



Windows Media and PowerPoint

How to link to remote media files in PowerPoint documents.

Among the virtues of Microsoft PowerPoint (PPT) are quick editing and almost total control over your desktop. Professional presenters know that these features make PPT the hands-down tool for visually rendering business ideas and complex processes on the PC and Mac. However, in the Web video world, presentation tools with such virtues are still rare—mainly because of browser and player diversity across the Web's wild and wooly user base. I still get depressed thinking about the hours I've spent on HTML and SMIL applications with embedded video clips that didn't look the same in IE and Netscape, or because a target audience at a corporate client only supported Real or Windows Media 6 (instead of version 7 or 8).

If you need to show video people clips that are stored on Web servers, you can build and edit PPT decks around those clips with only a few more steps than you use to create text-only PPTs. If you're already proficient at PowerPoint (as are most managers, marketers, and sales people), you won't need Web developers to set up HTML pages for you (unless you still like this time-consuming job). Of course, the video clips you want to show must be available on the Web—just like in browser-based presentations—and slow Internet connections will torpedo any Web-aware presentation regardless of the tools you're using. But for rapid, ad-hoc deployment of short, linear Web media applications over which you have sole authority, PPT can be a revelation.

Remember, we're talking about *embedding* video clips in PowerPoint documents—not *encoding* PPTs as Windows Media clips. Windows XP has some fine new tools for doing the latter (which I'll discuss shortly), but the point here is *linking* to remote media files that play in

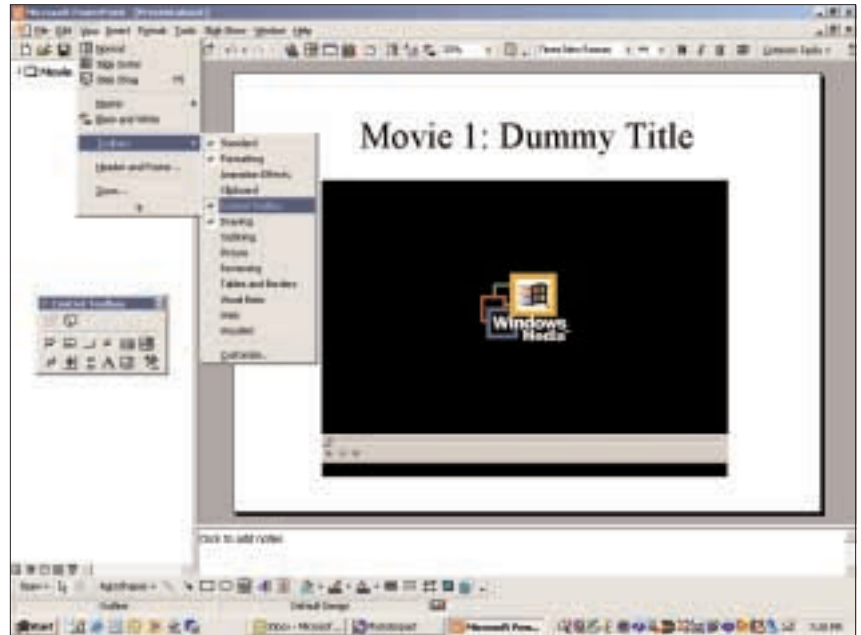


Figure 1—Instantiating an ActiveMovie control on a PPT slide.

desktop (PPT) documents without a lot of SMIL, JavaScript, and HTML overhead. For the record, PPT on the Mac does not yet support the techniques described here.

Embedding video clips in PPT docs

Straight out of the box, PowerPoint lets you insert Movies and Sounds and Video Clip Objects into PPT projects. Unfortunately, such media files are normally expected to reside on the same machine as the PPT executable. If you experiment with hyperlinks and suggestive object properties to get around this, you'll just get frustrated. Trust me. Hyperlinked media files will generally invoke their appropriate players *outside* of your PPT documents, thus defeating your purpose, with the possible exception of inserted audio clips. Most of this unwanted behavior is because the default media player is pre-ActiveX (despite the sophistication of PPT 2000 and 2002). However, I didn't test the techniques in Windows 98 or prior environments.

As it turns out, the key to embedding video clips in PPT slides is the Control Toolbox, not the Insert menu. Here's a quick example of how to embed video clips in slides in PPT 2000:

1. Open a new default document and key in a dummy title.
2. Click the View menu and mouse-over the Toolbars item.
3. Click the Control Toolbox to place the toolbox on your desktop.
4. Click the More Controls item (bottom right) in the toolbox; then click ActiveMovieControl Object.
5. When your cursor changes to a "+", drag your mouse on the PPT slide to create a movie window as shown in **Figure 1**.
6. Right-click on the ActiveMovieControl object to bring up its menu; then select Properties.
7. If you've developed Visual Basic applications, the resulting Properties menu will look familiar (see **Figure 2**). For now, the only property you need to edit is FileName,

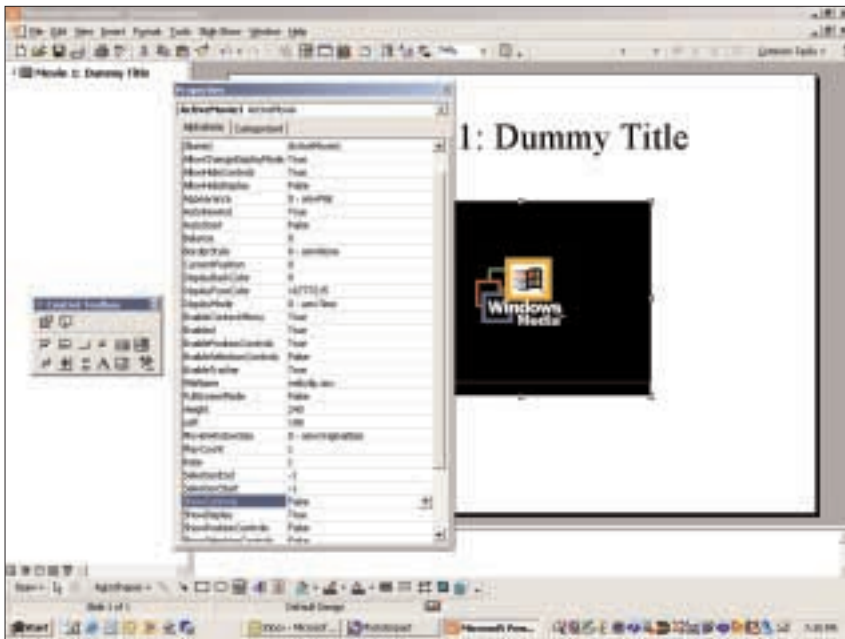


Figure 2—The ActiveMovie control property list.

which you should set to “WEB-CLIP.ASX.” You can also use a hard-coded URL here, such as <http://66.98.31.129/movie1.wmv>.

8. Aside from creating “Webclip.asx” (described next), you can now test the presentation by pressing F5. The embedded movie player should appear in your running PPT slide—not as a separate player outside of the PPT. If you go back to edit mode, you can resize and reposition the object, make it play full-screen, or auto-start without the control buttons (depending on how you want the user to navigate the show at runtime). In other words, have some fun but be careful how you scale your clips if you want them to look good. Native size is always best.

Using ASX files

Now for the ASX file. Any experienced Web video developer knows what these are and understands their value. The short story is that an ASX file is a tiny text document, playable by the Windows Media Player, which contains a reference to an actual media file, either local or remote. I use ASX files to point to new Web movies without needing to re-edit my PPT decks. If you keep the ASX in the same folder as your PPT, it will all just work (remember to include the ASX if you give someone else the PPT). Again,

you can use a movie’s URL in the property panel if you want to skip dealing with ASX files altogether.

The contents of a sample ASX file (such as “Webclip.asx”) are shown in **Figure 3** (below). The referenced movie clip can be a Windows Media file, an MPEG-1 file, or anything else playable by the Windows Media Player. The HTTP tag assumes this clip is available for progressive download from a Web site. For streaming video from a Windows Media server, you can use MMS instead of HTTP.

If you know the URL of a Web video clip, you can load that address into the Windows Media Player and play the movie. If, however, you want to provide context for that movie or create a professional presentation enhanced with video clips, PPT on the Windows desk-

```
<asx version="3">
<entry>
<ref href="http://66.98.31.129/movie1.wmv"/>
</entry>
</asx>
```

Figure 3—Basic ASX syntax for remote Web video clips.

top puts you in the driver’s seat. When you build PPTs with multiple Web clips, be sure to give each ActiveMovie control a unique name in its respective Properties list—not to mention a clear, serialized naming scheme for your multiple ASX files. For best results, only use one

movie per slide. If you send clients a Web-enabled PPT, tell them slide advancement requires a prior mouse click outside the movie window if they have clicked the movie or its start button to make the movie play.

Converting PPTs into Windows Media files

As I previously mentioned, the flipside to embedding video clips in PPT documents is encoding PPTs as Windows Media clips. Although the overall Windows 2000 world (including Windows Media 7 and Office 2000) offers some tools for this process, XP and Office XP take it to a whole new level (the official abbreviation for version 8 of the Media Player is MPXP). The name of the new preferred tool is MS Producer for PowerPoint 2002, which is documented at the address www.microsoft.com/office/powerpoint/default.htm.

So why aren’t more people using PPT to deliver presentations with Web video clips from remote servers? They’re certainly doing it successfully with local media on CD-ROM and DVD. I believe the answer is three-fold. First, HTML and SMIL developers in corporate services departments are more concerned with job security than ever, despite management’s desire just to show the darn movies. Second, mystery still surrounds the process of using ActiveX controls in PowerPoint. And third, browsers are considered the universal delivery vehicle for Web-based digital media.

I hope this column solves some of the mystery suggested in my second point. And although browsers are considered the delivery standard (as noted in my third point), PPT is nevertheless a universal presentation tool in the corporate sector. If you’re struggling with browser-based applications for short, linear product demos, using PPT with remote video links is a solid alternative. ■

Nels Johnson is president of Download Recordings, Inc. (www.downrecs.com), a San Francisco consulting firm engaged in digital media production and delivery. His books include *Web Developer’s Guide to Multimedia and Video* (Coriolis, 1996) and *Web Developer’s Guide to Multicasting* (Coriolis, 1997).