically, the structure is that of a magazine, where articles do not necessarily occupy contiguous pages. Part of an article may end with "Continued on page 152," and the part of the article on page 152 may indicate the page on which it is continued from there, and so on.

Because the files on a disk begin at random places on disk, an *index* records which files begin where on the disk. The index is itself another disk file, but one whose location on the disk can be found quickly. A disk index is very much like the index of a book—which always appears at the end, so readers know where to look for it. Having found the index, they can quickly find the page number of any item listed in the index and flip to that page.

Why aren't disks themselves organized like books, with documents laid out on consecutive blocks? Because disks are different from books in two important respects. First, they are dynamic. The information on disks is constantly being altered, augmented, and removed. A disk is less like a book than like a three-ring binder, to which pages are regularly added and removed as information is gathered and discarded. Second, disks are perfectly re-writable. A disk block may contain one string of `0`s and `1`s at one moment, and as a result of a single writing operation, a different string of `0`s and `1`s a moment later. Once a `0` or a `1` has been written in a particular position on the disk, there is no way to tell whether the bit previously in that position was a `0` or a `1`. There is nothing analogous to the faint traces of pencil marks on paper that are left after an erasure. In fact, there is no notion of "erasure" at all on a disk—all that ever happens is replacement of some bits by others.

Because disks are dynamic, there are many advantages to breaking the file into chained, noncontiguous blocks indexed in this way. For example, if the file contains a long text document and a user adds a few words to the middle of the text, only one or two blocks in the middle of the chain are affected. If enough text is added that those blocks must be replaced by five new ones, the new blocks can be logically threaded into the chain without altering any of the other blocks comprising the document. Similarly, if a section of text is deleted, the chain can be altered to "jump over" the blocks containing the deleted text.

Blocks that are no longer part of any file are added to a "pool" of available disk blocks. The computer's software keeps track of all the blocks in the pool. A block can wind up in the pool either because it has never been used or because it has been used but abandoned. A block may be abandoned because the entire file of which it was part has been deleted or because the file has been altered to exclude the block. When a fresh disk block is needed for any purpose—for example, to start a new file or to add to an existing file—a block is drawn from the pool of available blocks.

What Happens to the Data in Deleted Files?

Disk blocks are not re-written when they are abandoned and added to the pool. When the block is withdrawn from the pool and put back to work as part of another file, it is overwritten and the old data is obliterated. But until then, the block retains its old pattern of zeroes and ones. The entire disk file may be intact—except that there is no easy way to find it. A look in the index will reveal nothing. But "deleting" a file in this way merely removes the index entry. The information is still there on the disk somewhere. It has no more been eradicated than the information in a book would be expunged by tearing out the index from the back of the volume. To find something in a book without an index, you just have to go through the book one page at a time looking for it—tedious and time-consuming, but not impossible.

And that is essentially what the MIT researchers did with the disks they bought off eBay—they went through the blocks, one at a time, looking for recognizable bit patterns. A sequence of sixteen ASCII character codes representing decimal digits, for example, looks suspiciously like a credit card number. Even if they were unable to recover an entire file, because some of the blocks comprising it had already been recycled, they could recognize significant short character strings such as account numbers.

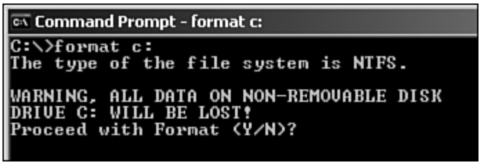
Of course, there would be a simple way to prevent sensitive information from being preserved in fragments of "deleted" files. The computer could be programmed so that, instead of simply putting abandoned blocks into the pool, it actually over-wrote the blocks, perhaps by "zeroing" them—that is, writing a pattern of all 0s. Historically, computer and software manufacturers have thought the benefits of zeroing blocks far less than the costs. Society has not found "data leakage" to be a critical problem until recently—although that may be changing. And the costs of constantly zeroing disk blocks would be significant. Filling blocks with zeroes might take so much time that the users would complain about how slowly their machines were running if every block were zeroed immediately. With some clever programming the process could be made unnoticeable, but so far neither Microsoft nor Apple has made the necessary software investment.

The Law Adjusts

Awareness is increasing that deleted data can be recovered from disks. The Federal Trade Commission now requires "the destruction or erasure of electronic media containing consumer information so that the information cannot practicably be read or reconstructed," and a similar provision is in a 2007 Massachusetts Law about security breaches.

And who has not deleted a file and then immediately wished to recover it? Happily for all of us who have mistakenly dragged the wrong file into the trash can, as computers work today, deleted files are not immediately added to the pool—they can be dragged back out. Files can be removed only until you execute an "Empty trash" command, which puts the deleted blocks into the pool, although it does not zero them.

But what about the Windows "FORMAT" command, shown in Figure 3.16? It takes about 20 minutes to complete. Apparently it is destroying all the bits on the disk, as the warning message implies. But that is not what is happening. It is simply looking for faulty spots on the disk. Physical flaws in the magnetic surface can make individual disk blocks unusable, even though mechanically the disk is fine and most of the surface is flawless as well. The FORMAT command attempts to *read* every disk block in order to identify blocks that need to be avoided in the future. Reading every block takes a long time, but rewriting them all would take twice as long. The FORMAT command identifies the bad blocks and re-initializes the index, but leaves most of the data unaltered, ready to be recovered by an academic researcher—or an inventive snooper.



```
C:\>format c:
The type of the file system is NTFS.

WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Proceed with Format (Y/N)?
```

FIGURE 3.16 Warning screen of Microsoft Windows FORMAT command. The statement that all the data will be lost is misleading—in fact, a great deal of it can be recovered.

As if the problems with disks were not troubling enough, exactly the same problems afflict the memory of cell phones. When people get rid of their old phones, they forget the call logs and email messages they contain. And if they do remember to delete them, using the awkward combinations of button-pushes described deep in the phone's documentation, they may not really have accomplished what they hoped. A researcher bought ten cell phones on eBay and recovered bank account numbers and passwords, corporate strategy plans, and an email exchange between a woman and her married boyfriend, whose wife was getting suspicious. Some of this information was recovered from phones whose previous owners had scrupulously followed the manufacturer's instructions for clearing the memory.

SOFTWARE TO SCRUB YOUR DISK

If you really want to get rid of all the data on your disk, a special "Secure empty trash" command is available on Macintosh computers. On Windows machines, DBAN is free software that really will zero your disk, available through `dban.sourceforge.net`, which has lots of other useful free software. Don't use DBAN on your disk until you are sure you don't want anything on it anymore!

In a global sense, bits turn out to be very hard to eradicate. And most of the time, that is exactly the way we want it. If our computer dies, we are glad that Google has copies of our data. When our cell phone dies, we are happy if our contact lists reappear, magically downloaded from our cellular service provider to our replacement phone. There are upsides and downsides to the persistence of bits.

Physical destruction always works as a method of data deletion. One of us uses a hammer; another of us prefers his axe. Alas, these methods, while effective, do not meet contemporary standards for recovery and recycling of potentially toxic materials.

Can Data Be Deleted Permanently?

Copies Make Data Hard to Delete

If your computer has ever been connected to a network, destroying its data will not get rid of copies of the same information that may exist on other machines. Your emails went to and from other people—who may have copies on their machines, and may have shared them with others. If you use Google's Gmail, Google may have copies of your emails even after you have deleted them. If you ordered some merchandise online, destroying the copy of the invoice on your personal computer certainly won't affect the store's records.

Rumors arise every now and then that engineers equipped with very sensitive devices can tell the difference between a 0 that was written over a 0 on a disk and a 0 that was written over a 1. The theory goes that successive writing operations are not perfectly aligned in physical space—a "bit" has width. When a bit is rewritten, its physical edges may slightly overlap or fall short of its previous position, potentially revealing the previous value. If such microscopic misalignments could be detected, it would be possible to see, even on a disk that has been zeroed, what the bits were *before* it was zeroed.

No credible authentication of such an achievement has ever been published, however, and as the density of hard disks continues to rise, the likelihood wanes that such data recovery can be accomplished. On the other hand, the places most likely to be able to achieve this feat are government intelligence agencies, which do not boast of their successes! So all that can be said for certain is that recovering overwritten data is within the capabilities of at most a handful of organizations—and if possible at all, is so difficult and costly that the data would have to be extraordinarily valuable to make the recovery attempt worthwhile.

How Long Will Data Really Last?

As persistent as digital information seems to be, and as likely to disclose secrets unexpectedly, it also suffers from exactly the opposite problem. Sometimes electronic records become unavailable quite quickly, in spite of best efforts to save them permanently.

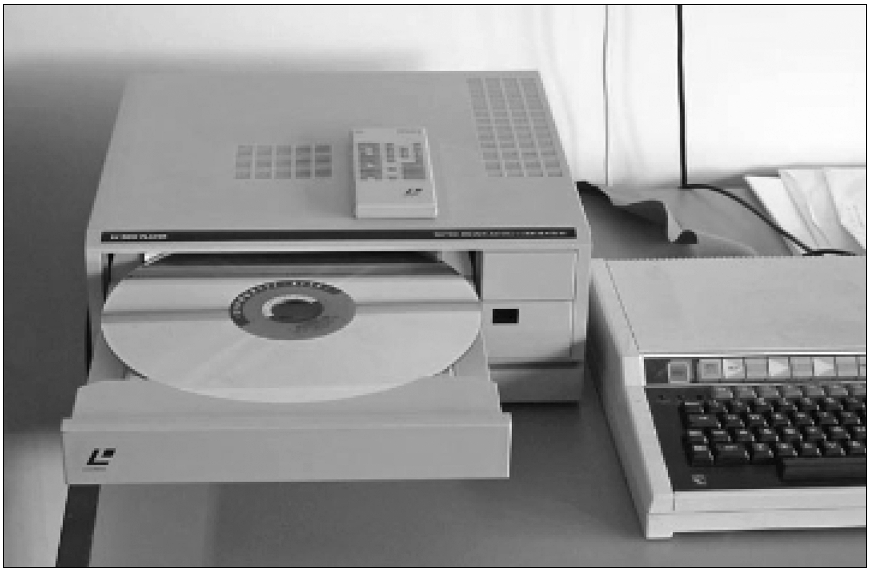
Figure 3.17 shows an early geopolitical and demographic database—the Domesday Book, an inventory of English lands compiled in 1086 by Norman monks at the behest of William the Conqueror. The Domesday Book is one of Britain's national treasures and rests in its archives, as readable today as it was in the eleventh century.

British National Archives.

FIGURE 3.17 The Domesday Book of 1086.

In honor of the 900th anniversary of the Domesday Book, the BBC issued a modern version, including photographs, text, and maps documenting how Britain looked in 1986. Instead of using vellum, or even paper, the material was assembled in digital formats and issued on 12-inch diameter video disks, which could be read only by specially equipped computers (see Figure 3.18). The project was meant to preserve forever a detailed snapshot of late twentieth-century Britain, and to make it available immediately to schools and libraries everywhere.

By 2001, the modern Domesday Book was unreadable. The computers and disk readers it required were obsolete and no longer manufactured. In 15 years, the memory even of how the information was formatted on the disks had been forgotten. Mocking the project's grand ambitions, a British newspaper exclaimed, "Digital Domesday Book lasts 15 years not 1000."



“Domesday Redux,” from Ariadne, Issue 56.

FIGURE 3.18 A personal computer of the mid-1980s configured to read the 12-inch videodisks on which the modern “Domesday Book” was published.

Paper and papyrus thousands of years older even than the original Domesday Book are readable today. Electronic records become obsolete in a matter of years. Will the vast amounts of information now available because of the advances in storage and communication technology actually be usable a hundred or a thousand years in the future, or will the shift from paper to digital media mean the loss of history?

The particular story of the modern Domesday Book has a happy ending. The data was recovered, though just barely, thanks to a concerted effort by many technicians. Reconstructing the data formats required detective work on masses of computer codes (see Figure 3.19) and recourse to data structure books of the period—so that programmers in 2001 could imagine how others would have attacked the same data representation problems only 15 years earlier! In the world of computer science, “state of the art” expertise dies very quickly.

The recovered modern Domesday Book is accessible to anyone via the Internet. Even the data files of the original Domesday Book have been transferred to a web site that is accessible via the Internet.

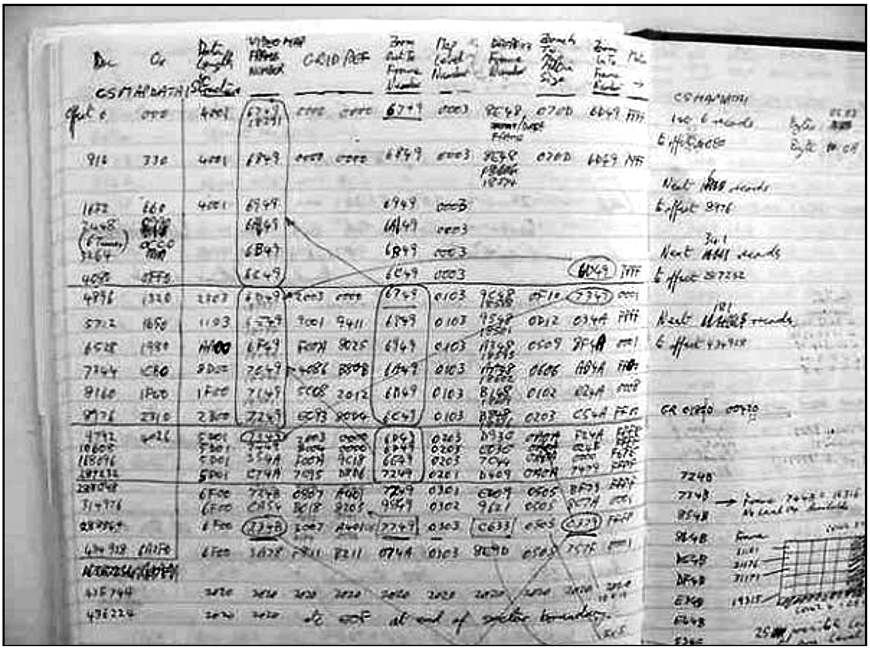


FIGURE 3.19 Efforts to reconstruct, shortly after the year 2000, the forgotten data formats for the modern "Domesday Book," designed less than 20 years earlier.

But there is a large moral for any office or library worker. We cannot assume that the back-ups and saved disks we create today will be useful even ten years from now for retrieving the vast quantities of information they contain. It is an open question whether digital archives—much less the box of disk drives under your bed in place of your grandmother's box of photographs—will be as permanent as the original Domesday Book. An extraordinary effort is underway to archive the entire World Wide Web, taking snapshots of every publicly accessible web page at period intervals. Can the effort succeed, and can the disks on which the archive is held

PRESERVING THE WEB

The Internet Archive (`www.archive.org`) periodically records "snapshots" of publicly accessible web pages and stores them away. Anyone can retrieve a page from the past, even if it no longer exists or has been altered. By installing a "Wayback" button (available from the Internet Archive) on your web browser, you can instantly see how any web page looked in the past—just go to the web page and click the Wayback button; you get a list of the archived copies of the page, and you can click on any of them to view it.

themselves be updated periodically so that the information will be with us forever?

Or would we be wisest to do the apparently Luddite thing: to print everything worth preserving for the long run—electronic journals, for example—so that documents will be preserved in the only form we are *certain* will remain readable for thousands of years?

✷

The digital revolution put the power to document ideas into the hands of ordinary people. The technology shift eliminated many of the intermediaries once needed to produce office memoranda and books. Power over the thoughts in those documents shifted as well. The authority that once accompanied the physical control of written and printed works has passed into the hands of the individuals who write them. The production of information has been democratized—although not always with happy results, as the mishaps discussed in this chapter tellingly illustrate.

We now turn to the other half of the story: how we get the information that others have produced. When power over documents was more centralized, the authorities were those who could print books, those who had the keys to the file cabinets, and those with the most complete collections of documents and publications. Document collections were used both as information choke points and as instruments of public enlightenment. Libraries, for example, have been monuments to imperial power. University libraries have long been the central institutions of advanced learning, and local public libraries have been key democratizing forces in literate nations.

If everything is just bits and everyone can have as many bits as they want, the problem may not be having the information, but finding it. Having a fact on the disk in your computer, sitting a few inches from your eyes and brain, is irrelevant, if what you want to know is irretrievably mixed with billions of billions of other bits. Having the haystack does you no good if you can't find your precious needle within it. In the next chapter, we ask: Where does the power now go, in the new world where access to information means finding it, as well as having it?