

Producing Desktop Video

by Nels Johnson, author of
“Web Developer’s Guide to Multicasting”

August 1997



Precept Software, Inc.
1072 Arastradero Road
Palo Alto, CA 94304
Phone: (650) 845-5200
Fax: (650) 845-5235
info@precept.com

<http://www.precept.com>

Producing Desktop Video

Since its debut in 1991, digital video on the PC (desktop video) has matured into a stable, easy-to-manage type of multimedia. Now desktop video is migrating to PC networks, specifically corporate intranets and, to some extent, the Internet itself—with quality as good as when played from CD-ROM. The problem is, for anyone new to desktop video production, there remains a lack of standard, documented techniques for capturing, editing and compressing such desktop video clips in the first place.

To help put such techniques in perspective, this white paper presents basic strategies for creating finished desktop movies suitable for playing over a network (or even from CD-ROM). These strategies can be implemented on standard PC equipment using standard video sources (such as videotape decks) and off-the-shelf digitizing equipment easily installed on most PCs. What's most important to understand is that networked video is still multimedia, and just as easy to produce and manage as it was prior to being distributed over networks.

In practical terms, networked video can be broken down into two basic categories: stored and live (as opposed to CD-ROM media, which is simply stored). Stored assets are digitized with appropriate hardware and software, encoded with suitable compressors, deposited on hard drives, Jaz cartridges or CD-ROM, then delivered to consumers at scheduled times, or on demand. Live content is digitized, compressed and streamed to consumers by live video servers all at the same time, with little (if any) actual production.

Note that the word live doesn't necessarily imply that the content comes from a live performance, although it might. Rather it means that the video is digitized with a capture card in real time (usually from a camera or a video tape deck) and not immediately stored on a mass storage device prior to distribution on the network. The balance of this document covers production of desktop video belonging to each of these two categories, although stored content receives the major focus.

Getting Started

One good way to approach the desktop video production process is to break it down conceptually (see Figure 1). Generally speaking, any PC that captures (digitizes) video from analog videotape has five generic components:

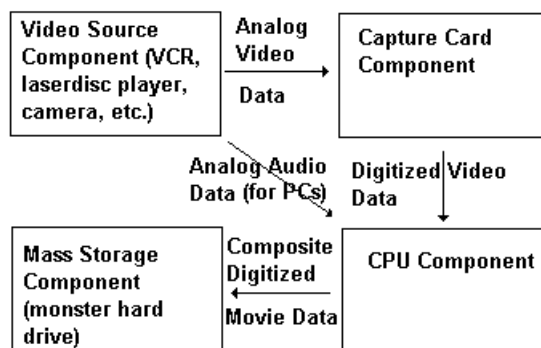


Figure 1

The CPU—The Computer itself.

The Capture Card—The device in the computer that digitizes the incoming analog video signal from the tape deck.

The Mass Storage Volume—Usually a large-capacity hard drive, used for temporary of raw, uncompressed digitized video.

The Analog Media Source—The video deck (or camera) that the video plays from at capture time.

Video Editing Software—A program (like Adobe Premiere or Ulead Systems' Media Studio Pro) for editing and compressing captured video clips.

Despite advances in peripheral equipment and raw computing power over the last few years, most PC-based video production stations are still composed of these five key components. To understand how they work together, let's look at the role of each one individually.

The CPU Component

The role of the CPU in the digitization process is to synchronize the activities of the capture card and the mass storage device. When selecting this component, keep in mind that, the more powerful the CPU, the more time you will save—especially when compressing your final video clips after you have captured and edited them. Don't forget that you'll need a sound card installed if you want to capture and play back your video clip's audio track.

The Capture Card

This is the component subject to the most controversy and marketing hype. At the high end are products like The Targa 2000 and the DPS Perception (\$2,000 to \$4,000, depending on overall configuration), although these cards are basically designed for outputting digital video back to tape at full-screen resolution—as opposed to making quarter-screen desktop video clips. Still, many multimedia production firms invest in these top-drawer cards for the overall high quality they provide.

In the mid-range are solutions like the Intel Smart Video Recorder (under \$200), Video Labs' Stinger (\$200), and Multimedia Access' Osprey card, all of which deliver very good production values for multimedia but are not suitable for outputting desktop video clips back to tape. Generally speaking, if a capture card can digitize 30 frames per second at 320 x 240 (quarter-screen resolution) it will be fine for almost all desktop video applications. When selecting this component, make sure that it is specifically supported by your PC's operating system (Windows 95 or Windows NT).

To capture and produce MPEG digital video clips, you will need an MPEG encoder card such as FutureTel's Prime View II product. Professional MPEG encoders capture and compress video simultaneously in real time. In other words, when you capture a clip with a card like the Prime View II, that clip can be considered finished and ready for distribution immediately—assuming you are happy with the production values and overall audio/video aesthetics.

The Mass Storage Volume

The main role of the mass storage volume is to store raw video data digitized by the capture card, although it is also important for holding video clips during the editing and compression processes. Because uncompressed video takes up huge amounts of storage space (up to one gigabyte for several minutes of raw, quarter-screen footage), most producers use external, multi-gigabyte SCSI hard drives for this purpose.

If your CPU doesn't already have a SCSI card (or built-in SCSI on the motherboard) you'll need to invest in one before attaching a SCSI mass storage volume. Several years ago, getting SCSI devices to work properly on Windows machines was often problematic but now they work fine on most Windows95 and Windows NT machines. If you are only making short clips (under one minute) you may be able to use your internal hard drive as your mass storage component (assuming it has at least a two-gigabyte capacity) but this approach will likely be unsatisfactory in the long run.

The Analog Media Source

This component may take several forms: VCR, VTR, camera, laserdisc player, etc. Just as varied are the tape quality standards inherent in these devices: VHS, SVHS, Hi8, Betacam SP, and so on. Although it is fair to say that the overall quality of a captured video clip depends on the quality of the original tape stock (e.g. VHS vs. Betacam), keep in mind that other factors are involved as well, such as the number of times the tape has been copied, how well the content is lit, and the integrity of the audio track.

Apples to apples, betacam tape is significantly superior to VHS, although you'll need a betacam *deck* to play such tapes back at capture time. Such decks are expensive to own and rent. Also, your capture card should support at least *S-video* (as opposed to standard composite video), if not *component* video connections. Clearly, these are confusing issues for newcomers to desktop video production, but there are several books and online resources available which cover them in sufficient detail. And, once you start working with different tape standards on a regular basis, these issues usually clear up immediately.

The bottom line is, once you understand a video capture system at the conceptual level presented above, you'll be ready to move on to the editing and compression strategies discussed below. Like any complex technical process, capturing video takes practice, patience and an appreciation of the strengths and weaknesses of all the components. Overall, your main goal should be getting the highest quality uncompressed video onto your mass storage volume (i.e. your hard drive). Once you have this, you can proceed with confidence to the editing and compression stages.

Video Editing Software

This final component consists of software, as opposed to the prior four *hardware* components. As noted above, professional-level products like Adobe Premiere offer sophisticated video *effects* as well as video compression facilities. Because such effects (fades, irises, titles, barn doors, chroma keying, etc.) require substantial computing power, a very fast computer (Pentium 200+) is highly recommended when using any type of desktop video editing software.

In most discussions of desktop video software, you will likely hear the words *non-linear editing*. If you have never edited video, this term may seem meaningless. If you have a background in traditional, tape based video production, you will grasp the significance immediately. In essence, non-linear means that all the raw clips comprising an in-progress production sequence are available instantly for editing—as opposed to rolling reels of tape backward or forward to get to specific sections. The reason for this instant availability is that digitized clips are normally stored on computer hard drives, which use random-access data storage.

Editing and Compressing Desktop Video

To dramatize these techniques most effectively, let's begin with a real-life example: production of a Precept H.261 video clip, in Microsoft's standard AVI format, to be multicast using Precept's IP/TV software. The analog version of the movie is stored on a VHS video tape. We'll describe the process beginning with capture on a Windows95 PC, then editing and final compression using Adobe Premiere for Windows, followed by audio track tweaking with an off-the-rack product named Sound Forge.

Capturing the Movie

For the capture process, we'll employ the Intel Smart Video Recorder III (ISVR3), which can digitize raw video clips at 320 x 240, 15 frames per second, fully uncompressed, on fast machines. The capture application we'll use is Asymetix's Digital Video Producer (DVP), since that product is often bundled with the ISVR3. Note that while the recipe for this capture operation is specific to the ISVR3/DVP combination, it accurately reflects the level of complexity involved in a generic Windows capture session.

Assuming all your gear is connected properly (with the tape in the deck and properly queued), launch the DVP capture program.

For our mass storage component, we have a 1-gig external SCSI hard drive attached. To employ this volume, click the *Set Capture File* option in the *File* menu, navigate to the drive in the resulting dialog, and key in a name the raw capture file (for example, RAW1.AVI).

Click the *Allocate File Space* item in the *File* menu.. The traditional wisdom here is that you should allocate just 1 megabyte and let the program and your system work together to allocate additional space as needed on the fly.

In the *Options* menu, click the *Audio Format* item. In the resulting *Audio Options* dialog, choose the lowest quality settings: 11 KHz, 8-bit, mono. Note that we're capturing lower than 44 KHz, 16-bit because of the requirements of our finished clip. In general, you should always capture with your audio settings as rich as possible—assuming your system won't compromise video quality in the process. If your don't capture audio with high quality, there are few ways you can improve it later. Tweaking rich audio tracks is covered below. Audio input volume should be adjusted with the built-in Windows volume control (the speaker icon normally at the bottom right of your Windows95 desktop).

Click the *Video Format* item in the *Options* menu to display the *Video Format* panel (see *Figure 2*). As noted above, the ISVR3 can digitize at 320 x 240, 15 fps, uncompressed (a.k.a. RGB 24) without dropping frames. Prior versions of the card had trouble achieving this performance level. To keep those earlier cards from dropping frames at capture time on slower CPUs, Intel supplied a semi-raw codec named YVU9, which is also included in the new package. If your CPU drops frames when capturing at RGB24, try switching to YVU9 mode.

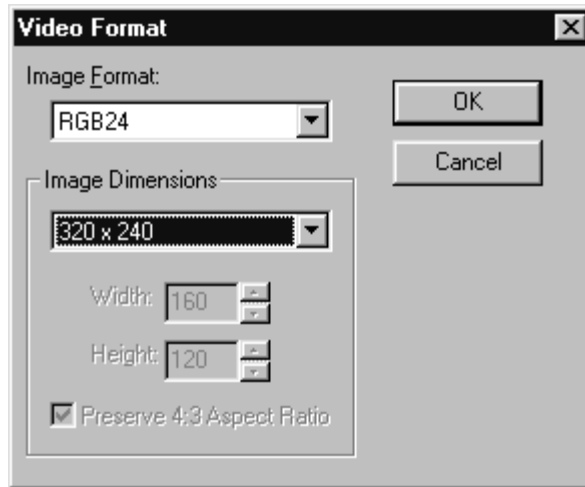


Figure 2

Click the *Video Source* item in the *Options* menu to see the *Video Source* dialog. If you wish, experiment with the *Brightness*, *Contrast* and *Saturation* settings to bring out the best quality in your analog video source. Sometimes you can make high quality VHS source look as good as, say, fair quality Betacam SP content by carefully calibrating these controls at capture time.

Check the *Preview Mode* item in the *Options* menu if you want to watch the video play in real time as it is digitized, or if you want to see where you are as you cue the video tape on your deck.

Click the *Settings* item in the *Capture* menu to bring up the *Capture Settings* panel. In general, it's always a good idea to capture at the frame rate which you expect to use for the finished movie. Note, however, that if you are creating rich raw archives from which to make final movies of yet undetermined quality, you should capture with video and audio attributes set as high as possible. This example requires us to set the frame rate to 15 fps.

The *Capture Storage* setting can seem confusing but it really isn't. Checking *Capture to Disk* makes the program do just that. *Capture to Memory* means capture to memory until memory runs out, then capture to disk. For general efficiency, go ahead and check *Capture to Memory*.

Cue the video tape in the VCR to a point several seconds prior to where you want to start capturing and press the play button on your video deck. Once the tape is rolling smoothly, click the *Capture Video* button in the tool bar. If preview mode is enabled, you'll see the video in the preview window. All the buttons in the tool bar should now be gray except for *Stop*. The activity light on your external SCSI hard drive should be flashing continuously.

Let the process proceed until the content you want captured has played out, then press the escape key or click the *Stop* button in the toolbar (don't forget to press the stop button on your video deck). If everything went normal, you should have a fully-formed raw video file stored on your external SCSI hard drive.

Editing the Movie

With a raw movie successfully captured and stored, we can now edit and compress it. Our video editing workbench program for this example will be Adobe Premiere for Windows 4.2, although there are other excellent products available for doing this kind of work, such as Ulead Systems' Media Studio Pro. If you're not familiar with Premiere, this section will acquaint you with the feel of its main features and general way of doing things. Aside from its current inability to handle MPEG directly, Premiere is an almost seamless environment for producing desktop movie clips.

Remember that our goal is to produce a Precept H.261 clip suitable for multicasting with IP/TV on an intranet. Again, the single biggest time saver when it comes to working with any desktop video editing program is a fast computer. Even as a talented amateur, you can seem like a seasoned pro if your movies take 20 minutes to render and compress rather than 4 hours (based on rough differences between, say, a 486 machine and a Pentium 166 or faster). In other words, if you're going to do serious desktop editing, get a serious workstation that won't slow you down.

Here is the basic strategy:

When you fire up Premiere for Windows, you are greeted with a splash screen and a configuration panel or two, but you ultimately wind up in the main work area. There are two basic top level windows: the *Project* window and the *Construction* window. Other windows can be opened to get at other important features, but this is where to concentrate at first.

The first thing we need to do is get our raw movie into an active Premiere project. Click on the *Import* item in the *File* menu, then select the *File* item to bring up the *Import* dialog. Navigate to the external SCSI drive and select the video clip to import—in this case RAW1.AVI. An icon for the imported clip will appear in the *Project* window. You can double-click this icon to bring up a new window in which you can set the in and out points for the clip (among other things).

Now we can put this project member into the *Construction* window. Drag the file icon for RAW1.AVI from the *Project* window to the desired timeline in the *Construction* window. Now is a good time to save the project (with the *File/Save* menu item). Saving it to the same location as the rest of the media in the project (in this case, the external SCSI drive) is generally recommended.

Let's add a quick title using Premiere's excellent titling facility. To run the title editor, click *File/New/Title*. In the interest of brevity, we'll dispense with the title creation details since they are very straightforward and readily available in the Premiere documentation. Store the finished title, H261.PTL, in the same place on the external SCSI drive as the raw movie and the project file, then import it and drag it onto the same time line in the *Construction* window as the raw movie clip.

Finally we're ready to render and compress the movie. First, save the project (which now includes the title). Now click the *Make Movie* item in the *Make* menu.

Navigate to the location of the raw capture file (RAW1.AVI). Key in a name for the finished movie, in this case H261.AVI. Don't worry about the attributes listed at the bottom of this dialog, since we're about to change them. There are two important sub-dialogs to deal with here. Go to the first one by clicking the *Output Options* button.

This panel normally comes up with default values. We are building the entire AVI Movie (as opposed to the Work Area) at 320 x 240 with 11 KHz / 8-bit / mono uncompressed audio, interleaved with video at the single frame level. Click *OK* to return to the *Make Movie* dialog.

From the *Make Movie* dialog, click the *Compression* button to invoke the *Compression Settings* panel (see Figure 3). Select the *Precept H.261 Compressor* (installed automatically when the Precept Viewer or Server is installed). Select *Thousands* of colors. Set the quality slider to 75% for now. Note that using a quality setting of 100% can produce unwanted side effects in an H.261 movie if your data rate is too low. It's worth experimenting with these two inter-related settings (Quality and Data Rate) for a while to get the hang of it. In this case, we'll limit the data rate to 100 kbs, which may seem low for a 320 x 240 movie, but don't forget that this clip will be played in a networked environment.

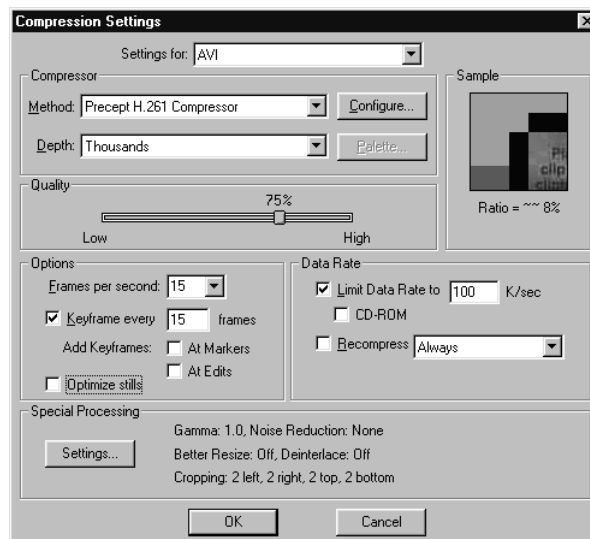


Figure 3

If you wish, click the *Settings* button to enter the *Special Processing* dialog. Here you can trim pixel fringe off the sides of your video frames (where lots of video noise lives). Don't forget to check the box to scale the video window back up to 320 x 240.

Click the *OK* button to return to the *Make Movie* dialog. The attributes at the bottom of the panel should now reflect our new settings. Assuming everything is in order, go ahead and click the *Save* button.

Premiere's *Making Movie* dialog now comes up with a progress bar and a read-out of the encoding time remaining. You can press the escape key at any time to stop the compression process and play that portion of the movie that has been compressed so far.

When the encoding is complete, play the movie natively by clicking on it via the Desktop Explorer. This makes for more reliable performance than playing the movie back from within Premiere. You may want to close Premiere first (after saving your project).

If the clip performs well on your local machine (good lip-synch, unbroken audio, etc.), it is ready for the next step. Remember that we are still making a Precept H.261 movie for multi-casting with Precept's Server and Viewer software. That next step is audio track processing, in this case with Sound Forge 4.0. At this point the video track encoding is complete.

Tweaking A Movie's Audio Track

Because the movie being made in this example requires an audio track calibrated to 8 KHz/8-bit / mono, we need a separate tool at this point. In general, most desktop video clips produced on PCs have so-called MPC audio tracks, such as 11 KHz/8-bit/mono. To reduce our 11 KHz audio sampling rate to 8 KHz, we'll use Sound Forge 4.0 from Sonic Foundry. This will give us the chance to look at some of Sound Forge's other features as well. For the record, you can also convert from 11 KHz to 8 KHz with the much simpler Windows95 Sound Recorder application.

Here's a sample recipe:

Launch Sound Forge and *Open* the AVI file H261.AVI. One of the nice things about Sound Forge is that it lets you load a whole AVI file to work on the audio track in place. Otherwise you would have to separate a movie's audio track, work on it, then rejoin the audio track to the video track of the AVI file.

To re-sample this audio track, select the *Resample* item from Sound Forge's *Process* menu (see *Figure 4*). You can key in 8,000 directly or use the up and down arrows to the right of the edit field to change the existing value. If you need information about the rest of the controls in this dialog (or any of Sound Forge's control panels) just click the help button. Sound Forge's help system is quite extensive.

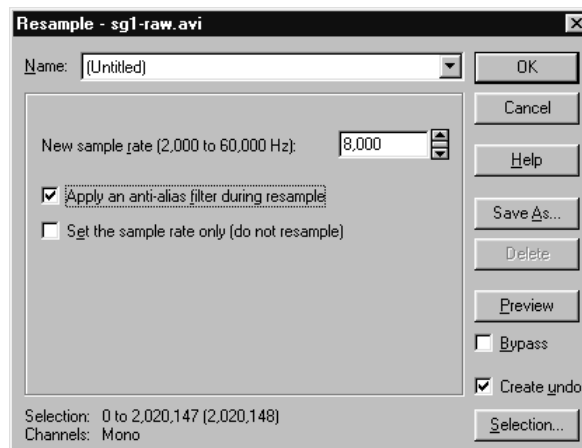


Figure 4

Another of Sound Forge's excellent features is its normalization facility. This ensures that the various frequencies in an audio clip are spread across as wide a band as possible. To normalize the audio track of H261.AVI, select the whole track with *Edit/Select All*, then click *Normalize* from the *Process* menu.

With these two operations complete, exit Sound Forge by doing a *File/Save*, then *File/Exit* (or just exit the program and confirm that you want to save the AVI file with the edited audio track). Serious desktop producers will likely want to edit their movies and audio-only clips in other ways as well, especially if they are work with very rich audio attributes (such as 44 KHz/16-bit /stereo).

Note: To make sure that a video compressor does the best possible job within the constraints of a specified data rate, some producers like to encode movies in Premiere without an audio track, then edit and attach the audio track after the video is compressed. Sound Forge is up to this task as well.

Producing MPEG-1 Content

As of this writing, capturing, editing and compressing MPEG files directly on a PC from video tape is still an evolving art. Prior to 1997, most desktop video producers did not own top of the line MPEG capture boards, nor did they have a reliable way to convert existing digital assets (like QuickTime and AVI movies) to and from the MPEG file format. Both of the above situations are now changing as professional MPEG capture hardware drops below \$5,000 and software companies like and Xing and Digigami ship high quality conversion tools for going from uncompressed AVI and QuickTime files to MPEG.

Digitizing video with a high-end MPEG board like FutureTel's PrimeView II (see Figure 5) allows you to capture and compress simultaneously in real time, which can give you finished MPEG clips at the end of your capture session. Unfortunately, professional MPEG editing suites (like Premiere in the AVI / H.261 world) have not fully converged on the PC desktop. In other words, if all you need to do is make finished MPEG clips from finished video tapes, a top-drawer MPEG capture board might make good sense. However, if you need to edit your video content after you capture it, especially with fades, custom transitions and special effects, you might have trouble finding a Windows-based editing suite as of July, 1997. Of course, this is sure to change in the near future, and some existing Windows products (such as FutureTel's ClipView) do splice MPEG files together with cuts only edits.



Figure 5

As noted above, the other principal method for producing high-quality MPEG clips is to convert them from uncompressed AVI (or QuickTime) format to MPEG using a software tool. One good strategy is to edit raw AVI clips together in Premiere, along with titles, transitions and other effects, then make the finished movie (as in the example above) as yet another uncompressed clip—which will keep the video quality very high. You can then convert this final uncompressed clip with, say, Xing's MPEG Encoder 2 (see Figure 6) to get an MPEG file arguably as good as one encoded in real time with an expensive MPEG capture card.

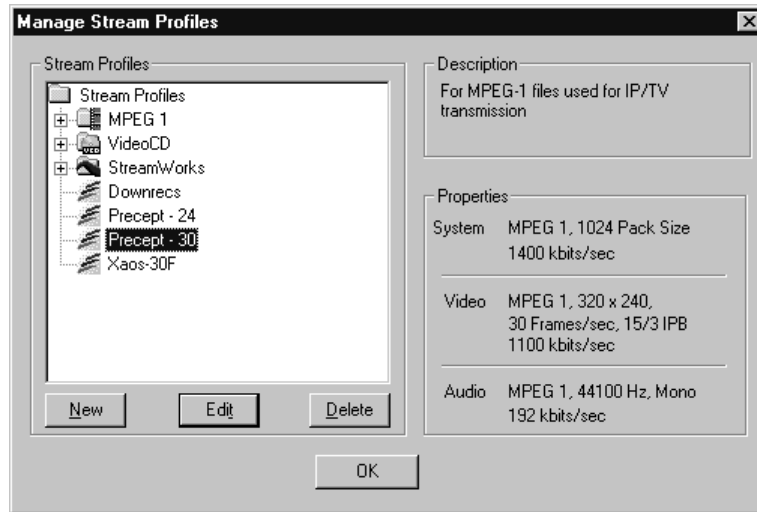


Figure 6

If you choose this method, it is highly recommended that you capture your clips at 30 frames per second and keep the entire production at this frame rate until final version to MPEG, since MPEG's native frame rate for playback is 30 frames per second. Bear in mind also that MPEG audio is normally sampled at 16-bit, 44.1 KHz, stereo.

Live Desktop Video

Earlier in this document we identified two basic types of desktop video: stored and live. While stored video files (the main focus of this document) are a mature part of the multimedia landscape, live desktop video streams are slightly more exotic. As noted, live means either taken right from a camera or digitized from tape and streamed immediately. Recipients of such streams can't normally tell the difference (depending, of course, on the nature of the content).

Production of live desktop video comes down to a couple of basic elements: Quality of the capture card and quality of the analog video signal. Since off-line editing (like in the Premiere/H.261 example above) does not play a part in this process, cards that compress to final format in real time are used exclusively. The VideoLabs Stinger and Multimedia Access Osprey cards work well in this environment. If you need to stream live MPEG data, the FutureTel PrimeView II is a good choice for this process. The IP/TV Server works well with this product.

When setting up cameras, microphones and lighting equipment for live encoding events, observing the following tips can make a crucial difference in the quality of the presentation to the user:

In general, favor tight shots and close-ups over medium and long shots.

Insist that actors with speaking parts use lapel microphones (as opposed to using the camera mic to capture audio across patches of noisy air). Run the audio signal from the speaker's lapel mic into the sound card in the PC housing the video capture card.

Light your subjects as well as possible with soft or reflected light (as opposed to harsh, direct or unnatural light). For important events, test various lighting strategies ahead of time with stand-in speakers.

Summary

The purpose of this document was to present an overview of the methods available for digitizing (capturing) desktop video files from videotape. Of special interest were formats suitable for network delivery, such as MPEG and H.261. As noted, there are few established strategies for performing such captures, mainly due to the evolving nature of the available hardware and software tools.

If you are setting up a facility for any kind of desktop video production, the key points to remember to remember from this document are:

Break down your digitization hardware into the four basic categories (CPU, capture card, mass storage devices and analog video source) to isolate the performance requirements for each piece of equipment.

Use a professional-level editing program (like Adobe Premiere) for editing and compressing your movies.

Depending on the nature of your analog content (on your video tapes) consider investing in a high quality MPEG capture card, but only if your content doesn't need much editing.

Again, almost all desktop video—networked or not—is essentially PC-based multimedia. As such, it can be captured, edited and made highly presentable using a variety of powerful yet easy to master tools. The key is using the right tool for the right part of the job.



All trademarks are properties of their respective organizations. © 1997 Precept Software, Inc. All rights reserved. Product specifications are subject to change without notice. Check with Precept's Sales department for the most current information. 048 WP-03 08/97
