A Streaming Media Primer

Enabling real-time or on-demand access to audio, video, and multimedia content via the Internet or an intranet.

from the Adobe
Dynamic Media Group

> *"…studies have shown that visitors to sites with streaming media remain up to twice as long as they do at sites without media content…*
>
> *"…interest by enterprises is changing rapidly as they begin to realize that utilizing streaming media can increase the reach, impact, timeliness, and cost-effectiveness of key communications with employees, customers, partners, and suppliers."*
>
> ***—The Aberdeen Group[1]***

Introduction

Streaming media adds engaging motion and sound to the Web experience, increasing site stickiness, interactivity, and retention. Streaming allows timely, dynamic content to be seen by a larger, even global audience, helping to cost-effectively disseminate information, to address new markets, and to bring your corporate culture closer to far-reaching constituencies. Streaming in its truest form, can help to protect video content from being "pirated" and misused.

Streaming media is no longer merely a promise. Streaming is here today.

This Primer won't tell you everything about this rapidly emerging technology, but it will give you an overview of the opportunities and the pitfalls, the costs, and the basics. You'll find out what makes streaming different, and how you can use compelling streaming media content to make your Web sites more dynamic. At the *very* least, you'll learn enough to be an informed member of a work group planning and executing streaming implementations. But, even if you are a complete beginner, you'll find out how easy it can be to edit, encode, and integrate streaming video into your own Web site. And, if you are already creating video productions, this Primer will introduce you to the state-of-the-art streaming media technologies you can use to extend your content to the Web, and to confidently share your productions online.

Contents

1 *http://www.microsoft.com/windows/windowsmedia/en/archive/casestudies/hewlett/default.asp*, "Using Streaming Media for Competitive Advantage," Aberdeen Group, Inc., December 2000

WHAT IS STREAMING MEDIA?

Streaming media enables real-time or on-demand access to audio, video, and multimedia content via the Internet or an intranet.

Streaming media is transmitted by a specialized media server application, and is processed and played back by a client player application, as it is received, leaving behind no residual copy of the content on the receiving device.

The evolution of streaming

1994

Accessible only by users with high-end workstations, the first true streaming videos run over the experimental Mbone (Multicast Backbone) network.

1995

Progressive download technology allows audiences with standard personal computers to hear audio and view video files as they are being downloaded.

1997

The first all-in-one, audio-video players are released.

1999

True streaming is introduced to the general market.

The World Wide Web Consortium (W3C) endorses Synchronized Multimedia Integration Language (SMIL), a text-based tag markup format for streaming multimedia. SMIL frees developers from proprietary formats, enabling multiple vendors to supply software tools.

2001

First implementations of MPEG-4, an open standard that should lead to player interoperability.

JUST WHAT *DOES* IT MEAN TO STREAM?

The primary characteristics of "streaming media"

Streaming is an emerging technology. You'll find that there are many diverse, and often quite confusing, definitions floating around. This primer deals with streaming **media** only—i.e., audio, **full-motion video**, and **multimedia** content—as opposed to other applications of streaming technology, such as the streaming of real-time stock quotes. For our purposes, three primary characteristics combine to define streaming media, as explained below:

1. Streaming media technology **enables real-time or on-demand access to audio, video, and multimedia content via the Internet or an intranet.**

   Streaming technology enables the near **real-time** transmission of events recorded in video and/or audio, as they happen—sometimes called **"Live-Live,"** and commonly known as **Webcasting.** Streaming technology also makes it possible to conveniently distribute pre-recorded/pre-edited media **on-demand**. In other words, media that is stored and published on the Web in streaming formats can be made available for access at any time.

2. Streaming media is **transmitted by a media server application, and is processed and played back by a client player application, as it is received.**

   A **client** application, known as **player**, can start playing back streaming media as soon as enough data has been received—without having to wait for the entire file to have arrived. As data is transferred, it is temporarily stored in a **buffer** until enough data has accumulated to be properly assembled into the next sequence of the media stream. When streaming technology was first available, the ability to begin playback before the entire file had been transferred was a distinct advantage. Now, however, **pseudo-streaming** techniques, such as **progressive download**, allow some other formats to begin to play before file download is completed. So, while the ability to begin playback prior to completing file transfer is a characteristic of streaming, it is not, in and of itself, a differentiating factor.

3. A streamed file is received, processed, and played simultaneously and immediately, **leaving behind no residual copy of the content on the receiving device.**

   An important advantage of streaming media (unlike either traditional or progressive download) technology is the copyright protection it provides. No copy of the content is left on the receiving device. Therefore, the recipient can neither alter nor redistribute the content in an unauthorized manner.

Streaming marries something old with something new

If you take away the references to the Internet and the computer from our definition of streaming media, it's clear to see that we have been "streaming" media, according to industry expert Steven M. Blumenfeld, "…since the dawn of the media age… Streaming media is not new, it has been around since the inception of the radio (Guglielmo Marconi 1897 - inventor of the radio). We just called it broadcast."[2] Broadcast, however, as we currently know it in the form of radio and television, does not yet provide the **rich media** experience that the Internet and the World Wide Web enable.

2 *http://broadcastengineering.com/html/2000/august/features/streamMedia_0800.htm*, "Streaming Media," by Steven M. Blumenfeld, *Broadcast Engineering*, August 2000

Streaming media is the convergence of broadcast and rich media, empowering both content providers and audiences with a whole new world of choices.

Streaming media is the convergence of broadcast and rich media, empowering both content providers and audiences with a whole new world of choices. Like broadcast, streaming media delivers the real-time or on-demand access to audio, video, and multimedia content that audiences want, while providing the copyright protection content providers demand. But streaming media can also weave interactivity into the experience. Unlike analog broadcasting, the digital nature of streaming media facilitates the integration of interactive capabilities such as the **chapterization** of segments, clickable **hotspots** within the video frame, **URL flips** that automatically launch Web pages at specific instants during playback, and the intelligent indexing of media content through the use of searchable keywords.

WHY DO *I* WANT TO STREAM?

Because, according to Arbitron/Edison Media Research, seven out of ten Internet users in the U.S. say Web sites would be more enjoyable if sound and video were included more often.[3] More than one in every four is already a "Streamie" (someone who has listened to streaming audio or watched streaming video online).[4]

"Today, the noteworthy successes of streaming video are happening over intranets… streaming video over networks is not typified by tiny and clumsy imposters for video, but rather, full-frame, full-motion MPEG-1 or MPEG-2 digital video… the excitement and the bandwidth for effective video streaming already exists over intranets, and… business, educational, medical, and entertainment users are now demonstrating that… [it] is [a] money-making and money-saving business."

—Jeff Sauer
"Today's Stream Believers"[6]

Enterprises are reaping the benefits of streaming now

Offering numerous case studies on their Web site, Microsoft shows that enterprise customers such as Aetna, Fujitsu, and Hewlett Packard are already gaining remarkable ROI (return on investment) for their streaming initiatives. The Aberdeen Group, an independent consultancy that prepared the Hewlett Packard case study, reports "[HP] was able to streamline product introductions, increase the effectiveness and reach of its communications, and significantly reduce the costs of key communications. The cost savings—derived primarily from reducing the use of multi-city roadshows, audio bridges, and satellite links—were approximately $1.2 million in the first year and generated an ROI of over 1800% and a one-month payback. Therefore, Aberdeen concludes that the business benefits are sufficiently compelling that enterprises should seriously investigate deploying streaming media technology."[5]

The three most common ways enterprises use streaming are for corporate communications, electronic learning, and sales and marketing. Enterprise customers should consider using streaming technology to:

- Send e-broadcasts across their corporations
- Give remote presentations to employees, customers, and partners
- Deliver end-to-end training to the sales force or retail locations
- Enhance professional development and communicate HR policies
- Conduct pre-sales demos
- Provide customer service

"Streamies" are twice as likely to click on Web ads and make online purchases.[8]

Streaming media attracts high-value e-consumers

The power of streaming is also becoming apparent for B2C (business-to-consumer) e-commerce entrepreneurs. According to a study conducted in the U.S., called "The Buying Power of 'Streamies,'"[7] and published by The Arbitron Company and Edison Media Research, "Streamies, those people who watch or listen to Internet audio and video… represent 43% of Web users and 24% of all Americans. These are the most experienced Web users, spending 46% more time online than the average person. Streamies, twice as likely to click on Web ads and make online purchases, spend lots of money

3 "Startling New Insights About the Internet and Streaming" by Arbitron/Edison Media Research, September 2000, PDF available at *http://www.arbitron.com* or *http://www.edisonresearch.com*
4 "Streaming at a Crossroads" by Arbitron/Edison Media Research, February 2001, PDF available at *http://www.arbitron.com* or *http://www.edisonresearch.com*
5 The Aberdeen Group, as quoted at *http://www.microsoft.com/windows/windowsmedia/en/business/others.asp*
6 "The Moving Picture: Today's Stream Believers" by Jeff Sauer, *eMedia Live*, December 2000
7 "The Buying Power of 'Streamies'," Arbitron/Edison Media Research, February 2000, PDF available at *http://www.arbitron.com* or *http://www.edisonresearch.com*
8 "The Buying Power of 'Streamies'," Arbitron/Edison Media Research, February 2000, PDF available at *http://www.arbitron.com* or *http://www.edisonresearch.com*

"DOWNLOAD" DEFINED

"Uploading" and "downloading" mean transferring data from one computer to another. The two words have the same relationship as exporting and importing —the correct choice depends on which side of the border you're on. Uploading is *sending,* or exporting, data from one computer to another; downloading is *retrieving,* or importing data from one computer to another.

Here's where it gets a little confusing…

Whenever you import data from one computer to another, you are, technically speaking, downloading. As you have already learned, streaming means that the file is processed and played as it is received—i.e., downloaded—and then the content is discarded. So, even though you have downloaded a file, it will not and cannot be saved to your hard drive.

However... in common usage, "downloading" has come to mean receiving *and saving* data to your hard drive. When you save, or write a file to disk, if it has not been "locked" or otherwise secured against tampering, and if you have the appropriate software application, you could alter the file and/or redistribute it.

So a distinction has been drawn between "downloading" and "streaming." While "traditional download" and "progressive download" both result in media files being written to disk (i.e., saved on the end-user's hard drive), when you receive (i.e., "download") streaming files, the media content is *not* saved.

online and plan to spend even more." The study concludes that when Web sites integrate streaming media into their content, they will attract more of these potentially high-value customers. In an even more recent study, "Startling New Insights about the Internet and Streaming,"[9] published in September 2000, Arbitron/Edison Media Research tell us that 56% of Streamies have made an online purchase, spending a mean amount of $768 per year online, as opposed to 33% of non-streamies making online purchases with mean online spending at $596 per year.

The advantages of streaming outweigh its shortcomings

Quality concerns will continue to challenge media streamed over the public Internet, until broadband is ubiquitous. Even now, however, streaming offers distinct advantages that other methods of media delivery do not provide:

- No waiting for complete downloads—well, not "much" waiting, anyway.
- Streamed files are not written to disk—they are processed and played as they are received, then discarded, leaving no residual copy of the content on the receiving device, and thereby alleviating copyright concerns.
- Streaming is capable of conveying live events around the globe in near real time.
- Supports interactivity, allowing content to be experienced in a non-linear manner. Video-on-demand, for example, can be chapterized, allowing users to jump to desired portions.
- Streaming content is an excellent way to enhance otherwise **static** Web sites.

WHAT MAKES STREAMING MEDIA DIFFERENT?

Traditional download makes you wait

The Internet and **downloadable media** formats have brought end-users access to a wide range of multimedia content published on the World Wide Web. But, in a **narrowband** world (as of this writing, most Internet users are still connecting at speeds of 56Kbps or lower), downloading large media files is quite tedious and, very often, inconvenient. It goes like this: Just click on a link to the file; choose "Save As;" designate a destination folder on your hard drive; and then… wait… as… the… file… is… down… load… ed…. You have to wait for the entire download to finish—and that could be minutes or even hours—before you can see or hear what you downloaded. So you might spend a lot of time downloading media files, only to find that when you play them, they are not what you really wanted or expected. .

"Until recently, audio and video on the Web was primarily a download-and-play technology. You had to first download an entire media file before it could play. It was like pouring milk into a glass and then drinking it. But because media files are usually very large and take a long time to download, the only content found on the Web was short 30-second clips—often even shorter. Even these files could take 20 minutes or longer to download. In other words, it took a long time to pour the milk, and then it would barely quench your thirst… Watching audio and video files that stream is more like drinking straight from the carton; streaming media files begin playing almost immediately, while the data is being sent, without having to wait for the whole file to download. Other than a few seconds of delay before the file starts to play, you don't have to wait to start watching, no matter if the file lasts 30 seconds or 30 minutes."

—Microsoft Windows Technologies Web Site[10]

9 "Startling New Insights About the Internet and Streaming," Arbitron/Edison Media Research, September 2000, PDF available at *http://www.arbitron.com* or *http://www.edisonresearch.com*
10 *http://www.microsoft.com/windows/windowsmedia/en/compare/webservvstreamserv.asp*

IS FLASH STREAMING?

The Macromedia® Flash™ (SWF) format was designed to deliver low-bandwidth motion graphics specifically for the Web. It's great strength lies in generating smooth vector animations.

According to our definition of "true streaming," Macromedia Flash animations and other SWF format files are *not* streaming media. However, SWF files do share some of the important characteristics of streaming media.

SWF files *do*:

- **Deliver content with motion and sound, as well as with interactivity**

***However,* SWF files do not deliver full-motion video. AVI and MPEG files cannot be imported into Flash. QuickTime (MOV) files can be imported, but do not become part of the Flash file and cannot be viewed with the Flash player. A utility can be used to reduce a video file to a sequence of still bitmap images which can then be used in a series of key-frames to *simulate* video, but the resulting movie does not deliver the same quality as streaming video optimized in a native streaming file format and played back on a video player. Furthermore, Flash files are limited in size to 16,000 frames. At 15 frames per second, that's approximately 17 minutes worth of material.**

- **Begin to play before the file is entirely downloaded**

***However,* SWF delivery is really progressive download. Served from a Web server and transmitted via HTTP, Flash playback will stall if network congestion is encountered, and transmission cannot be adjusted to accommodate real-time conditions as can true streaming.**

- **Can only play back using the media player that is part of the Flash architecture**

***However,* SWF files can be created using a variety of other software tools in addition to Macromedia Flash, including but not limited to Adobe LiveMotion and Adobe After Effects.**

According to the Macromedia Web site, "Flash was created to make small, streamable, vector-based files for Web delivery. This is… where Flash performs best.… it's important for users to ask themselves, 'Is this program designed to do what I am attempting"

Traditional download lets you save and redistribute media

The downloadable approach is good for the end-user who wants to ***keep*** the content and replay it whenever they like. In 1999, the phenomenal popularity of the 2 1/2-minute movie trailer for "Star Wars: The Phantom Menace"—a whopping big 25 MB file at 480x216 resolution (good quality)—was proof that media-hungry Internet audiences are willing to be forgiving of long waits for large files. But if an artist or publisher does not want end-users to be able to save and freely redistribute their media content, the downloadable approach is problematic—that's the story of MP3s and Napster.

Progressive download can get hung up

Unlike traditional download, **progressive download** (also known as **pseudo-streaming** and, in QuickTime, **fast-start streaming**) allows playback to begin before the file is completely downloaded. Progressive download allows playback of content that has been received to continue, while the remainder is still being downloaded. Progressive download, therefore, allows the end-user to view the beginning of the file, as the remainder is being downloaded, and to quit the download if the content is not worth the wait.

Like streaming, progressive download does allow audiences to see and hear the content immediately—as it is being downloaded—but ***only*** if the download speed can be maintained at the rate needed to keep up with playback. If the Internet connection becomes busy, causing the download to slow or stall, audio and video playback may not always stay synchronized, playback may become choppy or, as is often the case, may come to a halt, as the player waits for the download to catch up.

Progressive download also leaves a copy of content behind

Unlike streaming, progressive download writes the media file to disk, leaving a copy of the content in the memory of the receiving device. Streaming media is processed, played, and discarded, leaving no physical copy behind.

Despite the drawbacks of progressive download, it can be a good alternative if true streaming is not possible. Progressive download can be easier and less costly to implement than true streaming. All the production techniques described in this Primer should be applied to preparing media for progressive download, as well as for streaming.

True streaming is based on a different architecture

Streaming media requires a different *architecture* from that used for downloadable media—an architecture that supports a higher level interactivity over the Web, and offers a wealth of additional advantages. Streaming allows you to browse through and jump back and forth in the content—as opposed to progressive download, which requires viewing from beginning to end. Streaming supports both **on-demand** and live content to be transmitted over the Web in real time, allowing audiences anywhere in the world to playback media whenever and however they need or want it, and to experience live events as they happen.

True streaming requires a specialized media server

Downloadable media files—including those prepared for progressive download—are typically served from a standard **Web server** that does not have any ability to adjust transmission to the uncontrollable fluctuations in connection speed that are inherent in the Internet environment.

Streaming exploits new Web transfer protocols, as well as communication techniques between clients and servers, to facilitate the continuous playback of synchronized audio and video in real time. Streaming media files are usually **encoded** in several versions, optimized for different **data rates**. The **streaming media server** delivers the appropriate version, either as selected by the end-user manually or as indicated by browser default settings. In some cases, the server can automatically select the best version, based on information brought back from the client regarding the platform and the connection speed. Depending on the intelligence built into the media server and client player software, the stream can be dynamically adjusted, at one or both ends, if the connection slows down.

WHAT IS "REAL TIME?"

"Real time" is one of those techie terms we were all supposed to wake up one day (not so very long ago), simply understanding. The implication is that "real time" means "as-it-happens," or "concurrent with reality." Since streamed files are processed and played back *as* they are transferred from server to client, *all* streaming is essentially "real-time," as far as the *data* is concerned. There is, however, always some amount of delay between trans-mission and reception, making the use of the term "real time," when it comes to streaming, slightly inaccu-rate. Transmission of streaming media is dependent upon such factors as available bandwidth, connection speed, and network congestion. And there is often at least a momentary delay while the client-dependent buffer accumulates enough data to process and play. So, espe-cially when media is being streamed to a wide variety of platforms, via the public Internet, each recipient will experience playback in a slightly different timeframe— not actually concurrently.

WHEN IS "LIVE" LIVE?

The *content* of streaming media may be real-time— meaning "live," as in an event being Webcast as it occurs. Or, it may have been stored in an archive and be streamed "on demand." Streaming media may be categorized into three basic program types, based on the kind of content streamed and how it is made available to the audience:

VOD (Video-On-Demand): When pre-recorded streaming media content has been archived on a server, it may be accessed "on-demand" by individual audience members, at any time. Whenever an end-user clicks on the link to request the program, it starts playing from the beginning. In similar fashion to viewing a video cassette or DVD, the end-user can control the experience by mani-pulating controls on the player interface to move forward or backward, to pause, or to jump to any point in the program. Multiple users may be streaming the same program, but each user may be accessing different portions of the content concurrently.

SCHEDULED WEBCAST: Archived content may also be used for a "scheduled Webcast," sometimes called a "streaming broadcast." The designated URL is not valid before the Webcast begins nor after it ends. The host starts streaming the program at a scheduled time, con-tinuing to stream from start to finish. Just like with tele-vision broadcasts, viewers who don't start to access the stream at the appointed time join the program "in-prog-ress." Scheduled Webcasts are sometimes referred to as "live Webcasts," when the audience is "live"—for example, on-line meetings or distance learning situations in which parts of the presentation consist of pre-recorded stream-ing media and other parts are interactive exchanges (chat, panel discussions, Q&A, quizzes, etc.)

"LIVE" LIVE WEBCAST: When live content is streamed as it occurs, it is, of course, considered "live." Like any scheduled Webcast, everyone in the audience experiences the same thing at the same time. To differentiate a "live Webcast," (i.e., a scheduled Webcast streaming pre-record-ed content to a "live" audience), you'll sometimes hear "as-it-occurs" Webcasts referred to as "Live-Live."

The benefits of using a specialized streaming media server include:

- More efficient use of the network bandwidth
- Better audio and video quality to the user
- Advanced features like detailed reporting and multi-stream multi-media content
- Support for large numbers of users
- Multiple delivery options
- Content copyright protection

Files that have been encoded in a streaming media **format** *can* be served by a standard Web server. **Web server streaming**, also called **HTTP streaming** is, however, actually just a flavor of **progressive download**. HTTP streaming creates a local cached copy of the media file, so there is no way to prevent end-users from copying the file into a designated directory on their hard drive. In other words, HTTP streaming does not provide the copyright protection for media content that is, by definition, an essential characteristic of true streaming.

One interesting advantage of HTTP streaming, however, is its ability to transfer streaming media files across firewalls which often do not allow true streaming media files to pass. Many specialized streaming server software solutions provide streaming media publishers with the option of switching over to HTTP streaming when true streaming is prevented by a firewall.

Streaming is not used to transfer normal data

Because audio and video are time-dependent mediums, in order for them to play smoothly, **packets** of information being transferred must arrive on time and in good condition. The public Internet, however, is based on an **asynchronous** system, wherein it doesn't matter in what order packets arrive. Transfer Control Protocol—the "TCP" part of **TCP/IP** ("IP" stands for Internet Protocol)—typically ensures that files are assembled in proper sequence, and requests replacements for any packets lost or corrupted in transport. This is known as "error correction," and it is a luxury that is sacrificed for streaming as we know it today. When media is streamed, data that is lost or damaged is gone forever—at least as far as that stream is con-cerned—resulting in imperfect, sometimes even indecipherable, playback.

Fortunately, video is a very "forgiving" medium—we tend to get the gist of the content, even when it skips a beat or two. So streaming technology is still viable when the transmission isn't ideal. Besides, streamies seem to be just as forgiving an audience as the masses who supported the early imple-mentations of so many other technologies that have brought us motion and sound—scratchy-sounding phonograph records; grainy, black and white films; early talkies with less-than-perfect sound synchronization; and the frustratingly intermittent television reception some of us still recall from pre-cable days. Easily-delighted streamies are, apparently, willing to forgive a lot of the jittery, jumpy, jerky qualities of streaming media at its worst. But the point is that streaming is not a reliable way to transfer "normal" data— data that must be complete and uncorrupted to be acceptably reproduced on the receiving end. So streaming is not used to transfer text files, databases, intricate graphics, and other types of unforgiving information.

Streaming is bandwidth-dependent

When a non-streaming file is downloaded, **bandwidth** is relevant only in so far as how long it takes for the data to be transferred. A file transferred over a **narrowband** (low-bandwidth) connection will take much longer to download than the same file transferred over a **broadband** (high-band-width) connection, so long as all other factors (such as server load, network congestion, etc.) are equivalent. But however long the download takes, when all is said and done, the quality of the playback experience will be the same, assuming all other relevant factors (e.g., processing capability, video support, quality of monitor and speakers, popcorn, etc.) are equal.

For streaming media to play back smoothly, a continuous, steady "stream" of data must be delivered—a stream that won't "choke" the pipes. In other words, you need to produce a file that streams at a low enough **bit rate** (data transfer rate) for the connection to handle. Reducing the bit rate of a media file enough to flow smoothly through a narrowband connection means sending much less data than a broadband connection is able to handle—much less data than we are used to seeing when we watch broadcast television; much, much less than when we play a DVD. There are several ways in which the bit rate can be reduced:

- the physical dimensions (area) of the video frame can be made smaller
- the number of frames per second (fps) of the video can be lowered, and/or
- the amount of information in each frame can be reduced through *compression.*

When any or all of these strategies are employed, the quality of the playback experience suffers.

If a media file is encoded to optimize playback for the least common denominator—a typical narrowband connection at 28.8 **Kbps**, for example—the quality will be relatively low. A user with a higher bandwidth connection, such as cable modem, DSL, or a corporate T-1 line (which, with current technology might support "near-broadcast" quality) ***can*** play any streaming media clip that has been encoded for transmission at their connection speed or lower—but they will suffer poor quality unnecessarily. However, a user with a low bandwidth connection, who attempts to play a stream that has been encoded for higher bandwidths, will experience choppy playback and delays. The best solution the emerging streaming media industry has come up with to date, is to publish streaming media content as multiple streams, encoded for an assortment of bandwidths, in order to accommodate a wide variety of end-users. This is called **multi-bit rate encoding**, or **MBR**.

WHO'S STREAMING IN THE REAL WORLD?

There's a whole lot of streaming going on…

According to Nielson/NetRatings, streaming media consumption skyrocketed to an all-time high in November 2000, with 35 million Web users at home accessing streaming content, a 65% increase from 21 million in November 1999. Nielson/NetRatings indicates that figure accounts for 36% of all Internet users, as compared to 28% during the same period in the prior year. Nielson/NetRatings also reports that people with high-speed Web connections—11.2 million home users in November 2000—are more likely to consume streaming media content. Active users with a broadband connection in the home (over 56 Kbps) were 50% more likely to access streaming media than their dial-up counterparts (56 Kbps and below).[11]

Trends show streaming usage by Web users at home is skyrocketing

Audience Demographic	November 1999	November 2000	Percent Increase
Females	9 million	16 million	77%
Males	12 million	19 million	56%
Kids/Teens	4 million	7 million	65%
Seniors	700,000	1.4 million	95%

…And lots more streaming to come

A Q42000 report by U.S. Bancorp Piper Jaffrey indicates that the implementation of streaming media to meet consumer demand will drive the next wave of Internet growth. The report predicts that total spending on Web streaming media will grow from $9.7 million in 1999 to $21.6 million by 2004.

11 "Streaming Media Usage Spikes 65 Percent," *InternetNews.com*, December 2000, *http://www.internetnews.com/streaming-news/article/0,,8161_532781,00.html*

"The popularity of rich media grew steadily last year—to the point that a majority of Internet users are viewing some type of streaming media at least once a month. In January 2000, about 34 million Internet users viewed web video… Over the course of the year, that number grew steadily to November's high of nearly 49 million (spurred in part by the American presidential election, which drove Internet traffic to an all time high)."

—Dale Sorenson
***Streaming Media Market Report 2001*[14]**

"Consumers are not the only ones who will benefit from the streaming media revolution," the report states, "as e-tailers will find that streaming media allows them to target ads more effectively than previous methods… In addition, streaming media will benefit from an increase in business-to-business (B2B) broadcasting of events such as earnings reports, meetings and conventions, seminars, product and service announcements, and distance learning."[12]

Everybody's streaming now…

Spending on streaming media technology and services is expected to grow from $9.7 billion in 1999 to $15.2 billion in 2002.[13] So who's doing all this streaming?

- Television and radio stations
- Movie studios
- News organizations
- Music labels and musicians
- Corporations (intranet and extranet)
- E-tailers
- Educators and trainers
- Facilities managers and security providers
- Architects, engineers, construction managers
- Individuals

…Come along, we'll show you how!

There's a lot to know about streaming, and new technology is emerging every day. This Primer won't make you an expert. But we hope it will encourage you to get started.

- If you are a **business professional**, you may want to use streaming media to achieve a wide variety of objectives, from facilitating internal and external corporate communications to implementing more cost-effective training to bolstering your e-business effectiveness.
- If you are a **video professional**, your customers are probably already asking you to create original content and to repurpose existing content for streaming applications. It is quickly becoming apparent that you need to add streaming media to your capabilities. And, when you use streaming to deliver original media to Web audiences, you maintain control of your content with copyright protection.
- If you are a **Web designer or developer**, your clients will be looking to you to integrate streaming media into their Web sites to make them more compelling. Yep! It's one more capability you need to add to your continually expanding repertoire—sooner is better than later.

"Streaming media is scary… But streaming media is easy… These days, it is simple to create and to implement in your Web site. The only challenge is in understanding the range of choices and directions the streaming media industry offers."

—Tim Kennedy
***Streaming Media World*[15]**

12 "Report: Net Set for Streaming Media," by Lori Enos, *E-Commerce Times*, October 2000, *http://www.ecommercetimes.com/perl/story/4492.html*
13 "Streaming Media: The Next Wave for Digital Content Distribution," by C. Eugene Munster and Robert S. Drobinski, U.S. Bancorp Piper Jaffray
14 "Streaming Media Market Report 2001" by Dale Sorenson, *Macintouch Special Reports*, April 2001, *http://macintouch.com/stream2001.html#2000use*
15 "What is Streaming Media" by Tim Kennedy, *Streaming Media World*, June 1999, *http://www.streamingmediaworld.com/gen/tutor/whatis/*

SERVERS WITH A SMILE?

If you are a do-it-yourselfer who is not going to be streaming very much media to very many people, your Web server and media server might just be different processes (handled by different server software applications) running on the same machine.

If you are already going into techno-phobic shock at the very mention of "clients" and "servers," don't despair, read on… The techies who invent these things love to confuse the rest of us with tricky terminology, just to make sure we keep them around.

While the term "server" sometimes refers to hardware, as in "That old server is collecting dust," it may also mean software, as in "We installed a new Web server on that computer today." Both uses actually refer to the same principle—a server is a solution designed to receive requests from a third party and respond by delivering, or "serving up," a particular type of data. This process is also known as a "transaction."

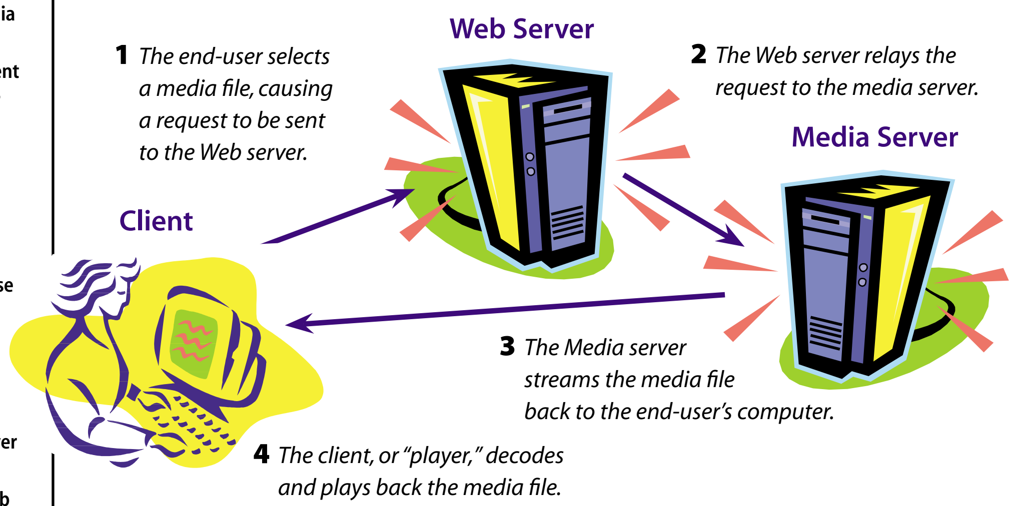
Requests are sent to a server by a "client." Again, the term client may be used to refer to a PC (hardware) or to a software application. A good example of a client is a Web browser, which is software installed on a PC that functions as a client to Web servers. When the client receives the data it has requested from the server, it may further process and/or display the data. Media "players" are software clients that process and play back video and/or audio.

Any computer on which the appropriate server software has been installed can function as a server. But typically, when a piece of hardware is dedicated to that role, it is made up of high-grade components that will deliver reliable 24/7/365 availability.

If you reach the point where you are serving so many streams to so many users that, in order to handle the traffic, you need a dedicated media server (and you decide to handle your media hosting in-house rather than outsourcing), you will probably want what is known as an "appliance server" or "network appliance." A network appliance is server hardware with "factory-installed" server software, designed specifically for a specific type of server function—such as handling media.

HOW DOES STREAMING WORK?

Although streaming technology can start to seem mighty complex mighty fast, it's all really based on this simple model:



WHERE DO STREAMS COME FROM?

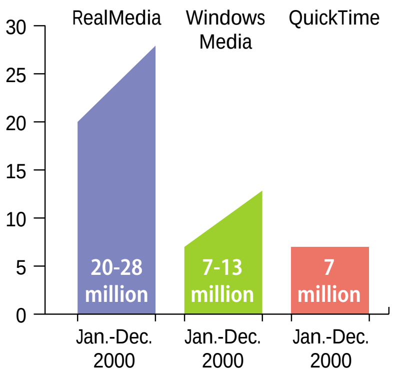
Streaming media architectures

An **architecture** is an interdependent system comprised of a variety of components that all work together to perform certain functions. Streaming media architectures are comprised of **encoding** and transmission methods, *server* software, and players (*client* software). Currently, the three most popular streaming media architectures are **RealMedia**, **Windows Media**, and **QuickTime**.

There are also a few interesting alternatives available, including architectures based on Java applets that do not require the end-user to have a proprietary **player** (or **plug-in**) installed. Instead, an ultra-light Java applet delivers an ephemeral player to the end-user's browser, enabling playback for the duration of the streaming session, after which the player is discarded. While such java-based players are usually less intrusive than proprietary players—some even virtually transparent—they typically do not offer the full range of end-user controls made available by the proprietary players.

Currently, however, and for a variety of technology and business reasons which are beyond the scope of this document, the streaming market is dominated by three architectures: **RealMedia, Windows Media**, and **QuickTime**. The unfortunate result is that there are now three parallel standards that do not work together. Imagine what it might have been like if, when television first hit the airwaves, each of the three major networks required a different device to translate their signal into video and audio—each home would have needed an ABC television set, a CBS television set, and an NBC television set, if the household wanted to watch programs on every channel. That's essentially what the situation is, in today's streaming media marketplace. While there is some limited cross-over in the players' ability to play back other architectures' formats, the technology is so competitive, that end-users cannot count on cross-over capabilities working, as the formats rapidly evolve. This can be confusing and irritating for end-users, who must continually download the latest plug-ins and then figure out which to choose as the default player in their browser, based on which is being supported by the majority of their favorite content providers.

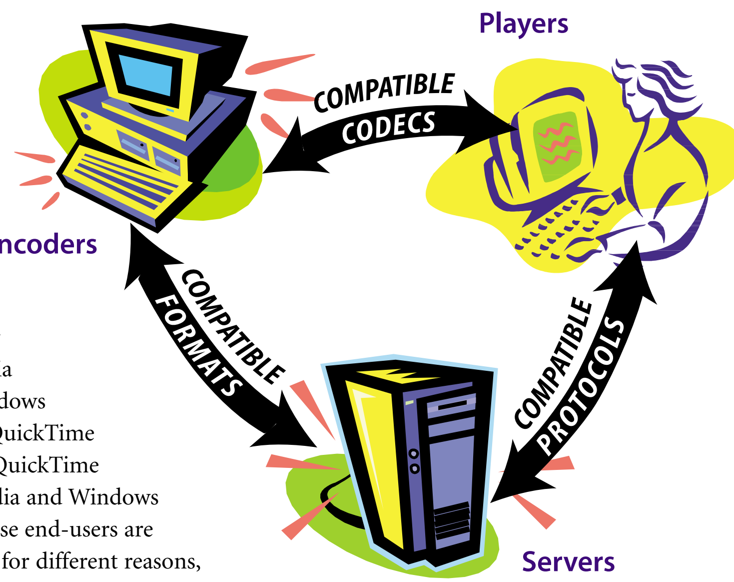
Year 2000: unique monthly audience by player, in millions of viewers per month[16]



"…most content providers support all the players—Microsoft's Media Player, Real Network's Real Player and Apple's QuickTime—in an effort to reach as many eyeballs as possible. There are some solutions available… that make it easier to handle multiple bit rates from one source, but encoding for the different formats continues to be a challenge. In addition to requiring more employees to handle the encoding, it also requires more space on servers…."

—Ken Kerschbaumer
Broadcasting & Cable Magazine[17]

It's all interrelated—in a streaming architecture, everything must be compatible



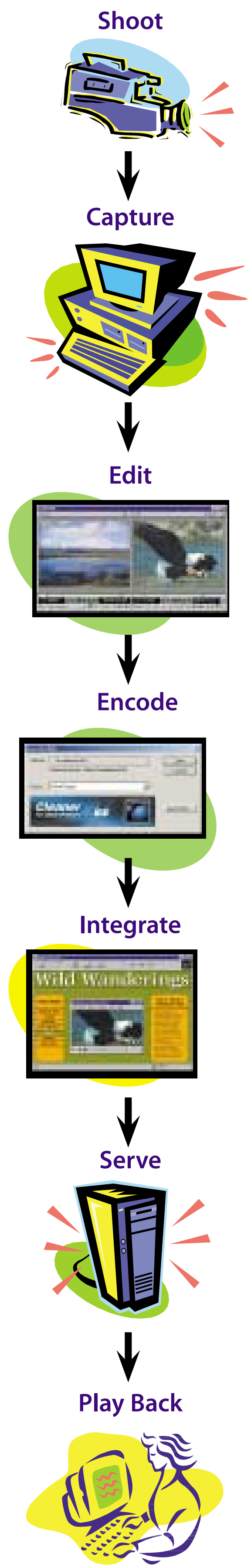
From the content provider's perspective, the situation is even more difficult: the RealMedia server cannot stream Windows Media and QuickTime files; the Windows Media server cannot stream QuickTime and RealMedia files; and the QuickTime server cannot stream RealMedia and Windows Media files. Meanwhile, because end-users are choosing their default players for different reasons, and no content provider wants their media to be overlooked simply because the end-user has selected one player over another, content providers usually choose to encode, serve, and stream their content in at least two, if not all three of the major formats. And, making it even more complex and costly, depending on whether they are delivering over narrowband or broadband or both, content providers may offer streaming files at multiple bit rates, within each format. So there may be as many as nine different versions to be encoded, stored, and streamed.

Each architecture has pro and cons, as well as champions and critics. Decisions you make about which architectures to use, when, and how, will depend upon the parameters of your project, your audience, your objectives, and other factors. Don't be surprised if you find it quite difficult to decide—this is the subject of great debate, even among the experts. But one thing upon which the experts agree, is the need for an open, extensible standard that will satisfy the needs of technology providers, content providers, and audiences alike.

Streaming media formats

Architectures are often mistakenly called "formats." An architecture is much more than just a format. A **format**, also known as a **file format**, is simply the file structure an architecture creates with its **codecs** (short for compressor-decompressor—you'll learn more about codecs later in this Primer). The file format and, therefore, the architecture, can usually be recognized by the filename extension (a dot plus three characters) tagged onto the name of the file.

16 Source: "Streaming Media Market Report 2001" by Dale Sorenson, *Macintouch Special Reports*, April 2001, *http://macintouch.com/stream2001.html#2000use*
17 "The Impossible Stream," by Ken Kerschbaumer, *Broadcasting & Cable Magazine*, December 11, 2000



The "big three" streaming architectures and their native file formats

Architecture	Native Formats	Streaming Media File Extensions
QuickTime	QuickTime Format	.mov (sometimes .qt or .qti)
RealMedia	RealMedia Format	.rm
Windows Media	Advanced Streaming Format or Windows Media Video/Audio	.asf, .wmv, .wma

MPEG (denoted by .mpg and several other file extensions) is a standard file format for video, and it is also a codec that is used to create formats, but it is not a complete architecture. MPEG-1, originally designed for multimedia formats on CD-ROM, does not support true streaming but only progressive download. MPEG-4 is a brand new, international open standard for Web video, based around the QuickTime format. Windows Media Video v1 is a derivative of the MPEG-4 codec, which has been renamed to avoid confusion. QuickTime 5, just released as of the finalization of this Primer, is the first full implementation of MPEG-4 for streaming media.

WHAT ALL DO I NEED TO MAKE MY PRODUCTIONS STREAM?

This section merely outlines the tools and techniques involved in streaming media—you'll find more detailed explanations in the section of this primer entitled "How Do I Make (Good) Streaming Media." The process from planning and shooting your video, through outputting the appropriate file formats for the selected medium are, essentially, the same steps you would go through for developing any kind of digital video project: pre-production (planning), production (shooting video and recording audio), and post-production (editing, adding effects and titles, mixing and synchronizing audio, and outputting for your medium of choice). But for streaming media, the process doesn't end there—you still have to publish your streams.

First, you'll need some media you want to share

Whether it's a live-live Webcast from a Web-cam trained on your baby's first steps… or a scheduled Webcast of your CEO giving a keynote speech that's pushed to the extended enterprise… or an effects-heavy movie trailer that audiences can stream on-demand to build anticipation for the next "big summer blockbuster"… if you can **capture** it, you can stream it.

The term "video capture" may be used to refer to the digitization of *analog* video, from a connected camcorder (video camera) or from a connected tape deck, as it is saved onto a computer hard drive via a *video capture card* installed in the computer. The captured video clips, once digitized, can then be manipulated (edited and/or encoded) with computer software.

The very ***same*** term—"capture"—is also used to refer to the simple ***transfer*** of **DV** (a video format typically shot with a digital camcorder and, therefore, already in digital form), from a camcorder or tape deck, onto a computer via an **IEEE 1394** connection (also known as FireWire or iLink). For Webcasting, analog video and DV can both be captured "live" while filming, to be streamed "live"—i.e., in real time (although a specialized video capture card designed to handle streaming is necessary for the live capture and streaming of analog video).

Similarly, audio, whether recorded live or transferred from a CD or other sound storage medium is also said to be "captured," when it is saved onto a computer.

You'll probably want some video editing and visual effects software

If you are not doing a **live feed,** you have the opportunity to edit and develop your video story with titles, motion graphics, transitions, and visual effects, as well as with music, voice-overs, sound effects, and other audio enhancements. Or, you may be repurposing content that was produced for other mediums, that needs to be recut and simplified for best results on the Web. *Adobe Premiere®* and *Adobe After Effects®* software provide all the tools you need for digital video editing and visual effects.

You'll need tools for pre-processing and encoding streaming media

Outputting your production as streaming media is likely to include two basic steps: **pre-processing** and **encoding** (also referred to as **compression**, although compression is only part of encoding). Pre-processing filters out non-essential information from your video and audio—information that is difficult to encode and is not required for generally accepted Web-quality media.

If you are doing a live Webcast, you'll need to set up automatic pre-processing functions between capture and encoding. If, however, you edited your project for VOD (video-on-demand), you may have already done most, if not all, of the pre-processing in the editing stage. The compression part of encoding reduces the **bit rate** of the digital media file so that it can be efficiently streamed at the proper bandwidth. Encoding also translates the media file into the architecture's format, and may add other functionality such as **digital rights management features.**

WHY DO WE NEED PLUG-INS?

When you click on a Web link for a media file, you depend on your browser to automatically "know" how to play it.

It "knows" because most files on the Web use a MIME (Multipurpose Internet Mail Extension) format. Originally developed for transmitting attachments to e-mail messages, MIME is a standard system for identifying the type of data contained in a file, based on its extension. MIME types that are automatically recognized and displayed by most browsers include formatted text files like HTML, graphics files like GIF and JPEG, and interactivity-enhancing files like those created in JavaScript or ActiveX. When a browser encounters an "unknown" MIME type, it does one of the following:

- **Begins downloading the file and opens it with the application designated in your browser preferences.**
- **If it does not find a corresponding application, it may ask you with which application you want to open the file, or it may ask if you want to download an appropriate application.**
- **Allows you to save the file to disk.**
- **Allows you to cancel the transfer.**

Browsers can handle practically any file type, even those that they cannot display themselves, by using external, or "helper," applications known as plug-ins. These auxiliary programs must be installed for a browser to load and display (or play) certain MIME types. Adobe Acrobat® PDF files, Macromedia Flash SWF files, and video and audio files formatted for RealMedia, Windows Media, and QuickTime players are just a few of the more commonly encountered file types requiring a plug-in for display or playback.

Browsers usually come with common plug-ins already installed and, when they encounter a file requiring a plug-in that has not been installed, will often prompt you to download and install it, if you choose to do so.

Gotta have a streaming media server… or two… or three…

The term "**server**" can either be used to refer to server software, or to computer hardware dedicated to serving processes. When we say that you need a specialized server to stream media, we are primarily referring to the server software. You'll recall from our definition of streaming, that ***true*** streaming requires a specialized streaming server. Although files formatted for streaming *can* be served from a Web server, files cannot truly be streamed. In the next section, and later in this Primer, you'll learn more about why this is so.

If you choose to deliver your streaming media in only one format, you need only use the server software that supports that format. But if you decide to offer ***more*** than one format, remember that ***each format must be served by its compatible server software***. So, if you want to offer your streaming content in all three of the popular streaming formats—QuickTime, RealMedia, and Windows Media—you will need to run, or have access to, servers for all three architectures.

New options for serving and hosting streaming media are emerging on an almost daily basis. The pivotal question for every "streamer," large or small, is whether to serve their own streams or to outsource. A **hosting provider** may prove to be an excellent alternative to setting up your own server(s), if you are an individual or small business expecting to stream just a little. If you intend to serve a ***lot*** of streaming media—as part of an enterprise initiative, in support of e-commerce, or for most any type of commercial endeavor—you ***will*** need dedicated hardware, and/or a hosting provider.

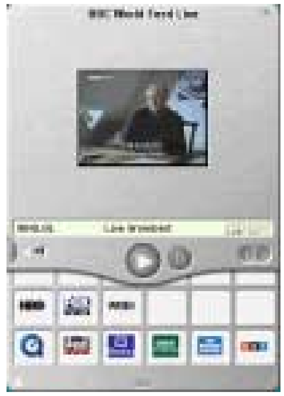
And a streaming media player… or two… or three…

Streaming media requires **client** software on the receiving end—in other words, each of the end-users in your audience needs a **player** to see and hear your streaming media content. These players stream pre-recorded and live media over the Internet, and can play back media that has been saved to disk in various formats.

There are two basic types of streaming media clients:

- **Java clients**—Java clients quickly download a Java applet to the user's machine before the streaming content begins, so that any Java-enabled browser can play back the stream. Java clients can provide a virtually transparent experience for the end-user, playing content that appears to be seamlessly embedded into a Web page, a banner, or an e-mail, without invoking a pop-up window for a plug-in player that may interrupt the experience. Java clients do not, typically, offer the full range of end-user controls provided by plug-in clients. For the moment, the plug-in client model remains the popular standard.
- **Plug-in clients**—A **plug-in** is an application that adds functionality to your Web browser. In the case of a media player, the plug-in provides the client software needed to play back and control media that is either downloaded or streamed to the browser. The three major streaming media architectures use plug-in clients.

Standalone RealMedia Player offers exclusive streaming coverage of Major League Baseball



Standalone QuickTime Player displaying Favorites "drawer"

QuickTime Player embedded into Web page

Standalone Windows Media Player displaying portal mode

Users can acquire these and other "skins" to personalize their Windows Media Player:

- *A nostalgic skin resembling an old-fashioned radio*
- *A futuristic silver ball with sliding elements and hidden features*
- *A cute penguin named "Melvin" who plays your music and videos*

"The issue for consumers will be the irritation of having to download the player and figuring out which one is going to become the default player… Streaming media content almost certainly requires media companies to convert to multiple formats."

—Jeremy Schwartz, consultant[18]

RealNetworks, Apple, and Microsoft each offer players that are available via free download from their Web sites (although the Windows Media Player comes pre-installed on PCs shipped with the Windows operating system, while the QuickTime Player is pre-installed on the Mac OS platform). All three of these major competitors sell downloadable player upgrades, offering premium models that can enhance playback controls with such features as video contrast and color adjustment, audio graphics equalizers, the selective recording and saving of streaming content (if authorized by the content provider), and more.

While the three major architectures' media players will all play back some competitive formats, they do not always support the newest releases of a competing format. This is a complication that makes it difficult for consumers to decide which player to choose as the default in their Web browser. Meanwhile, content providers who want to reach as many eyeballs as possible are forced to support all three players.

With competition between the three big players heating up, the contenders have begun to incorporate portal-like functionality into their players to add value, offering categorized content choices such as movie trailers, news, and other vertical content, as well as allowing end-users to add conveniently clickable access to their own favorite "channels." One of the ways in which the big three streaming media contenders compete is by vying for exclusive rights to broadcast popular content—especially sports, entertainment, and news—only via their own portal-player. As a primary means to generate revenue, portal-players offer advertisers the opportunity for sponsorships, streaming commercial messages prior to playing user-selected content, and making the experience of streaming video on the Internet more and more like traditional broadcast radio and TV.

18 As quoted in "The Impossible Stream," by Ken Kerschbaumer, *Broadcasting & Cable Magazine*, December 11, 2000

While the essence of the player is its capability to support media playback, the experience of the player may take a variety of different forms, either as a standalone window, or embedded into a Web page to enhance static content. In either case, the player's UI (user interface) often looks and works much like a familiar physical device—either a television set or a radio or a jukebox, of sorts. The player can, in most cases, have a customized look and feel, known as a **skin.** Skins may be delivered by content providers, in order to "brand" the streaming media experience. Skins for some players may also be user-selected, allowing customization by end-users.

In the corporate world, streaming media players are evolving beyond just organizing and playing media. Players are becoming comprehensive presentation engines, with the ability to present live or recorded multimedia content with slides, video, music and narration. Such presentations can be streamed over a corporate intranet or, via the Internet, to customers and constituencies around the globe.

HOW DOES IT GET THERE FROM HERE?

"Although the terms Web and Internet are often used synonymously, they're actually two different things… The Internet is the global association of computers that carries data and makes the exchange of information possible. The World Wide Web is a subset of the Net—a collection of interlinked documents that work together using a specific Internet protocol called HTTP… In other words, the Net exists independently of the Web, but the Web can't exist without the Net."

—CNET[19]

Streams are sent from a streaming media server to a client using a **protocol** known as **RTP** (Real-time Transport Protocol). RTP is similar to **HTTP** and to **FTP**—protocols used by Web servers—but there are some essential differences…

Excuse me… you don't *know* what HTTP and FTP are?

What's the matter? Have you been living under a rock for the past eight years? (Just kidding!) The first Web browser, called "Mosaic," was created by Marc Andreeson in 1993. He went on to cofound Netscape in 1994, and the rest is history. The fact is, we've only had the ability to surf the Web with **GUI**-enabled browsers for a few short years, so don't be embarrassed if you need a little education on the way the **Internet** works—its history and technology has hardly made it into school curriculums yet, and most of us have been too busy ***using*** the **Internet** and the **Web** to be bothered to learn how they work.

In the computer world, just like in the "real" world, a "protocol" is a set of standards that defines how information is to be conveyed and how parties are to interact. Unless the conventions are strictly adhered to, one party will not recognize the other, and the information will not be transferred. The Internet is a virtual Tower of Babel, connecting a great many different types of computing platforms via a vast array of different communications mediums. So many types of protocols are used, on several different levels, all at once. This may seem complicated, but it is not an unfamiliar concept. When you make a telephone call, many protocols are also employed: the country code, area code, exchange, and the identifying numbers are all a part of the addressing protocol; there may be an automated protocol in use to locate the individual within the organization you are calling that requires pressing the correct sequence of buttons; and there is even a protocol for your verbal exchanges over the phone that includes critical success factors we now take for granted, such as identifying yourself, communicating without reference to visual aids, and ending the call with some form of "Goodbye."

TCP/IP is the most dominant protocol suite on the Internet, comprised of two main protocols—IP and TCP. TCP/IP might be likened to a global air traffic control network that makes sure data goes to the right destination and gets there intact. **IP** (Internet Protocol), the basis of most Internet protocols, breaks up large chunks of information into digestible *packets.* In addition to the data being conveyed, each packet (also known as a **datagram**) carries a header containing the source and destination **IP addresses**, as well as a sequence number that allows the destination computer to reconstruct the packets in the correct sequence, when they arrive. This sequencing information is critical, as the packets may not arrive in proper sequence, since they each find their own way to the final destination—along whatever path is necessary, depending on continually fluctuating network traffic conditions. If a telephone line breaks down along the way, a packet will find another route by which to travel. IP focuses mostly on the location of hardware, getting the information across the vast network, from one device to another.

19 http://coverage.cnet.com/Content/Features/Techno/Networks/ss03.html

ALPHABET SOUP: A PROTOCOL PRIMER

FTP (File Transfer Protocol)
- Uses TCP/IP (one-to-one, reliable)
- Transfers files across the Internet
- Typically used to transfer Web pages from the computer on which they were created, to the server from which they will be hosted

IP (Internet Protocol)
- Underlying protocol which transmits data across the Internet
- Supports real-time transfer of data

HTTP (Hypertext Transfer Protocol)
- Uses TCP/IP (one-to-one, reliable)
- Protocol for transmitting hypertext files, typically used for transmitting Web pages

RTP (Real-time Transfer Protocol)
- Layered on top of IP or UDP which support real-time transfer of data
- No error handling, no reliability, no real-time guarantee
- Information is time-stamped for synchronization
- One-to-one or one-to-many

RTSP (Real Time Streaming Protocol)
- Protocol for handling streaming media, which supports VCR-like operations
- Uses TCP; layered on top of RTP

SMTP (Simple Mail Transfer Protocol)
- Protocol for handling e-mail
- E-mail client software uses the SMTP protocol to send mail to an e-mail server and POP3 or IMAP4 protocols to fetch mail from an e-mail server.

TCP (Transfer Control Protocol)
- Layered on top of IP
- Establishes a virtual connection between a destination and a source
- Handles congestion and reliability
- One-to-one

TCP/IP
- Protocol suite comprised of IP, TCP, and some other protocols
- The protocol which handles most Internet traffic

UDP (User Datagram Protocol)
- Layered on top of IP
- Unlike TCP, no virtual connection
- Provides very little error recovery
- One-to-many
- Used primarily for broadcasting information over the Internet

TCP (Transfer Control Protocol) focuses on software, assuring that the data actually gets sent, is delivered, and is correctly reassembled at the other end. If any errors occur during transport, such as degradation or loss of some packets, TCP will call for those packets to be resent. TCP also works to optimize network bandwidth by dynamically controlling the flow of information, slowing it down as the network becomes congested.

HTTP (HyperText Transfer Protocol) is the protocol used to transfer **hypertext** files on the Web. Hypertext, invented by Ted Nelson in the 1960s, is the standard for content on the Web—i.e., Web pages are hypertext files. Hypertext is a database system by which related "objects" (text, pictures, music, video, etc.) can all be linked together, offering a non-linear experience of the information. In general usage, hypertext has come to refer to any text that contains "links" to other documents. Since hypertext is interactive, it requires a two-way communication protocol that allows the receiver to communicate back through the network, to call up the URLs for the objects to which the hypertext is linked. Take care not to confuse HTTP—a protocol—with **HTML** (HyperText Markup Language), which is the computer **language** typically used to code Web pages, defining how they look and how they behave.

FTP (File Transfer Protocol) is an efficient protocol used for transferring files between two devices over the Internet, and does not rely on the Web. FTP is used for a wide variety of purposes, as it is handy for the upload or download of most any type of file. FTP is typically used for transferring Web pages from the computer on which they were created to the server that will host the site of which they will be a part.

HTTP and FTP cannot be used for true streaming

HTTP and FTP are both layered on top of TCP, a protocol that ensures the retransmission of lost or damaged packets. Audio and video content transmitted using these protocols will be received intact—eventually. Lost or damaged packets are simply resent. This works just fine for the traditional "download it and then play it" modality, where the file will not be played back until it is received in its entirety.

When HTTP is used for progressive download, the very reliability of the protocol can cause problems, as the retransmission of lost or damaged packets may disrupt pseudo-streaming playback. The media file will still—eventually—arrive in its entirety, can be saved, and can then be smoothly played back from the destination hard drive, just as it could be with a traditional download. So, as reliable as they are for document transmission, HTTP and FTP cannot be used for true streaming, because their error-correction schemes undermine the temporal relationship between video and audio packets. They provide no facility for accommodating the data transfer rate to the audio or video data rate—that is, the rate at which the data comprising the media plays back. Using HTTP or FTP, a one-minute movie might download in one second, one minute, or even take an hour, depending on the size of the file and the speed of the connection. If the connection speed is less than the media's data rate, the media still makes it to the client, progressive download just won't play smoothly.

RTP enables streaming

RTP (Real-time Transfer Protocol) transmits data in real time. This means that a one-minute movie is transmitted in one minute. Layered on top of **UDP** (User Datagram Protocol), rather than on top of TCP, RTP provides very little error-correction; lost, late, or damaged packets are simply dropped. Whatever data arrives is what plays. So, when we say that streaming is a "forgiving" medium, we are referring to the fact that it will keep on playing, even if it "skips a beat," and that these losses are tolerable—even preferable, to the delays that would be imposed using TCP. In fact, "Live-Live" streaming would be impossible without this tolerance. Of course, if the connection speed is lower than the data rate, playback will be "jerky," or the media might not play at all.

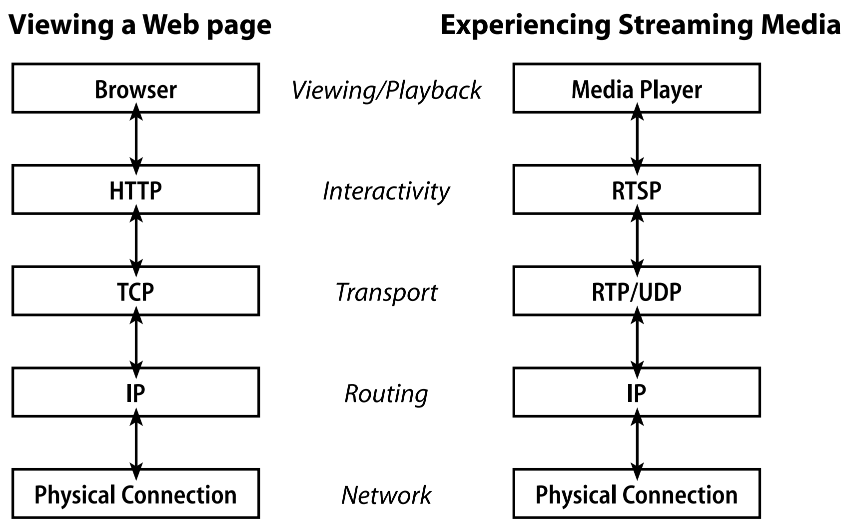
The other side of the coin, however, is that the media is always streamed at the constant data transmission rate needed to match the media's data rate—i.e., in real time. So, with a faster connection, there may be bandwidth left over. One of the advantages of streaming media is the predictable load per stream, on both the server and on the network.

One of the disadvantages of RTP-enabled streaming is that many network firewalls block the transmission of UDP information, although this is changing as the corporate world embraces streaming.

RTSP controls streaming media

While RTP enables a one-way stream, transmitting media from the server to the client, **RTSP** is a two-way protocol (similar to HTTP) which uses TCP to communicate, and is usually layered on top of RTP. According to its developers, "RTSP acts as a 'network remote control' for multimedia servers."[20] Like a VCR remote control, RTSP provides mechanisms that allow individual end-users or designated Web conference participants to specifically request streams from one or more servers, as well as a specific transport type and destination(s) for delivery of the data; request information about the data in a format-specific fashion; start, stop, and pause the delivery of the data; and gain random access to various portions of the data (where applicable—not, for example, in the case of a real-time live feed).

This very simplified chart shows how protocols are layered one on top of another and, on the protocol level, how the system for viewing a Web page relates to the system for experiencing streaming media.



Unicast and Multicast

When media is **broadcast**, a single stream is transmitted to all clients on the **network**. To understand what broadcast means, one need only think about our traditional use of the term "broadcast media," where the network is a cable or satellite system, or the airwaves, and the clients are television or radio receivers. Because only a single stream is transmitted, all the clients that are "tuned in" experience the same portion of the media program, at the same time, whether the program is live or pre-recorded.

When media is streamed over the Internet, it may be either **unicast** or **multicast**.

Unicast: A unicast is a one-to-one "narrowcast," in which each end-user gets a separate stream—even if they are experiencing the same media simultaneously. Because they each get their own stream, end-users *can* be given options for controlling the media, such as the ability to pause the stream, replay portions, or to jump to a different part of the program. This type of control is, however, only possible with pre-recorded content made available on-demand, and this flexibility comes at the expense of both server capacity and bandwidth—every end-user must be served and sent a discrete stream.

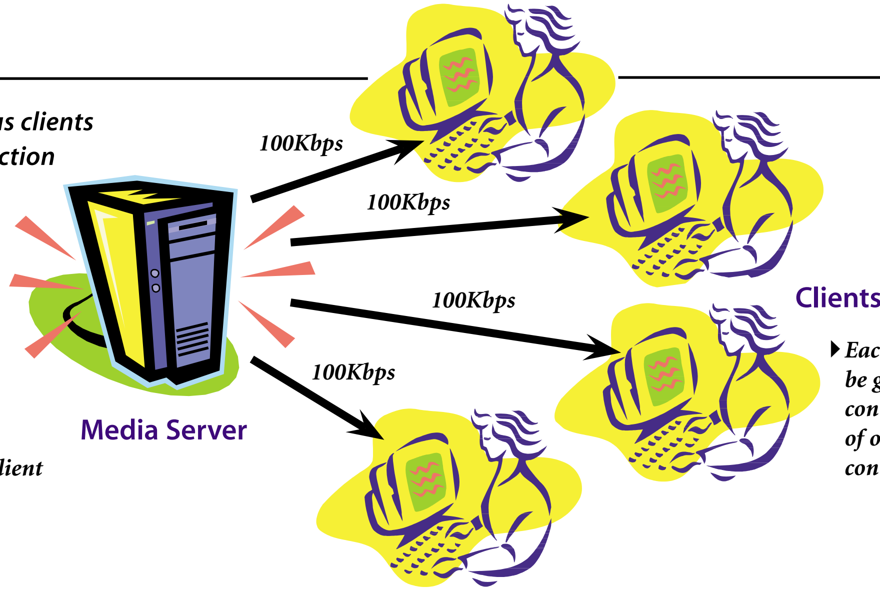
Multicast: Multicasting, or "IP multicasting," is also considered a narrowcast strategy, and it is designed to conserve both server processing capacity and bandwidth. The server transmits only one stream, which is replicated by special routers throughout the network, to be distributed to groups of multiple end-users. Multicasting does not allow the same flexibility for the end-user as does unicasting—every end-user must experience the same content concurrently.

20 "Real Time Streaming Protocol (RTSP)", an Internet-Draft, by H. Schulzrinne, A. Rao, R. Lanphier, Columbia U./Netscape/RealNetworks, Internet Engineering Task Force, 1998, PDF available at http://www.cs.columbia.edu/~hgs/rtsp/

Unicasting

4 x 100Kbps simultaneous clients requires 400Kbps connection from server

- *Best choice for on-demand media*
- *Each client gets a different stream, even if they are watching the same movie*
- *Heavier load (CPU and bandwidth) on server per client*

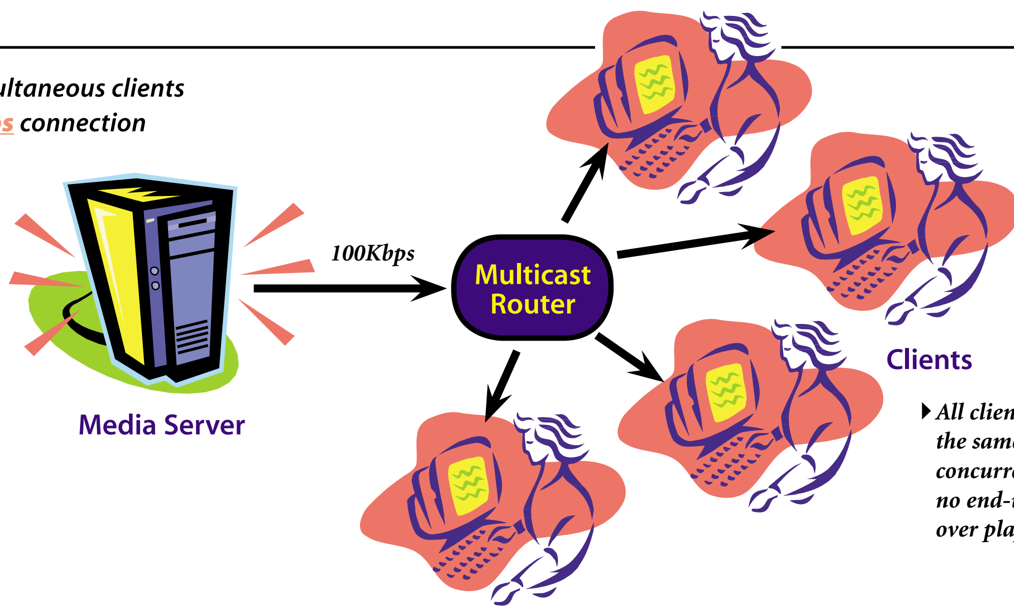


- *Each client can be given individual control over playback of on-demand content*

Multicasting

4 x 100Kbps simultaneous clients requires 100Kbps connection from server

- *Best for live or scheduled media*
- *Each client gets the same stream*
- *Conserves CPU processing power and bandwidth at server*



- *All clients playback the same content concurrently, with no end-user control over playback*

Multicasting is an excellent method for delivering the same content to multiple clients at the same time. The server sends only one data stream, whereas using unicast, the server must send a redundant stream for each connected client. So unicast results in a high server CPU load as well as increased network bandwidth demands at the server. Broadcast would solve the problem of duplicated streams, but would end up flooding the entire network, even if only a few end-users wish to receive the content.

Multicasting is only possible if both the streaming software and the network support it—not all systems do. On the software side, the latest versions of the big three architectures—Apple QuickTime, Microsoft Windows Media, and Realmedia—do support multicasting. In terms of the network, to multicast over an intranet, routers must be upgraded or replaced with multicast-enabled devices. While this adds yet another expense to streaming implementation, for those organizations looking to reduce the high costs of training and/or other essential enterprise-wide communications, the return on investment may be well worth the expenditure. Furthermore, multicasting can be used for a wide variety of other applications, in addition to streaming media.

HOW DO I MAKE (GOOD) STREAMING MEDIA?

WHAT AFFECTS STREAMING MEDIA QUALITY?

If you had never tasted ice cream, you would have no idea how to judge the quality of your first cone. Just like early television viewers, who were delighted with black and white, difficult to tune-in, often fuzzy transmissions, your first ice cream might seem truly delectable—but your estimation of quality might change dramatically after you had sampled a richer, creamier brand. Before we can address how to achieve the best quality streaming media, we have to consider what kind of quality to expect.

"Someday, when you go online, you'll enjoy the same quality and variety of video you now see on cable TV… Someday. But not today!… As yet, there is no elegant, cost-effective, and entirely satisfactory appliance that integrates TV, phone, and computer. However the goal of integrating and standardizing all media is overwhelmingly guiding hardware and software developers' efforts in all three media… So don't worry, that someday is coming very soon."

—Video Software Laboratory[21]

Full-screen, broadcast-quality: It's not completely science fiction—if you and your organization have the funds and the wherewithal to upgrade your intranet to a fully streaming-enabled network, complete with multicast routers, adequate media server capacity, and enough bandwidth to support your unicast and/or multicast requirements, then you can expect to achieve full-screen, broadcast-quality video on every up-to-date, streaming-optimized desktop in your organization. You can also expect this type of quality if you are streaming very selectively, only to those end-users who will be connected through a reliable, outsourced, streaming-enabled broadband network, and who will be viewing on broadcast-quality-streaming-capable desktops. Chances are, if you are in this category today, you work for a sizeable enterprise that is ready, willing, and able to commit to streaming as a business-critical component of its communications strategy. If you are one of those few, you can pretty much expect the premium, gourmet-quality ice cream experience of your favorite buy-it-by-the-pint brand.

Not anywhere near broadcast-quality: The rest of the world is in the "Wow!-It's-Web-video!-Right-here-right-now!-So-who-cares-if-it's-kind-of-small-and-choppy" category. Don't expect high production values to come across. Expect the satisfaction level of the cheap, off-brand, sundae-in-a-cup you might find in the freezer case of an out-of-the-way convenience store, on a long, hot, dusty, rural road trip—it's really great because that's all there is. At least for now. Of course there is a substantial difference between the quality you can expect when the client is on a powerful PC connected via broadband, vs. the quality you might expect on a slow machine connected via a 28.8 dial-up modem. But, as the world continues to embrace broadband, as more of the Internet becomes native multicast-enabled, and as the instrumentality of the streaming infrastructure becomes more commonplace, your expectations for better quality should rise. Meanwhile, it's worth doing what you can to optimize the experience from every angle, because audiences out there ***do*** want streaming media now—even though it is not the broadcast-quality experience they know as television… yet.

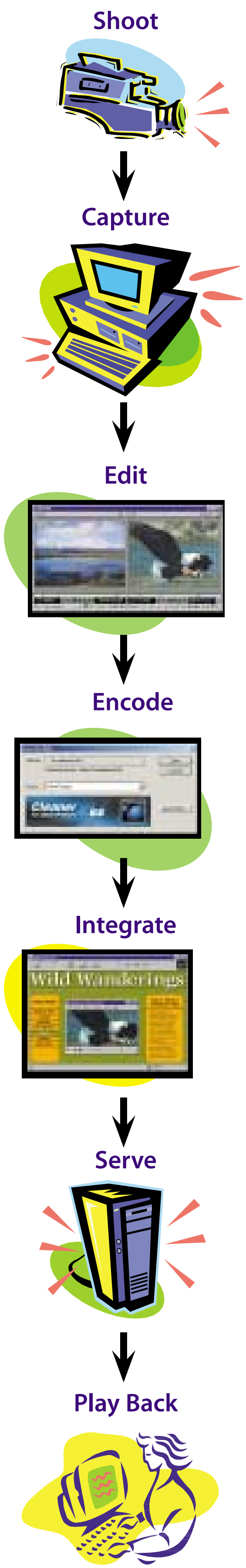
What affects streaming media quality? Everything! Video content, video quality, the type of compression used, server types and connections, how many people are viewing simultaneously, the Internet, the user connection, and the user platform—all these, and others, are factors that will affect the quality of your streams. The fact is, every step in your workflow can affect the quality of your streaming media, from planning the project to how you choose to transmit the data streams across the network, whether over the Internet or your own intranet. There ***are*** ways to help compensate for the some of the peculiarities of streaming media. The next several sections discuss the steps in the streaming media workflow, with particular emphasis on how to keep quality as high as possible.

"It is important to decide what constitutes acceptable delivery standards. If the limit is a delivery platform for narrowband, you will be accepting severe performance limitations. Conversely, if a higher standard is set, such as residential broadband, you will have more flexibility."

—H. Peter Alesso, e-Video[22]

21 *http://www.video-software.com/proj17.htm*

22 *e-video, Producing Internet Video as Broadband Technologies Converge*, by H. Peter Alesso, Addison Wesley, 2000



THE STREAMING MEDIA WORKFLOW

The first stages of the streaming media workflow are, essentially, identical to those used for creating any type of digital video production. To learn more about digital video, check out the *Adobe Digital Video Primer* that can be found on the Adobe Web site.[22]

There *are* some special considerations to bear in mind as you plan, shoot, and edit video specifically for the Web—considerations that can mean the difference between "Compelling!" "Classy!" and "Way cool!" versus "I wouldn't waste my time watching that." This section covers some of the basics, and provides some tips to enhance the quality of your streaming media productions.

Pre-production → *Production* → *Post-production* →

Plan the Project	Create/Gather Raw Material	Capture/ Import	Create/ Composite	Assemble/ Edit	Output and Distribute
• Outline • Script • Storyboard • Budget/Finance • Cast • Costumes • Sets/Props • Locations • Logistics	• Shoot raw video/stills • Procure stock footage • Create elements such as 2D/3d animations and titles • Record/ procure audio	• Digitize analog video • Import DV • Import elements • Import audio	• Motion graphics • Visual effects • Sophisticated, multi-layered compositions	• Develop a rough cut • Perform non-linear editing (NLE) • Author multi-media and Web content • Mix audio	• Output to a variety of media: • Video • Film • CD-ROM • DVD • The Web • Output an edit decision list (EDL)

The chart above shows the steps in the digital video workflow. These steps are fully detailed in the *Adobe Digital Video Primer.* In this primer, we will only cover considerations that are critical to creating good quality streaming media.

The chart below expands the workflow that happens in the last box in the chart above. It shows greater detail in regard to publishing and serving your streams, integrating your streaming media content into the Web visitor experience, and managing your Web site and your streaming media content, monitoring usage and adjusting accordingly.

Prepare → *Output* → *Distribute* → *Manage* →

Assess/Plan	Pre-process	Encode	Publish	Integrate	Manage
• Assess target audience re: > platform > connection speeds • Select parameters including: > target bit rate > balancing audio/ video bit rates > image size > frame rate, > keyframes	• Pre-process live feeds • Perform any pre-processing not completed in post-production—get proper image parameters, adjust color and brightness, filter and enhance audio, etc. • Test early, test often	• For an Internet audience, output for multiple formats at multiple bit rates • For a homogenous intranet audience, output in selected format at optimal bit rate • Don't forget to test	• Post encoded files to media server (if available in-house, or work with hosting outsource) • Test streams on minimum system you plan to support	• Develop Web navigation and UI support for streams • Create interactive UI elements needed to access and augment streaming content • Assemble Web pages incorporating interactivity and links to streaming content	• Monitor Web site traffic • Manage content

22 *Adobe Digital Video Primer*, PDF available for download at *http://www.adobe.com/motion/main.html*

IF IT ABSOLUTELY, POSITIVELY HAS TO BE STREAMED LIVE

If it's really got to be live-live, you'll be capturing video as it's shot, mixing audio, switching input signals from multiple devices, adding titles, pre-processing and encoding, and sending your production out to the host for streaming over a network or the Internet—all "on the fly." As you might imagine, this can complicate matters somewhat… You will probably want to invest in a specialized system—or turn to an expert outsource equipped to handle the rigors of live Webcasting.

Systems range from the multi-million dollar, state-of-the-art setup CNN uses… all the way on down to the "turnkey home webcasting" system for $99 (add your own computer) described by John Townley for *Streaming Media World*—it's a solution, says Townley, that "can give you anything from a lot of fun for yourself, family, and friends to the beginning of a career as a content provider and broadcaster… you'll have it up and rolling in no time… a hands-on intro to the ins and outs of what video streaming is all about."[24] Such turnkey home webcasting systems, often touted as "Webcams," are the perfect solution for cat lovers seeking a remote monitoring system for the litterbox (check out *http://cats.about.com/pets/cats/cs/catcams*) and *maybe* even for streaming your kid's birthday party live to the grandparents—*if* you can keep the little tyke close enough to the camera and in the frame for long enough to be recognized.

In between the multi-million dollar solution and the family fun pack are enterprise-class solutions, for tens of thousands of dollars, that can capture source media in real time; handle pre-processing, from real-time color correction to digital watermarking; and perform single-pass encoding into multiple formats and bit rates.

Or, if you want to go on location and stream a live show, you'll find there's even an all-inclusive system-in-a-suitcase that integrates an entire TV production studio into a rugged, portable package for under $25,000.

Outsourcing with an experienced Webcasting firm is generally a safe choice. Choose one with the equipment and the now how to make sure your live Webcast event all goes as planned.

"One of the first misconceptions about streaming video is that it is just an add-on. By that, I mean that many people think that you can take any video short and just 'add-on' streaming at the end of the process… The reality with streaming video is the reality you would find when producing content for any medium. You get out of it what you put into it. Good video doesn't just happen. Good video is designed. The choices you make when planning, shooting, and editing your production will have a profound impact on what happens when you try to make it stream."

—Tim Kennedy
Streaming Media World[23]

PRE-PRODUCTION

Don't skip the planning

Production is the actual shooting of video—and all the related activities, such as lighting, blocking, and sound recording. It follows, then, that **pre-production** is everything that should be done ***before*** you begin to shoot. In the typical digital video workflow, pre-production includes such activities as generating the concept and, often, "selling" it; developing an outline and/or a script; budgeting; and planning the shoot. Even if you are shooting a very short bit of video—such as a brief message intended for streaming—it is important to plan. Plan schedules, transportation, personnel, equipment rentals, etc. Scout locations. Make sure you have a background that will work well for Web compression. Make sure there is enough power to handle your cameras and lights. Make sure everyone knows what to do, and when to do it. Make sure your actors or subjects are fully briefed, know what to wear, and understand how to behave, based on the limitations of the medium.

Plan to keep it short

Your audience is likely to have a short attention span for media in the limited size and compromised quality you'll be delivering. Unless you are developing streaming media for a "controlled environment," such as an intranet, or you know that all of your viewers will have a high-speed connection, we would recommend that you limit your streaming presentations to no more than 3-4 minutes, or even less.

Should it stream live or as VOD?

"Does it have to be live? Why do you think it has to be live?" According to an interview in *DV Web Video Magazine,* these are the first questions Pat Paulson, one of the industry's foremost Webcast experts, asks his clients. Paulson spent years covering live events for television and film, before venturing into Webcasting in 1998. Since then, he has overseen the Webcasting for the American Express concert series and for artists such as Melissa Etheridge, Sheryl Crow, and Ozzy Osborne. Phil Fracassi, interviewed by *DV Web Video* for the same article, is senior vice president of digital programming and operations for House of Blues Digital, a business unit of the burgeoning media empire founded in 1992 by comedian Dan Akroyd and his cronies. House of Blues expects to Webcast between 800 and 1000 concerts in 2001. These guys ***know*** what they're talking about!

"Most of the events we do are to tape, and then we schedule like any network would. There are a lot of reasons for that," says Fracassi. "…for a lot of the festivals… we'll have a mixture of realtime live and tape delay… we prefer just to tape it all, get it back to the office and encode it…"

23 "Streaming Basics: Shooting Video for Streaming," by Tim Kennedy, *Streaming Media World,* January 12, 2001, *http://smw.internet.com/video/tutor/streambasics1*
24 "Turnkey Home Webcasting," by John Townley, *Streaming Media World*, January 22, 2001, *http://smw.internet.com/video/reviews/spotlife/index.html*

MANAGING EXPECTATIONS: A TRUE STORY

NASA (the U.S. National Aeronautics and Space Administration) is incorporating streaming media into its communication strategy. With 15 separate centers across the country, reports **Streaming Media Magazine,** ***the agency hopes to use streaming media to help weave together the efforts of its far-flung staff, while adhering to the fallout of post-Cold War budget cuts that mandates finding ways to do everything "faster, better, cheaper."***

"On January 11, 2001, NASA chief administrator Daniel Goldin and chief engineer Brian Keegan entered the auditorium of the government agency's Washington, D.C. headquarters for a streaming videoconference. After screening a pre-recorded interview segment, the webcast went live, and Goldin and Keegan responded directly to staff questions about a failed Mars mission and a report highlighting NASA's communications problems.

"Agency engineers encoded the video feed from NASA's Betacams and sent it through the agency's wide area network to more than 300 employees in 13 facilities throughout the United States… This was the first time Goldin had used webcasting to discuss a controversial agency-wide report with his staff. It was a test case for NASA's use of streaming media."

The **Streaming Media Magazine** ***article recounts how NASA Webmaster Charlie Redmond and a NASA computer engineer, Alan Federman, got the agency on-track for streaming. Managing the expectations of the "head honcho" was a critical part of their planning process.***

"…In December, Redmond and Federman showed Goldin streaming video at several different bandwidths. When he spoke on camera about the Mars report, Goldin wanted staff to 'see his eyes move.' Clearly, 56Kbps would not achieve the desired result. 'When you're talking to someone about a crucial aspect of their job, gestures like frowns and squints are significant,' says Redmond. 'Dan likes meeting small groups of staff in person, face to face, and he wasn't happy with jerky, low-bandwidth video…' "

—Jason Thomson
"2001: A Streaming Odyssey"
Streaming Media Magazine[26]

"I think we'll definitely be moving to more VOD [video-on-demand]," states Paulson. "If you think about it, how much of the TV that you watch is live, besides the news? There's a reason for that, and the reason is quality control. [Unless it's] breaking news… you want to take time to promote the event, you want to take time to optimize the encoding. That way, when you deliver it, you're delivering the best-quality product, and you can definitely remove some of the uncertainty that goes along with the live Webcast."[25]

Manage expectations

Perhaps the most important difference in planning for streaming media vs. traditional media productions is in the area of "managing expectations." Make sure everyone has the same objective—to put a specific video and/or audio production on the Web, for reasons that make sense—e.g., the Web is really the best way to distribute the production to your intended audience. Make sure everyone with a vested interest in the success of your streaming media venture knows what to expect in terms of the quality of the finished production, as it is likely to be seen by your intended audience.

Know your audience; know your limitations

Before you pick up your camcorder to shoot, establish at what data rate your video will be streamed. If you are targeting a broad range of Internet users, the overwhelming majority of whom, in 2001, are still connecting with 56K or 28.8K modems, then you should definitely stick to a very simple video design plan. If your audience is comprised solely of viewers on a corporate intranet, or only those with high bandwidth connections, you can plan to take a little bit more license with motion and transitions. But even high-speed connections sometimes suffer from network congestion, so you should still plan your streaming production to be comprised of just a few, fairly static, shots.

PRODUCTION

The next phase in the workflow is "Creating and Gathering Raw Material." This phase may involve selecting pre-recorded footage to incorporate into your streaming project or shooting original material. Creating or selecting video can be very exciting, but it's extremely important to remember that when it comes to streaming media, not all "good" video is going to shine—temper your enthusiasm and your creative impulses with an understanding of what will and won't work well.

It's a balancing act: complexity vs. quality

Minimalism is a an artistic style or technique that is characterized by extreme spareness and simplicity. A minimalist approach most often makes the best streaming media. In other words, ***keep it simple!***

One reason is the small size at which most streaming media will appear on the end-user's computer monitor. Lots of busy detail simply won't be seen.

The other big reason has, at its core, compression—a full discussion of which we'll reserve for a later section. But, in order to understand some of the whys and wherefores of shooting good video for the Web, you need to understand some basic principles of compression. Succinctly worded by Logan Kelsey and Jim Feeley in their article for *DV Web Video Magazine,* "Shooting Video for the Web," the following two statements are all you really need to know about compression in order to shoot good video for streaming:

- The simpler the information is within an image, and the more each frame in a sequence resembles the frames before and after it, the easier the image and the sequence are to compress and the better the resulting compressed video will look.
- The more complex the information is within an image and the less each frame in a sequence resembles the frames before and after it, the harder the image and sequence are to compress and the worse the resulting compressed video will look.[27]

25 "Webcasting Live Events," by Kevin Seal, *DV Web Video Magazine*, November 2000
26 "2001: A Streaming Odyssey," by Jason Thomson, *Streaming Media Magazine*, May 14, 2001, *http://www.streamingmedia.com/article.asp?id=7315*
27 "Shooting Video for the Web," by Logan Kelsey and Jim Feeley, *DV Web Video Magazine*, May 2000

"Here's the scenario: The opening scene of a nature documentary shows helicopter shots of antelope running across the African landscape. The sun is beginning to rise and the camera swoops over the savanna as clouds of dust trail the stampeding herd. If the production company is preparing to stream this nature video across a 28.8kbps or 56kbps modem, the most important feature to convey to the viewer is:

A: The feel of the African savanna at sunrise

B: The pounding antelope hooves contrasting the serenity of a rising sun

C: You are looking at antelope

If you chose C, you are correct. Cinematographers and Directors have a selection of zooms, fades, and fancy camera tricks to choose from— streaming media producers do not (especially at low bandwidths). The best streaming media is produced from very simple shots."

"Attempt to correct problems while shooting video, rather than attempting a resurrection during the edit or encode. Remember, streaming video production should not be approached the same as television or film production. The technology requires producers to meet these challenges with new techniques."

—Michael Long
Streaming Media.com[28]

As Kelsey and Feeley warn, artsy "*Blair Witch*, MTV, and even VH1 'looks' are out." You've got to avoid the stumbling blocks that might lead to poor quality, but you don't want to make everything so static that you bore your audience with dull content.

A bit of out-of-the-box thinking can go a long way. For example, Kelsey and Feeley suggest using a short depth of field and a lot of distance between the subject(s) and the background. While the subject stays crisp, the background elements are thrown into soft-focus. The result is recognizable faces or objects, an interesting composition, and video that's not difficult to compress.

Streaming media *can* be "forgiving"

Here's a piece of good news: You're ***not*** shooting for a 20-foot high screen filled with a gargantuan close-up of your CEO's face, so the flaws about which he or she might be self-conscious are likely to be miniaturized and compressed right out of existence. What this really means is that you ***can*** produce video of high enough quality to make good streaming media without extremely high-end equipment; without an army of makeup artists, hairdressers, and other behind-the-scenes support crew; and without a Hollywood-size budget. While the old, familiar video-production maxim, "Garbage in—Garbage out" is still valid, streaming media can be much more forgiving than film or video destined for larger formats, higher resolutions, and faster frame rates.

If possible, keep it digital

If you are shooting original footage for your streaming media project, you can use either a **DV** (digital video) camcorder or, literally, any video camera you can lay your hands on. Today, you can get a good DV camcorder for under $1,500, and these reasonably-priced devices shoot and record video as digital information—on DV casettes—offering the advantage of the IEEE 1394 connection. IEEE 1394 (aka FireWire or iLink) lets you transfer DV directly onto the hard drive of your computer for editing and encoding, without risking any of the compression-related **generation loss** or **artifacts** you might encounter when capturing analog video. You'll find IEEE 1394 built into many newer model computers, or you can connect using an inexpensive IEEE 1394 **capture card**.

If you do shoot with a traditional analog video camera, or you want to incorporate analog video from tapes into your streaming media project, you'll need to capture it—translate it from analog to digital form as you transfer it to your computer—using an analog-to-digital capture card. These cards are a bit more costly than IEEE 1394 cards, and the conversion process can degrade the quality of the video somewhat, but due to the nature of the streaming process, the differences between analog and DV source material can be minimal.

Will a better camera make a difference?

Maybe. Some people say it's a waste to shoot for streaming on a high-quality video **format**. According to experts Logan Kelsey and Jim Feeley, "Those people are wrong." Kelsey and Feeley suggest that you do ***not*** shoot Hi8, S-VHS, 3/4-inch, or VHS. "At a minimum, shoot in DV. Consider shooting in DVCAM, DVCPRO, Digital-S, or Betacam SP. The advantage of these last formats is that you get access to really good cameras… The DSP and other advanced features in these cameras help compensate for poor lighting conditions. Cameras with skin-tone filters and the resultant smooth facial detail produce especially easy-to-compress images… professional cameras get you access to better lenses. Industrial and broadcast lenses use better glass and give you more control over focus, iris, and zoom…" But Kelsey and Feeley go on to say, "If you work mostly with small camcorders, don't despair… definitely shoot in DV, DVCPRO, or DVCAM. And pay extra special attention to lighting and framing. Modern DV camcorders are quite nice, and you can always rent a better camera for key shoots."[29]

28 "How to Shoot Audio and Video for Streaming Over the Internet," by Michael Long, Overdrive Media, Inc., *Streaming Media.com, http://www.streamingmedia.com/tutorials/prod_shooting.asp*
29 "Shooting Video for the Web," by Logan Kelsey and Jim Feeley, *DV Web Video Magazine*, May 2000

"Video in essence is about light, and no matter what quality-level you're shooting for, you need to consider how you light your subject. If you're shooting beaches in the Greek islands under the world's most stunning natural light, you have few worries a little sunscreen won't cure. But if you shoot a talking head in a flourescent-lit office, or a band on a stage rigged with a panoply of different colors and levels of lamps, you're going to spend thought and effort getting good footage."

—Tim Tully
Streaming Media.com[30]

"…remember that the detrimental effects of strong colors and complex patterns will be even more severe than those same effects on tape."

—Logan Kelsey and Jim Feeley
DV Web Video Magazine[31]

"…you'll always get the best results if you start with a clear, clean audio recording, preferably using a separate microphone that isn't mounted on the camera. Keep the audio level as loud as possible without driving the meter into the red—this is especially critical in DV, where too much level can result in some particularly nasty distortion."

—Steve Cunningham
Streaming Media.com[32]

If possible, shoot exclusively for streaming

In many situations, video will be shot for a wide variety of uses, and intended for output to several different mediums. But the "keep it simple" rule that works so well for a short streaming media project, might yield footage that would translate into a very boring production for a longer, larger format production. Conversely, if streaming media is not the primary objective of the shoot, you might not end up with any footage at all that will stream well.

So, if you are shooting an event or other production with the intention of creating multiple versions for a variety of uses, and if you have enough budget and peoplepower to do it, you should consider shooting concurrently with multiple cameras and/or crews. Have one camera or crew shoot for the best streaming media footage possible, while the other camera or crew shoots the wider angles, pans and zooms, and gets more of the action you'll want for the videotape or DVD version.

This is not such an unrealistic notion when you consider that adequate-quality DV camcorders are so well-suited to capturing footage for streaming, and that they are so reasonably priced for either purchase or rental.

Keep the lighting as even as possible

While lighting effects such as complex shadows or lens flare may add interest to the composition of many of the movies you admire, streaming is not the place to experiment with your tricky lighting repertoire. Those complex shadows, hotspots, and other forms of contrast just make the footage more difficult to compress, leaving unwanted artifacts rather than added interest. The more even the light is, the better it will compress. Use bounce cards to minimize hard shadows outdoors; indoors, keep your lighting soft. Avoid camera-mounted lights—they generate hotspots unless heavily diffused. A standard 3-point lighting kit that can be rigged with umbrellas, bounce cards, and/or diffusion is a worthwhile investment that will handle most any situation you'll encounter when shooting tight subjects for streaming media.

Action! (just don't move around very much)

The whole point of putting video on the Web, rather than just still images, is the added dimension that action brings to the experience. But, when compressed, more action in your video—whether that action is due to subject movement, camera movement, or to complex transitions—equates to a higher data rate in the file you want to stream. So, in order to successfully stream action-packed video, you would need an audience you're sure will be connected at a high bandwidth, and/or you would need to make the size of your image frame smaller. Unfortunately, the current limitations of the Internet mean that streaming a video of an athlete's "talking head" to endorse your footwear product will work much better than streaming a video of the team in-action on the playing field. Video with relatively stationary subjects will produce better streaming media than video with lots of motion or rapid scene changes.

Consider using "bluescreen"

If you have the equipment and the expertise, **bluescreen** can be an excellent technique for achieving quality streaming video. According to Kelsey and Feeley, "By shooting on bluescreen [or greenscreen]—dropping the background and compositing your subject on top of a digital still image—you'll have a background that will be noise-free and rock-steady."[33] A background that does not change from frame to frame is easier to compress, and leaves more **bits** available for what does change—your subject. But Kelsey and Feeley also warn that lighting and shooting "good" bluescreen takes lots of experience, and the post work required can be time-consuming and/or costly. (Although we'd like to note that ***Adobe Premiere*** and ***Adobe After Effects*** software provide the tools you need to do it yourself with relative ease).

30 "Shooting Video for the Web: Equipment," by Tim Tully, *Streaming Media.com, http://www.streamingmedia.com/tutorials/view.asp?tutorial_id=35*
31 "Shooting Video for the Web," by Logan Kelsey and Jim Feeley, *DV Web Video Magazine*, May 2000
32 "Creating Audio for the Web," by Steve Cunningham, *Streaming Media.com, http://www.streamingmedia.com/tutorials/view.asp?tutorial_id=38*

SOME GUIDELINES FOR SELECTING OR SHOOTING GOOD STREAMING VIDEO

- **Minimize movement**—The less subject movement, the better. Avoid scenes with lots of activity or motion.
- **Use a tripod**—The less camera movement the better. If you really *need* to pan, a tripod with a fluid head will reduce motion to just one dimension.
- **Keep the background simple**—Within a small frame, subjects moving on a simple background are easier to distinguish than subjects on a complex or moving background.
- **Avoid complex textures and stripes**—Patterned clothing or scenic elements can cause distortion effects.
- **Avoid trees or foliage with moving leaves**—Very small details in continuous motion compress horribly, even at high data rates.
- **Avoid "hot" colors**—Bright whites, blues, reds and yellows glare under video lights, and will look even worse on the Web than they do on NTSC tape; darker colors and earthtones will work better. If you can control what the subject is wearing, recommend dark, solid colors.
- **Avoid harsh contrast**—Typically, video that is well-lit and does not have a lot of contrast will compress better for streaming. Brightly lit subjects shot at a constant exposure will help keep the foreground detail crisp.
- **Tight shots are best**—Streaming video is seen on a miniature screen. To your Web audience, a telescopic shot of a cast of thousands storming a hill will look like "chaos on an anthill." Close-ups work best, allowing Web viewers to recognize faces and expressions, or to distinguish objects and their details. People engaged in conversation should be positioned very close together, and be prompted to remain that way.
- **Keep your subjects within the frame**—Be sure your subjects or actors understand the boundaries within which they can move and still remain within the frame.
- **Create plenty of background space for titles**—The small image size means you'll need a larger percentage of the frame as a background for readable titles than you would for a production with a larger image size.
- **Stick to hard cuts**—Pans and zooms add movement that will not compress well.
- **Slow your shutter speed** —A slow shutter speed causes moving objects to blur, but leaves still objects in focus, making your video easier to compress. Try using a shutter speed that's about half your frame rate: shoot 24 fps film at 1/48 second; shoot 30 fps video at 1/60 second.
- **Don't neglect the audio**—Don't rely on a built-in microphone—use wireless remote mics or boom mics. Pre-test audio levels. Get a sound expert, if possible.
- **Test whenever possible**—Shoot some test video, encode it, and stream it across the system you plan to employ. If you have had trouble managing expectations, showing an early test can convince clients that compromises or concessions may need to be made.

Sometimes a *word* is worth a thousand *pictures*

You will be forgiven for bad video—but you will be abandoned for bad audio. While those reasonably priced DV camcorders do yield fine imagery for streaming, remember that consumer cameras have consumer microphones built-in. Poor quality **mics** tend to pick up lots of background noise, which can be virtually impossible to filter out later. Don't use a mic intended for rock and roll. Do use a remote lapel, handheld, or boom microphone with either a strong proximity effect or a directional pick-up pattern. If you are shooting DV, record 16-bit (not 12-bit) audio—the more information you start with, the better result you are likely to get when you encode your audio. Keep the audio level as loud as possible without driving the meter into the red—with DV, too much level can result in distortion. Be sure to test your audio levels prior to recording. Better yet, get an experienced audio engineer to help. If feasible, record as much audio as possible in post-production, rather than live. A voice-over recorded in a studio will always produce much better audio than sound recorded on location.

Shooting for live Webcasting

The operative word here is "live." Make sure your subjects, and the audience if there is one, know that your are Webcasting live. Because what your cameras "see" and what your microphones pick up is exactly what is being sent to the audience at large—on your corporate intranet or on the Internet.

POST-PRODUCTION

In the digital video workflow, **post-production** includes everything from capturing through outputting. However, for the purposes of this primer, we will address only capturing through editing in this "Post-Production" section, in order to cover the critical topic of outputting more thoroughly in a section of its own.

Capturing video for streaming

Once you've shot or otherwise collected your video clips, you need to get them into your computer, in digital format, so that you can edit and/or encode your streaming production. ***Adobe Premiere*** software, with native DV support for dozens of DV devices, as well as support for most analog capture cards, is an excellent choice for controlling and automating the capture process. If you are planning to use ***Adobe Premiere*** or other non-linear video editing software to craft your production for VOD (video-on-demand), you'll follow the same basic capture procedures that you would for any type of digital video project. You'll find a description of how to configure your system and what capturing your clips entails, in the *Adobe Digital Video Primer*.[33]

"If you shot DV, it's as easy as 'plug 'n play'—just connect your DV camcorder to your computer via an IEEE 1394 connection and you're ready to edit."

—Adobe Digital Video Primer

33 *Adobe Digital Video Primer*, PDF available for download at *http://www.adobe.com/motion/main.html*

SOME CAPTURE POINTERS FOR STREAMING MEDIA

- **Use AV hard drives**—SCSI or RAID
- **Research your video capture card**—Test it, if possible, before making a final decision.
- **Conserve CPU power**—Disable all anti-virus programs, screen savers, and extensions during the capture process.
- **Optimize your hard drive**—Defragment your hard drive before capturing.
- **Capture high quality**—Capture your source material (both video and audio) at the highest quality settings your system can handle—30 fps is recommended for video; 16-bit for audio.
- **Avoid dropped frames**—Test and improve your system until you reach a zero-dropped-frames standard.
- **Capture mono audio**—Unless you really, truly *need* stereo output.
- **Keep an archive of your raw video file**—For future use, and because "accidents" happen.

Compared to the final files you would output for other mediums, such as DVD-ROM or VHS tape, the final files you'll output for the Web will be small—a smaller frame size, a lower frame rate, and more heavily compressed. However, it is recommended that you capture your source material (both video and audio) at the highest quality settings your system can handle—30 fps is recommended for video; 16-bit for audio. You'll be using up lots of hard-disk space, but it's better to start with high quality (i.e., more data) and, therefore, have more choices about what information to discard when you reach the encoding stage, than to start with low quality (less data) and potentially regret having fewer options down the road.

Window size (resolution)	Frames per second (fps)	Color depth (bits)	Data rate per second
640 x 480	30	24	27.65 MB
640 x 480	15	24	13.83 MB
320 x 240	15	24	3.46 MB
160 x 120	15	24	865 KB

Make sure your system has adequate capacity. A 3.46 MB-per-second data rate is standard for DV. In terms of disk space, you'll probably want at least a 30 GB hard drive. The larger your resolution (video window size), the more colors, and the higher the frame rate captured… the bigger your captured file will be.

Dropped frames, during capture, are something you want to avoid. You can't recover dropped frames after capture, so you'll want to test and improve your system until you reach a zero-dropped-frames standard. If you are using *Adobe Premiere* to control capture, make sure to select the "Report Dropped Frames" option, so you'll know when frames are dropped. You can also set *Adobe Premiere* to automatically abort capture when frames are dropped. Dropped frames are usually the result of a configuration problem, or are caused by trying to capture video that exceeds the capture rate for your system. DV requires a hard disk capable of sustaining a 3.6 MB per second data rate.

Capture cards for streaming

You'll find there are capture cards designed specifically for streaming media. Your decisions will depend on your needs and your budget. But the industry is now mature enough that there is, indeed, something that's right for every requirement.

The most basic capture cards are for input only. Some popular, inexpensive ones (under $200) let you hook up a variety of sources at once (although only one of those devices can be used at a time) including S-video and full-duplex audio input. Such cards provide basic capture software. "But in order to stream, you'll need a solid video editing package, such as *Adobe Premiere*, and streaming software like Real Producer Plus, to refine and process the video before broadcasting from a site," reports Joel Strauch in his article for *Streaming Media.com,* "Video Capture on a Shoestring."[34]

More expensive capture cards may have input and output capabilities, and may perform a variety of ancillary tasks such as scaling (resizing), color conversion, de-interlacing, closed captioning and more. One popular capture card priced under $2,500 has such on-board pre-processing capabilities, leaving the bulk of your CPU power for creating higher quality video streams. These specialized streaming cards enable real-time (live) streaming from a DV source, with the ability to transfer the digital video directly to the host.

41 "Video Capture on a Shoestring," by Joel Strauch, *Streaming Media.com, http://www.streamingmedia.com/article.asp?id=6794*

Pre-processing your video

Pre-processing—sometimes called "**optimizing**"—removes non-essential information from your video and audio—information that is difficult to encode and/or does not substantively add to the quality of the streamed media. So pre-processing prior to encoding reduces the burden on the compressor, potentially saving time and CPU capacity. It also allows you to make some of the decisions about what information will be included in the encoded media, rather than relying solely on your compression software to make all the determinations.

SOME TYPICAL PRE-PROCESSING TASKS FOR STREAMING MEDIA

GETTING THE PROPER IMAGE PARAMETERS:

- **Cropping your video**—Removing 5% from the frame edges is a good starting point to get rid of edge artifacts you might not even be able to see without zooming in very tightly, such as tearing or black bands around any of the frame edges. These anomalies can occur during capture and create unnecessary pixels that would waste precious bandwidth.

  Pay attention to the *aspect ratio* when you crop, either maintaining the original and/or keeping the dimensions to multiples of four. Compression algorithms often divide images into blocks of 4x4 or 16x16 pixels. Keeping the dimensions to multiples of four assures that your aspect ratio won't be reconfigured during encoding to fit the inherent constraints of the compression software. If you need to scale the image to a particular window frame, make sure you maintain that aspect ratio—4:3, for example, for a 160x120 window.

  To crop your footage in *Adobe Premiere*, drag and drop the Crop filter from the Video Effects palette to the Timeline. Then, in the Effect Controls palette, adjust the crop setup to reduce the picture by 5% on each side, or to whatever amount you choose. You can also crop your movie globally by using the Special Processing functions in the Export Movie Settings when you output.

- **Image size**—Video is typically captured at a larger frame size than you are likely to output for streaming. If your audience is comprised only of broadband users and want large frames, you can compress your video at capture size. But, if you are streaming to users with slower connection speeds and/or you want to improve image quality and/or achieve higher frame rates, you may want to scale your video to a smaller size. The Cleaner 5 module of Adobe Premiere offers convenient scaling options. The pull-down the Scaling Quality option in Media Cleaner EZ 5 allows you to choose Fast, Normal or Accurate scaling. Unless you're under time constraints, use the Accurate function to anti-alias and to reduce jagged edges.

- **Deinterlacing**—Traditional video systems "paint" images on the screen in alternating lines, in a process called interlacing. Computer monitors do not use interlacing to display video and it should be removed from video clips that will ultimately be viewed on the computer. You can deinterlace in *Adobe Premiere* by selecting a clip in the Timeline, pulling down the Clip menu and choosing Video Options, then Field Options, and checking the Always Deinterlace box. Or you can deinterlace a project globally by choosing the Deinterlace option in the Special Processing area of the Export Movie Settings.

- **Inverse telecine**—When film (shot at 24 fps) is converted to video (at 25 fps for PAL; 29.97 fps for NTSC), additional frames are added, in a process called telecine. You can use inverse telecine to discard those extra frames that would just burden your encoder. You'll find information on how to do this with *Adobe Premiere*, in the Support Knowledgebase at *www.adobe.com*.

- **Two-dimensional filtering**—Low-pass, bandpass, and high-pass filters use mathematical functions to adjust the pixels of which images are comprised, in order to alter image qualities. By using these filters to selectively or uniformly blur images or sharpen edges, you can effect the differences, from frame-to-frame, that the compression software will have to deal with. Cleaner 5 offers three such filtering options: Blur, Unsharp Mask and Adaptive Noise Reduction, each with a variety of different settings for a wide range of flexibility.

ADJUSTING COLOR AND BRIGHTNESS:

- **Brightness, contrast, hue, saturation**—These are often subjective settings. Any adjustments needed depend on the source material and how it was captured. Experimentation is the key to finding the right settings for each project.

- **Gamma correction**—Applies a non-linear function to the amplitude values of the pixels, helping to create more whites without brightening the entire range or more darks without affecting the bright areas.

- **Black and white restore**—This is an option in Cleaner 5 that lets you adjust only the very white or very black portions of images, forcing values to pure black or pure white. The Amount slider determines what values will be affected by the operation, while the Smoothness slider adjusts values near the transition points. A typical use of Black and White restore is reducing noise in flat areas such as black backgrounds behind titles.

AUDIO PRE-PROCESSING:

- **Noise removal**—Extraneous noise makes the audio signal more difficult to encode and uses up precious bits. When encoded for streaming, the effects of noise are usually amplified, and can result in an unintelligible soundtrack.

- **Enhancing the signal**—Depending on how well your audio was recorded, you may need to uniformly or selectively boost or cut the signal to arrive at the best level to encode for the Web. You can also enhance the signal using a variety of effects, or filters, that alter different qualities of the audio, such as reverb or echo.

Adobe Premiere 6.0 includes a sound studio-style Audio Mixer and 20 audio filters that make it easy for you to pre-process your audio. The Media Cleaner EZ 5 module built into Adobe Premiere 6.0 also provides tools for optimizing your audio.

"As you mourn the loss of your effects, remember what streaming your content is all about. Your goal is to communicate a message. Effects cause unnecessary change in your streaming video. They force the video stream to put more resources into something other than what the true subject is of your video. Your subject is what communicates that message. Lose the effects. Focus on the goal.

Is there hope for the editor/artist who thinks special effects actually add to their message? Maybe. Increasingly, streaming players are beginning to incorporate built-in effects. Rather than rendering the effect into video pixels and encoding the digital mash for streaming, a player might take two streaming clips and render the effect directly on the viewer's machine. The result is a smooth dissolve or motion wipe. The catch is the viewer's machine might not be up to the task of figuring out all that effects math. As connection and computer speeds grow faster, you will see more and more options to use effects. Until then, stick to cuts."

—Tim Kennedy
Streaming Media World[43]

Pre-processing tasks may be performed at different points in the workflow:

- **As part of the capture process**—For live Webcasting, you'll need to set up automatic pre-processing functions between capture and encoding. Some sophisticated capture cards provide built-in features for handling and automating pre-processing tasks, whether for live Webcasting or for preparing captured video for editing.
- **As part of the editing process**—If you are capturing media specifically for use in a VOD (video-on-demand) project that you are going to edit, you'll probably do most, if not all, of your pre-processing at the beginning of or during the editing process.
- **As part of the encoding process**—If you are putting together a single production for multiple uses, you are likely to do your Web-specific pre-processing after editing, but before encoding your media for streaming. Similarly, if you are going to repurpose a previously edited project for streaming delivery, you may need to do some pre-processing before encoding, to make the production more suitable for Web delivery. But even if you do pre-process a multi-purpose production prior to encoding, it may still not stream well. Beware of trying to stream productions with too much action, lots of detail that will disappear within a small image, fancy transitions, motion-laden effects, or complex audio tracks. You are likely to have much better results if you capture and pre-process suitable raw footage and re-cut a short version of your production, specifically designed for Web delivery.

Adobe Premiere 6.0 provides a variety of flexible tools for pre-processing your video and your audio that can be conveniently used at whichever point in your workflow you choose. Additionally, ***Adobe Premiere*** includes a special, built-in version of Media Cleaner EZ 5, Terran Interactive's popular software for pre-processing and encoding media for the Web. Typical pre-processing tasks fall into three broad categories: 1) getting the proper image parameters, 2) adjusting color and brightness, and 3) audio pre-processing. The box below provides some details.

Editing video for streaming

Editing activities include cutting and pasting clips together to tell your story, inserting transitions between clips, incorporating visual effects, mixing and synchronizing audio with your video, and adding titles. The *Adobe Digital Video Primer*[42] provides a general introduction to the editing process.

"Non-linear editing (NLE) makes editing and assembling your production as easy and as flexible as word processing."

—Adobe Digital Video Primer

Assembling a video production for streaming is, essentially, the same process as for any other medium. ***Adobe Premiere*** software, with its robust "DV in, Web out" features, is an excellent choice for editing your video for streaming—or for any medium. ***Adobe After Effects*** software adds the industry-standard compositing toolset to your capabilities. And these two powerful, professional products deliver tight integration that streamlines your workflow.

But, bear in mind as you edit and develop your production, as you did when shooting for streaming, that you need to be cognizant of the limitations of the Web, avoiding unnecessary motion or change that will bog down streaming files.

42 *Adobe Digital Video Primer*, PDF available for download at *http://www.adobe.com/motion/main.html*
43 "Streaming Basics: Editing Video for Streaming," by Tim Kennedy, *Streaming Media World*, March 8, 2001, *http://smw.internet.com/video/tutor/streambasics2/index.html*

A FEW TIPS FOR EDITING STREAMING MEDIA

- **Skip the fancy stuff**—Complex transitions add motion. Try to stick with straight cuts as much as possible. Focus on content, not on effects.
- **Use slow motion or even still shots**—If you've got some shaky, handheld video you want to stream, try slowing it down, or even using still images. Slow motion footage does not change as quickly, and that makes it easier to encode; still images are, of course even that much easier.
- **Try matte techniques**—If you didn't shoot blue-screen or greenscreen, but you have suitable footage to mask your subject, you can composite in a still background to achieve the same end—conservation of motion for more efficient encoding.
- **Try matte techniques**—If you didn't shoot blue-screen or greenscreen, but you have suitable footage to mask your subject, you can composite in a still background to achieve the same end—conservation of motion for more efficient encoding.
- **Make titles larger and simpler, and let them linger longer**—Remember that Web viewers are looking at your titles in a potentially blurry, possibly as small as 160x120 pixel window. Choose large font sizes, simple bold typefaces, and add a little space between characters. Keep backgrounds simple behind your type, and be sure there's a significant difference in brightness between the type and the background. By keeping your titles onscreen longer, the encoding software can recognize them as unchanging "content and will better maintain details.
- **Try titling outside of the video frame**— Use event markers within your streaming video to launch titles in the form of still or animated GIFs, as part of the Web page surrounding the video window.

Processing and mixing audio for streaming with video

"You've just finished recording your latest audio masterpiece for the Web. But on playback the audio track sounds like an AM radio with dying batteries, and you realize it will sound even worse after encoding. What can you do?" Steve Cunningham answers his own question in his *Streaming Media.com* Tutorial "Creating Audio for the Web.[43] Check it out for descriptions of many of the different techniques you can use in post-production to improve a recording that isn't all you had hoped for.

"Streaming media, in its present form, is not about being subtle. In an audio mix of narration and background, always remember what pays the bills. That music is nice for establishing mood but it is the voice-over that sells, informs, or educates.

I am a big believer in clear mono sound. Stereo may be nice, but I don't think it currently adds much to the viewing experience for most streaming users. I would rather provide my viewers with more data resources dedicated to streaming the video than providing a stereo soundtrack."

—Tim Kennedy
Streaming Media World[44]

You'll find all the tools you need for processing and mixing your audio—including a full range of audio effects (aka filters)—in ***Adobe Premiere*** software. ***Adobe Premiere 6.0*** provides a professional-level audio mixing tool—the Audio Mixer, so there's no need to purchase a separate software package to handle your audio. The ***Adobe Premiere*** Audio Mixer resembles a sound-studio-style, multichannel mixer featuring gain and pan adjustments for up to 99 audio tracks, a VU meter on each track that lights up to indicate **clipping**, and automation features you can use to streamline your workflow. The Audio Mixer lets you make adjustments in real time, playing the audio while watching the synchronized video. You can also make adjustments in the Timeline, using convenient rubberband controls for volume, panning, or balancing audio. ***Adobe After Effects*** offers a full range of audio processing tools, as well, and allows you to generate compressed audio and MP3's for use in your streaming media productions.

The audio you've mixed may sound great on your computer's sound system, but by the time it gets encoded and streamed to an end-user listening with a 28.8 Kbps connection and a dinky little built-in computer speaker, it may have turned to mush. How will you know? Expert Tim Kennedy recommends that you "follow an old recording studio trick. Go out and get a small worthless speaker and plug it into your system. Then turn off those stereo speakers or put down your headphones and listen closely to what that little mono speaker is putting out. Grab a co-worker (who doesn't know the material) and get a second opinion."

When it comes to audio for streaming with video, experts agree on the following:

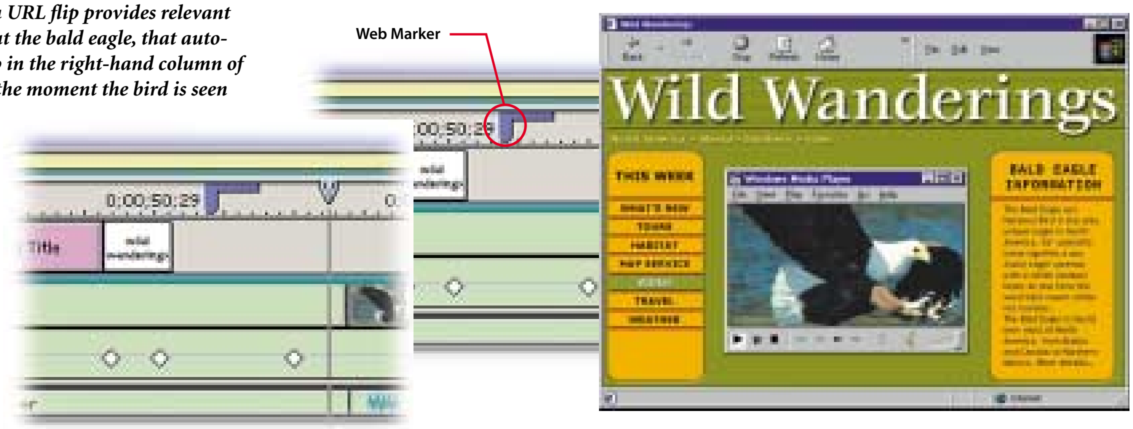
- **Avoid mixing multiple channels**—opt for clarity.
- **Make sure the message gets through**—Sacrifice or subdue background music in favor of clear narration that informs, educates, or sells.
- **Stick to mono**—If you're streaming your audio with video, don't waste the bandwidth on stereo.

44 "Streaming Basics: Editing Video for Streaming," by Tim Kennedy, *Streaming Media World*, March 8, 2001, *http://smw.internet.com/video/tutor/streambasics2/index.html*

Adding interactivity to your video production

Streaming allows you to take advantage of the interactivity made possible by the Web. **Chapterization** allows viewers to jump to the beginning of a sequence, or chapter, that is specified in your production with timeline markers. Adobe Premiere and Adobe After Effects also offer the ability to set Timeline markers that include links to HTML pages. Using these markers, you can develop streaming videos that automatically launch Web pages at precise points during playback. These **URL flips** may provide different backgrounds for your video presentation, even titles, or offer additional information that is relevant to the video content being viewed.

In this example, a URL flip provides relevant information about the bald eagle, that automatically pops up in the right-hand column of the Web page, at the moment the bird is seen in the video.



A single frame of uncompressed video takes about 1 megabyte (MB) of space to store. You can calculate this by multiplying the horizontal resolution (720 pixels) by the vertical resolution (486 pixels), and then multiplying by 3 bytes for the RGB color information. At the standard video rate of 29.97 frames per second, this would result in around 30 MB of storage required for each and every second of uncompressed video! It would take over 1.5 gigabyte (GB) to hold a minute of uncompressed video!

—Adobe Digital Video Primer

OUTPUTTING STREAMING MEDIA

Encoding compresses and formats media for streaming

You've finished editing and assembling your movie. Now it's time to produce the final file—or files—for streaming distribution via the Internet or your intranet. Outputting—or exporting—your production for streaming requires **encoding**. Encoding accomplishes two main objectives: 1) it reduces the size of your video and audio files, by means of **compression**, making Internet delivery feasible, and 2) it saves your files in a format that can be read and played back on the desktops of your targeted audience. Some encoding solutions may also be configured to provide additional processing functions, such as digital watermarking, for example.

What is compression?

The goal of compression is to reduce the data rate while still keeping the image quality high. Video for the Web might need to be compressed at a ratio of 50:1 or even more. **Lossless** compression, whereby, no essential data is lost (used for the transmission of financial data, for example) can compress no more than 30 percent. Fortunately, because human perception is based not only on what we actually see and hear, but also on what we infer, video and audio can withstand **lossy** compression and still maintain enough quality to be acceptable. Lossy compression can compress to any level, but the more compression is applied, the more quality is sacrificed.

There are many different methods of compressing video. One way is to simply reduce the size of each video frame. A 320x240 image has only one-fourth the number of pixels as a 640x480 image. Or we could reduce the frame rate of the video. A 15 frame-per-second (fps) video has only half the data of a 30 frame-per-second video. Both of these methods are used to compress video for the Web. For narrowband connections, a very small frame size—usually 240x180—is standard. While larger image sizes are often transmitted via broadband connections, the recommended frame rate for streaming video is still 15 fps.

43 "Creating Audio for the Web," by Steve Cunningham, *Streaming Media.com, http://www.streamingmedia.com/tutorials/view.asp?tutorial_id=38*
44 "Streaming Basics: Editing Video for Streaming," by Tim Kennedy, *Streaming Media World*, March 8, 2001, *http://smw.internet.com/video/tutor/streambasics2/index.html*

HOW MUCH COMPRESSION IS ENOUGH? UNDERSTANDING BANDWIDTH MATH[45]

In order to be sure your streaming media plays smoothly, you'll need to determine how much data you can send to your audience, without clogging up the system. The good news is that there are tools you can use so you don't really need to do any math, but it's a good idea for you to understand the underlying principles.

THE FUNDAMENTALS

File sizes are measured in "K" or "KB"—short for kilo*bytes* (1,024 bytes; usually rounded to 1,000 bytes). Since a byte is 8 bits, a kilo*byte* is (approximately) 8,000 bits.

Transmission rates are also measured in "K", but the "K" here is short for "Kbps"—kilo*bits* per second—not kilo*bytes*, but kilo*bits*. Note that there are exactly 1,000 bits in a kilo*bit*.

Don't mistake kilobytes for kilobits. A 56K (i.e., Kbps, or kilobit per second) modem won't download a 56K (i.e., KB, or kilobyte) file in one second. And it gets more confusing…

THE LIMITATIONS

A 56Kbps modem is rated to move 56 kilo*bits* of data per second. That's 56,000 bits. Divide by 8 (the number of bits in a byte) and you get 7KB (kilo*bytes)* per second—a more familiar, and therefore more meaningful measure of file size for most of us.

THE VARIATIONS

According to the quick calculation we've just done, a 56 KB image file would need *at least* 8 seconds to stream over a 56K modem...

Why did we say "at least?" Well, for all you math whizzes, a 56K modem won't download a 56K file in 8 seconds, because, as you'll recall, a *modem* K is exactly 1,000 bits, while a *file* K is actually 1,024 bytes and, with "start bits," "stop bits," and modem compression, a byte can use more or less than eight bits of bandwidth.

Furthermore, 56K modems do not consistently provide 56 kilobits of throughput—56K is the upper limit. The actual bandwidth fluctuates continually, frequently dropping well below 56 kilobits per second.

THE RECOMMENDATIONS

Check with the provider of your streaming architecture to ascertain what bandwidth is recommended. For example, Real recommends using 34Kbps for 56K modems, while Apple (QuickTime) recommends using 53Kbps.

THE MATH

Now let's suppose you've got 50KB of video to stream to a 56K modem. To determine how much time the sequence needs to stream:

1. Convert the "storage" measurement (kilobytes) to the "shipment" measurement (kilobits) by multiplying by a factor of 8, since there are 8 bits to a byte.
   50KB (kilobytes) X 8 = 400Kb (kilobits)
2. Divide your shipment by the bandwidth recommended for the targeted connection. We'll use Real's recommendation of 34Kbps for a 56K modem.
   400Kb ÷ 34 Kbps = 12 (11.76) seconds
3. Divide your shipment by the bandwidth recommended for the targeted connection. We'll use Real's recommendation of 34Kbps for a 56K modem.
   400Kb ÷ 34 Kbps = 12 (11.76) seconds

So it will take a little less than 12 seconds to stream the 50KB. If your video is comprised of three different images, each about the same size, and your transitions are equally spaced at 4-second intervals, it should stream quite nicely.

But what if your images are different sizes? For example: 17KB, then 25KB, then 8KB... If we do the math, everything will be fine for the first four seconds. But, for the next image, we may start to get into some trouble:

25KB X 8 = 200Kb
200Kb ÷ 34Kbps = 6 (5.88) seconds

Trying to push 6 seconds worth of material through the pipe in 4 seconds could clog up the pipe and/or choke the streaming player...

HELPFUL TOOLS

You could use a calculator and graphing software to carefully plot out kilobits vs. time... But, fortunately, clip analysis tools in software such as Adobe Premiere and Terran Interactive's Media Cleaner will graph the data rate of your movie for you. Using these tools, you can check portions of your program as you develop it, to be sure you stay within the required bandwidth limitations.

HELPFUL HINTS

- Don't forget to include the audio in your bandwidth calculations.
- Keep up with the latest documentation for the streaming format you are using—different formats handle bandwidth in different ways.
- Buy some time—put lightweight graphics and text that consume little bandwidth up front, to buy loading time for heavier-weight material that comes later. Your audience will be distracted while content is being loaded and buffered in the background.
- Don't disregard CPU capacity. Like most video or graphics professionals, you probably work on a relatively fast, powerful computer. Everyone in your audience may not. Even if you are careful to spread out the data rate, complex transitions or effects can bog down the client computer—especially if the user has other applications running in the background.
- Test the limits. Don't throw away or donate those old dial-up modems or slow machines. Keep them on hand for testing the limits. How low can you go without completely compromising your content?

RULES OF THUMB

Even if you were to calculate and graph precisely how much bandwidth a file requires, the result would be only an approximation of the actual data rate as it streams over the Internet. The bottom line is that you need to use "rules of thumb" and "leave some slop." See the chart below for "Safe bets" using Apple QuickTime.[46]

Connection	Rated	Typical throughput	Safe bets
28.8 modem	28.8 Kbps	2.4 KB/second	2 KB/second
56K modem	53 Kbps	4.8 KB/second	4 KB/second
Dual ISDN	128 Kbps	12 KB/second	10 KB/second
DSL	384 Kbps	35 KB/second	30 KB/second
T1	1.54 Mbps	150 KB/second	50 KB/second
Cable modem	6 Mbps	300 KB/second	50 KB/second
Intranet/LAN	10 Mbps	350 KB/second	35 KB/second
100base-T LAN	100 Mbps	500 KB/second	50 KB/second

Kbps = kilobits per second • Mbps = megabits per second • KB/second = kilobytes per second

45 The term "Bandwidth Math" and examples cited in this section derived from "Don't Be Scared of Bandwidth Math," by Tim Kennedy, *Streaming Media World*, May 26, 2000, *http://smw.internet.com/symm/tutor/bandmath/index.html*

46 *QuickTime for the Web*, by Steven W. Gulie, May 1, 2000, Morgan Kaufmann Publishers

"…bloated streaming media… has a tendency to alienate your site visitors. Your clip may look good off your LAN on your fast CPU. But if it looks like junk to your user, you have just told them that you do not need their business... Do the bandwidth math. Test on a slower CPU. Take as many people along for the ride as you possibly can."

—Tim Kennedy
Streaming Media World[47]

A **codec** *(set of mathematical algorithms)* is used by an **encoder** *(component of an architecture that produces final streaming files)* to **compress** *(reduce the data rate of)* digital video and/or audio. The **player** *(client component of an architecture that plays back media files)* must use the <u>same</u> codec to decompress the file that was used to compress it.

"…the best way to learn to take advantage of these [different codecs] is to play around with them. Take the same source clip, run it through different codecs, data rates, and frame rates, and carefully explore the results. This is still more of an art than a science."

—Ben Waggoner
DV Web Video[48]

But even reducing both the frame size and the frame rate does not reduce the bit rate of video files enough for efficient streaming. So additional methods for compressing the amount of data that needs to be transmitted are used. We haven't space here to go into too detailed a discussion, but the following general descriptions will give you a general understanding of compression methods.

<u>Intra-frame compression:</u> Derived from still formats, **intra-frame compression**, also known as **spatial compression**, treats each frame individually. Some spatial compression schemes work by discarding much of the color information in the picture. As long as this type of compression is not too severe, it is generally unnoticeable. In fact, in even the highest quality "uncompressed" video used by broadcasters, some of the original color information has been discarded. Avoiding "hot" colors and harsh contrast when shooting helps ensure the best results from this type for compression. Spatial compression works best on continuous-tone images without sharp edges, such as photography and videography (as opposed to animated graphics). The softer and smoother the imagery, the better results you'll achieve with systems that use spatial compression.

<u>Inter-frame compression:</u> **Inter-frame compression,** also known as **temporal compression**, takes advantage of the fact that any given frame of video is probably very similar to the frames around it. So, instead of storing all the information for each and every frame, temporal compression schemes store **keyframes** in their entirety, while **delta frames** (comprising the majority of frames in Web video) contain only the information that is different from the keyframes that come before and/or after. You can usually specify how frequently keyframes occur—less frequent (and, therefore fewer) keyframes mean smaller files, but will also result in lower quality, since more frames will need to be inferred, or "predicted," from keyframe information.

What is a codec?

Compression and decompression are accomplished by a set of mathematical algorithms called a **codec**—short for "compressor/decompressor." Codecs compress data as it is encoded, and decompress it for playback. In order to be played, a file compressed with a certain codec must be decompressed by the same codec.

While the codec itself is comprised of software code, a codec may be a part of either a hardware or a software solution. For example, an "onboard" codec may be found in a video capture card, or in a specialized encoding workstation used for capture and encoding of real-time (live-live) streaming video. Encoding software, such as that included in the three major streaming architectures, provides codecs, as well.

Different codecs for different tasks

Video codecs and audio codecs work differently. You'll need to choose a video codec to compress your imagery, and an audio codec to compress your sound. And some codecs are more appropriate for certain kinds of work than others, since different codecs use different compression strategies. The Adobe Premiere Technical Guides, which can be found in the Support area of the Adobe Web site at www.adobe.com, offer some more detailed descriptions of different types of compression and codecs.

Which streaming media codecs are available to you depends on which architecture(s) you choose. You'll find some of the same codecs (MPEG-4, for example, for video; MP3, for example, for audio) in different architectures. But remember, since encoding not only compresses the file but also formats it for a specific architecture, even if the same codec is used, streaming files are likely to play back only on the player that is part of the encoding architecture.

Currently, the three dominant architectures offer just a few options for streaming video codecs. While QuickTime provides some variety, RealNetworks and Windows Media deliver little choice in codecs for streaming video. In terms of audio codecs, Real offers a wide variety of choices, with QuickTime

47 "Don't Be Scared of Bandwidth Math," by Tim Kennedy, *Streaming Media World*, May 26, 2000, *http://smw.internet.com/symm/tutor/bandmath/index.html*
48 "Choosing a Streaming Video Technology," by Ben Waggoner, *DV Web Video,* May 2000

"[...after applying our basic settings...] Then we start tinkering... Usually, there are some parts of the movie that don't compress well in each [bandwidth] version. We make a clip of the part that looks the worst, including about 5-10 seconds before and 5-10 seconds after the problem area, taking the clip from the uncompressed version. We recompress this section as many times as necessary, adjusting the settings each time until we get it right. Sometimes we have to go to a higher data rate. When the clip of the worst part looks right, we recompress the whole movie with those settings... We use half a dozen computers at a time for tinkering, each one with slightly different settings. That saves a lot of time... We keep our default settings on one machine, so they don't get overwritten, and drag them onto the others for tinkering..."

—"Secrets of the Apple Compressionist" QuickTime Book[49]

Save for Web in Adobe Premiere features a special edition of Terran Interactive's Media Cleaner EZ, making it easy to encode your productions for streaming in QuickTime, RealMedia, and/or Windows Media formats.

and Windows providing fewer options. Because streaming is an "emerging" technology, codecs are continually being improved. It is important to keep up with the latest developments, so check the Web sites for the dominant architectures, as well as trade publications and other resources on a regular basis.

Multiple bit rate encoding

When media is encoded as individual files for multiple bandwidths, end-users must select which file they wanted to stream, often from among nine or more choices if you encode for all three of the dominant architectures. Multiple bit rate (MBR) encoding solutions alleviate such confusion for the end-user, and can also make the process of encoding easier. There are, however, some trade-offs.

With the MBR approach, several different streams are encoded at different bit rates and combined into a single file. This is called SureStream in the RealMedia architecture and Intelligent Streaming in Windows Media. In the QuickTime architecture, the strategy is slightly different—separate files called "alternate data rate" movies are stored in the same folder or directory. When the client calls for the media file, a negotiation between client and server determines the available bandwidth, the appropriate bit rate stream is automatically transmitted and, if the network becomes congested, the transmission will automatically switch to a lower bit rate stream.

Critics of this "adaptive" streaming approach say that servers are sometimes oversensitive to network congestion, often streaming a lower bandwidth and, therefore, poorer quality file than necessary.

Multiple bit rate (MBR) encoding vs. individual streaming files

Issue	MBR Encoding	Individual Encoding
Deployment	Requires the appropriate streaming media server software	Files can be hosted on either a streaming server or Web server
Network congestion	Server software monitors each connection, automatically switching to a lower bandwidth stream to prevent media player from pausing to re-buffer.	May cause temporary disruptions to playback
Storage	MBR files often require more space than the sum of the individual files	Files require less storage space
Web page integration	Only one "Play" button graphic or link needed on Web page	Multiple "Play" button graphics or links are required
Frame size	Video frame size must be the same for all bandwidth streams so, while quality can be better, size can't be larger to take advantage of higher bandwidth capacity	Individual files can be optimized to take full advantage of higher bandwidths, e.g. larger frame sizes
Audio	Audio settings must be the same for all bandwidth streams to avoid disruptions to the soundtrack as the stream is adapted to conditions	Individual files can be optimized to take full advantage of higher bandwidths, e.g. higher quality audio

How do I encode my media productions?

While all of the choices make encoding sound complicated, it *can* be very, very easy to output your productions for streaming. The "one-step Web output" Save for Web feature in Adobe Premiere, and wizards available in other post production tools, deliver simple-to-use solutions. And, like most film and video techniques, advanced tools are available which can, when combined with experience and expertise, dramatically improve the quality of the final production. You can also look to a growing number of specialists to outsource encoding. Your decision to encode your own video projects for streaming or to call upon the services of a specialized outsource depends on your objectives, your ambitions and, of course, your budget.

49 *QuickTime for the Web*, by Steven W. Gulie, May 1, 2000, Morgan Kaufmann Publishers

HOW DO I PUT MY STREAMING MEDIA ON THE WEB?

A URL that begins with HTTP:// points to a Web server; a URL that begins with RTSP:// points to a streaming server.

Like any other resource file you publish on the Web, you post your media file to a server that has a connection to the Internet and an accessible IP address. Like any other resource file served over the Web, your streaming media file is referenced by a URL (uniform resource locator). But there are some significant differences between publishing Web pages and downloadable files vs. publishing streaming media.

PUBLISHING STREAMING MEDIA

Web server vs. streaming server—what's the difference?

Earlier in this primer, we distinguished between a Web server and a streaming server. Publishing your streaming media files on a Web server is possible, but it results in pseudo-streaming (aka progressive download, aka HTTP streaming, aka fast start streaming in the QuickTime architecture). For true streaming (aka hinted streaming in the QuickTime architecture), streaming media files must be posted to a streaming server.

Are you absolutely, positively sure you want to do true streaming?

Web server pros	Web server cons	Streaming server pros	Streaming server cons
▸ No special server software needed; just use your Web server ▸ Media gets through no matter how slow the connection, because lost packets are retransmitted until they are received ▸ With fast connections, media plays as it downloads, so it seems like streaming to the audience ▸ End-user gets to keep a copy of the file to play and/or share ▸ No problems with firewalls	▸ Can't transmit live feeds ▸ Can't broadcast or multicast ▸ No flexibility for end-user—e.g. end-user cannot skip "chapters"—audience must download file in its entirety, from beginning to end ▸ Leaves a copy of the media file on the end-user's hard disk, risking unauthorized alteration and/or redistribution	▸ Only way to transmit live feeds ▸ Broadcasts and multicasts (one stream to many viewers) ▸ VOD (video on-demand) files can provide flexible access for end-users ▸ Doesn't leave a copy of the movie on the viewer's hard disk ▸ Does not use up space on end-user's hard disk ▸ Bandwidth usage can adjust to conditions	▸ Requires a streaming server ▸ Playback may be disrupted or delayed if data rate exceeds connection speed or available bandwidth ▸ Media typically loses some data in transmission, and lost packets are gone for good ▸ Firewalls may not allow streaming files to pass

What kind of streaming media server(s) do I need?

If you do not plan to stream too many files to very many people, and you can be sure that your entire audience will be happy to use the same, single architecture, then you probably need only one server. And it ***is*** possible to run your streaming server software on the same computer which is functioning as your Web server.

But if your audience is large and you want to offer the opportunity for the end-user to choose which player to use, remember that you need a different streaming server for each architecture you plan to support. To serve streams that are compatible with the three dominant architectures, you'll need server software from Microsoft for Windows Media (.asf) files, from Real for RealMedia (.rm) files, and from Apple for hinted QuickTime (.mov) files. What this means in terms of machines, is that you'll need a Windows or Unix operating system to run all three servers, since Windows Media and RealMedia servers do not run on MacOS.

In terms of hardware, the amount of storage you need can be calculated using the bandwidth math basics you've already learned in this primer—multiply the bit rate of your streaming media (in kilobits per second) by the length of the media (in seconds) and divide by 8, to arrive at the amount of disk space (in kilobytes) you need for your streaming media files:

$$\frac{(\text{Kbps X seconds})}{8} = \text{KB}$$

"…If you're delivering a lot of streaming content over the Internet, telecommunications bandwidth is typically the major expense. Leasing costs for the high-bandwidth digital lines you need can cost more per week than your server and software cost annually."

—"Setting up a Streaming Server" *QuickTime Book*[50]

If your files are MBR (multiple bit rate) encoded, use the above equation for each stream contained in every MBR file. Add the results of your calculations for all of your streaming media files together, then don't forget to add the amount of storage you need for the server software itself, and whatever incremental capacity you anticipate needing.

What about my Internet connection?

Perhaps surprisingly, the capacity of the streaming server is not typically the limiting factor for throughput—the bandwidth of your Internet connection is what really counts. "A single Macintosh running QuickTime Streaming Server can deliver up to 1,000 simultaneous streams at 56 kilobits each, or an aggregate of over 50 megabits per second. That's enough to saturate 40 T1 connections to the Internet, or to completely swamp five 10Base-T Ethernet LANs."[51]

But the good news is that you only need enough bandwidth to accommodate the amount of streaming you plan to do… To serve a small number of streams from your home or office, 128K DSL is the minimum you should consider. Cable modem can be used, but continual traffic fluctuations make the network unreliable, potentially compromising the quality of your streams. In fact, attempting a live Webcast over a cable modem connection would be unwise. A single T1 line can carry enough streaming data to serve 25 end-users simultaneously connected at 56K. Whatever your connection, be sure to check with your ISP—your throughput will be limited by your ISPs backbone carrier; some ISPs are better prepared for the anticipated explosion of streaming media demand than others.

So just how much bandwidth *do* you need? Let's use a bit more bandwidth math to figure it out—the bandwidth required for serving content can be calculated by multiplying the data rate (in kilobits per second) of your streaming production(s) by the maximum number of simultaneous streams you expect to serve:

Kbps X maximum number of simultaneous streams = Kbps of bandwidth needed

Example bandwidth calculations

Data rate of streams	Max. audience size	Bandwidth needed	Example connection
20 Kbps	60	1,200 Kbps or 1.2 Mbps	T1
60% 20 Kbps plus 40% 80 Kbps	100	4,400 Kbps or 4.4 Mbps	Fractional T3
80 Kbps	100	8,000 Kbps or 8 Mbps	10 Mbps fractional T3
20 Kbps	2,000	40,000 Kbps or 40 Mbps	T3

Enough math already! I just want to *make* streaming media…

We've got good news for you... If you don't want to mess around with servers, you don't have to. Just as there are Web hosting providers, there are hosting services that can serve your streaming media files for you. Should you serve your own streaming media or outsource?

"…Define your target audience: Are they dial-up or broadband customers? Is your content video, audio or both? Live or on-demand? Is your content mission critical? How much redundancy do you require? Do you require support or do you have the expertise to run the hardware and software? Do you want yourself or someone else to take the page at 3am when the service is down?"

—Steve Stevenson
***Streaming Media.com*[52]**

50 *QuickTime for the Web*, by Steven W. Gulie, May 1, 2000, Morgan Kaufmann Publishers
51 ibid.
52 "Choosing a Hosting Provider," by Steve Stevenson, *http://www.streamingmedia.com/tutorials/view.asp?tutorial_id=52*

"…quantify what you will need from a host and do your due diligence in investigating hosting companies. Ask for a free evaluation to test their service and talk to your friends in the industry of their experiences… make sure you have weighed the factors of service, availability, speed, reliability, support, and quality."

—Gene Shin
Streaming Media.com[53]

What do streaming host providers have to offer?

A streaming service or an ISP that hosts streaming media is a very attractive choice for the hobbyist or small business—you get to do the creative part and leave most of the technicalities (and capital expenditures) to experts. For industrial-strength, global implementations, there are enterprise-focused co-locations that maintain streaming server farms and dedicated bandwidth. Like top-notch Web hosting co-locations, they provide security, dedicated or shared servers, load balancing, edge-caching, e-commerce transaction capabilities, metrics, and reporting. Some specialized streaming host providers also offer encoding and Web casting services.

How do the costs compare?

Several popular consumer-focused ISPs host streaming content at very reasonable prices—we've seen advertisements for as low as $9.95 per month. The costs to set up a streaming server in-house can range from $3,500 for hardware and software (really a consumer solution) up to $25,000 or more for an adequate corporate system, and into the hundreds of thousands of dollars for an enterprise-level, live Web-casting facility. Commercial quality hosting costs are likely to be thousands of dollars per month, but before you make the capital investment to do it in-house, remember that streaming technology is rapidly advancing. Be sure to do the math to determine the scope of your hosting needs and investigate potential providers with care. Ask direct questions about set-up fees, surcharges (apparent and hidden), length of contract, upgrade/downgrade fees, and reporting costs. And don't forget that this is an emerging industry—*everything* is negotiable!

SHOULD YOU SERVE YOUR STREAMS IN-HOUSE OR OUTSOURCE?

FOR ENTHUSIASTS:

If you are a hobbyist or have a very small business, and you want to stream a few, short clips, the easiest option is to outsource. Check with your ISP—in addition to offering Web hosting, they may provide streaming media hosting services ,as well. There are also some, primarily consumer-targeted, businesses offering streaming services for free or for very low monthly rates—check out iClips, SpotLife, StreamCode, and QuickStream, to name just a few.

If you're an intrepid "stream-it-yourselfer," you may want to set up your own server. To stream media yourself requires a combination of hardware, software, and bandwidth—here are some minimum basics:

- **500MHz processor**
- **128K RAM**
- **Video capture card**
- **Minimum 7200rpm hard drive**
- **Minimum connection 128K DSL**
- **Video editing software (e.g., Adobe Premiere)**
- **Encoding and server software (e.g., Windows Media, RealMedia, QuickTime)**

To serve a small number of streams from your home or office, you will need at least 128K DSL speeds. Cable modems will suffice, although the amount of traffic on the network is unpredictable and may degrade the quality of the streams you provide. You definitely won't be supporting a live Webcast using a cable modem, because you need a reliable, dedicated connection that does not fluctuate connection rates. By doing it yourself, you are limited to the backbone carrier of your ISP where your streams may suffer from network congestion or limited bandwidth availability due to the ISP's network.

FOR CORPORATE IT DEPARTMENTS:

You probably do *<u>not</u>* want to set up in-house streaming capabilities if:

- **You lack a well-developed internal network and a high-speed (T1+) Internet connection with excess bandwidth**
- **Management wants existing IT staff to focus on core competencies and/or progress on mission critical tasks cannot be derailed by a new initiative and/or existing IT staff has significant project backlog**
- **Staff has little or no experience with Web hosting, video streaming, or video conversion**
- **No network management function is in place to ensure availability, performance, and bandwidth utilization for a streaming system**
- **No access to video editing suite and editing/conversion tools**
- **No existing server resources available to provide video streaming service**
- **Number of concurrent video streams is likely to vary significantly**
- **Bandwidth usage during peak video demand is likely to exceed surplus bandwidth or compete/degrade performance of mission critical applications**
- **Monthly cost with no capital or up-front expenditures for equipment or software is preferred; organization willing to pay higher monthly costs to reduce short-term risk**
- **Very short timeline to implement; desire is to begin streaming in days**
- **Organization is unsure of the role of streaming media**
- **Company wishes to pilot a streaming media program, and may bring it in-house after pilot is successful**

FOR CORPORATE IT DEPARTMENTS:

It *<u>may</u>* make sense to set up in-house streaming capabilities if:

- **You already have a significant internal network and high-speed (T1+) Internet connectivity with significant surplus bandwidth available on Internet uplink**
- **Staff has experience in web hosting, video streaming, and video conversion**
- **Network management ensures effective bandwidth utilization**
- **Organization already owns video editing suite and editing/conversion tools**
- **Data center has adequate existing server resources available to provide video streaming service**
- **Number of concurrent video streams is expected to be fairly consistent**
- **Bandwidth usage during peak video demand will be significantly less than available surplus bandwidth**
- **Preference is to incur capital expenses (servers, tools, software, etc.) to avoid recurring charges; organization willing to make long-term investments for lower costs**
- **Organization is willing to spend time and money to purchase equipment/software/tools, and to build infrastructure**
- **IT department has existing excess staff to implement, manage, and monitor the streaming media solution.**
- **Organization has a clear vision of media streaming strategy**

53 "Do It Yourself vs. Outsourcing," by Gene Shin, *http://www.streamingmedia.com/tutorials/view.asp?tutorial_id=51*

INTEGRATING STREAMING MEDIA

How do I link my streaming media to my Web page?

If you use a standard HTML <a href> tag, browsers will download the streaming media file just as if it was being served by a Web server—that is, the file must download completely before being played from the user's hard drive—as you know, this is *not* streaming.

To differentiate between a downloadable media file and a streaming media file, a media **metafile** is used. The link you insert into your Web page points to the metafile, which is posted on the Web server. The metafile, which is a simple text file, invokes the correct media player on the client desktop, locates the selected streaming media file on the streaming media server and, as the streaming media file downloads, redirects it from the browser to the media player.

Metafile extensions and streaming media file extensions

Architecture	Metafile Extensions	Streaming Media File Extensions
RealMedia	.rpm or .ram	.rm
Windows Media	.asx, .wax, or .wvx	.asf or .wmv (.wma for audio)
QuickTime	.mov or .sdp	.mov (sometimes .qt or .qti)

The QuickTime approach is a bit different from RealMedia and Windows Media. In the QuickTime architecture, the streaming instructions are embedded right into the movie as a **hint track**, and the link from the Web page goes directly to the streaming server. But some browsers will automatically attempt to open an RTSP URL with the specified default player, which may not be QuickTime. To be sure the QuickTime player is invoked, a form of metafile which Apple calls a **reference movie** can be employed; to take advantage of QuickTime's answer to MBR—alternate bit rate files—a reference movie *must* be used. A reference movie is a separate, tiny .mov file published on your Web server. When invoked, the reference movie calls up the appropriate streaming (i.e., hinted) movie file from the streaming media server. Live feeds in QuickTime use a more standard metafile scheme, with a text file format called **SDP** (Session Description Protocol); the SDP file is published on the Web server and points to the proper location on the broadcasting media server.

Integrating streaming media into a rich media experience

A Web page is often like a three-ring circus, with different things going on, in different places, on different timelines. It has been difficult for Web developers to synchronize presentations comprised of assorted components, especially when those components are served up from different locations via connections that are each subject to their own congestion patterns. **Synchronized Multimedia Integration Language**, or **SMIL** (pronounced "smile"), lets Web developers choreograph rich media presentations, including streaming media, with comparative ease.

"[SMIL] is a simple but powerful markup language that lets you coordinate the clips so that they each play how, when and where you want them to… If you are familiar with other Web-based markup languages, such as HTML, you will pick up SMIL quickly. You'll need to be careful, though, because SMIL is less forgiving than HTML. Lapses that may not matter in HTML markup, such as missing quotations marks… will prevent a SMIL file from working properly. "

—SMIL Basics
realnetworks.com54

SMIL, based on the language XML, is reputed to be as easy to learn and use as HTML. It can, of course, be written with a text editor, but there are several convenient SMIL editors now available for this relatively new technology. In essence, SMIL is to RTSP and media servers what HTML is to HTTP and Web servers—it is the language that best takes advantage of the benefits of the streaming protocol. And, as HTML is the standard language used to coordinate the content displayed on Web browsers, SMIL is the language that is rapidly becoming the standard for coordinating the content played on media players. Once your "clips" (e.g., video, text, still images, animations, etc.) are encoded in their streaming (or other) formats, you assemble your presentation using SMIL.

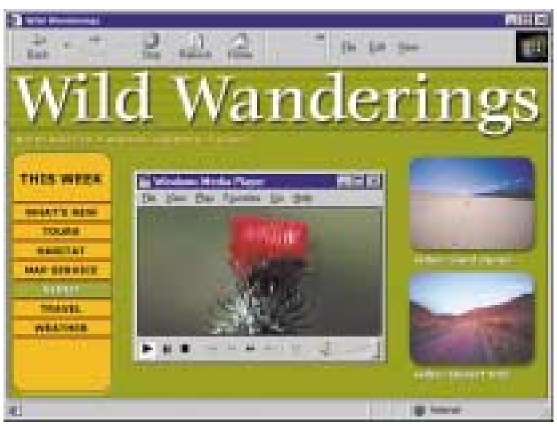
The end-user must have an SMIL-enabled media player or browser to take advantage of SMIL files. The latest versions of all three dominant architectures' players and several others', as well as the newest version of Internet Explorer, are compliant.

55 *http://www.realnetworks.com/devzone/tutorials/authoring/smil/smil_basics.html*

SMIL offers numerous benefits:

- Bandwidth negotiation—Similar to QuickTime's alternate bit rate scheme, you can individually encode and optimize your media files to get the best result at different bit rates. SMIL can automatically negotiate the appropriate file to stream, depending on available bandwidth, without the quality compromises inherent in MBR files.
- Personalized content—A SMIL file can list different language or content options for clips. For example, to create a video with soundtracks in different languages or soundtrack content targeting different audiences, you could produce just one video clip, with no soundtrack, then create separate audio clips for each language or version. Your SMIL file has just one link, and the appropriate soundtrack is synchronized and streamed with the video, based on the preferences each end-user set-up in his/her media player.
- Interactivity—Using SMIL, you can easily create interactive media presentations, such as an audio or video jukebox that plays a different clip each time the user clicks a button.
- Flexibility and control—SMIL puts the developer in control of the presentation. When your presentation includes multiple clips, you can use SMIL to define the layout. You can start playing a clip at any point in its internal timeline, without changing the encoded clip itself. And, since a SMIL file lists a separate URL for each clip, you can assemble presentations from clips stored on any server (media and/or Web servers), anywhere.
- An open standard—SMIL can be used to choreograph a wide variety of file formats for a variety of different media players and browsers.

This Web page, with its embedded media player, was assembled with Adobe GoLive and other tools in the Adobe Streaming Media Collection



" It is easier to run the streaming player as a stand-alone application, but if you are creating content that needs to be very interactive or needs to be integrated with other Internet technologies, embedding is the way to go… Lately, most popular player platforms have been implementing interactive content through information included in metafiles (SMIL, ASX, RAM). This new trend has made it easier to create the interactivity that developers are looking for, but it is still not as extensible as embedding the player.

—Gene Shin
Streaming Media.com[53]

What about embedding streaming media into my Web pages?

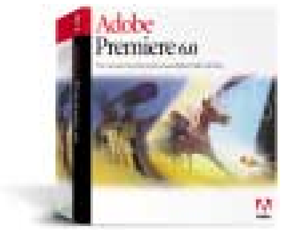
This question should really be, "What about embedding a streaming media *player* into my Web pages?' When your end-user clicks on a link to access streaming media, the media player first has to load, usually opening a new window that is distinctly separate from the browser window. If you want to provide an experience whereby your streaming content is more seamlessly integrated with your other Web content—including text, still images, animations, music and other audio—as well as high levels of interactivity, you may want to embed a media player into your Web page.

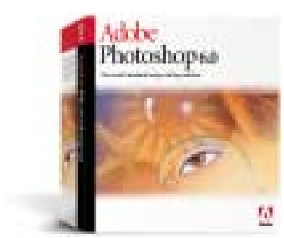
An embedded player can have a custom skin (look and feel), as well as customized controls (functionality). You can also include functions that trigger on mouse-over events, or player states. URL flips triggered by Web markers inserted into the video timeline can synchronize other elements on the Web page with the streaming content.

The drawback to embedding a media player is that you must choose *which* player to embed. The quality of the rich media experience you create for your audience comes at the expense of end-user preferences. If your audience preferences are controlled (e.g., by corporate or institutional dictate) or predictable (e.g., by platform or content), then embedding a player is likely to be a good choice. If, however, you are targeting a widely diverse demographic (e.g., consumers), embedding a player will be sure to diminish the size of the audience you actually reach.

If you plan to embed a media player, especially for implementations where end-users may employ a variety of browsers and/or platforms, you'll need some fairly advanced Web development skills and an understanding of the differences between the different browsers' implementations of JavaScript, Active X controls and plug-ins. To learn more about the specifics of embedding streaming media players, a good place to start is the SDK (software developer kit) downloadable (typically for free) from the Web site of your preferred architecture.

55 "Embedding Players," by Michael Long, *http://www.streamingmedia.com/tutorials/view.asp?tutorial_id=54*





HOW CAN ADOBE HELP ME CREATE STREAMING MEDIA?

The best way to learn more about streaming is to actually start creating your own streaming media content. Powerful, professional **Adobe®** software products are an excellent choice for creating streaming media, Web graphics, and animations, as well as for integrating all kinds of content into dynamic and compelling Web sites.

Output to all the most popular streaming formats: One of the best reasons to choose **Adobe** software for creating streaming media is the broad range of output options available to help ensure that everyone in your Web audience gets the files they need to enjoy your productions. In addition to flexible output options for the traditional video mediums, **Adobe Premiere** offers one-step Web output to all the most popular streaming media formats including RealMedia G2, Windows® Media, and Apple® QuickTime™ streaming formats, through a special, built-in version of Terran Interactive's Media Cleaner EZ 5. Plus, **Adobe Premiere** provides advanced output options for both RealMedia and Windows Media, offering even more precise control. With **Adobe After Effects**, you can render to both the QuickTime and Flash (SWF) formats, as well as outputting for film, broadcast, and multimedia. Tight integration also makes it easy for you to import your **Adobe After Effects** compositions and animations into **Adobe Premiere** to take advantage of all the streaming media output options there.

DV in, Web out.: It can be just that easy. With a DV camcorder and **Adobe Premiere**, any reasonably savvy software user can "Shoot it! Cut it! Web it!" But you'll also find that **Adobe Premiere** software delivers an extraordinarily powerful toolset for every aspect of digital video editing. **Adobe Premiere 6.0** has been rebuilt from the ground up, making this long-popular digital video editing software an even more powerful tool that novices and experienced professionals will all appreciate. **Adobe Premiere** lets you work the way you want to, with more customizable features than ever before, more precise creative control, more time-saving automation, and even a sophisticated new sound-studio-style audio mixer that lets you view your video while mixing multi-track audio in real time. Cross-platform interoperability and tight integration with industry-standard tools, including **Adobe After Effects, Adobe Photoshop®,** and **Adobe Illustrator®**, enhance productivity and streamline your workflow. The award-winning Adobe user interface makes **Adobe Premiere** easy to learn and use, sharing many of the same intuitive toolbars, palettes, menus, and other features found in familiar Adobe software products.

A complete streaming solution: Combining comprehensive streaming media and interactive animation with powerful Web capabilities, the **Adobe Streaming Media Collection** delivers the cost-effective toolset professionals need to make Web sites more dynamic. The **Adobe Streaming Media Collection** includes four full-featured Adobe software products:

- **Adobe Premiere**—The essential tool for professional digital video editing
- **Adobe After Effects**—The essential tool for motion graphics and visual effects
- **Adobe LiveMotion™**—Professional web graphics and animation
- **Adobe GoLive®**—Professional web authoring and site management

Cross-platform interoperability between the components, as well as tight integration with other familiar Adobe products including **Adobe Photoshop** and **Adobe Illustrator** software, supports a highly productive, collaborative workflow and speeds learning. With every component in the **Adobe Streaming Media Collection**, you'll enjoy the extraordinary depth of features and functionality you have come to expect from award-winning **Adobe** products.

HOW TO PURCHASE ADOBE SOFTWARE PRODUCTS

Via Web:

http://www.adobe.com/store

Via Phone:

Adobe After Effects Hotline: **(800) 685-3504**

Adobe Premiere Hotline: **(888) 724-4507**

Education customers:

Contact an Adobe Authorized Education Reseller at:

http://www.adobe.com/education/purchasing/resellers.html

Free Tryouts:

Adobe After Effects: **http://www.adobe.com/products/aftereffects/tryreg.html**

Adobe Premiere: **http://www.adobe.com/products/premiere/tryreg.html**

To find the reseller nearest you, visit:

http://www.adobe.com/store/otherplaces/uscanada/main.html

FOR MORE INFORMATION

There are a variety of resources available for obtaining technical information about *Adobe Streaming Media Collection* products. There also is an abundance of places to share your experiences with other users. For a complete list of these resources go to **http://www.adobe. com.** Additionally, the following digital video resources may be helpful:

VIDEO CAPTURE CARDS:

To find a video capture card that suits your needs and is certified with Adobe Premiere, go to www.adobe.com/premiere; click on the "Capture Cards" link.

BOOKS ON DIGITAL VIDEO:

Adobe Products **"Classroom in a Book"** by Adobe Press
(800) 428-5331
www.adobe.com/products/adobepress

Digital Video for Dummies by Martin Doucette, published by IDG Books Worldwide, Inc., 1999

Real World After Effects by Sherry London and Eric Reinfeld, published by Peachpit Press, Inc.

Creating Motion Graphics with After Effects: High Impact Animation for Video and Film by Trish Meyer and Chris Meyer, published by Miller Freeman Books

DIGITAL VIDEO CONFERENCES:

BDA (Broadcast Designers Association)
www.bdaweb.com

DV Expo
T: (415) 278-5300/F: (415) 278-5200
www.dvexpo.com

IBC (International Broadcasting Convention)
www.ibc.org.uk/

NAB (National Association of Broadcasters)
T: (202) 429-5300/F: (202) 429-5343
www.nab.org

newMedia
www.newmedia.ca/

SIGGRAPH
T: (312) 321-6830/F: (312) 321-6876
www.siggraph.org

WEVA (Wedding & Event Videographers Association)
www.weva.com

DIGITAL VIDEO TRAINING:

Adobe Certified Training Program
Adobe certified trainers listing
www.adobe.com/products/partnerfinder/searchtraining.html

Total Training Inc.
Video training (760) 944-3900
www.totaltraining.com

Digital Institute of Video Arts
Video training (510) 932-2282
www.knowpath.com

Mac/Windows Academy
Training on video cassette
(800) 527-1914/(904) 677-1918
www.macacademy.com

Payne Media Inc.
Video: "Up and Running with Digital Video"
(425) 455-1025
www.paynemedia.com

Straight Scoop Enterprises
CD: "Secrets of Adobe Premiere 5"
(503) 643-3976
www.straight-scoop.com

Virtual Training Company (VTC)
CD-based training for Adobe Premiere
(888) TRAIN-CD/(408) 492-1051
www.vtco.com

VideoSyncrasies: The Motion Graphics Problem Solver
Videotapes by Trish and Chris Meyer
www.deskimages.com/ae.shtm

DIGITAL VIDEO WEB BASED FORUMS:

Adobe User to User Forums
www.adobe.com/support/forums/main.html

Helptalk Online www.helptalk.net/audiovideo

Digital Editor Online
www.digitaleditor.com

The World Wide Users Group Forums for After Effects Users
www.wwug.com/forums/adobe_aftereffect/index.htm

The World Wide Users Group Forums for Premiere Users
www.wwug.com/forums/adobe_premiere/index.htm

LIST SERVERS:

Adobe After Effects List-serverwww.fido.se/Pages/maillist.html

Adobe Premiere List-server
www.wwug.com

The DV-L List Server
www.dvcentral.org

After Effects Mail Group
www.softmotion.com/Pages/start.html; Click on Mailists link; click on Info button for After Effects list

DIGITAL VIDEO NEWSGROUPS:

rec.video.desktop
rec.video.production
rec.video.professional

ADOBE SYSTEMS DIGITAL VIDEO RESOURCES:

Adobe Systems Incorporated
345 Park Avenue San Jose, CA 95110
T: 408-536-6000/F: 408-537-7000

Adobe After Effects Customer Service
800-685-3504

Adobe Premiere Customer Service
888-724-4507

Adobe After Effects for Mac OS Technical Support
206-675-6210

Adobe After Effects for Windows Technical Support
206-675-6310

Adobe Premiere for Mac OS Technical Support
206-675-6205

Adobe Premiere for Windows Technical Support
206-675-6305

Adobe Premiere for SGI Technical Support
206-675-6405/www.adobe.com

Main Customer Support Website
www.adobe.com/support/main.html

Searchable Support Databases Website
www.adobe.com/support/database.html

Techdocs e-mail (automated system)
techdocs@adobe.com

Index of Adobe Premiere and Adobe After Effects documents
310099

Index of all available documents:
100099

GLOSSARY

adaptive streaming: See *MBR*.

analog: The principal feature of analog representations is that they are continuous. For example, clocks with hands are analog—the hands move continuously around the clock face. As the minute hand goes around, it not only touches the numbers 1 through 12, but also the infinite number of points in between. Similarly, our experience of the world, perceived in sight and sound, is analog. We perceive infinitely smooth gradations of light and shadow; infinitely smooth modulations of sound. Traditional (non-digital) video is analog.

animated GIF: GIF is a bitmap file format often used on the Web. GIF (which may be correctly pronounced as either "giff" with a hard "g" or "jiff") stands for Graphic Interchange Format. Because it applies data compression, GIF is often used for photographic images. An animated GIF combines several images into a single GIF file. Applications that support the animated GIF standard (GIF89A) cycle through the images, creating the impression of motion. The animated GIF format does not provide as much control and flexibility as other animation formats but, because it is supported by nearly every Web browser, has become very popular.

architecture: The structure of the software, or the system of software components, responsible for the creation, storage, and playback of multimedia content. An architecture may include such things such as encoders, compression/decompression support, file formats, server software, and browser plug-ins. Different multimedia architectures offer different features and options, and store data in different file formats. ***QuickTime***, ***RealVideo***, and ***Windows Media*** are examples of streaming media architectures.

artifact: Visible degradations of an image resulting from any of a variety of processes. In digital video, artifacts usually result from color compression, and are most noticeable around sharply contrasting color boundaries such as black next to white.

ASF: Defined by Microsoft, "ASF" stands for *Advanced Streaming Format*. ASF is the file format in the Windows Media architecture.

ASP: An ***application service provider*** (ASP) is a third party entity that manages and distributes software-based services and solutions to customers across a wide area network from a central data center.

aspect ratio: The ratio of an image's width to its height. For example, a standard video display has an aspect ratio of 4:3.

asynchronous: Not coordinated in time; contrasted with synchronous. In a typical synchronous protocol, each successive transmission of data requires a response to the previous transmission before a new one can be initiated. An asynchronous protocol allows transmissions to occur independently of one another. In computer communications using asynchronous protocols, each piece of data usually has a start bit at the beginning and a stop bit at the end, so that the valid data can be distinguished from random noise. Most communications between computers and devices are asynchronous; the public Internet is based on an asynchronous system.

AVI: Defined by Microsoft, "AVI" stands for ***Audio Video Interleave***. AVI is the standard file format for digital video on the Microsoft® Windows® platform. AVI is not a streaming format; ASF (***Advanced Streaming Format***) is the Windows Media streaming format.

bandwidth: The data-carrying capacity of a network. Bandwidth is the maximum amount of data that can travel a communications path in a given time, usually measured in kilobits per second (Kbps). If you think of the communications path as a pipe, then bandwidth represents the width of the pipe, which determines how much data can flow through it at once. Connections of 56 Kbps or lower (typical dial-up connection rates) are considered low-bandwidth, aka ***narrowband***. High-bandwidth, aka ***broadband*** connections are higher than 56 Kbps (e.g., ISDN, DSL, cable modem, T-1).

bit: The smallest unit of data used by computer systems. A bit (short for binary digit) has a value of either 0 (nil) or 1. Bits are the "building blocks" of binary data.

bit depth: See *bitmap*.

bitmap: Also known as ***raster***, bitmap data comprises a set of binary values specifying the color of individual ***pixels*** that make up an image. Bitmap data is characterized by ***resolution*** and ***bit depth***. Resolution relates to the detail in an image, and is expressed in dots per inch (dpi) or pixels per inch (ppi). The higher the resolution (i.e., the more dots used to describe the image), the more detail possible. Bit depth defines the number of colors the image can display. A high-contrast (no grey tones) black and white image is 1-bit, meaning it can be off or on, black or white. As bit depth increases, more colors become available:

Bit depth	Maximum colors
1	2
2	4
4	16
8	256
16	32,768
24/32	16.7 million

For image detail and quality, bit depth is as important as resolution, since the bit depth determines the colors available in the palette. When fewer colors are available, areas that may have shown a subtle shift of tones and hues are rendered instead as single blocks of solid color, eliminating image detail. Bitmap data is indispensable for continuous tone images, such as scanned or digital photographs, and for anti-aliased images. However, bitmap data is consistently larger than vector data. Each pixel in a bitmap image has to be defined. A relatively small 150-pixel x 150-pixel graphic requires 22,500 discrete bits of information plus the palette, or color lookup table (CLUT), that is usually included. For more information see the online Adobe Technical Guides at ***http://www.adobe.com/support/techguides/livemotion/lmobjects/page2.html***

bit rate: In a digital network, the number of bits that pass a given point, in a given amount of time, usually a second. Bit rate is usually measured in kilobits (Kbps), or thousands of bits per second. While the term "bit rate" is a synonym for ***data transfer rate*** (aka ***data rate***), bit rate seems to be used more often when referring to telecommunications transmission technology, and data transfer rate (or data rate) is used more often when referring to computing systems.

bluescreen: Video editing technique used to combine a subject with a background shot separately. The subject is shot against a solid blue (or sometimes green) color screen. The blue color can be selected on a frame-by-frame basis, through the use of a video editing system. A matte can then be generated to isolate the subject, in order to composite it onto the desired still or motion background clip.

broadband: High-bandwidth network (Internet or intranet) connections higher than 56 Kbps (e.g., ISDN, DSL, cable modem, T-1).

broadcast: A single stream transmitted to all clients on a network, whereby all the clients experience the same portion of the media program, at the same time, whether the program is live or pre-recorded.

broadcast quality: Standard of quality associated with current expectations of clearly-received network or cable television.

browser: A ***client*** application, usually with a graphical user interface (GUI), that provides a way to experience and interact with files hosted on a network, such as the ***Internet*** or an intranet. Web browsers provide access via the Internet to files posted on the ***World Wide Web***. Web browsers use the ***Hypertext Transfer Protocol (HTTP)*** to make requests of ***Web servers*** connected to the Internet, on behalf of the browser user. The two most popular browsers in use today are Netscape Navigator and Microsoft Internet Explorer.

buffer: A temporary storage area, or holding place, usually in the RAM (random access memory) of a computer or peripheral device (e.g., a printer), where data can be collected and/or held for ongoing or later processing.

byte: Eight (8) ***bits***.

camcorder: A video camera, i.e., a device that records continuous pictures and generates a signal for display or recording. To avoid confusion, it is recommended that the term "camcorder" be used rather than "camera"—in contrast, a digital camera records still images, while a digital camcorder records continuous video images.

capture: If the source footage is analog, "capture" refers to the act of digitization (conversion to a digital format) to make the video usable on a computer and, usually, the simultaneous application of ***compression*** to reduce the video to a manageable ***data rate*** for processing and storage. If the source video is ***DV***, "capture" typically refers to the simple transfer of video from an external device, such as a digital camcorder or tape deck, to a computer hard drive.

capture card: A printed circuit board that fits into an expansion slot on a computer and provides functionality for capturing video from an external source.

chapterization: The process of placing markers into the timeline of a video to demark the beginning points of sequences, or chapters, to which a viewer may "jump," at will, during playback, so long as chapterization is supported by the video architecture in use. ***True streaming*** of video supports chapterization; ***pseudo-streaming***, a.k.a. ***progressive download***, does not.

client: In a client/server architecture, the client is a software application that makes requests of the ***server*** on behalf of the end-user. A Web ***browser*** is a client application; a media ***player*** is also a client application. Sometimes the term "client" is also used to refer to a PC or workstation (hardware) on a network being used by an individual to access data and or applications hosted on a server.

clip: A digitized portion of media, typically video or audio.

clipping: Clipping occurs when audio volume reaches or exceeds 0 decibels (db) for 3 consecutive seconds. If you don't lower the volume so it is within the clipping threshold, data will be lost.

codec: Short for ***compressor/decompressor***; comprised of algorithms that handle the compression of video to make it easier to work with and store, as well as the decompression of video for playback.

co-location: An outsource, usually a dedicated facility or facilities, that provides physical space for and/or shared usage of essential computer equipment such as Web servers and, often, mission-critical managed services. Co-locations typically provide a high level of security and protection against fire, theft, vandalism, power outages or irregularities, and other risks which might, potentially, devastate a business that is reliant on 24/7/365 computer operations.

compositing: The process of combining two or more images to yield a resulting, or "composite" image.

compression: Algorithms used by a computer to reduce the total amount of data in a digitized frame or series of frames of video and/or audio.

datagram: See ***packet***.

data rate *or* data transfer rate: Amount of data moved over a period of time, such as 10MB per second. Often used to describe a hard drive's ability to retrieve and deliver information. Also see ***bit rate***.

de-interlacing: The process, usually performed by video editing software, of removing ***interlacing*** from video originally intended for display on television monitors, in order to make it suitable for display on computer monitors.

delta frame: In ***interframe*** (a.k.a. ***temporal***) ***compression***, periodic ***keyframes*** store all the information that comprises a frame, while delta frames store only the information that changes from frame-to-frame in between keyframes.

digital: In contrast to ***analog***, digital representations consist of values measured at discrete intervals. Digital clocks go from one value to the next without displaying all intermediate values. Computers are digital machines employing a binary system, i.e., at their most basic level they can distinguish between just two values, 0 and 1 (off and on); there is no simple way to represent all the values in between, such as 0.25. All data that a computer processes must be digital, encoded as a series of zeroes and ones. Digital representations are approximations of analog events. They are useful because they are relatively easy to store and manipulate electronically.

digital rights management (DRM): Server software that protects against the illegal distribution (a.k.a. pirating) of copyrighted content over the Internet.

digitizing: Act of converting an analog audio or video signal to digital information.

download: The act of one computer requesting and receiving data, via a network, from another computer.

downloadable media: *Media* (i.e., video and/or audio) that can be downloaded and written to disk (i.e., saved onto a computer hard drive); typically refers to media files hosted on a ***Web server*** (as opposed to a ***media server***) which can, and often must, be entirely downloaded and written to disk in order to be played (as contrasted to ***streaming media***).

DV: Generally refers to a ***digital*** video format that incorporates DV ***compression***, the type of compression used by DV systems or by a DV ***camcorder*** that employs the DV format. A standard consumer DV camcorder uses mini-DV tape, compresses the video using the *DV25* standard, and has an *IEEE 1394* port for connecting to a desktop computer. The DV designation is also used to for a special type of tape cartridge used in DV camcorders and DV tape decks.

DV25: The most common form of DV ***compression***, using a fixed data rate of 25 megabits/sec.

dynamic media: Refers to design elements that incorporate motion and/or sound such as video, animation, and audio.

effect: Distortion of a frame or frames of video to change its appearance; distortion of audio to alter the quality of sound.

encoder: A software application or a device (hardware) used to encode—i.e., compress and format (see *encoding*) digital video or audio.

encoding: Encoding accomplishes two main objectives: 1) it reduces the size of your video and audio files, by means of compression, making Internet delivery feasible, and 2) it saves your files in a format that can be read and played back on the desktops of your targeted audience. Some encoding solutions may also be configured to provide additional processing functions, such as digital watermarking, for example. Encoding may be handled by a software application, or by specialized hardware with encoding software built in. Note that the term ***compression*** is often used interchangeably with the term "encoding" when referring to the final step in preparing media files for Web distribution; but compression is only a part of the encoding process.

fast-start streaming: Term used in Apple's QuickTime architecture for ***progressive download***.

fields: The sets of upper (odd) and lower (even) lines drawn by the electron gun when illuminating the phosphors on the inside of a standard television screen, thereby resulting in displaying an ***interlaced*** image. In the *NTSC* standard, one complete vertical scan of the picture—or ***field***—contains 262.5 lines. Two fields make up a complete television ***frame***—the lines of field 1 are vertically interlaced with field 2 for 525 lines of resolution.

FireWire: The Apple Computer trade name for *IEEE 1394.*

Flash: Macromedia Flash is a popular architecture for vector-based Web animation. Often referred to as "streaming," Flash does not fit the definition of "true streaming" used in this Primer. While Flash shares some characteristics with *streaming media,* and can simulate video by animating sequential frames, it does not deliver standard, full-motion video file formats over the Web and is constrained to files containing no more than 16,000 frames (approximately 17 minutes of material at 15 fps). The Flash file format, SWF, can be created using Adobe After Effects and Adobe GoLive software, as well as other applications.

FPS: Number of "frames per second;" a method for describing ***frame rate***.

frame: A single still image in a sequence of images which, when displayed in rapid succession, creates the illusion of motion—the more frames per second (***FPS***), the smoother the motion appears.

frame rate: The number of images (video frames) shown within a specified time period; often represented as *FPS* (frames per second). A complete ***NTSC*** TV picture consisting of two fields, a total scanning of all 525 lines of the raster area, occurs every 1/30 of a second. In countries where ***PAL*** and ***SECAM*** are the video standard, a ***frame*** consists of 625 lines at 25 frames/sec.

FTP: File Transfer Protocol is the simplest way to exchange files between computers on the Internet. Like the Hypertext Transfer Protocol (***HTTP***), which transfers displayable Web pages and related files, and the Simple Mail Transfer Protocol (SMTP), which transfers e-mail, FTP uses the Internet's TCP/IP protocols.FTP is commonly used to transfer Web page files from the on which they were created to a Web server.

hinted movie: Term used in Apple's QuickTime architecture for video files that are formatted for ***true streaming***.

hint track: Term used in Apple's QuickTime architecture for the set of streaming instructions which, in the QuickTime streaming format, are embedded right into the movie. Other architectures provide streaming instructions as ***metafiles*** which are often downloaded separately from the actual movie file.

host: A computer system providing data to be accessed by a user working on a remote system, via a network such as the Internet; can be a synonym for ***server***. The term "host" is also used to refer to businesses that provide such systems. A hosting services provider (HSP) is an application service provider (ASP) dedicated to providing hosting services. A hosting services provider typically operates a Web server farm, at a data center or co-location facility. The term "host" may also be used as a verb, as in "to host" files on a server.

HTML: Hypertext Markup Language is a lexicon of formatting commands typically used to create Web pages. HTML is the language that describes all the basic elements of a page (such as text and graphics), but its current incarnation can't do much to make a page interactive; Java and ActiveX are often used to build interactivity into Web pages.

HTTP: HyperText Transfer Protocol is comprises the underlying set of rules used by the World Wide Web to define how messages are formatted and transmitted, and to communicate what actions Web servers and browsers should take in response to various commands. For example, when a URL is specified in a browser, an

HTTP command is sent to a Web server directing it to fetch and transmit the requested Web page.

HTTP streaming: See *progressive download.*

hypertext: A database system wherein objects (text, images, video, audio, and applications) can be linked to each other. For example, while reading an article about jazz music, you might click on the name of a musician to see a photograph, biography, or discography; you might click on the name of a song to hear it played or to view score. A Web page is simply a graphical user interface (GUI) used to convey a hypertext document.

IEEE 1394: The interface standard that enables the direct transfer of DV between devices such as a DV camcorder and a computer; also used to describe the cables and connectors utilizing this standard.

i.LINK: The Sony trade name for ***IEEE 1394.***

inter-frame compression: Also known as ***temporal compression,*** inter-frame compression reduces the amount of video information by storing only the ***differences*** between frames.

interlacing: System developed for early television and still in use in standard television displays. To compensate for limited persistence, the electron gun used to illuminate the phosphors coating the inside of the screen alternately draws even, then odd horizontal lines. By the time the even lines are dimming, the odd lines are illuminated. We perceive these "interlaced" ***fields*** of lines as complete pictures.

intra-frame compression: Also known as ***spatial compression***, intra-frame compression reduces the amount of video information within each frame individually.

IP: Internet Protocol, the basis of most Internet protocols, breaks up large chunks of information into digestible ***packets.*** In addition to the data being conveyed, each packet (also known as a ***datagram***) carries a header containing the source and destination ***IP addresses,*** as well as a sequence number that allows the destination computer to reconstruct the packets in the correct sequence, when they arrive.

IP address: A numeric identifier for a computer or device on the Internet. An IP address consists of four numbers separated by periods, or dots (e.g., 192.168.0.1), representing a unique 32-bit address. An IP address consists of a network portion and a host portion; how many bits designate the network and how many designate the host varies.

KB: Kilobytes

Kbps: Kilobits per second

keyframe: A frame selected at the beginning or end of a sequence of frames, that is used as a reference for any of a variety of functions. In inter-frame video compression, keyframes typically store complete information about the image, while the frames in between may store only the differences between two keyframes.

live feed: Data which is fed to a server in real time (i.e., as it is recorded), rather than being pre-recorded.

live-live: Sometimes used to refer to the streaming of a ***live feed.***

lossy: Generally refers to a compression scheme or other process, such as duplication, that causes degradation of signal fidelity.

lossless: A process that does not affect signal fidelity; e.g. the transfer of DV via an IEEE 1394 connection.

MB: Megabytes

M-Bone: The MBONE (Multicast Backbone) is a virtual network layered on top of the physical Internet to support the routing of multicast packets. For more information, see www.mbone.com.

Mbps: Megabits per second

MBR: Short for ***multiple (or multi) bit rate***, and also known as ***adaptive streaming,*** MBR is a technique by which several streams, compressed at different bitrates, are encoded together into a single file. When the client calls for the media file, a negotiation between client and server determines the available bandwidth, and the appropriate stream is transmitted.

media: A term with many different meanings, in the context of ***streaming media***, it refers to video, animation, and audio. The term "media" may also refer to something used for storage or transmission, such as tapes, diskettes, CD-ROMs, DVDs, or networks such as the Internet.

media server: Specialized server software that takes advantage of appropriate Web transfer protocols such as RTSP (real time streaming protocol), as well as special communication techniques between clients and servers, to facilitate the continuous playback of synchronized audio and video in real time, adjusting the streams transmitted to the actual bandwidth available. Media server software may be running on discrete hardware, or can be deployed in combination with Web server software running on the same device.

metafile: A file containing information that describes or specifies another file. Some streaming media formats use metafiles to invoke the client media player and/or to specify the location of a streaming file on a media server.

mic: Pronounced like "mike;" short for microphone.

MPEG: *Motion Pictures Expert Group* of the International Organization for Standardization (ISO) has defined multiple standards for compressing audio and video sequences. Setting it apart from ***JPEG*** which compresses individual frames, MPEG compression uses a technique where the ***differences*** in what has changed between one frame and its predecessor are calculated and encoded. MPEG is both a type of ***compression*** and a video ***format.*** "MPEG-1" was initially designed to deliver near-broadcast quality video through a standard speed CD-ROM. Playback of MPEG-1 video requires either a software decoder coupled with a high-end machine, or a hardware decoder. "MPEG-2" is the broadcast quality video found on ***DVD***'s. It requires a hardware decoder (e.g.; a DVD-ROM player) for playback. "MPEG-4" is a relatively new standard, now being used in some architectures for streaming media.

multiple (or multi) bit rate: See *MBR*.

multicast: Multicast is an efficient way to transmit the same media stream to many recipients simultaneously, by replicating the stream at router hops where the path to different multicast group members diverges. Multicast end-users experience the same portion of the media, at the same time (in contrast to ***unicast***, which can allow each end-user to control their own experience when accessing pre-recorded, or on-demand, files). The infrastructure to handle multicasting, known as the ***M-Bone*** (multicast backbone) is still emerging; the Internet is not yet ready for the popular proliferation of multicasting. The terms "multicast" and ***narrowcast*** are sometimes used interchangeably, although "multicast" more specifically refers to the actual technology inherent in the process.

narrowband: Low-bandwidth (typically dial-up) network connections usually 56 Kbps or lower.

narrowcast: Transmission of media to multiple end-users but, as differentiated from ***broadcast***, not to everyone on a network

NTSC: *National Television Standards Committee* standard for color television transmission used in the United States, Japan and elsewhere. NTSC incorporates an ***interlaced*** display with 60 ***fields*** per second, 29.97 ***frames per second***.

on-demand: Media which is not transmitted live, as it is recorded, but is made available as an archive on a server, for end-users to access when they wish. A television broadcast is "live;" renting a video and watching it at home is "on-demand."

optimize: Another term for ***pre-process.***

packet: Although computers and modems can send data one character at a time, it's more efficient to send information in larger blocks called data "packets," or ***datagrams***. When using the standard Internet protocol, TCP/IP, packets are typically around 1,500 characters. Packets consist of the data being transmitted plus the IP address information of the sender and the recipient

PAL: *Phase-alternating line* television standard popular in most European and South American countries. PAL uses an ***interlaced*** display with 50 ***fields*** per second, 25 ***frames per second***.

pixel: An abbreviation for ***picture element***. The minimum computer display element, represented as a point with a specified color and intensity level. One way to measure image ***resolution*** is by the number of pixels used to create the image.

player: In a multimedia ***architecture***, the client software application, typically a ***plug-in***, that enables playback of the media.

plug-in: A plug-in extends the capabilities of a Web browser, enabling the client to display or playback a file type which the browser itself cannot handle.

Pre-processing: Sometimes called ***optimizing***, pre-processing removes non-essential information from your video and audio—information that is difficult to encode and/or does not substantively add to the quality of the streamed media. So pre-processing prior to encoding reduces the burden on the compressor, potentially saving time and CPU capacity.

post-production: The phase of a film or video project that involves editing and assembling footage and adding effects, graphics, titles, and sound.

pre-production: The planning phase of a film or video project—usually completed prior to commencing production.

production: The phase of a film or video project comprised of shooting or recording raw footage.

progressive download: Also known as ***pseudo-streaming,*** "progressive download" allows end-users to experience media accessed via a network such as the Internet, while the media file is still in the process of downloading; as opposed to ***downloadable media,*** which cannot be played back until the entire file is received. Unlike ***true streaming,*** progressive download leaves a copy of the media file on the client. Progressive download is also called ***HTTP streaming*** because Web server software using standard protocols (HTTP servers) can deliver progressive download files, unlike true streaming, which takes advantage of the special protocols used by ***media server*** software to adjust transmission to match the actual available bandwidth.

protocol: A formal description of, i.e., rules for, how specific types of computer systems interact.

pseudo-streaming: See *progressive download.*

QuickTime: Apple's multi-platform, multimedia software ***architecture***; used by software developers, hardware manufacturers, and content creators to author and publish synchronized graphics, sound, video, text, music, VR, and 3D media. "QuickTime 4" includes strong support for "real" (RTSP) ***streaming.***

RealMedia: *Architecture* developed by RealNetworks to deliver audio and video content on the Web. Along with QuickTime and Windows Media, one of three dominant streaming architectures.

real-time: In computing, refers to an operating mode under which data is received, processed and the results returned so quickly as to seem instantaneous. In terms of streaming media, refers to content streamed live, or as-it-happens.

reference movie: A *metafile* in the QuickTime architecture, used to invoke the QuickTime player and/or to refer the browser to an adaptive bit rate (MBR) file.

resolution: The amount of information in each frame of video, normally represented by the number of horizontal ***pixels*** times the number of vertical pixels (e.g. 640x480). All other things being equal, a higher resolution will result in a better quality image.

rich media: Another term for ***multimedia.***

RTP: Realtime Transfer Protocol is used as the transfer protocol for RTSP streaming (true streaming), and can deliver a streaming media file to one or more end-users simultaneously (i.e., can ***unicast*** or ***multicast***).

RTSP: Realtime Streaming Protocol provides the functionality allowing end-users to randomly access and control the delivery of streaming media content.

server cluster or **server farm:** A group of networked servers that streamline internal processes by distributing the workload and expedite computing processes by harnessing the power of multiple servers. Load-balancing software tracks demand for processing power from different machines, prioritizing the tasks, and scheduling and rescheduling them depending on priority and demand users put on the network. Redundancy ensures that if one server in the farm fails, another can step in as backup.

SECAM: Similar to *PAL* at 25 FPS, the SECAM format is employed primarily in France, the Middle East, and Africa. It is only used for broadcasting. In countries employing the SECAM standard, PAL format cameras and decks are used.

skin: A custom GUI (graphical user interface) designed for a specific media player.

SMIL: Pronounced like "smile," SMIL stands for Synchronized Multimedia Integration Language. Like HTML, it is a markup language designed to be easy to learn and deploy on Web sites. SMIL was created specifically to solve the problems of coordinating the display of a variety of multimedia on Web sites. By using a single timeline for all of the media on a page, display can be time-coordinated and synchronized.

spatial compression: See *intra-frame compression.*

static media: Refers to design elements that do not incorporate motion or sound such as still photos or graphics.

stream: Data of a distinct type sent from server to client at a rate defined by the server. For a typical video broadcast, for example, one stream could consist of the video signal, one stream could consist of the audio data, and one stream could contain the closed caption information. In most cases, each stream is served at the rate that it should be rendered by the client.

streaming: Process of sending media over the Internet or other network, allowing playback on the desktop as the video is received, rather than requiring that the entire file be downloaded prior to playback.

SWF: The SWF graphic file format is a version of the Macromedia Flash Player vector-based graphics format introduced in 1997. The SWF file format is ideal for presenting vector-based interactive and animated graphics with sound for the Web. Vector images are ideal for graphics with solid areas of color and distinct object definitions. Because a SWF file is vector-based, its graphics are scalable and play back smoothly on any screen size and across multiple platforms. A vector animation usually has a smaller file size than a bitmap animation. For more information see the online Adobe Technical Guides at: ***www.adobe.com/support/techguides/webpublishing/flash/***

TCP/IP: These two protocols (Transmission Control Protocol/Internet Protocol) were developed by the U.S. military to allow computers to talk to each other over long distance networks. IP is responsible for moving packets of data between nodes. TCP is responsible for verifying delivery from client to server. TCP/IP forms the basis of the Internet, and is built into most every common operating system.

temporal compression: See *inter-frame compression.*

timeline: The graphical representation of program length onto which video, audio and graphics clips are arranged.

true streaming: Affording real-time access to content via the Internet or an intranet, true streaming is enabled by a specialized server application, that relies on streaming protocols to adjust the rate of transmission to accommodate available bandwidth.

tunneling: The use of specially designed paths to carry multicast traffic over the Internet.

24-bit color: Type of color representation used by current computers. For each of the Red, Green, and Blue components, 8 bits of information are stored and transmitted—24 bits in total. With these 24 bits of information, over a million different variations of color can be represented.

UDP: User Datagram Protocol, like TCP, runs on top of IP networks. Unlike TCP/IP, UDP/IP provides very little error recovery, offering instead a direct way to send and receive datagrams over an IP network. It is primarily used for streaming over the Internet.

unicast: The technique whereby a single stream is transmitted to a single end-user; i.e., each end-user gets a unique stream. Bandwidth-hogging unicasting is not as efficient as ***multicasting***.

upload: The act of one computer sending data, via a network, to another computer.

URL: A Uniform Resource Locator is the "address" used to find a document or resource on the World Wide Web. The first part of the address specifies the protocol (typically HTTP for Web pages, FTP for files not residing on a Web server, or RTSP for streaming files); the second part specifies the IP address, or domain name; and the rest specifies the directory structure for finding the discrete file on the host computer.

URL flip: A coded marker embedded in the timeline of video or audio that calls up a link to a Web page during playback.

video capture card (or board): Installed inside a computer, adds the functionality needed to ***digitize*** analog video for use by the computer. Using a hardware or software ***codec***, the capture card also compresses video in and decompresses video out for display on a television monitor.

VOD: Video On Demand refers to streaming files that are archived on a streaming server and may be accessed by an end-user at any time, as contrasted with live, or real-time, content.

Webcasting: The technique of broadcasting media over an intranet, extranet, or the Internet.

Web server streaming: Yet another term for ***HTTP streaming***, ***pseudo-streaming***, or ***progressive download***; see ***progressive download***.

Windows Media: Offered by Microsoft, one of the three dominant architectures for distributing media on the Web, including streaming media.