

Topic: 1.1.3 Data storage

In a computer, storage is the place where data is held in an electromagnetic or optical form for access by a computer processor. There are two general usages:

- 1) Storage is frequently used to mean the devices and data connected to the computer through input/output operations - that is, hard disk and tape systems and other forms of storage that don't include computer memory and other in-computer storage. For the enterprise, the options for this kind of storage are of much greater variety and expense than that related to memory. This meaning is probably more common in the IT industry than meaning than the second meaning.
- 2) In a more formal usage, storage has been divided into: (1) primary storage, which holds data in memory (sometimes called random access memory or **RAM**) and other "built-in" devices such as the processor's L1 cache (2) secondary storage, which holds data on hard disks, tapes, and other devices requiring input/output operations.

Primary storage is much faster to access than secondary storage because of the proximity of the storage to the processor or because of the nature of the storage devices. On the other hand, secondary storage can hold much more data than primary storage.

In addition to RAM, primary storage includes read-only memory (ROM) and L1 and L2 cache memory. In addition to hard disks, secondary storage includes a range of device types and technologies, including diskettes, Zip drives, redundant array of independent disks (RAID) systems, and holographic storage. Devices that hold storage are collectively known as storage devices.

A somewhat antiquated term for primary storage is *main storage* and a somewhat antiquated term for secondary storage is *auxiliary storage*. Note that, to add to the confusion, there is an additional meaning for primary storage that distinguishes actively used storage from backup storage.

FORMATS FOR STORAGE OF DATA:

The following are the various types of data that you might find:

TEXT

Text can be represented easily by assigning a unique numeric value for each symbol used in the text. For example, the widely used ASCII code (American Standard Code for Information Interchange) defines 128 different symbols (all the characters found on a standard keyboard, plus a few extra), and assigns to each a unique numeric code between 0 and 127. In ASCII, an "A" is 65, "B" is 66, "a" is 97, "b" is 98, and so forth. When you save a file as "plain text", it is stored using ASCII. ASCII format uses 1 byte per character 1 byte gives only 256 (128 standard and 128 non-standard) possible characters. The code value for any character can be converted to base 2, so any written message made up of ASCII characters can be converted to a string of 0's and 1's.

GRAPHICS

Graphics that are displayed on a computer screen consist of pixels: the tiny "dots" of color that collectively "paint" a graphic image on a computer screen. The pixels are organized into many rows on the screen. In one common





Topic: 1.1.3 Data storage

configuration, each row is 640 pixels long, and there are 480 such rows. Another configuration is 800 pixels per row with 600 rows, which is referred to as a "resolution of 800x600." Each pixel has two properties: its location on the screen and its color.

A graphic image can be represented by a list of pixels. Imagine all the rows of pixels on the screen laid out end to end in one long row. This gives the pixel list, and a pixel's location in the list corresponds to its position on the screen. A pixel's color is represented by a binary code, and consists of a certain number of bits. In a monochrome (black and white) image, only 1 bit is needed per pixel: 0 for black, 1 for white, for example. A 16 bit color image requires 4 bits per pixel. Modern display hardware allows for 24 bits per pixel, which provides an astounding array of 16.7 million possible colors for each pixel!

COMPRESSION

Files today are so information-rich that their size may become very large. This is particularly true of graphics files. With so many pixels in the list, and so many bits per pixel, a graphic file can easily take up over a megabyte of storage. Files containing large software applications can require 50 megabytes or more! This causes two problems: it becomes costly to store the files (requires many floppy disks or excessive room on a hard drive), and it becomes costly to transmit these files over networks and phone lines because the transmission takes a long time. In addition to studying how various types of data are represented, you will have the opportunity today to look at a technique known as data compression. The basic idea of compression is to make a file shorter by removing redundancies (repeated patterns of bits) from it. This shortened file must of course be de-compressed - have its redundancies put back in - in order to be used. However, it can be stored or transmitted in its shorter compressed form, saving both time and money.

ANIMATION

Somewhere between the motionless world of still images and the real-time world of video images lies the flip-book world of computer animation. All of the animated sequences seen in educational programs, motion CAD renderings, and computer games are computer-animated (and in many cases, computer-generated) animation sequences.

Traditional cartoon animation is little more than a series of artwork cells, each containing a slight positional variation of the animated subjects. When a large number of these cells are displayed in sequence and at a fast rate, the animated figures appear to the human eye as if they are moving.

A computer-animated sequence works in exactly the same manner. A series of images are created of a subject; each image contains a slightly different perspective on the animated subject.

When these images are displayed (played back) in the proper sequence and at the proper speed (frame rate), the subject appears to move.

Computerized animation is actually a combination of both still and motion imaging. Each frame, or cell, of an animation is a still image that requires compression and storage. An animation file, however, must store the data for hundreds or thousands of animation frames and must also provide the information necessary to play back the frames using the proper display mode and frame rate.

DIGITAL VIDEO:

One step beyond animation is broadcast video. Your television and video tape recorder are a lot more complex than an 8mm home movie projector and your kitchen wall. There are many complex signals and complicated standards that





Topic: 1.1.3 Data storage

are involved in transmitting those late-night reruns across the airwaves and cable. Only in the last few years has a personal computer been able to work with video data at all.

Video data normally occurs as continuous, analog signals. In order for a computer to process this video data, we must convert the analog signals to a non-continuous, digital format. In a digital format, the video data can be stored as a series of bits on a hard disk or in computer memory.

The process of converting a video signal to a digital bit stream is called analog-to-digital conversion (A/D conversion), or digitizing. A/D conversion occurs in two steps:

1. **Sampling** captures data from the video stream.
2. **Quantizing** converts each captured sample into a digital format.

Each sample captured from the video stream is typically stored as a 16-bit integer. The rate at which samples are collected is called the *sampling rate*. The sampling rate is measured in the number of samples captured per second (samples/second). For digital video, it is necessary to capture millions of samples per second.

DIGITAL AUDIO

All multimedia file formats are capable, by definition, of storing sound information. Sound data, like graphics and video data, has its own special requirements when it is being read, written, interpreted, and compressed. Before looking at how sound is stored in a multimedia format we must look at how sound itself is stored as digital data.

All of the sounds that we hear occur in the form of analog signals. An analog audio recording system, such as a conventional tape recorder, captures the entire sound wave form and stores it in analog format on a medium such as magnetic tape.

Because computers are now digital devices it is necessary to store sound information in a digitized format that computers can readily use. A digital audio recording system does not record the entire wave form as analog systems do (the exception being Digital Audio Tape [DAT] systems). Instead, a digital recorder captures a wave form at specific intervals, called the sampling rate. Each captured wave-form snapshot is converted to a binary integer value and is then stored on magnetic tape or disk.



Topic: 1.1.3 Data storage

MIDI STANDARD

Musical Instrument Digital Interface (MIDI) is an industry standard for representing sound in a binary format. MIDI is not an audio format, however. It does not store actual digitally sampled sounds. Instead, MIDI stores a description of sounds, in much the same way that a vector image format stores a description of an image and not image data itself.

Sound in MIDI data is stored as a series of control messages. Each message describes a sound event using terms such as pitch, duration, and volume. When these control messages are sent to a MIDI-compatible device (the MIDI standard also defines the interconnecting hardware used by MIDI devices and the communications protocol used to interchange the control information) the information in the message is interpreted and reproduced by the device.

MIDI data may be compressed, just like any other binary data, and does not require special compression algorithms in the way that audio data does.

ERROR DETECTION TECHNIQUES:

Errors introduced by communications faults, noise or other failures into valid data, especially compressed data where redundancy has been removed as much as possible, can be detected and/or corrected by introducing redundancy into the data stream.

Error detection and correction or **error controls** are measures to ensure consistent delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data.

Safety in Redundancy:

More redundancy detects more errors, at the cost of more data transmitted. We could simply send each message 3 times, and discard any copy that disagrees with the other two. This is a simple example of a "perfect" code, although it is far from perfect, it is called that because it adds exactly enough redundancy to detect or correct some number of errors. In this case, not that many. Note that although it is unlikely, it IS possible that the two identical copies both happened to have the exact same error, and the third copy is the correct one. If each copy arrived different in some way, we might have detected two errors. Error detection and correction systems are rated by how much redundancy they cost, and how many errors they can detect or correct. In this example three "symbols" are used, each is the length of the original message. It can detect two errors, and correct one. We can do much better.





Topic: 1.1.3 Data storage

CHECKSUM:

A *checksum* of a message is a modular arithmetic sum of message code words of a fixed word length (e.g., byte values). The sum may be negated by means of a “ones'-complement” operation prior to transmission to detect errors resulting in all-zero messages.

Checksum schemes include parity bits, check digits, and longitudinal redundancy checks. Some checksum schemes, such as the “Damm algorithm”, the “Lund algorithm”, and the “Verhoeff algorithm”, are specifically designed to detect errors commonly introduced by humans in writing down or remembering identification numbers.

A checksum is determined in one of two ways. Let's say the checksum of a packet is 1 [byte](#) long. A byte is made up of 8 bits, and each bit can be in one of two states, leading to a total of 256 possible combinations. Since the first combination equals zero, a byte can have a maximum value of 255.

- ☛ If the sum of the other bytes in the packet is 255 or less, then the checksum contains that exact value.
- ☛ If the sum of the other bytes is more than 255, then the checksum is the remainder of the total value after it has been divided by 256.

Let's look at a checksum example:

- ☛ **Bytes total 1,151**
- ☛ **$1,151 / 256 = 4.496$ (round to 4)**
- ☛ **$4 \times 256 = 1,024$**
- ☛ **$1,151 - 1,024 = 127$ checksum**

PARITY:

For example, if we send some specific sequence of ones and zeros, and then count the number of ones that we sent and send an extra 1 if that count is odd or an extra 0 if that count is even, then we have introduced a small amount of redundancy into the transmission. The extra bit is called the “parity bit”, it is even parity because it makes the total number of 1's in the transmission become an even number, and is a simple example of what is called an “extended” code. The receiver can then count up the number of 1 bit they received, perform the same calculation, and if the result does not match the extra bit we sent them, they will know that an error occurred. If by chance, two errors occur in the byte being transmitted for example “00001111” changes to “00000011” the parity bit will still read this as a one which is theoretically correct, but the data is different, this error won't be detected by even parity.

The sender while creating a frame counts the number of 1s in it, for example, if even parity is used and number of 1s is even then one bit with value 0 is added. This way number of 1s remains even. Or if the number of 1s is odd, to make it even a bit with value 1 is added.





Topic: 1.1.3 Data storage

The receiver simply counts the number of 1s in a frame. If the count of 1s is even and even parity is used, the frame is considered to be not-corrupted and is accepted. If the count of 1s is odd and odd parity is used, the frame is not corrupted and accepted.

CHECK DIGIT:

Number added to a code (such as a bar code or account number) to derive a further number as a means of verifying the accuracy or validity of the code as it is printed or transmitted. A code consisting of three digits, for example, such as 135 may include 9 (sum of 1, 3, and 5) as the last digit and be communicated as 1359.

Check digits can identify 3 types of error:

- (1) If 2 digits have been inverted e.g. “23459” instead of “23549”
- (2) An incorrect digit entered e.g. 23559 instead of 23549
- (3) A digit missed out altogether e.g. 2359 instead of 23549

Now you will learn how check digits are calculated. The ISBN-10 (used on books) has been chosen as the example; this uses a module 11 system which includes the letter X to represent the number 10.

Example 1

To calculate the check digit for the ISBN “0 - 2 0 1 - 5 3 0 8 2 - ?”

- (i) the position of each digit is considered:

“10 9 8 7 6 5 4 3 2 1” ← digit position
0 - 2 0 1 - 5 3 0 8 2 - ? ← Number

- (ii) Each digit is then multiplied by its digit position and the totals added together

$$(0 \times 10) + (2 \times 9) + (0 \times 8) + (1 \times 7) + (5 \times 6) + (3 \times 5) + (0 \times 4) + (8 \times 3) + (2 \times 2) \\ = 0 + 18 + 0 + 7 + 30 + 15 + 0 + 24 + 4 \\ = 98$$

- (iii) The total is then divided by 11 (modulo – 11) and the remainder, if any, is subtracted from 11 to give the check digit.

$$98 \div 11 = 8 \text{ remainder } 10$$

$$11 - 10 = 1$$

This gives a check digit of “1”

Final ISBN becomes “0 - 2 0 1 - 5 3 0 8 2 - 1”





Topic: 1.1.3 Data storage

Example 2

To check the correctness of a check digit the computer re-calculates it as follows:

The ISBN to check is: 0 - 1 3 1 5 - 2 4 4 7 - X

(i) The position of each digit is considered:

10	9	8	7	6	5	4	3	2	1	← digit position
0	-1	3	1	-5	2	4	4	7	-X	← number

(ii) Each digit is then multiplied by its digit position and the totals added together

$$\begin{aligned} & (0 \times 10) + (1 \times 9) + (3 \times 8) + (1 \times 7) + (5 \times 6) + (2 \times 5) + (4 \times 4) + (4 \times 3) + (7 \times 2) + (X \times 1) \\ & = 0 + 9 + 24 + 7 + 30 + 10 + 16 + 12 + 14 + 10 \text{ (recall that } X = 10\text{)} \\ & = 132 \end{aligned}$$

(iii) The total is then divided by 11; if there is no remainder then the check digit is correct:

$$132 \div 11 = 12 \text{ remainder } 0$$

Hence the check digit is correct

ERROR CORRECTION:

Error correction may generally be realized in two different ways:

ARQ

ARQ is also called **Automatic repeat request** which is an error control (error correction) method that uses error-detection codes and positive and negative acknowledgments. When the transmitter either receives a negative acknowledgment or a timeout happens before acknowledgment is received, the ARQ makes the transmitter resend the message.





Topic: 1.1.3 Data storage

FILE TYPES:

Multimedia formats, however, are much more complex than most other file formats because of the wide variety of data they must store. Such data includes text, image data, audio and video data, computer animations, and other forms of binary data, such as Musical Instrument Digital Interface (MIDI), control information, and graphical fonts. Typical multimedia formats do not define new methods for storing these types of data. Instead, they offer the ability to store data in one or more existing data formats that are already in general use.

For example, a multimedia format may allow text to be stored as PostScript or Rich Text Format (RTF) data rather than in conventional ASCII plain-text format. Still-image bitmap data may be stored as BMP or TIFF files rather than as raw bitmaps. Similarly, audio, video, and animation data can be stored using industry-recognized formats specified as being supported by that multimedia file format

MIDI

The MIDI file format is used to store MIDI song data on disk. The discussed version of the MIDI file spec is the approved MIDI Manufacturers' Associations format version 0.06 of (3/88). The contact address is listed in the addresses file. Version 1.0 is technically identical but the description has been rewritten. The description was made by Dave Oppenheim, most of the text was taken right out of his document.

MIDI files contain one or more MIDI streams, with time information for each event. Song, sequence, and track structures, tempo and time signature information, are all supported. Track names and other descriptive information may be stored with the MIDI data. This format supports multiple tracks and multiple sequences so that if the user of a program which supports multiple tracks intends to move a file to another one, this format can allow that to happen.

The MIDI files are block oriented files, currently only 2 block types are defined, header and track data. Opposed to the IFF and RIFF formats, no global header is given, so that the validation must be done by adding the different block sizes.

MP3

Filename Extension: .mp3

Format Type: Lossy Compressed

When Internet file-sharing boomed into popularity with Napster and the iPod, the MP3 cornered the market for one reason: it had a small footprint. Without broadband connections, it was impractical at the time to share file sizes larger than the MP3 standard 2 – 3 Megabytes.

And that preference has stuck for some time now even though MP3 does not have nearly the same amount of quality as WAV or AIFF files. But despite this growing base of people using higher quality formats, [there are still those who prefer the MP3](#) – whether out of nostalgia or quality, who knows.

What does this mean for you? Well, the MP3 format uses compression which actually removes data from a song using complicated algorithms. The reason for removing this data is to save space and make the file smaller.

So, if you have a slower internet connection or limited hard drive space, MP3 could be your file format of choice. If





Topic: 1.1.3 Data storage

you're worried about quality loss, don't fret too much about it. While, yes, there is a noticeable drop off in sound quality, MP3 files fall square under the "good enough" umbrella.

JPEG

JPG files, also known as JPEG files, are a common file format for digital photos and other digital graphics. When JPG files are saved, they use "lossy" compression, meaning image quality is lost as file size decreases. JPEG stands for Joint Photographic Experts Group, the committee that created the file type.

JPG files have the file extension .jpg or .jpeg. They are the most common file type for images taken with digital cameras, and widely used for photos and other graphics used on websites.

Unlike GIF files, which show significant loss in photo image quality, JPGs allow for some degree of file size reduction without losing too much image quality. However, as file sizes get very low, JPG images will become "muddy." When saving photos and other images as JPG files for the web, email and other uses, you must decide on this compromise between quality and file size.

MP4

MP4 is an abbreviated term for **MPEG-4 Part 14**. It may also be referred to as **MPEG-4 AVC**, which stands for **Advanced Video Coding**. As the name suggests, this is a format for working with video files and was first introduced in 1998. The **MPEG** refers to Motion Pictures Expert Group who is responsible for setting the industry standards regarding digital audio and video.

The MP4 is a container format, allowing a combination of audio, video, subtitles and still images to be held in the one single file. It also allows for advanced content such as 3D graphics, menus and user interactivity.

Because MP4 was a reliable application that required a relatively low amount of bandwidth, just about everyone could take advantage of using the tool. This was especially true as technology made it possible to create more powerful desktop and laptop systems that had a larger hard drive and could command more power.

The enhancement of the speed of various types of Internet connections also helped to make MP4 **more accessible** to a greater audience. MP4 works in a similar although much more complex way to MP3s, by compressing the files without losing any of the quality. MP3 technology revolutionized the way in which music and audio files are used and it's looking like the MP4 format will do the same for the video market.





Topic: 1.1.3 Data storage

DATA COMPRESSION

Storing data in a format that requires less space than usual.

Data compression is particularly useful in communications because it enables [devices](#) to transmit or store the same amount of data in fewer [bits](#). There are a variety of data compression techniques, but only a few have been standardized. The [CCITT](#) has defined a [standard](#) data compression technique for transmitting faxes ([Group 3](#) standard) and a compression standard for data communications through [modems](#) (CCITT V.42bis). In addition, there are [file](#) compression formats, such as [ARC](#) and [ZIP](#).

Data compression is also widely used in [backup](#) utilities, spreadsheet, and [database management systems](#). Certain types of data, such as [bit-mapped graphics](#), can be compressed to a small fraction of their normal size.

LOSSY COMPRESSION

Lossy compression refers to discarding irrelevant information. Generally this means compressing images, video, or audio by discarding data that the human perceptual system cannot see or hear.

Lossy compression is a hard AI problem. To illustrate, speech could theoretically be compressed by transcribing it into text and compressing it with standard techniques to about 10 bits per second. We are nowhere near that!

Even worse, we could imagine a lossy video compressor translating a movie into a script, and the decompressor reading the script and creating a new movie with different details but close enough so that the average person watching both movies one after the other would not notice any differences. We may use a result by Landauer (1986) to estimate just how tiny this script could be. He tested people's memory (over a period of days) over a wide range of formats such as words, numbers, pictures and music, and concluded that the human brain writes to long term memory at a fairly constant rate of about 2 bits per second. Currently we need 10^7 bits per second to store DVD quality MPEG-2 video.

The state of the art is to apply lossy compression only at a very low level of human sensory modeling, where the model is well understood.

IMAGE COMPRESSION

All image formats, even BMP, may be regarded as a form of lossy image compression. An uncompressed image is normally a 2 dimensional array of pixels, where each pixel has 3 color components (red, green, blue) represented as an integer with a fixed range and resolution. A pixel array is an approximation of a 2 dimensional continuous field where the light intensity at any point would be properly described as a continuous spectrum. Note how lossy compression is applied:

The eye can't see detail much smaller than 0.1 mm, so there is no need for an image to have more than a few thousand pixels in each dimension.

The eye can't detect differences in brightness of less than about 1%, so there is no need to quantize brightness to more than a few hundred levels.

The eye has 3 types of cones sensitive to red, green, and blue. Combinations of these colors can reproduce every color that the eye can see. There is no need to distinguish pure spectral yellow emitted by a rainbow from the apparent





Topic: 1.1.3 Data storage

yellow from a monitor produced from a mixture of red and green light, even though there are instruments such as a spectrograph that can make such distinctions.

The eye detects brightness on a logarithmic scale, so there is no need to use more bits to represent brighter lights. Sunlight is 1000 times brighter than room light, but doesn't look like it.

THE FILE TYPES

TIFF is, in principle, a very flexible format that can be lossless or lossy. The details of the image storage algorithm are included as part of the file. In practice, TIFF is used almost exclusively as a lossless image storage format that uses no compression at all. Most graphics programs that use TIFF do not use compression. Consequently, file sizes are quite big. (Sometimes a lossless compression algorithm called LZW is used, but it is not universally supported.)

PNG is also a lossless storage format. However, in contrast with common TIFF usage, it looks for patterns in the image that it can use to compress file size. The compression is exactly reversible, so the image is recovered exactly.

GIF creates a table of up to 256 colors from a pool of 16 million. If the image has fewer than 256 colors, GIF can render the image exactly. When the image contains many colors, software that creates the GIF uses any of several algorithms to approximate the colors in the image with the limited palette of 256 colors available. Better algorithms search the image to find an optimum set of 256 colors. Sometimes GIF uses the nearest color to represent each pixel, and sometimes it uses "error diffusion" to adjust the color of nearby pixels to correct for the error in each pixel.

GIF achieves compression in two ways. First, it reduces the number of colors of color-rich images, thereby reducing the number of bits needed per pixel, as just described. Second, it replaces commonly occurring patterns (especially large areas of uniform color) with a short abbreviation: instead of storing "white, white, white, white, white," it stores "5 white."

Thus, GIF is "lossless" only for images with 256 colors or less. For a rich, true color image, GIF may "lose" 99.998% of the colors.

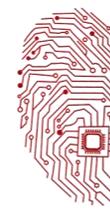
JPG is optimized for photographs and similar continuous tone images that contain many, many colors. It can achieve astounding compression ratios even while maintaining very high image quality. GIF compression is unkind to such images. JPG works by analyzing images and discarding kinds of information that the eye is least likely to notice. It stores information as 24 bit color. **Important: The degree of compression of JPG is adjustable.** At moderate compression levels of photographic images, it is very difficult for the eye to discern any difference from the original, even at extreme magnification. Compression factors of more than 20 are often quite acceptable. Better graphics programs, such as Paint Shop Pro and Photoshop, allow you to view the image quality and file size as a function of compression level, so that you can conveniently choose the balance between qualities and file size.

RAW is an image output option available on some digital cameras. Though lossless, it is a factor of three or four smaller than TIFF files of the same image. The disadvantage is that there is a different RAW format for each manufacturer, and so you may have to use the manufacturer's software to view the images. (Some graphics applications can read some manufacturer's RAW formats.)

BMP is an uncompressed proprietary format invented by Microsoft. There is really no reason to ever use this format.

PSD, PSP, etc., are proprietary formats used by graphics programs. Photoshop's files have the PSD extension, while





Topic: 1.1.3 Data storage

Paint Shop Pro files use PSP. These are the preferred working formats as you edit images in the software, because only the proprietary formats retain all the editing power of the programs. These packages use layers, for example, to build complex images, and layer information may be lost in the nonproprietary formats such as TIFF and JPG. However, be sure to save your end result as a standard TIFF or JPG, or you may not be able to view it in a few years when your software has changed.

Currently, GIF and JPG are the formats used for nearly all web images. PNG is supported by most of the latest generation browsers. TIFF is not widely supported by web browsers, and should be avoided for web use. PNG does everything GIF does, and better, so expect to see PNG replace GIF in the future. PNG will not replace JPG, since JPG is capable of much greater compression of photographic images, even when set for quite minimal loss of quality.

LOSSLESS COMPRESSION

Lossless data compression is a class of [data compression algorithms](#) that allows the original data to be perfectly reconstructed from the compressed data. By contrast, [lossy data compression](#), permits reconstruction only of an approximation of the original data, though this usually allows for improved [compression rates](#) (and therefore smaller sized files).

Lossless data compression is used in many applications. For example, it is used in the [ZIP](#) file format and in the [GNU](#) tool [gzip](#). It is also often used as a component within lossy data compression technologies (e.g. lossless [mid/side joint stereo](#) preprocessing by the [LAME MP3](#) encoder and other lossy audio encoders).

Lossless compression is used in cases where it is important that the original and the decompressed data be identical, or where deviations from the original data could be deleterious. Typical examples are executable programs, text documents, and source code. Some image file formats, like [PNG](#) or [GIF](#), use only lossless compression, while others like [TIFF](#) and [MNG](#) may use either lossless or lossy methods. [Lossless audio](#) formats are most often used for archiving or production purposes, while smaller [lossy audio](#) files are typically used on portable players and in other cases where storage space is limited or exact replication of the audio is unnecessary.

