

MP3 and MPEG-4 with the Darwin Streaming Server

STREAMING AHEAD

Apple's Darwin Streaming Server is a free version of the commercial Quicktime server. It can handle the seminal MP3 and MPEG-4 formats, and it has a web front-end. **BY OLIVER FROMMEL**

Video streaming is still a fairly uncommon application on Linux. Free servers often have the disadvantage of supporting only their own obscure formats, whereas commercial products are often prohibitively expensive. The Real server, for instance, works really well, but it costs a few thousand dollars. And the Basic server is free, but it supports a maximum of only five clients and is time-bombed to quit after a year's service [1].

Free Server for MP3 and MPEG 4

On a more positive note, Apple has released major components of its Quicktime server under an open source license, the Apple Public Source License [2]. The Darwin Streaming Server (DSS) can handle the popular MP3 audio format, Quicktime, and MPEG-4 video. Before you can download the server software, you first need to register on the Apple website [3].

Then navigate to the DSS page at [4] and click on *Streaming Server 5.0.1* below *Source Code* to download the server. Alternatively, you might like to try the Red Hat 9.0 package, which, incidentally, worked fine with Debian, Suse 9.1, and Fedora Core 3. If you prefer to build from the source code, the Do-it-Yourself Box has a few useful tips.

Although the program will run fine on many distributions, the install script may not. For example, the required user group may be missing – and this means adding the group manually as the root user: `groupadd qtss`. The configuration file that accompanies the source code distribution is also quite useless. You should use the configuration file from the binary package instead, as this gives you a few useful settings.

After setting up the configuration file and the user and group accounts, the administrator can launch the server by typing `/usr/local/sbin/DarwinStreamingServer`. The `-d` option tells the server

to run in the foreground. [Ctrl-C] quits the server. The `/usr/local/sbin/streaming-adminserver.pl` script enables the web administration interface.

Browser-based Setup

If you now see the *Launching Streaming Server...* message, you can try out the Web front-end. To do so, type the `http://server_address:1220/` URL in your browser location tab. If the server and the browser are running on the same machine, you can type `localhost` as the server address; otherwise use the computer name or IP address. If you prefer, you can launch *DarwinStreamingServer* without the web front-end, edit the configuration file manually, and enter `killall -HUP DarwinStreamingServer` to tell the server to parse the configuration file for the changes.

Log on to the web front-end using the credentials you entered during the installation. The Darwin server will prompt you for another password on the next

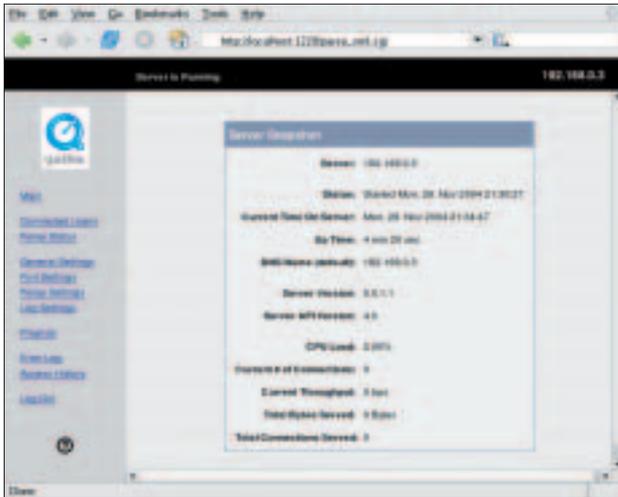


Figure 1: The main window of the web interface with statistics for the active server.

page; this password is used for incoming MP3 streams, which the server is able to distribute.

The server will then ask you if you want SSL encryption for the connection. The Darwin server needs a few additional software packages to enable SSL: the OpenSSL library, the Perl module Net::SSLeay, and above all, a valid SSL certificate. The secure Web front-end listens on port 1240. Of course, if you are running Darwin at home or in a trusted environment, you can do without SSL.

One Directory for All

You then need the path to the video files, which defaults to `/usr/local/movies` – that is, the location where the install script placed the sample files. Finally, Darwin offers to enable streaming via Port 80, which can be a big help if you have to navigate firewalls. You can change all of these options later in the web interface. After completing these

settings, the main window of the web front-end should appear (Figure 1).

As previously mentioned, the sample video files are placed in the master directory, `/usr/local/movies`, during the install. To play these files on Linux, you need a player that supports the MPEG 4 format and the RTSP streaming protocol.

If you intend to use MPlayer [5], you will need to rebuild the

application with the LIVE library [6]. Most players, such as the VideoLAN client *vlc*, [7] need the Ffmpeg [8] library for MPEG 4 files.

Do-it-Yourself MPEG 4

Encoding your own MPEG 4 files is a science in its own right. Having digital video material on your hard disk takes you a step closer to your goal, but you need some more mileage to get to MPEG 4:

- Extract the audio data from the original and store it in a separate file (using *ffmpeg -vn*, for example)
- Encode the audio track using *faac* to give you an AAC formatted file.
- Use *ffmpeg* to convert the video to an MPEG 4-compliant video format.
- Add the audio and video files to an MPEG 4 container using *mp4creator* from the MPEG4IP package [9].

The technical term for this is multiplexing. The *-hint* option of *mp4cre-*

ator adds so-called hints to the file, that is, tags that the server references for fast forwarding and reversing. Incidentally, the MPEG4IP package includes MPEG 4 players with or without a GUI. Listing 1 shows the output of the *mp4info* utility for one of the MPEG 4 sample files.

The first track is the AAC format audio track, followed by the video track, and finally the hints. These tracks are then followed by scene data and object descriptors, which we do not need at this point.

Playlists for MP3

The Darwin server can also stream MP3 files, but it needs a playlist to do so. To create a playlist, click on *Playlists* in the left-hand column of the web front-end. On the top right, next to the larger field, you will note two links: *New MP3 Playlist* for MP3 files and *New Movie Playlist* for movies (Figure 2). If you click on *New MP3 Playlist*, a website like

Do-it-Yourself

To build the Darwin server from the source code, first unpack the zip file by typing *unzip DarwinStreamingSrc5.0.1.1.zip*. Then change to the new directory created by this step, *DarwinStreamingSrc5.0.1.1* and type *./Buildit*. The *./Install* script will move the files to the correct locations: */usr/local/bin* and */usr/local/sbin* for the programs and */etc/streaming* for the configuration files. The install script also prompts you to enter a login name and password for the Darwin streaming server administrator.

Unfortunately, the script is buggy: *Buildit* fails to put the files where *Install* expects them:

```
copying qtpasswd to
/usr/local/bin/qtpasswd
cp: cannot stat `qtpasswd': No
such file or directory
```

Check the installation script output for the correct location, and copy the files to this location manually. If you can't locate the files, run the *find* tool:

```
# find -name "qtpasswd"
./qtpasswd.tproj/qtpasswd
```

Copy this file to where install expects it to be: *cp ./qtpasswd.tproj/qtpasswd /usr/local/bin*. Repeat this step for all the required files.

Listing 1: Output from mp4info Utility

```
01 $ mp4info sample_100kbit.mp4
02 mp4info version 1.1
03 sample_100kbit.mp4:
04 Track Type Info
05 1 audio MPEG-4 AAC LC, 70.031 secs, 28 kbps, 22050 Hz
06 2 video MPEG-4 Simple @ L3, 70.000 secs, 63 kbps, 192x240 @
15.00 fps
07 3 hint Payload MP4V-ES for track 2
08 4 hint Payload mpeg4- generic for track 1
09 5 scene BIFS
10 6 od Object Descriptors
```

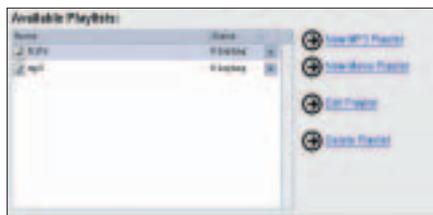


Figure 2: The Darwin streaming server has links for creating movie and MP3 playlists on the top right of the web interface.

the one shown in Figure 3 appears. The text fields at the top are for the playlist name and mountpoint, that is, for the part of the URL that player programs will use to discover the playlist. And for good measure, you might like to select the right musical genre for the song in the box below the fields.

You can set the play mode for the list at the top right of the window; the play mode can be *Sequential*, *Sequential Looped*, or *Weighted Random*. To ensure that Darwin really does play the tracks in random order, enter a zero in the *Repetition* text box. Otherwise, the server will play each track at least once before it repeats a track.

The field on the left of the interface shows you a list of available audio tracks, and the field on the right shows you the tracks in the current playlist. Clever use of Javascript by the Apple programmers allows you to drag and drop tracks from one field to another. You can double click to change to a subdirectory, or click on the *Open Folder* link below the text field.

Random or Sequential Play

There is a field for the weighting in *Weighted Random* mode following the track name in the field on the right. The bigger the number you enter here, the more often Darwin will play the MP3 file. After completing your playlist, click on *Save Changes* at the bottom of the window.

The web page shown next has an overview of the playlists that you have setup but are not actually playing. Clicking on the icon for the

| Table 1: Important Files and Paths | |
|---|-------------------------------|
| Name | Function |
| /usr/local/sbin/DarwinStreamingServer | Main program |
| /usr/local/sbin/streamingadminserver.pl | Perl script for web front-end |
| /etc/streaming/ | Configuration directory |
| /etc/streaming/streamingserver.xml | Server configuration file |
| /var/streaming/logs | Logfile directory |

playlist status will launch the playlist. MP3 players can access the stream as *http://servername:8000/mountpoint*.

Darwin stores logfiles to match in a subdirectory below */var/streaming/playlists* which uses the playlist name. The file extension *.err* is for the error log, *.config* points to the configuration, *.log* is the logfile, and *.playlist* is the playlist itself.

By default Darwin only uses the central */usr/local/movies* directory for video and audio files. A *createuserstreamingdir jim* command would create a *Sites/Streaming* directory tree below the home directory for a user called *jim*. Jim can store his media files here. Players can access these files using addresses such as *rtsp://Servername/~jim/movie.mp4*.

The Trouble with Troubleshooting

The complexity of this whole affair makes it quite difficult to find out exactly where things are going wrong if they do go wrong. Players are not renowned for the clarity of their error messages, and this often leaves users wondering whether the player was unable to find the file on the server, or if

the problem is simply the case of a missing codec.

The server side does not help to shed any light on the issue. Although the Darwin server generates logfiles, their content is mostly useless. For example, */var/streaming/logs/Error.log* did not have a single error report, not to mention tips on how to remedy the issue. The *Error Log* entry in the web interface takes you to the same nonsensical data. At least the playlist logfiles had some details that could help point you in the right direction when troubleshooting.

Conclusion

From a technical point of view, the Darwin streaming server has a lot of potential. And you can set up a free streaming solution, which will not die overnight, in combination with free tools such as FFMpeg and MPEG4IP.

Unfortunately, usability is still a big issue – despite Darwin’s well-designed web interface. Ironically, the installation and troubleshooting issues don’t look difficult to fix for someone familiar with the software, but Linux support probably isn’t at the top of Apple’s priority list. ■

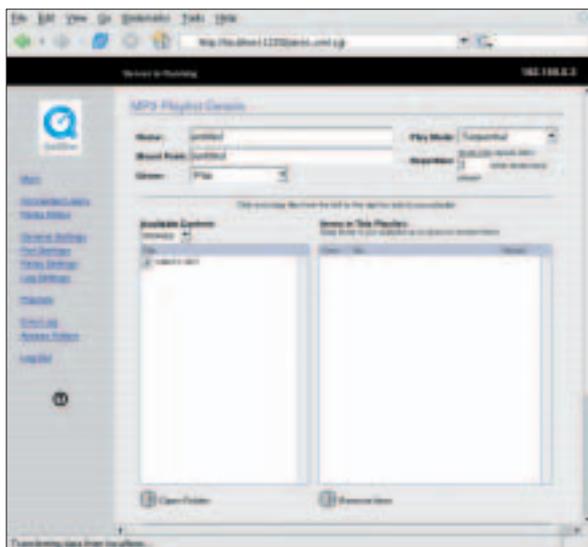


Figure 3: The user-friendly web interface for the playlists lets you drag and drop tracks to add them to a list.

INFO

- [1] Helix Basic Server: <http://forms.real.com/rnforms/products/servers/eval/mbps.html>
- [2] Apple Public Source License: <http://www.opensource.apple.com/apsl>
- [3] Registration: <https://signin.apple.com>
- [4] Darwin Streaming Server: <http://developer.apple.com/darwin/projects/streaming/>
- [5] MPlayer: <http://www.mplayerhq.hu>
- [6] LIVE library for RTSP: <http://www.live.com/liveMedia>
- [7] VideoLAN: <http://videolan.org/vlc/>
- [8] FFMpeg: <http://ffmpeg.sourceforge.net>
- [9] MPEG4IP: <http://mpeg4ip.sourceforge.net>