

Integrating QT into Web Sites

Dynamic Media Technologies
Fall 2002

Doug DeGroot

10 Oct 2002



QuickTime

What we *will* and will *not* be looking at today...

We *will be* looking at:

- HTML code for integrating QT files with HTML files.

We *will not be* looking at:

- Audio and Video compression.
- Multi-targeting code.

There are multiple ways to embed QT movies into a web page.

- Using the QuickTime Plug-In
- Using the QuickTime Player application
- So that the movie is invisible (hidden)
- You can support MIME-type hijacking or defeat it.

Some References and Sources of the following Materials:

- **QuickTime for The Web, for Windows and Macintosh,** 2nd Edition, Apple, QuickTime Developer Series

An online copy of Chapter 5 can be found at:

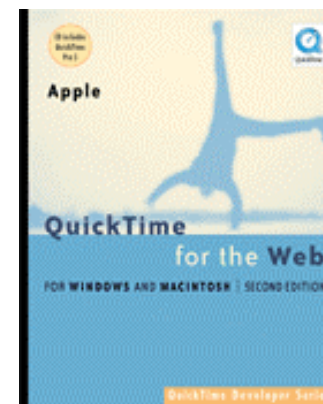
- <http://developer.apple.com/techpubs/quicktime/qtdevdocs/QT4WebPage/sample.htm>

- **QuickTime Technical Support**

- <http://www.info.apple.com/usen/quicktime/>

- **Embedding QuickTime for Web Delivery**

- <http://www.apple.com/quicktime/authoring/embed.html>
- <http://www.apple.com/quicktime/authoring/embed2.html>



A class copy can be found in the library.

QuickTime Support



Special Features of the QT Plug-in

- The QuickTime Plug-in has a lot of special features that you can control from HTML.
- You can tell the QuickTime Plug-in
 - to start playing a movie automatically
 - or to wait until the user clicks the Play button.
 - not to let users save the movie to disk, (provides some basic copy-protection)
 - to hide the movie controller
 - to set the default audio volume
 - to have the plug-in play a series of movies one after the other
 - And more.
- The QuickTime Plug-in has over 25 special features that can be controlled from HTML.
- These features can be combined in hundreds of ways.
- You control the special features by inserting parameters in the `<EMBED>` tag.

The <EMBED> and <OBJECT> Tags

- The <EMBED> tag allows media file types other than those directly supported by the browser to be handled with an external application or plugin.
- In the case of QuickTime, the external application is QuickTime Player and the plugin is the QuickTime Plugin.
- The <OBJECT> tag:
 - With the release of Internet Explorer 5.5 SP2 and later, you must use an <OBJECT> tag in addition to the <EMBED> tag for your web pages to be compatible with both Netscape and Internet Explorer on Mac and Windows systems.
 - *(We will come back to the <OBJECT> tag later in this discussion.)*

The <EMBED> & Tags

- The <EMBED> and tags are quite similar.
 - They both accept SRC, WIDTH and HEIGHT parameters.
 - These parameters are required for the <EMBED> tag.
- The parameters tell the browser which media to display and the width and height of that media.
- SRC = the URL of the media file to display
 - either by an absolute or relative URL.
- HEIGHT = the vertical size in pixels of the SRC file.
- WIDTH = the horizontal size in pixels of the SRC file.
- Example:

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

For space saving purposes, I generally omit the trailing </embed> tags in the rest of these notes. You should not do so in your HTML code, though.

Attributes of the <EMBED> Tag

- [AUTOHREF](#)
- [AUTOPLAY](#)
- [BGCOLOR=HEX](#)
- [BGCOLOR=NAME](#)
- [CACHE](#)
- [CONTROLLER](#)
- [CORRECTION](#)
- [DONTFLATTENWHENSAVING](#)
- [ENABLEJAVASCRIPT](#)
- [ENDTIME](#)
- [FOV](#)
- [GOTO \$n\$](#)
- [HEIGHT](#)
- [HIDDEN](#)
- [HOTSPOT \$n\$](#)
- [HREF](#)
- [KIOSKMODE](#)
- [LOOP](#)
- [MOVEID](#)
- [MOVIEName](#)
- [NODE](#)
- [PAN](#)
- [PLAYEVERYFRAME](#)
- [PLUGINSPage](#)
- [QTNext \$n\$](#)
- [QTSRC](#)
- [QTSRCCHOKESPEED](#)
- [QTSRCDONTUSEBROWSER](#)
- [SCALE](#)
- [SRC](#)
- [STARTTIME](#)
- [TARGET \$n\$ =FRAME_NAME](#)
- [TARGET=MYSELF](#)
- [TARGET=QUICKTIMEPLAYER](#)
- [TARGETCACHE](#)
- [TILT](#)
- [TYPE=MIMEtype](#)
- [URLSUBSTITUTE \$n\$](#)
- [VOLUME](#)
- [WIDTH](#)

Movie Controller Options

- **IMPORTANT:** To add a controller to your movie, you must add 16 pixels to the height of the movie.
- For example, if your movie height is 240 pixels, set HEIGHT="256":

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



Address: http://www.apple.com/trailers/wb/t3/medium.html

go

Google Live Home Page Apple iTools Apple Support Apple Store MS MacTopia MSN Office for Macs InformIT Internet Explorer

Favorites History Search Scrapbook Page Holder

T3 RISE OF THE MACHINES

07.02.2003
www.terminator3.com



THIS FILM IS NOT YET RATED

For additional ratings information please visit: www.mpaa.org | www.filmratings.com | www.parentguide.org

TM AND © 2002 © 2 AND ITS RELATED ENTITIES. ALL RIGHTS RESERVED. PRIVATE USE ONLY. SALE, DUPLICATION OR OTHER TRANSFER OF THIS MATERIAL IS STRICTLY PROHIBITED.

PRIVACY POLICY

Hiding the Movie Controller

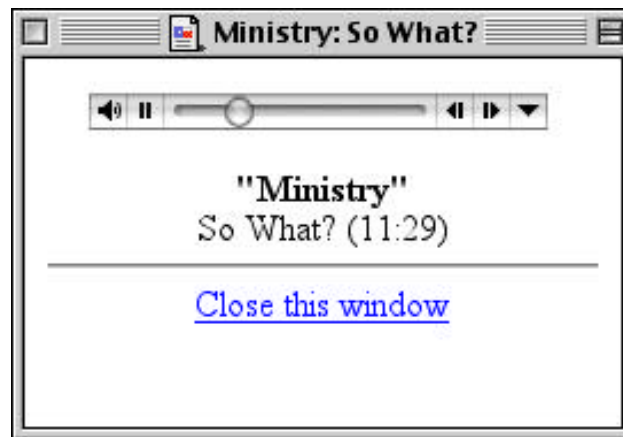
- Of course, you can choose to completely hide the Movie Controller.
- CONTROLLER = value (*true or false*)

```
<embed src="sample.mov" width="200" height="100"  
  controller="false">
```
- WARNING: Never set HEIGHT or WIDTH to less than 2, even if the movie is hidden
(avoids some problems with early browsers).
- For poster movies, you should normally set CONTROLLER = “false”.

Sound-Only Movies

- For sound-only movies (movies with no text or video tracks)
 - use a height of 16 for the controller
 - and any width that looks good for the page layout.
- Example:

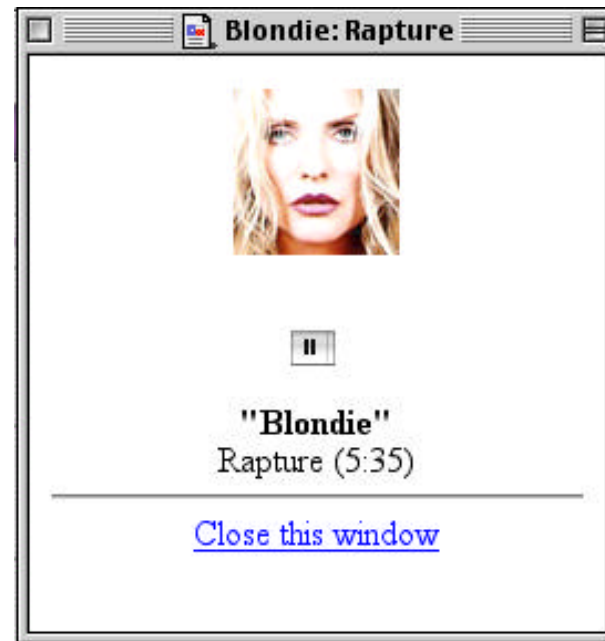
```
<embed src="sample.mov" width="200" height="16">
```



Sound-Only Movies, continued

- If you set both HEIGHT and WIDTH to 16*
 - the controller is minimized to a single Play/Pause button.
- Example:

```
<embed src="sample.mov"  
width="16" height="16">
```



* I sometimes find that I have to use a width of 20 to get this to work on Mac IE 5.0 or higher.

Volume = *value*

- The VOLUME attribute sets the movie's initial audio volume.
- Only integer values between 0 through 100 are allowed for the volume value.
- The default VOLUME value is 100.
- A setting of 0 effectively mutes the audio, whereas a setting of 100 sets the volume to the maximum.
- If you mute the movie, make sure the controller is visible!
- Example:

```
<embed src="sample.mov" width="200" height="240"  
  volume="50">
```

Hidden Movies

- Tag Parameter: HIDDEN
 - There are no values to supply for this attribute.
 - It controls the visibility of the movie.
- If you do not specify HIDDEN, then the movie will be visible.
- If you do specify HIDDEN, the movie is not visible on the page.
- Useful for sound-only movies which are hidden in the background.
- Note: If you add the HIDDEN attribute, you should generally be sure to set AUTOPLAY="TRUE" as well.
- Example:

```
<embed src="sample.mov" height="2" width="2" hidden  
  autoplay="true">
```

Scaling your movies:

- SCALE = value
- Value is either TOFIT, ASPECT, or a number.
- The default value of SCALE is 1.
- If set to TOFIT, the movie is scaled to fit the embedded box as specified by the HEIGHT and WIDTH attributes.
- If this is set to ASPECT, the movie will scale to fit the embedded box while maintaining the aspect ratio.
- Examples:

```
<embed src="sample.mov" width="200" height="240" scale="1.5">
```

```
<embed src="sample.mov" width="400" height="480" scale="tofit">
```

Scaling Uniformly

- QuickTime movies can be scaled uniformly with great results.
- You can create a movie that is 160 x 120 and show it at 320 x 240.
- The file size is 1/4 the size of a true 320 x 240 image, but the displayed image is nearly as good due to QuickTime's image quality.
- Great reductions in file size can be achieved with this technique.

HREF = *url*

- **HREF=*url*** (optional...requires QuickTime 3 or later)
- When set, the HREF attribute provides a link to another page or movie when the movie is clicked on.
- **Note:** If you are using a relative path name for the HREF then it should be relative to the location of the movie specified in the SRC attribute.
- Example:

```
<embed src="sample.mov" width="200" height="240"  
      href="http://www.apple.com/quicktime">
```
- See also: [TARGET](#) and [TARGET="QUICKTIME PLAYER"](#)

Autoplay

- AUTOPLAY = value
- If “true”, then the movie begins playing automatically. Otherwise, a user event is required to start it.
- Example:

```
<embed src="sample.mov" width="200" height="240"  
      autoplay="true">
```

Starting & Ending Times

You can tell the QT Plug-in to play only a certain part of a movie.

```
STARTTIME = "00:00:00.0" and ENDTIME = "00:00:00.0"
```

The time values are specified as

Hours: Minutes: Seconds:Frames (30ths of a second).

Example: Use different buttons to play different portions of a the same movie (viz. DVD interfaces). Also useful with streaming movies.

The frames component can be specified as a fixed point number so that more specific times can be specified.

For example:

3:30:22.5 means 3 minutes, 30 seconds and 22.5 frames.

```
<embed src="sample.mov" width="200" height="240"  
  starttime="00:01:10.0" endtime="00:02:10.0">
```

CACHE = Value

- **CACHE=***value* (optional...requires QuickTime 3 or later)
- If **CACHE=TRUE** is specified, the browser will **CACHE** movies.
- If the movie has been **CACHED** when the user returns to the page, the movie will be played from **CACHE** instead of being downloaded again.
- **CACHE** can either be **TRUE** or **FALSE**, or simply **CACHE**. **CACHE** by itself implies **TRUE**.
- The default behavior is set by the user in the 'QuickTime Plug-in Settings.'
- **Note:** The functionality of this attribute is supported only by Netscape Navigator, version 3.0 or later, and Communicator.
- For example:

```
<embed src="sample.mov" width="200" height="240" cache="true">
```

Click Here, Play There



(see QuickTime for the Web, Ch. 5; in the library)

- You can associate a URL with a QuickTime movie, so the URL loads when someone clicks inside the movie's display area. This lets you do a lot of interesting things.
- The URL can specify a Web page, a JavaScript function, or a second movie. If the URL specifies a Web page, that page can include an embedded movie of its own.
- You can specify a target for the URL, so it loads in another frame or another window. You can also target the URL to replace the current movie in the QuickTime Plug-in, or to load in the QuickTime Player application, launching it if necessary.
- This last is particularly useful. You see, you can't directly tell the browser to launch an application using HTML; that's an Internet no-no. So if you want a movie to launch in QuickTime Player, this is how you do it. You embed a movie for the plug-in, typically a single image with no controller, and tell the plug-in to launch a different movie in QuickTime Player when someone clicks the image.
- **href=" url " target=" FrameName " targetcache="True" | "False"**
- Example:

```
<EMBED SRC="Poster.mov"
HEIGHT="256" WIDTH="320"
CONTROLLER="False"
href="Actual.mov"
TARGET="quicktimeplayer">
```
- Note: an external MP3 player, such as iTunes, can also be set by the user to play MP3s, if wanted.

Playing Multiple Movies

- You can tell the QuickTime Plug-in to play a series of movies, one after the other. When the current movie finishes, the next movie in the list starts. This can be a good way to deliver a long movie; users never need to store more than a small piece of it on their computers at any time.
 - `QTNEXT n = "< Filename > T< Target >"` or `"GOTO n "`
- Example:

```
<EMBED SRC="First.mov" HEIGHT="176" WIDTH="120"  
QTNEXT1="<Second.mov> T<myself>" QTNEXT2="<Third.mov> T<myself>">
```
- You have to append a number between 1 and 255 to each `QTNEXT`. They execute in numerical order.
- Note that you must place angle brackets around the `< Filename >` and `T< Target >` values, and quotes must surround both values jointly:

```
QTNEXT4="< Filename > T< Target >".
```
- Set `< Filename >` to the name of the file you want to play next. You can include a relative path or the full URL.
- The `T< Target >` parameter is optional. It specifies where the movie plays. If it's not specified, the next movie replaces the current browser window or frame, and any subsequent `QTNEXT` statements are lost.
- The special value of `T<myself>` targets the QuickTime Plug-in, and is normally the value you want; the next movie replaces the current movie, and subsequent `QTNEXT` statements are executed in turn.
- The target can also be the name of a browser window or frame. If no window or frame of that name exists, a new browser window with that name is created.
- You can also specify **`T<quicktimeplayer>`** as a target, which opens the movie in QuickTime Player.

Playing Multiple Movies - cont.

- The special targets T<myself> and T<quicktimeplayer> are case-sensitive in some versions of QuickTime. Use all lowercase.
- Each QTNEXT statement has an index number, which is the order in which the movies play. If you specify a QTNEXT1 , then a QTNEXT5 , and then a QTNEXT10 , for example, the three of them execute in numerical order, and their index numbers are 1, 5, and 10.
- Want to keep your sanity? Number the QTNEXT statements in simple numerical order: 1, 2, 3...
- The special index value of 0 is assigned to the original movie named in the SRC parameter (or the QTSRC parameter, if it was specified).
- Why have index numbers? Because instead of a filename, you can specify <GOTO n > , where n is the index of the QTNEXT you want to go to. You normally do this to create a simple loop:
- This example plays `First.mov` , then `Second.mov` , then `Third.mov` , then goes back to `First.mov` (index 0), in an endless loop.

```
<EMBED SRC="First.mov" HEIGHT="176" WIDTH="120"  
  QTNEXT1="<Second.mov> T<myself>"  
  QTNEXT2="<Third.mov> T<myself>"  
  QTNEXT3="<GOTO0>">
```

Playing Multiple Movies - cont.

- You can use values other than 0 to cause QTNEXT statements to play in arbitrary order, or to start looping from some midpoint. For example,
 - ```
<EMBED SRC="Intro.mov" HEIGHT="176" WIDTH="120"
QTNEXT1="<First.mov> T<myself>"
QTNEXT2="<Second.mov> T<myself>"
QTNEXT3="<GOTO1>"
```
- would play the Intro.mov just once, then play First.mov and Second.mov in an endless loop.
- **Note:** A bug in QuickTime 4.1 caused GOTO0 to play the most recent movie instead of the first movie in the sequence, although any subsequent movies in the sequence played correctly. The workaround is to specify the original movie at the beginning and end of the loop, and use GOTO1 instead:
  - ```
<EMBED SRC="First.mov" HEIGHT="176" WIDTH="120"  
QTNEXT1="<Second.mov> T<myself>"  
QTNEXT2="<Third.mov> T<myself>"  
QTNEXT3="<First.mov> T<myself>"  
QTNEXT3="<GOTO1>"
```
- This sequence plays correctly whether the bug is present or not.