

# 13 | multimedia

AUDIO ON THE WEB WORKS VERY NICELY. Video is still a bit of a boondoggle. And making video accessible is so difficult you had best leave the job to the experts. And at present, there is no way for you the Web developer to become an expert.

## Goals

In this chapter:

- We'll understand the Big Four access techniques of captioning, audio description, subtitling, and dubbing.
- We'll explore the state of the art outside the Web and how the Web compares.
- We'll come up with a list of options for low- and medium-budget accessibility.

## What's the problem?

Well, that depends on how we define “problem.”

The accessibility problem is simple: Deaf people can't hear audio and blind people can't see video. The infrastructure problem is trickier: There are too many player and file formats for the various operating systems. While essentially every player can handle “universal” formats like MP3, each player's specific file format is proprietary to that player. It's much worse than the VHS/Beta discrepancy canonically cited as a parallel: Back in the day, it was always possible to find an A-list movie in both home-video formats. But, as if to spite real-world users, media files online are routinely offered in only one format, absolutely forcing you to run multiple players. Considering

accessibility specifically, different players and versions have different capacities and are incompatible, to varying degrees, with assistive technology like screen readers.

The *appropriateness* problem is intractable. Despite a few promising experiments here and there, there isn't enough bandwidth in the world to duplicate even the quality of broadcast television online. Nor is there any reason even to make that attempt, a fact lost on executives at media juggernauts, whose quest for some kind of ill-defined "convergence" threatens to ruin the Web as we know it in an ill-conceived effort to make it just like television.

We have the Web. We have television. Like matter and antimatter, the two should remain separate.

I am not wild about the use of video on the Web. (Audio is fine by me.) Yet I am not so religiously opposed to its use that I refuse to recognize that accessibility must be taken into account. But the utility and practicality of online video access actually mirror those of online video itself. Just as online multimedia aren't even remotely as good as TV, online multimedia accessibility isn't remotely as good as TV's. In fact, it's not as good as TV's was in 1989.

Closed accessibility features are hidden until they are activated. Open access features are always present and cannot be turned off.

## Defining our terms

As you read back in Chapter 4 ("What is media access?"), the broad categories of accessible media predating the Web are *captioning* for deaf and hard-of-hearing viewers and *audio description* for blind and visually-impaired audiences. These are two of what I call the Big Four access techniques; the other two are *subtitling* and *dubbing*, which this book does not really cover.

*Closed* accessibility features are hidden until they are activated. *Open* access features are always present and cannot be turned off.

## The basics

*Note: If you're already familiar with basic HTML, you can skip this section.*

HTML coding for multimedia leaves a lot to be desired. The oldschool technique is widely compatible but officially "deprecated," while the standards-compliant technique is poorly supported and has been known to crash browsers.

The oldschool technique is the `embed` element, which takes attributes similar to those of `img`. (In fact, technically you can use `embed` rather than `img` to specify a graphical image if you want.) An easy example:

```
<embed src="announce2.mov" width="320" height="256" />
```

It will not surprise you to learn that the width and height specifications govern the size of the window in which the file identified by `src` will play.

Netscape devised the embed element. It was never actually approved in a World Wide Web Consortium “recommendation.” It is of course widely used nonetheless.

The other oldschool technique, reserved for Java applets, is the `<applet></applet>` element, with a vast panoply of parameters. Life is too short to list them all in a section entitled “The basics,” but here’s one example:

```
<applet codebase="http://img.socks-online.co.uk/applets/classes"
code="sizes.class" width="350" height="200" alt="SockSizer&trade;"
align="left" >
</applet >
```

`<applet></applet>` has a few advantages, like the ability to include marked-up text or even graphics between the opening and closing tags (in theory, a browser unable to display the Java applet could display such content instead). You can and should also add an alt text inside the tag itself.

However, both those oldschool elements are now “deprecated” in favour of the allegedly superior `<object></object>` element, which is so generic it can encompass essentially anything – any “object,” including images, imagemaps, video, audio, lists, plain text, entire HTML documents. You can, for example, set up multiple `<object></object>` elements that enclose:

- a QuickTime video
- an ordinary still image that can be displayed if the video cannot
- a still image in a different file format if the previous file were undisplayable
- plain text that can be displayed if the image cannot

Such an example could be written as follows (and is not entirely farfetched given today’s limited browser support for the PNG format used here):

```
<object data="conf.mov" type="video/quicktime" title="Press conference (May
7/01)" width="300" height="300" >
<param name="pluginspage" value="http://quicktime.apple.com/" />
<param name="autoplay" value="true" />
<object data="desrosiers2.png" type="img/png" title="Yves-Ëacute;tienne
Desrosiers, CEO" >
Yves-Ëacute;tienne Desrosiers, CEO
<object data="desrosiers2.jpg" type="img/jpg" title="Yves-Ëacute;tienne
Desrosiers, CEO" >
Yves-Ëacute;tienne Desrosiers, CEO
</object >
</object >
</object >
```

Note the nested objects. You can place one object inside another in (usually) descending order of desirability or technical sophistication: A movie file, then maybe a sound file, then an image in a high-quality format like PNG (with enclosed alternative text), then maybe a JPEG image (ditto). The oft-cited principle of graceful degradation is at work here, or ought to be, if the element actually functions properly.

(The param elements, by the way, specify “initialization data”; compliant devices could, for example, automatically refer you to the download page for the QuickTime plug-in if you didn’t already have it.)

You don’t add an alt text or a longdescription to `<object></object>` per se. Any text that an `<object></object>` element might enclose will become the alternative text.

The bad news is that, for all its marvels, `<object></object>` is so poorly supported by real-world browsers – actually crashing Internet Explorer 4.01 for Windows in some unusual cases – that it is quite unusable. For video and audio files, you’re stuck with embed. Your choice then becomes writing a standards-compliant page that breaks browsers and doesn’t actually do what you want or writing a noncompliant page that works just fine. For the foreseeable future, noncompliance is the way to go. Yes, I know, I promised never to authorize or promote nonstandard HTML use, but if I stuck to that faithfully here, online video would essentially disappear.

## Technical infrastructure

Online video can hardly be considered “new media” in any strict sense. Video is not exactly a new addition to homes and businesses. Video on computer screens reaches all the way back to 1993, actually predating the Web. (Remember the Macintosh TV, with its cable-TV tuner and remote control?)

Unlike the entirely new task of making text-and-graphics Websites accessible, we have decades of experience in accessible video *outside* the Web. This history has rather raised expectations of what should be possible online. We should at least be able to exceed the capacities of “old media.”

So let’s consider television and home video.

- In North America, the so-called Line 21 captioning system, in use since 1979, gives us two usable streams of captioning. (Technically there are four channels, but only two are practicable. The first pair of channels – CC1 and CC2 – is sent down one pipe, while the other pair – CC3 and CC4 – has a separate pipe. To use CC1 and CC2 together, each gets half the total bandwidth. But if you use CC1 and CC3, each can have all the bandwidth of their respective pipes.)
- We also get two usable channels (out of four, with the same distribution as above) of text information that occupies all or half the screen depending on the device you’re watching.



DVD subtitles (from Tom Tykwer's *Run Lola Run*). True to the subtitling idiom (as opposed to captioning), the titles are an edited translation; they don't move to show who's speaking (in fact, who is speaking in this scene?); and no "non-speech information," like sound effects ("phone rings," "footsteps receding") or manner of speech ("sarcastically," "whispering") is shown. Subtitles are insufficient as an accessibility technique for deaf viewers.



DVD closed captions (also from *Run Lola Run*). Here we see the vaunted non-speech information crucial to making a film understandable to a deaf audience. One could dispute the caption positioning here. Note the white-on-black typography, and the lousy caption font my television set sticks me with.

- In Europe, the U.K., Australia, and other countries using World System Teletext (WST), several streams are available, with hundreds of available channels of full-screen text.
- Also in certain WST countries, a separate captioning system – Line 22, a variation of the one used in North America – has been available since 1992, with one caption stream.

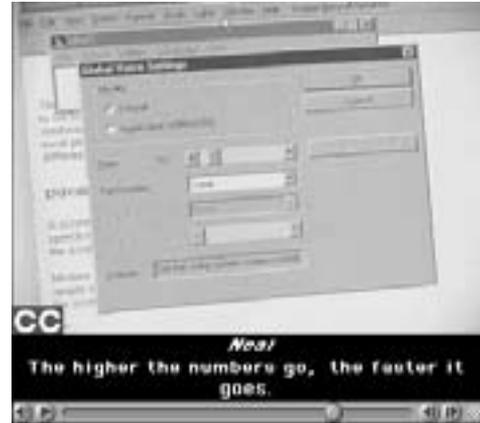
Flash:  
Nothing whatsoever.

- Home videotape devices in North America have always been able to record and play back captions. Videotape devices in WST countries cannot record World System Teletext captions (with rare exceptions); the transplanted North American Line 22 captioning system works with any VCR. In North America, then, home video has offered closed captions for nearly a full human generation, and for almost a decade in certain WST countries.
- Stereo television is widely used, in analogue and digital formats, around the world. In North America, it is uncommon but quite possible to use the second audio track (its actual name is Second Audio Program, or SAP) to deliver audio descriptions mixed in with main audio.
- Digital television systems – even those that are little more than tarted-up present-day analogue TV – offer at least two and usually many more audio channels. It then becomes possible to run a program with original audio, descriptions for the blind, dubbed dialogue, and descriptions in the language of the dubbing.
- DVDs offer up to 32 subtitle tracks (not exactly the same as captions, but the tracks can be put to equivalent use) and up to eight audio tracks. (The number of bits available on a DVD is finite; given that video is the whole point of DVDs and that video eats up a lot of space, you usually run out of available bits well before you run out of possible audio tracks. Subtitle tracks occupy far less space, but even I find it hard to imagine 32 titling variations.) NTSC DVDs can also carry closed captions – chiefly DVDs in Region 1 and in Japan (Region 2).

That's what we get to play with in the real world. What does the virtual world give us?

- QuickTime: No predefined limit to the number of text and audio tracks.
- RealVideo: Captions (one stream) and audio description (one stream), though the exact numbers are muddled given that you get one of each stream per language.
- Windows Media: "Multiple" captions, but no audio descriptions.
- Flash: Nothing whatsoever built into the data structure, but you can add text in a way that functions as captions.

Even if you wanted to duplicate the degree of accessibility available in North America (let alone in a medium like DVD), you couldn't do it in all the multimedia formats currently in use online. Pick one that seems to work well (like QuickTime) and you leave out everyone who doesn't have that player. (Admittedly, this would be more pressing if you chose Windows Media, which is new even for Windows users and essentially unused on Macintosh.)



Captioning as seen in RealPlayer (left) and QuickTime. Fonts and positioning are terrible (in some ways worse even than on TV); QuickTime makes it difficult to turn captions on and off (the “CC” button was added to make that possible), though doing so is much easier in RealPlayer. On the plus side, unlike on television, you can position captions outside the frame. It is not easy to “author” the separate text files necessary to create the captions and synchronize those files with the online video.

## SMIL

There does exist a platform-neutral, industry-standard markup language with which to create files for accessible media, including captioning and audio description. The so-called Synchronized Media Integration Language or SMIL is a World Wide Web Consortium “recommendation,” which is about as forceful and standardized as the W3C gets.

SMIL lets you cue text, audio, and video together in any combination. It is, in effect, a metalanguage (indeed, it’s based on XML) that describes what should appear when in a “time-based medium.” We’re not just talking about cinema, TV, radio, and music here: A slideshow or a PowerPoint presentation falls under the same category even if it contains nothing but words. (So would an animated GIF, theoretically.) SMIL handles anything that doesn’t just load once at a random moment unforeseen by the designer or author and sit there.

SMIL has been around since 1998. Player support is pretty good: QuickTime 4.1 and later and RealPlayer 8 and later support at least SMIL 1.0, while SMIL 2.0 support is present but incomplete in Internet Explorer 5.5 and later for Windows.

When it comes to creating a SMIL file, though, we harken back to the early days of the Web’s commercial boom. Is it better to code by hand (preferred by oldschool Web programmers; allows precision and full standards compliance, but is as slow as it is error-prone) or use a graphical

This all-QuickTime captioning experiment by WGBH gives you English or Spanish captions (true captions even in Spanish, not just foreign-language subtitles) and/or English or Spanish “enhancements”—paragraphs of description of the musical style, instruments used, and the like. Very slick, and a massive programming task.



editing program (preferred by neophytes; tends to produce nonstandard, hard-to-maintain markup but is noticeably faster)?

A great many professional Web developers mix and match, and indeed the seeming duality of text vs. graphical HTML editing barely scratches the surface of all the software involved. You might design your comps in Photoshop, touch them up in ImageReady or Fireworks, create HTML layouts in Dreamweaver, write CGI and database programs in a text editor, and test in a range of browsers.

Now you need to ask yourself: On top of all that, do I want to learn a new markup language known as SMIL, or do I want to learn new *software* that will produce SMIL files for me?

You actually have quite a few options for authoring programs; a good listing is found at the W3C site itself, at [w3.org/AudioVideo](http://w3.org/AudioVideo). Your source video track, any descriptions, and any captions or transcripts you create (which you will learn about shortly) are all files that SMIL can manipulate.

Naturally, Microsoft marches to the beat of a different drummer. Support for SMIL in Windows Media Player is somewhat nonstandard. (Of course, to be “somewhat” nonstandard equates with “nonstandard, period,” but when it comes to accessibility, standards have a tendency to be imperfectly supported.) Windows Media supports a subset of SMIL 2.0, which Microsoft has given the charming name of HTML+Time. A standards-compliant SMIL document will not necessarily work in Windows Media, but captioning and description files are simple enough that they will usually work. (Or so I gather. Quite a bit of research, including questions posed directly to Microsoft, failed to provide a definitive answer.)

Further, Microsoft has its own SMIL-like markup language for time-based media, the Synchronized Accessible Media Interchange or SAMI. Like SMIL,

SAMI resembles HTML and is not particularly difficult to learn. Yet there are no authoring programs for the format; it's nonstandard; and it works only in Windows Media Player, and even then only imperfectly. (A Microsoft Web page tell us simply: "Windows Media Player 5 supports a subset of the full SAMI specification." It's Microsoft's own file format and a Microsoft player supports only "a subset" of it?)

SAMI has been rendered obsolete by SMIL and its stepchild, HTML+Time. Unless you are creating accessible media for an intranet or some other installation that you absolutely know uses Windows Media Player and you are entirely sure you will never, ever need to migrate accessibility files anywhere else online, you should do no authoring at all in SAMI. Use SMIL instead.

### Interface issues

There's also a wee problem with user interfaces. While this is not technically your problem as a developer or designer, it is a necessary detail in understanding the practical obstacles standing in the way of accessible multimedia.

A deaf or hard-of-hearing person has no particular difficulty using the visual controls of media players. A blind or visually-impaired or a mobility-impaired person definitely does have difficulty given that it is normal for players to emulate VCR-style control buttons that you have to click with a mouse. There are two classes of player: Stand-alone (running as its own application) and embedded in a browser. It is quite often impossible to use the **Tab** key to move from a surrounding Web page to the player application and within the regions of the player. Keyboard equivalents are incomplete and insufficient in all players except Windows Media on the Windows platform per se, though all players, even Windows Media, suffer from the separation of stand-alone use vs. embedding in HTML pages. (How do you traverse the boundary?)

Screen-reader users are particularly ill-served. They're already dealing with layer upon layer of abstraction:

- The computer hardware. (With screen readers, extensive and unusual keystrokes are the norm.)
- The computer operating system.
- The screen reader (and/or Braille display, for that matter).
- The application software, like a browser or player, with possibly numerous open windows.
- The content within the software.
- The accessibility features of that content.

Yes, the tendrils of accessibility really do extend that deep. How else do you gain access to the audio descriptions of the video file running in your RealPlayer window inside Microsoft Internet Explorer for Windows under the

Multimedia players can pop up as free-standing miniapplications (even if still under the control of the browser), as in the example at top, the trailer for the Australian film *Let's Get Skase*.

Or the multimedia player (actually, the multimedia itself – the player is merely spawned by the browser) can be embedded inside an HTML document, typically within `<frame></frame>`, `<iframe></iframe>`, `<embed></embed>`, or `<object></object>` or in a table cell. The *Bourne Identity* example at bottom shows the embedded approach, which is admittedly nicer to look at.

At time of writing, by the way, I had never seen a single online movie trailer with captions or descriptions (both of which are readily achievable given movie-studio budgets and levels of expertise).



ægis of a screen reader you operate using the split keyboard you find pleasant to type on?

At time of writing, leading screen readers provided incomplete access to all the controls and features of multimedia player applications. There's nothing you the author can do about it. In fact, it is in no one's interest for authors to view this layer cake of incompatibilities as a technical challenge to be solved for their own particular Websites. It is not all that helpful for you to beaver away at programming a kludge that blind visitors can use to gain access to audio descriptions on your site, for example, to overcome some technical limitation of the sort just described. That same kludge probably relies on nonstandard methods (as a lot of JavaScript programming could be described) and definitely won't work anywhere else.

Even the seemingly simple case of an incompatibility of interface for captioning shows us how deep the technical obstacles run. The free QuickTime Player and the QuickTime Pro variant you must pay for will handle text tracks differently. If you encode hidden text tracks in a QuickTime movie, only the Pro version can turn them on. The free version has no access to the text at all through the player's native controls; it might as well not be there. (If you save a version with text tracks visible, they stay visible forevermore. Then the titles are open, not closed.)

*As you can see, even simple tasks face multilayered technical barriers.*

You can work up a little onscreen button that turns captions on and off through calls to the QuickTime scripting language known as ActionScript. In fact, this is so convincing a solution to the problem that contacts at Apple and the WGBH Educational Foundation all enthusiastically support it (rather than the more obvious permanent solution of fixing the players).

You'd think this approach would be just fine given that deaf people can see the nice onscreen button you have programmed. Here's a question for you: How does someone who can't use the mouse click the button? The question is somewhat unfair, actually, since you can assign a keyboard equivalent to an ActionScript, though you then have to include a very clear visible statement explaining which key to press (reminiscent of the accesskey problem discussed in Chapter 7, "Text and links").

As you can see, even simple tasks involve multilayered technical barriers. The players themselves have technical limitations; interfaces to actuate access features are inconsistent; and even if you solve those problems for one disabled group you might not have solved them for another.

### **Closed or open?**

Why are captions and descriptions *closed* on television and home video? Why, in other words, do you have to explicitly turn them on? Why isn't accessibility *open* or available as an intrinsic feature that cannot be deactivated?

The philosophy, backed up by essentially no research, holds that nondisabled people cannot stand captioning (especially) or audio description and

Until players are all at least as capable as existing TV and video, fully support certain standards, and are completely and effortlessly compatible and manipulable with screen readers and other adaptive technology, open accessibility is the way to go.

hate having either form of accessibility rammed down their throats. Accordingly, we have invented heavily-compromised and expensive technical systems to hide captions and descriptions from the vulnerable, delicate eyes and ears of nondisabled people.

On TV and in video, there's exactly one signal to play with. (Yes, if you have a satellite dish, you may indeed be able to watch multiple feeds of MTV or CBC or the BBC, but that is the rare exception, not the rule.) When you watch a TV station, you watch exactly the same feed every other viewer sees. That single feed must accommodate sensitive nondisabled people and also the deaf and the blind. Bandwidth is a scarce resource in TV and video, you might say. Main picture and sound and all access features must be bundled together.

But that is not the case online. We can run as many separate feeds as we like, each of them as different or as similar as we like. Outside the access field, we find a couple of examples. If you want to download, say, a video player, you may be asked which server is closest to you (North America, Europe, Australia, Africa); those are multiple feeds as we understand them here. Or you may use Akamai or a similar technology to distribute the load of serving multiple copies of the same file across many different servers in different locations.

In accessibility, we can use the capacity for multiple feeds to get around the shortcomings of closed accessibility. Instead of replicating the television model by providing, say, a single QuickTime file with every access feature hidden inside, why not give us a number of different but related QuickTime files?

- Plain – no access features at all.
- With open captions.
- With open descriptions.
- With both.

Sure, you're encoding and saving four files rather than one, but disc space is cheap. Sure, you're transmitting four files rather than one, but we have a range of ways to distribute server load at our disposal. In any event, individual visitors will tend to choose only one of the available file options; you're rarely serving more than one file per visitor. You're going to the trouble of creating captioning and descriptions anyway, so there is no real added effort there. And it is often technically simpler to add open captions and descriptions than closed ones.

We are faced with a thicket of incompatibilities in making "closed accessibility" work online. Recall that various players have various abilities. But it's actually worse than you think: Some players can't even do basic things right, like aligning text. The choice of fonts alone is a severe stumbling block: Untrained people generally have poor taste in typography; they do not

understand the special demands of reading onscreen text, particularly onscreen text overlaid onto video; their available fonts are limited; and finally, untrained people have an annoying habit of considering the font known as Arial appropriate for any purpose. (It is not appropriate for any purpose.) Worse, with rare exceptions the ultimate viewer must own the actual font in order to see it.

Until players are all at least as capable as existing TV and video, fully support certain standards, and are completely and effortlessly compatible and manipulable with screen readers and other adaptive technology, *open accessibility is the way to go*.

There's a notable disadvantage to open accessibility: Caption text is rendered as bitmaps or pictures and cannot be scanned, searched for, captured, downloaded, or printed the way a separate text track can be. It's a problem, but not such a pressing problem that it invalidates the whole approach. (Incidentally, a separate transcript, easily derived from the task of creating the open captions, is helpful to provide.) You needn't worry about a similar problem with audio description; as I explain later (see "Text descriptions"), the only kind of descriptions you ever want to deal with involve recordings of actual human speech, which itself cannot be scanned, searched for, captured, downloaded, or printed as text can be, and that is just the way things are.

### **The knowledge gap**

Why am I not teaching you how to do captioning and description right here?

That was originally the plan, I must confess. I then had quite a lot of time to consider what would be involved. And that turns out to be an entirely separate training course that is longer, more involved, and more taxing than this entire book. It is difficult to explain captioning and audio description without sitting down in person with tapes rolling. It is impossible to teach either discipline without extensive actual video to use as practice material. As with typography, it takes a very long time to develop an eye for it.

A written manual that attempts to teach media access, without actual media at hand to work with, could never do much good. The inclusion of captioning guidelines on this book's CD-ROM is meant purely as a gesture; reading them, you will come to understand the magnitude of the job.

In effect, in producing accessible media your room for creativity is severely hindered by the source material. You *cannot* do whatever you want. Cinema is a blank canvas; accessibility is a paint-by-number set.

This isn't merely my opinion. Captioning has been widespread since the 1980s (and dates back to the 1940s); audio description of film and television is nearly fifteen years old. While there is room for improvement in the existing practices, the fact remains that *there are existing practices*. You simply cannot make it up as you go along.

You need to ask yourself a few questions, then:

- Do I have these kinds of skills?
- Are these skills even remotely relevant to the rest of my work in development for the Web?
- Is this even the sort of thing I want to learn well enough to do a good job?
- Where would I go to learn these skills in the first place?

The answers to the first three questions rest with you. The answer to the last question is “Nowhere,” and it pretty much queers the whole enterprise.

#### LEARN BY WATCHING

I would suggest that everyone hoping to perform, oversee, manage, delegate, or simply understand any form of captioning or audio description spend a couple of weeks doing nothing but watching captioning and audio description.

Readers in the U.S. and Canada have the most options. First, turn on captioning on your television set. Virtually all televisions built since mid-1993 (and ancillary TV-receiving devices like computer video cards) have included caption-decoder chips as standard equipment under U.S. law. (Canada receives the same TV sets and equipment, by and large, but there’s no legal requirement.)

It is still possible to buy an external decoder for an old TV set, but you get better results (including colour captions) by using a new TV. It could be time to upgrade. (If you buy a television for the office for the purpose of learning captioning, it could be a deductible business expense.) Some sources for external decoders are United TTY Service ([UnitedTTY.com](http://UnitedTTY.com)), Harris Communications ([HarrisComm.com](http://HarrisComm.com)), and Hear-More ([HearMore.com](http://HearMore.com)).

You can watch audio descriptions on a few U.S. television networks. At time of writing, the complete list appears to be ABC, CBS, Fox, Lifetime, NBC, Nickelodeon, PBS, TBS, TNT, USA Network, and Turner Classic Movies. Schedules are extremely hard to come by; the searchable listings at [TV.Yahoo.com](http://TV.Yahoo.com) appear to be the only listing of U.S. described programs (look for the malapropist acronym “DVS”). Global and CTV regularly air described programming in Canada, as do a few other networks and broadcasters on occasion.

You can and should buy, rent, or borrow home videos with audio description. I can provide a firm recommendation for DVS Home Video, found at [DVS.WGBH.org](http://DVS.WGBH.org). At press time, nine DVDs in Region 1 (U.S., Canada, U.S. territories) and about three dozen in Region 2 (Japan, Europe, South Africa, Middle East) offer descriptions. (Eight of the Region 2 discs are in German.) That’s not really a lot. I maintain a list online: [joelclark.org/dvd/](http://joelclark.org/dvd/).

For readers in the U.K. and Ireland, Western Europe, Australia, and other areas served by the World System Teletext technology, captioned TV broadcasts

are widely available. You generally need a teletext television set (typical midrange and high-end models offer that feature), though you can find a very few external decoders and VCRs able to decode teletext captions. There are a few sources of open-captioned home videos, but not many – the Australian Caption Centre is one ([auscap.com.au](http://auscap.com.au)). It's also possible to watch closed-captioned home videos that use the Line 22 system, for which you need a separate decoder (the so-called Videocaption Reader) or a VCR or TV with that separate decoder chip. (Line 22 is not the same as World System Teletext. Yes, you need two decoders to watch all forms of closed-captioned programming.) Sarabec.com sells Videocaption Readers.

Audio description is present but rare on analogue and digital television in the U.K. and Germany. The Royal National Institute for the Blind in the U.K. sells a line of described home videos ([RNIB.org.uk](http://RNIB.org.uk)).

Japan uses the same television system as the U.S. and Canada, and Line 21 captions are in reasonably wide use there, though you generally need a separate decoder.

Readers pretty much everywhere in the world can watch subtitled DVDs, though the conventions used in subtitling aren't even remotely comparable to those used in captioning. That is true even for the technique euphemistically known as “subtitles for the deaf and hard-of-hearing,” about which I will refrain from launching into a diatribe.

This is not a short-term commitment. You can't just watch one or two shows and then turn the damn things off. It takes two solid weeks of watching captioning and audio description before it becomes comfortable. Rather like breaking in a new pair of shoes, the task of assimilating main audio and video and captions and/or descriptions all at once is foreign and unsettling at first. (For people over 40, anyway. Kids today are much more adept at handling multiple simultaneous streams of information.)

You will undoubtedly notice a wide divergence of captioning styles (if not in audio description). Who's doing it right? To answer that question would require another book, plus a full-on training course, and I have to take things one step at a time. Consider your task one of learning the range of acceptable practices. Take note of who captioned whichever programs you like, and keep watching for that firm's work; if you're going to emulate anybody, emulate only one style, not a mishmash of styles from this captioner and that.

### **Low-budget access**

After gaining all this knowledge, you may wish to try your hand. You can do captioning and audio description yourself, but do not underestimate the difficulties involved.

There's more than one way to publish on the Web, and sites of every description, with budgets of zero on up, can be made accessible to varying

*It takes two solid weeks of watching captioning and audio description before it becomes comfortable.*

degrees. Think of the Priority 1 through 3 guidelines from the Web Accessibility Initiative, or this book’s Basic, Intermediate, and Advanced accessibility advice.

Out in meatspace, for example, there’s a whole underground movement of home subtitling of Japanese *anime* videos. Fans go so far as to write their own software to do it. No one particularly cares how good or bad the subtitles are. They’re better than a Japanese soundtrack you cannot understand.

If your Website has just a few videoclips, which themselves aren’t exactly of Stanley Kubrick quality, does it particularly matter that amateurs are doing the captioning and description?

I would certainly endorse this kind of homegrown captioning and description, with reservations. The reservations have little to do with the probably low quality of the captioning (for small-time applications, that is not altogether important) and more to do with the enormous effort required. And audio description is another matter entirely.

#### **CAPTIONING AND TRANSCRIPTION**

Captioning starts with text. Transcribing the video accurately is the first problem. You are unlikely to have experience in transcription. Neither is anyone else nearby, for that matter. Among neophytes, there is a tendency to be at once too literal in transcription (“Um, yes. Um, I think, um –”) and too free (“No!!! [laughs hysterically!!] I DIDN’T SAY THAT!!!”).

The best way to learn how to transcribe is to watch captioned television. Failing that, reading existing transcripts and comparing them to the original audio is a good way to learn. (And where are you going to find those?) The problem with this advice is the fact that a great many rules of written English are literally invisible to most of us: At least at the level of grammar and punctuation, well-written English doesn’t call attention to itself, and the mechanisms it uses are hidden. It actually takes a lot of work to make a transcript, or anything else, read effortlessly.

There are a couple of shortcuts available in transcription. If your videoclip is based on a script, start with that, taking care to note any deviations from the script.

If you’re transcribing from scratch, there are a few good practices to follow.

Every transcription file should state what is being transcribed, ideally with links back to the source page, the homepage of the entire Website in question, and the original audiovisual file. Give an E-mail address for questions about the transcript; you may wish to set up an alias for this purpose (*transcripts@yourcompany.com*), and you may also wish to credit transcribers by name. If you hired an outside transcribing firm, definitely credit them. Provide a copyright declaration.

Indicate a change of speaker. Cascading stylesheets come in handy here (consistent with the advice of T.V. Raman, as found in Chapter 11, “Stylesheets”). Give each speaker his or her own paragraph style:

- `<p class="george">`
- `<p class="announcer">`
- `<p class="man1">` (for an unnamed first male speaker)
- `<p class="man2">` (for a second male speaker)

In this way, you assign styles to each paragraph based on who’s speaking. Inside a paragraph, however, you should mark up the name of the speaker by using a single `<span></span>` class. All actual character or speaker names inside a document share the same class; there is no need to differentiate them.

```
<span class="s"></span> (s for speaker; keeping things short is helpful
but not required)
```

A transcribed paragraph might begin like this:

```
<p class="george"><span class="s">George</span>: Transcribed words</p>
```

This method marks the paragraph as being the words of George and the word *George* as being the name of a speaker. Later on, it becomes possible to search for and extract only George’s speech, or to eliminate the names of all speakers, or to assemble a list of all the people who spoke. Isn’t that a bit more useful than a plain-text transcript where none of these transformations are possible? (The added effort is not great, as we’ll see shortly.)

You can define typographic attributes for the class known as `s` to give it, say, a bold sans serif font. You could also cause these speaker names to appear in capital letters:

```
span.s { font-family: Verdana, Geneva, sans-serif; font-weight: bold; text-
transform: uppercase }
```

For browsers and devices that do not understand stylesheets, it is not wrong to nest a `<strong></strong>` element inside `<span></span>`, taking care to redefine the `s` class to avoid redundancy. A transcribed paragraph could begin like so:

```
<p class="george"><span class="s"><strong>George</strong></span>:
Transcribed words</p>
```

using a class like this:

```
span.s { font-family: Verdana, Geneva, sans-serif; text-transform: uppercase }
```

What about the paragraph classes for speaker identification? They don’t have to look any different from other paragraphs per se. There is no requirement to

actually define these classes in your stylesheet. Heretical, isn't it? "Why else do we declare styles?" you ask. Well, in this case the goal is future manipulability rather than present-day appearance. A screen-reader user could remap the interpretation of such paragraph styles to speak in a different voice or volume. Or, later on, you could do a search of the file to extract everything that George said.

If it seems like a lot of work to type something like `<p class="george"><span class="s">George</span>` in front of every paragraph, you can take the easy way out. Just type the speaker's name and a colon at the opening of each paragraph in a simple text editor or word processor (yes, even Microsoft Word). As long as there's a consistent structure to your paragraphs – for example, two blank lines, then the character name, then a colon and a space – you can do a search-and-replace later on. You can even close the preceding paragraph (using `</p>`). Example:

- Search for: `[Return][Return]George:[space]`
- Replace with: `</p>[Return][Return]<p class="george"><span class="s"><strong>George</strong></span>:[space]`

Here is a final detail that is strictly optional. It helps to start each sentence on a new line. Web browsers will ignore such a linebreak (unless it's inside an unwise tag like `<pre></pre>`, which I rather doubt you will use), but if you encode sentences separately, even through the innocuous method of separating them with a carriage return, it will be easier to transform the transcript into chunked-up or scrolling captions later. If you were truly keen on this, you could use a linebreak element, `<br class="sentence" />`, which you could later search for and replace. This is not quite the best way to do it, since you are only marking a sentence boundary and not the beginning and end of each sentence, but it could be useful. If you'll permit me to pursue this detail even more exhaustively, a usually-invisible HTML character entity like the zero-width non-joiner – `&zwnj;` or the more reliable `&#8204;` – can act as a sentence boundary.

### **Non-speech information**

It is necessary to transcribe all relevant non-speech information. What does "relevant" mean? It's similar to evaluating which parts of a page must be modified for colourblind people: If you missed a notation of the sound effect, would you be confused, fail to understand a later event or statement, or make a mistake?

For example, if the video shows a person walking up to a podium and the floorboards creak en route, there is no particular reason to note that sound in writing. If, however, a floorboard *breaks* and the person stumbles, the sound is suddenly more important. If the speaker bumps the microphone out of frame

(invisibly, in other words) and says “Oops! Sorry!” then it is necessary to explain why the speaker is apologizing. (A notation like [Bumps the microphone] will do.)

Some sound effects are obviously always important (or nearly so): A ringing phone, a knock on a door, a crying baby.

Because sound effects can occur as events unto themselves or right in the middle of dialogue, you need to use markup for both cases. A paragraph- and a class-level style declaration will do. Some examples:

- `<p class="nsi">[Phone rings]</p>`
- `<p class="george"><span class="s"><strong>George</strong></span>: Good morning and welcome to the first day of our – <span class="nsi">[Bumps the microphone]</span> Oops! Sorry! Good way to get the ball rolling. Welcome to the first day of our AGM. Transcription continues</p>`

Here, `nsi` means “non-speech information.”

You don’t necessarily have to assign typographic attributes to these styles. Why? Because you absolutely must use some kind of delimiter – parentheses () or brackets [], but never angle brackets <> or braces {}, which are not used in English writing – to surround the text that conveys the non-speech information. This redundancy obviates the absolute necessity to define styles. (Screen readers, Braille displays, and other adaptive technology can recognize such punctuation.)

If you want to define a style, though, it’s perfectly fine. Italics are nice.

```
.nsi { font-style: italic }
```

You don’t have to transcribe words in a language you do not understand. Annotate a foreign-language segment inside a larger file:

- [Speaking Japanese]
- [Asks question in Japanese]
- [Responds in Japanese]

If an interpreter is used, state that the resulting text is translated:

- George (translated):
- George (through interpreter):
- Interpreter:

(Despite common usage by American newscasters, who seem to think “translator” sounds grander or more objective than “interpreter,” keep in mind that translators work with the written word while interpreters work with spoken and/or signed language.)

If the entire segment contains nothing but foreign-language dialogue, you can either send it out to a translation house for a proper transcript or

The way to make a video accessible is to work on it, not to produce a separate analogue. Separate hasn’t been equal for rather a long time, has it?

provide an excuse on your actual Website along the lines of “This audio file consists exclusively of dialogue in Japanese which we cannot transcribe.” Don’t pretend there isn’t any dialogue; tell us what’s happening. Being up-front and honest about your limitations is the way to go.

#### **Why bother with transcription?**

The goal here is to provide a captioned videoclip. But there are other forms of accessibility when captioning is impossible.

Under battle conditions and as an absolute last resort, it is permissible to provide a separate, free-standing text transcript of a videoclip. The practice is to be discouraged except where utterly unavoidable. The way to make a video accessible is to work on it, not to produce a separate analogue. Separate hasn’t been equal for rather a long time, has it?

If, however, you just don’t have the time, money, or expertise to produce even homegrown captions, you are not off the hook. You do have to provide a transcript, which ideally should be available as soon as the original clip is available but can be delayed a reasonable period while you put it together.

Since online video is usually of short duration, I doubt it would take your company more than a day or two to produce even a rudimentary text-only transcript. If you’re providing hours of video online, your budgets are already pretty high, and presumably you could at least send an audiotape of the video feed out of house for transcription. Or you could transcribe it in-house in chunks – two hours per day, for example.

Your obligation is to come up with some method of making your video accessible to deaf visitors. A number of options are at your disposal, not all of them technically onerous or expensive.

Even if you *do* provide a captioned clip, by the way, it is a good idea to give people a separate transcript, too. It can be scanned, searched for, captured, downloaded, or printed. Your captioners or your captioning software can provide a dump of plain text with a minimum of fuss; it’s preferable to convert that plain text into proper HTML, but you could get by without it.

#### **Linking**

Place a link to the transcript near the link to the source video file. It may be helpful to use a standard filename convention for transcripts, like using the video filename prefixed or followed by trn:

- announce2-trn.html
- trn-announce2.html

### AUDIO DESCRIPTION

There's an asymmetry involved in making video accessible to the blind or the deaf. Blind people can follow a videoclip with no picture much more easily than deaf people can follow it without sound. Nondisabled people can run their own experiment: Watch TV for a day with the volume all the way down (and no captions). Then watch TV the next day with your back turned (and no descriptions).

The most pressing need, then, is captioning, not audio description. If you have to choose one over the other (as will often be the case in small business), choose captioning. However, if you have more time than money, I expect you to do captioning first and pick away at the task of audio description gradually. You may be able to produce a captioned clip (or at least a separate transcript) for publication at the same time the uncaptioned clip goes up on your site, with a described version published two weeks later. What you may not do is ignore description completely.

That kind of staggered accessibility schedule is not completely kosher. We are violating the principle of equal access. Blind people aren't less important than deaf people, nor are they more important, nor is either group more or less important than others. But the intrinsic difference in understanding an audio-visual medium when you have access to only the audio or the video means we have to set priorities.

What about posting just an audio track, complete with descriptions, in lieu of a videoclip whose audio contains the descriptions? That would not be equivalent access; it's not as though you are presenting deaf people with visuals only and no sound.

What you can do, though, is provide both of the following:

- Your actual videoclip with descriptions.
- A separate file (in, for example, MP3 format) of main audio plus descriptions.

Some visitors may have so little vision that they don't need the images at all. They can download the much smaller MP3 file. But the option of video plus descriptions is always available.

### How to do it

Frankly, I always hate it when authors take the easy way out, explaining away their refusal to document a specific topic by calling it "beyond the scope of this book."

For better or worse, that is genuinely true in this case. Teaching you the mechanics of crafting a comprehensible text transcript is one thing. Training you to provide full-on audio description is quite another – something else indisputably beyond the scope of this book.

I acknowledge I am committing a sin, however venial, by telling you to add audio descriptions but not telling you how to go about it. I am also at risk of accusations of hypocrisy here given that I have spent a couple of decades arguing that lousy accessibility is worse than no accessibility at all. Through this book's advice or lack thereof, it is virtually certain that the captioning and description you create will not be up to professional standards. But for small-budget productions and for online video that is of low quality to begin with, professional standards are beside the point and some kind of accessibility is better than none at all.

### **Text descriptions**

Another option, which I discourage altogether, involves writing text descriptions of videoclips. Since we're creating descriptions for blind people, and blind people online have adaptive technology like Braille displays, screen magnifiers, and screen readers, if we provide understandable text then we've solved the access problem, right? No need to bother with voice recordings, right?

Not really. It's theoretically possible to synchronize such text with video: It's called captioning, or maybe subtitling, depending on the application. But try to imagine how this would work. You have a videoclip running in your player (with main audio and video), and somehow your speech synthesizer is supposed to keep tabs on a hidden text track and read it out loud at just the right moment. The speech must finish at just the right moment, too.

Screen-reader users already have to sit there listening to computer speech all day. (It's not uncommon for blind computer users to prefer to do things on the phone or face-to-face as much as possible just to relieve the tedium of that droning voice yammering at them all day.) Now we want to add dreary computer speech, through a technical apparatus that isn't as reliable as a simple human recording, to a videoclip that already contains human voices and other high-quality sound.

Why muddy the waters? The correct way to provide descriptions of a videoclip is to use human narrators.

In any event, the technical infrastructure I have just described does not exist. Screen readers have no way to read bits of text aloud at just the right moment – not when they're somehow embedded (likely through nonstandard means) inside a videoclip. We've got enough incompatibilities to deal with already. And if your player offers exactly one text track, who gets to use it – blind people or deaf people? If there are two or more tracks, how does a person using adaptive technology find out they are available and select them?

“Well, can't we just write up a text file full of descriptions?” you now ask. (The Web Accessibility Initiative actually recommends doing so.) How will that work, exactly? How do you explain which descriptions relate to which sections of the video? The entire concept is oxymoronic and ridiculous on its face. While separate transcripts suffice as a form of accessibility for deaf

viewers if there is no other choice, they are a last resort in that setting. There is never a case where a separate text description file suffices as a form of access for blind viewers. Don't even think of it.

The only way to describe video is with an actual human voice. Accept no substitutes.

### Software

Transcription is complicated enough that it's possible to earn a community-college degree in the discipline, and thus transcription actually is beyond our scope. But in this section, we're concentrating on a sort of amateur or home-grown transcription that doesn't have to be Pulitzer-quality. You may have to get the hang of transcription on your own, or simply choose not to worry about quality if you're only transcribing a handful of clips.

OK. Once your transcript is completed, what do you do with it?

One option is MAGpie, the so-called Media Access Generator, a Windows and Mac OS X software application by the National Center for Accessible Media at the WGBH Educational Foundation in Boston. (This book's CD-ROM has a link to the current version of the software.) MAGpie lets you add captions (and, with difficulty, audio descriptions) to RealVideo, QuickTime, and Windows Media files. (Not Flash. Not yet, anyway.)

You can use the transcript you previously created as a source file in MAGpie. You'll love this part: After I just finished telling you that you should prepare your transcript in HTML, not only will you need a plain-text version for MAGpie, you'll have to chunk up the transcript into sentences. For this and all other MAGpie tasks, you'll need to follow the directions in the MAGpie documentation. MAGpie can add audio descriptions to a videoclip. The version available at time of writing could add descriptions that you have *already recorded*; a later version, which had not been released at press time, lets you actually record your own descriptions as well as add them to a file.

(What's the easiest way to create a plain-text variant of an HTML file? Do a Save As Text from your browser, or Select All, copy, and paste into another program, like a text editor. Or use a program like BBEdit to remove HTML markup. Or – and here's a little-known feature – upload the file to a server and mail it to yourself using the Print command in Lynx, resulting in a superb and pristine text-only rendering.)

Moreover, this book's CD-ROM contains a demo version of CCaption, an application for Macs and Windows machines that can caption QuickTime and other video files. You can use your plain-text transcript as a source file with CCaption.

Given that two of the Big Three multimedia players and later Internet Explorer versions support SMIL files, and given that SMIL is a W3C recom-

mendation, and given that the inventor of the SAMI file format doesn't even support it properly, it behooves you to use SMIL whenever possible, which pretty much means always.

### **Big-budget access**

What if you are required, as under the U.S. Section 508 regulations (described in Appendix A, "Accessibility and the law"), to provide accessible video? Or if you decide to do it anyway, for reasons that should be familiar by now?

The following advice applies to any prosperous organization, firm, charity, or enterprise. If you have a budget large enough to carry the costs of serving a large quantity of discrete video files, or any number of files to a very large audience (possibly using Akamai or a similar technology to distribute the load), then you have more than enough money to make your video accessible properly. And to do it properly, you hire outside professionals.

I am not providing a hard-and-fast income or earnings or asset or wealth cut-off point below which you are excused from doing things the right way (i.e., the expensive way). That is not how the assessment of undue hardship or other forms of financial appropriateness is ever undertaken. It is always a relative or comparative analysis.

In the present case, though, this is a time for you to be honest with yourself: Deep down, do you know perfectly well that your wealthy organization – perhaps with hundreds of thousands or millions in annual revenues – can afford to do accessibility right? I'm going to leave it up to you to decide if you fit the description.

If you do, then you are well-advised to send out your video for captioning and audio description by recognized professionals in the field. Depending on the length, difficulty, and turnaround time, such captioning and description could cost hundreds or thousands of dollars per item. Yes, that much. It's still cheap in production-budget terms.

Whom should you hire? I have my favourites, but I'm not going to plug them. Some guidelines to follow:

- Never under any circumstances use a postproduction house, dubbing operation, ad agency, or any other business that considers captioning or description Just Another Add-On Service We Provide for the Convenience of Our Clients. Hire only firms that do nothing but captioning or description.
- Don't hire any captioning company with less than five years in the business. (Description is too new for that time restriction.)
- Treat Canadian vendors with great skepticism, since I know firsthand just how poor Canadian captioning and description actually are. (Exception: Real-time English-language captioning of live events, where Canadian captioners are generally good to excellent.)

- Be prepared to send the work out of town, out of state or province, or out of the country. In particular, Canadians may be forced to bite the bullet and hire American vendors, who will bill in painfully expensive American dollars.

Some advanced vendors can make a good-faith but insufficient attempt at producing a file with closed accessibility features. You're much better off asking for open captions and descriptions (in separate files, with both of them added to yet a third file).

#### **WORKING WITH VENDORS**

Here is how things will probably transpire when dealing with outside vendors. In this section, the phrase *online files* refers to a videoclip in an electronic format, or possibly on a CD or a DVD, but in any event, no videotape is involved.

For captioning, you should be able to provide your uncaptioned master in essentially any tape format, or even as an online file. You can ask for a closed-captioned “submaster” tape in return, if you are using a physical tape format, but you will probably simply ask for decoded closed captions, which can be added to any source of video footage. Decoded closed captions don't look very nice, but they are perfectly functional. They are burned into the tape, thus making them open.

Some captioning houses can use a character generator or titling software instead of a caption decoder to produce the captions, which will *sometimes* result in a nicer appearance and better legibility. That is not by any means guaranteed, though; see the discussion of “Screenfonts” below. An advantage of this technique, certainly, is how easy it is to use colours. Yellow captions are often very nice. Sometimes a background colour or “mask” is desirable, at least part of the time (over a white image on the screen, for example). You can talk this over with your captioner and come up with a few ideas.

It is actually quite possible that your vendor of choice can handle *only* videotape and not online formats. You may be faced with digitizing from videotape yourself after you have received the captioned and/or described submasters. If your vendor can accommodate online file formats, they should be able to give you exactly the file format and degree of compression you require. Insist, in fact – this is a case where the captioner or describer can't be a little bit pregnant. Either they offer cradle-to-grave service with online files or they don't.

For audio description, you will receive a new file or tape with audible descriptions. Or you could ask for an audio file by itself without video, which you may use as an adjunct to a video file with descriptions but not as a replacement.

Theoretically, your vendor could provide either descriptions mixed in with main audio (that's what is used most of the time in the field of described TV and video) or a file consisting only of descriptions, which are of course timed to start and stop at just the right moment and silence at other times. It is hard to imagine an actual use for the latter on the Web, and indeed the typical real-world use of such a format is for description in first-run movie theatres – you listen to the main soundtrack of the movie via the cinema's normal speakers and follow the descriptions through a headphone.

If you have opted to offer a video file with open captions and open descriptions, you need to plan for that up front and tell your captioning and description providers what you want to do. It is not particularly difficult to mate the description track to the open-captioned master, but you may want your vendors to take care of that for you rather than messing around in a video-editing program.

(I am not really discussing closed accessibility here. Consistent with my general advice, closed access doesn't work very well online. If you want to give it a whirl, though, at time of writing only the Caption Center and the Descriptive Video Service at WGBH could attempt closed online accessibility, and even then pretty much only in English; see [access.WGBH.org](http://access.WGBH.org).)

Copyright is an issue. The holder of the copyright must authorize the creation of the derivative works known as captions or descriptions. Most of the time the copyright owner will be you, and you simply sign a standard work order authorizing the process. If your videoclip includes songs or music, you will, I assume, have the right to reproduce such songs or music online; you may not have the right to transcribe the lyrics, which is what captioning does. Licenced works contained within your videoclip may require an explicit sublicense permitting them to be captioned or described.

### SCREENFONTS

Typography is important in captioning, though it is possible to provide only general advice in a general-interest book. The cheapest method uses closed-caption fonts decoded and burned into the picture. Such fonts are ugly everywhere on earth. They are, however, tolerable. (I've done tests with real-world video using decoded Line 21 closed captions. They're unsightly but perfectly readable.) Your vendor may be able to use subtitling software that offers better fonts.

We don't have a range of well-tested fonts at our disposal. The Royal National Institute for the Blind in the U.K. commissioned the design of a face named Tiresias engineered to be readable for visually-impaired people. It's fine, except, insanely enough, there is no such thing as Tiresias Italic. A typeface without italics is like a knife without a fork – unusable by itself here in the real world.

AG Schoolbook™ Regular A  
1966-1988: Günther Gerhard Lange

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

Bell Centennial™ Sub Caption  
1978: Matthew Carter

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

Bell Gothic™ Bold  
1938: Chauncey H. Griffith

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

FF DIN™ Regular  
1995: Albert-Jan Pool

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

FF Info Text™ Book  
1996: Erik Spiekermann & Ole Schäfer

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

FF Letter Gothic Text™  
1997: Albert Pinggera

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

Lucida Sans™ Roman  
1985: Charles Bigelow & Kris Holmes

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

FF OCR F™ Regular  
1995: Albert-Jan Pool

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

FF Schulbuch Nord™ Normal  
1991: Just van Rossum

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

Tiresias PCfont™ Regular  
2000: Royal National Institute for the Blind

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

Tiresias Infofont™ Regular  
2000: Royal National Institute for the Blind

**LILLIPUTIAN COGS**  
**11 Illicitly coerce gypsy**  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz1234567890

The choice of fonts for captioning is not an obvious one. Any typeface that looks classy and elegant in the pages of a book will end up spindly and mushed on low-resolution monitors. Nor are conventional (overused) sans serif faces like Helvetica suitable because too many characters are confusable. (Do not use Arial for any purpose.)

The faces illustrated here are at least marginally well-suited for captioning, if only for short video segments. Yes, you have to go out and buy them; fonts are not all free. On the plus side, Tiresias is included at no charge on this book's CD-ROM. Though it lacks an italic, it's certainly a bargain.

The good news is that italics and other variations are under development. In the meantime, the RNIB has licenced the inclusion of two Tiresias variants, Screenfont and PCfont (the latter for computer displays), on the CD-ROM accompanying this book. Yes, you get two fonts for free.

In any event, custom-engineered screenfonts not specifically intended for titling are also rare. Georgia and Verdana were designed for Microsoft by the undisputed greatest living practitioner of typeface design, Matthew Carter. They're free to download at [microsoft.com/typography/](http://microsoft.com/typography/); if you've installed anything by Microsoft in the last five years, you own one or the other, and new computers ship with either or both. They'll do fine for open captioning. (Bold, italic, and bold-italic variants are available, and, at least on Windows, the character set is gigantic.) Tahoma is a slightly narrower variant of Verdana that may be better for titling. Trebuchet, another Microsoft screenfont (designed by Vincent Connare), also works nicely.

Don't use Helvetica. Typographic neophytes think Helvetica is "legible." Try running a few tests with confusable characters like `l11i;|`, `00Q`, `aeso`, `S568`, or quotation marks. Related grotesk typefaces like Univers suffer similarly. (One more time: Don't use Arial. It's a bastardized variant of Helvetica, it's ugly, it bespeaks unsophistication, and it sticks you with all the same confusable characters as other grotesks.) Sans serif faces like Franklin Gothic, News Gothic, Officina Sans, Info Text, Thesis, Syntax, and Cæcilia do a better job of solving the problem of confusable characters.

Serif fonts work, but require care. So-called *slab serifs* or *egyptians* (with serifs cut square and perpendicular), like Stymie, Rockwell, Lubalin Graph, Boton, and Serifa, are surprisingly effective. So are a couple of novelty fonts entirely dismissed by graphic designers, like Souvenir and Benguiat. Don't even think of using traditional book typefaces like Times, Bookman, or Century. In fact, any typeface that would look classy and elegant in small sizes in a very serious and expensive book must be avoided like the plague in titling. Resolution is poor; you're looking at the type from a greater distance; the words move; foreground and background colours mix and move; and displays are luminous while print is reflective.

Colour choices? White characters with a slight black edging; white characters on a black background, as is the norm in Line 21 captioning; and yellow characters with or without edging all work well. It is possible to colour-code captions for different speakers; leave that up to the pros, and only if they have years of experience doing it. In all cases, open up the *tracking* (or *letterspacing*), the space between letters generally, as distinct from *kerning*, the space between pairs of letters. Glowing letters tend to bleed into each other; letters that are nicely spaced for print are too close together in video.

DVD subtiting houses, software, and techniques are generally usable in producing open captioning.

### LANGUAGE VARIANTS

Canadians may have to live with American spellings in their captioning; since Canadian captioning of prerecorded programming is substandard, the use of American captioners may be wise. Putting up with American accents in audio description is less onerous; American newsreader accents are similar to Canadian ones, and those are the sort of voices heard in audio description. I assume that the use of British audio-description narrators for American English materials is somewhat annoying, as the converse would be. (The British already describe U.S. programming and vice-versa; perhaps I am overstating the case.)

If you're American, could you tolerate Canadian, British, or Australian spellings? I know the answer already: You didn't know there was such a thing as Canadian spelling, and now that you do know, nothing but American orthography will do. Fortunately for you, the U.S. offers a vast selection of professional and qualified captioners, so your linguistic gene pool need never be tainted, at least in accessible video.

Can you handle British or Australian spellings or accents if you aren't British or Australian? Decide for yourself.

## Flash accessibility

Something quasi-miraculous came to pass while I was writing this book: Macromedia Flash went from completely inaccessible to quite accessible overnight.

I can and will take partial credit for this event, since I had written an article in December 2000 explaining all that was wrong with Flash from an accessibility perspective. I also chatted on the phone with Macromedia and yentaed its developers to various (other) luminaries in the accessibility biz.

A year and a half later, Flash MX (the development platform) and Flash 6 (the player program) were released, and lo and behold the single biggest deficiency had been remedied: Suddenly Flash "content" was accessible to screen readers. At press time, there remained quite a lot of work to do, but what Macromedia managed to accomplish is nonetheless impressive.

### The screen-reader problem

The Flash MX "authoring environment" and the Flash 6 player solve a few accessibility problems.

Screen-reader compatibility is the first Macromedia access milestone. In ordinary HTML Web sites, screen readers can read text on the page, plus text equivalents like alt, title, and longdesc.

Nearly every blind or visually-impaired person online who uses a screen reader does so on the Windows platform. Apart from the large general installed base of Windows machines, the reason for Windows' dominance

traces back to a Microsoft software infrastructure known as Active Accessibility. MSAA acts as an intermediary between the structure and appearance of Windows software programs (including Windows itself and various browsers) and adaptive technology like screen readers.

Adaptive technology can poll MSAA to find out where the cursor is located, where text, toolbars, and icons are located and what they say and mean, and more.

In order to make a computer accessible, a screen-reader manufacturer merely has to write software compatible with MSAA calls, plus the usual caveats about compensating for individual programs' incompatibilities (including Microsoft's own software). This is not a small task, but it is a much easier task with MSAA than it would be if adaptive-technology makers were forced to reinvent the wheel, which is actually the case on, say, Mac OS, which offers nothing in the way of an accessibility infrastructure. The Gnome Accessibility Project is an ongoing but incomplete effort to write an access infrastructure for Linux.

#### **MX/6: The first hurdle**

Macromedia's "authoring environment," Flash MX, and the new Flash 6 player offer substantial, real, and only slightly incomplete screen-reader support. Among other things, you can assign text equivalents (similar to alt and longdesc in HTML) to buttons, input fields, movies, and a few other items, all of which screen readers can find and read out.

Text per se is automatically "exposed" to screen readers, meaning that many parts of many existing Flash sites are instantly made accessible if you're using Flash 6 and the right adaptive technology. Authors don't have to lift a finger.

#### **HTML equivalence**

HTML is itself not completely up to the task of making Web pages accessible. But the capabilities of HTML are a useful baseline of comparison.

Among the things you can do in HTML that you can't do in Flash:

- Set and change text languages (though you can detect a language setting in Flash using ActionScript)
- Add titles to nearly everything
- Add long descriptions to certain data types (like frames and iframes); Flash does not use equivalent data types, but you can nonetheless make frame- or iframe-like components in Flash
- Mark up acronyms and abbreviations (dubiously useful in HTML, but the capability is there)
- Include multiple levels of alternative content (like nested `<object><object></object></object>`, or the many alternatives in `iframe`)

- Group and annotate form elements (using input, legend, fieldset, and the like)

(Some commentators accuse Macromedia of pulling a Microsoft by developing self-contained proprietary programming realms that undermine the universality of HTML and standardized Web technologies. Macromedia denies it, but if that happens, Flash has to be at least as accessible as HTML. At present, it isn't.)

### **Unfair testing**

The list of what's possible in HTML are in many ways an unfair comparison. Flash isn't HTML, and even some of the HTML-specific access capabilities are not very useful (like acronym and abbr). Colourblindness is poorly understood, and the existing requirements, which call for essentially random or arbitrary colour replacement, not only are absurd in the real world but don't necessarily solve the inaccessibility for people with colour deficiencies.

HTML has been around long enough that its capacities have influenced accessibility requirements. Accessibility experts are, moreover, generally hostile to good visual design. There's a considerable bias within Web accessibility toward "universal" HTML and away from "proprietary" software like Flash and PDF. People are just gonna have to get over that. DVDs, home videotapes, television, and the movies are all accessible in slightly different but functionally comparable ways. HTML, Flash, PDF, and whatever new technology comes along can all be accessible in their own ways.

This issue may clarify the general objections of some Flash critics. Instead of complaining about Flash-only Web sites, shouldn't we be concerned about appropriate alternatives? An HTML site should be available in parallel with a Flash site; the HTML site should be as HTML-like as possible, with the Flash site as Flash-like as possible. You can have similar but not identical content and functions in both sites.

Similarly, Flash-only sites should be as accessible as possible in Flash-specific ways, while HTML-only sites should have HTML-like accessibility.

### **Multimedia**

The most significant deficiency in accessible Flash is the absence of primitives – built-in procedures and capabilities – for captioning (for deaf viewers) and audio description (for blind viewers).

Flash animations – even very discreet, tasteful, highly usable animations, including those that do nothing but move text around onscreen – are a form of cinema. Cinematic works are already made accessible in a variety of media and settings (TV; tape and disc; movie houses; online). There is no such thing

as a perfect system in any of those media; some access provisions are only barely adequate.

Nonetheless, data structures are already in place for captioning and audio description in non-Flash media. There are, in effect, slots into which you can stick caption text or a recording of an audio description. In “traditional” online video of the QuickTime/Real/Windows Media ilk, we suffer from a profusion of data structures, including RealText, QTtext, SMIL, and SAMI.

It is not particularly easy to add captions and descriptions to traditional online video, which in many ways is significantly worse than very old media like TV. But it is at least possible, using, for example, the WGBH Educational Foundation’s MAGpie software, a link to which is included on this book’s CD-ROM. You can hack your way through the existing text primitives in Flash to create a captioned animation; it is merely difficult and clumsy. It is also theoretically possible to add a second audio track using the existing Flash sound structures that will function as descriptive narration.

But the reality is that it remains impossible to caption or describe a Flash animation within a Flash authoring program itself. You the viewer cannot simply select a standardized, universal command in the Flash player itself to turn on captions or descriptions.

Macromedia knows all this, in part because I have talked to them at great length to make sure they don’t overlook captions and descriptions and don’t blow it when they try to implement those features. The issue is that the development team for accessibility at Macromedia is small (never more than four people full-time, usually more like 2½ people). The company wisely chose to get screen-reader access working first and worry about everything else later.

I am told that rudimentary captioning support will appear in a dot-level Flash upgrade this year or next. It is a fair supposition that MAGpie will play a large role. Freebie suggestion: Add support for SMIL, which is a full-fledged W3C recommendation.

There remains the general problem, applicable to all audiovisual media, of the lack of training in accessibility, which Macromedia developers will not be able to solve but must eventually be solved anyway by someone, somewhere, somehow. Even if we had a perfect technical infrastructure for audiovisual accessibility, there’s no training on how to do it properly.

### **A good start**

Macromedia has taken serious steps to fix its accessibility deficiencies. There’s still a lot that’s missing, but Macromedia is aware of nearly everything that needs to be done and will presumably fix it all. Still, the Macromedia case is a concrete example of a high-profile company with a kewl product embracing accessibility in an unbegrudging way.

## Audio

Online audio is much easier to make accessible than video.

First, only audio files that are entirely or largely comprised of words, narration, or dialogue need to be made accessible – not, in other words, your entire MP3 music collection. That includes music with lyrics. However, a documentary *about* a musical group must be made accessible. You may not have the rights to reproduce actual song lyrics; in cases like those, use annotations like [Playing “Don’t Fear the Reaper”] or, if you absolutely must, something as vague as [Music plays].

Second, synchronization is not necessary. Some will tell you otherwise. Synchronization is nice. It is quite possible to accomplish using SMIL. Go ahead, knock yourself out.

But all you need to do is provide an accurate transcript. The link to the transcript must be near the link to the source audio file. The techniques used are the same as with transcribing video. Do not unreasonably delay producing such a transcript.

## Live feeds

All the foregoing advice applies to prerecorded audio and video files. But what if you’re Webcasting a live event?

For nearly 20 years, we have enjoyed *live* or *real-time* captioning or *stenocaptioning* of events as they happen. The technique does not involve someone gamely typing away at a standard computer keyboard; there isn’t a typist on earth who can keep up with human conversation that way. Instead, trained court reporters, using stenotype keyboards, listen to the audio and enter what they hear in phonetic shorthand. Software translates the keystrokes into visible words by looking up the phonetic shorthand in a dictionary.

Stenocaptioning can be and is being done in English, French, Spanish, German, and Italian. It’s a tremendously demanding profession, which nearly anyone can learn but few can master. Start with the hardware: Stenotype machines have very few keys (24, in fact), which you must press in combination just to produce one phrase, word, syllable, or phoneme. Then you have to press another combination of keys for the next phrase, word, syllable, or phoneme. (Repetitive-strain injuries were commonplace among court reporters before anyone who used a personal computer ever heard of them.)

According to interviews with experienced pros in the field, it is easy to learn *stenotypy* (an actual term in use) well enough to keep up with, say, a 120-words-per-minute conversation. But, in the real world, conversations run at 180 to 220 words a minute (“wam” – again, an actual term in use). Your ability to keep up with such speeds is preordained: You top out at some maximum speed, and only if you have a hereditary predisposition can you

reach 180 to 220 wpm. In other words, you have to go through extensive training just to find out if you can actually do the job.

Furthermore, said training requires you to learn the keystrokes for thousands of syllables and words, and in fact you must devise separate keystrokes for homonyms (like *there/their/they're*) so that those words will be translated into correct spellings by the software. Also, proper names and unusual or foreign-language words require their own keystrokes, which, save for rare cases, you cannot look up; you simply have to know them.

It is quite possible to use real-time captioning online. People have tried to send out captions to browsers using JavaScript, and I suppose it is theoretically possible to transmit real-time captions in a closed format in SMIL or some player-specific format, but why bother? The technical incompatibilities here are even worse than with prerecorded video.

I strongly advise you to hire qualified, experienced real-time captioners to caption your live event. They will interface their software with a standard television captioning decoder to produce open-captioned video, which you can then stream online. You may need to set up two video streams (captioned and uncaptioned) if you wish to spare sensitive hearing people the agony of watching captions.

Every real-time captioner can deliver a plain-text file after the fact – instantly so, in fact. You can post that file as a separate transcript (though I recommend adding proper HTML markup, as described previously).

In North America, very advanced captioners can stenocaption in mixed case, though most of the time you'll see captions in capital letters only. Text editors can do a reasonably intelligent conversion from all-caps to mixed case, or you can just leave the file as-is if you have to.

I should note that a product called eScription from Speche Communications (pronounced “Speech”; [www.speche.com](http://www.speche.com)) can intercept real-time captions emanating from the captioning software and insert them into the closed text-track or caption fields in RealPlayer, Windows Media, or QuickTime formats. Does it work? I don't know; I've never tried it, having merely spoken with staff from Speche. It is an available option.

There is no way to make live audio Webcasts accessible to deaf and hard-of-hearing visitors – not without turning them into live video Webcasts whose video segment consists of a blank screen with visible captions. That is not necessarily a bad way to do things. A workable alternative is to run the live audio feed inaccessibly, but transcribe it later, or, better yet, hire real-time captioners to transcribe the audio as it happens and post the transcript immediately after the event is over.

An extended audio feed, like that of an all-day conference, should be chunked into segments if you choose the latter approach. Post the chunks as you get them (e.g., after each speaker; after morning and afternoon sessions;

or once a day). Don't save up the entire huge set of transcripts until absolutely everything is finished.

You can post text files first and marked-up HTML versions later if necessary. Posting a raw text file (with, say, a .txt filename extension) is not the best idea because browsers handle plain-text files unpredictably. The chief problem is linebreaks; browsers do not always wrap lines of text to fit inside the browser window. Add at least minimal HTML coding, like paragraph tags `<p></p>` around paragraphs. (A quickie search-and-replace will usually suffice to add those.) Don't use the `<pre></pre>` (preformatted) element, since it will cause lines to scroll offscreen in many browsers.

## Bottom-Line Accessibility Advice

### Basic accessibility

- Set up a schedule to provide at least a transcript of the dialogue and meaningful sound effects of any posted online video or audio.
- Use all available accessibility features in Flash.

### Intermediate and Advanced accessibility

- Provide captioning and audio description for online video.