

shairport-sync

Synchronised Audio Player for iTunes / AirPlay

Synopsis

```
shairport-sync [-dvw] [-a name] [-A latency] [-B command] [-c
configurationfile] [-E command] [--forkedDaapdLatency=latency] [--get-cover-
art] [-i latency] [-L latency] [-m backend] [--meta-dir=directory] [-o backend] [--
password=secret] [-r threshold] [--statistics] [-S mode] [-t timeout] [--
tolerance=frames] [-- audio_backend_options]
shairport-sync -D
shairport-sync -h
shairport-sync -k
shairport-sync -h
shairport-sync -R
shairport-sync -V
```

Description

shairport-sync plays audio streamed from iTunes or from an AirPlay device to an ALSA-compatible audio output device.

A feature of shairport-sync is that the audio is played synchronously. This means that if many devices are playing the same stream at the same time, all the outputs will stay in step with one another. This allows multiple devices play the same source without getting out of phase with one another, enabling, for example, simultaneous multi-room operation.

shairport-sync can additionally be compiled and configured to stream raw audio to a pipe or to stdout.

Settings can be made using the configuration file (recommended for all new installations) or by using command-line options.

Configuration File Settings

You should use the configuration file for setting up shairport-sync. This file is normally */etc/shairport-sync.conf*. You may need to have root privileges to modify it.

Settings are organised into groups, for example, there is a "general" group of standard settings, and there is an "alsa" group with settings that pertain to the ALSA back end. Here is an example of a typical configuration file:

```
general = {
```

```
name = "Mike's Boombox";  
interpolation = "soxr";  
password = "secret";  
};  
alsa = {  
output_device = "hw:0";  
mixer_control_name = "PCM";  
};
```

Most settings have sensible default values, so -- as in the example above -- users generally only need to set (1) the service name, (2) a password (if desired) and (3) the output device. If the output device has a mixer that can be used for volume control, then (4) the volume control's name should be specified. It is highly desirable to use the output device's mixer for volume control, if available -- response time is reduced to zero and the processor load is reduced. In the example above, "soxr" interpolation was also enabled.

A sample configuration file with all possible settings, but with all of them commented out, is installed at */etc/shairport-sync.conf.sample*.

To retain backwards compatability with previous versions of shairport-sync you can use still use command line options, but any new features, etc. will be available only via configuration file settings.

The configuration file is processed using the *libconfig* library -- see http://www.hyperrealm.com/libconfig/libconfig_manual.html.

"GENERAL" SETTINGS

These are the settings available within the **general** group:

```
name="service_name";
```

Use this *service_name* to identify this player in iTunes, etc. The default name is "Shairport Sync on <hostname>".

```
password="password";
```

Require the password *password* to connect to the service. If you leave this setting commented out, no password is needed.

```
interpolation="mode";
```

Interpolate, or "stuff", the audio stream using the *mode*. Interpolation here refers to the process of adding or removing frames of audio to or from the stream sent to the output device to keep it exactly in synchrony with the player. The default mode,

"basic", is normally almost completely inaudible. The alternative mode, "soxr", is even less obtrusive but requires much more processing power. For this mode, support for libsoxr, the SoX Resampler Library, must be selected when shairport-sync is compiled.

statistics=*"setting"*;

Use this *setting* to enable ("yes") or disable ("no") the output of some statistical information on the console or in the log. The default is to disable statistics.

mdns_backend=*"backend"*;

shairport-sync has a number of modules of code ("backends") for interacting with the mDNS service to be used to advertise itself. Normally, the first mDNS backend that works is selected. This setting forces the selection of the specific mDNS *backend*. The default is "avahi". Perform the command **shairport-sync -h** to get a list of available mDNS modules.

output_backend=*"backend"*;

shairport-sync has a number of modules of code ("backends") through which audio is output. Normally, the first audio backend that works is selected. This setting forces the selection of the specific audio *backend*. The default is "alsa". Perform the command **shairport-sync -h** to get a list of available audio backends. Only the alsa backend supports synchronisation.

port=*portnumber*;

Use this to specify the *portnumber* shairport-sync uses to listen for service requests from iTunes, etc. The default is port 5000.

udp_port_base=*portnumber*;

When shairport-sync starts to play audio, it establishes three UDP connections to the audio source. Use this setting to specify the starting *portnumber* for these three ports. It will pick the first three unused ports starting from *portnumber*. The default is port 6001.

udp_port_range=*range*;

Use this in conjunction with the previous setting to specify the *range* of ports that can be checked for availability. Only three ports are needed. The default is 100, thus 100 ports will be checked from port 6001 upwards until three are found.

drift=*frames*;

Allow playback to drift up to *frames* out of exact synchronization before attempting to correct it. The default is 88 frames, i.e. 2 ms. The smaller the tolerance, the more likely it is that overcorrection will occur. Overcorrection is when more corrections (insertions and deletions) are made than are strictly necessary to keep the stream in sync. Use the **statistics** setting to monitor correction levels. Corrections should not greatly exceed net corrections.

resync_threshold=*threshold*;

Resynchronise if timings differ by more than *threshold* frames. If the output timing differs from the source timing by more than the threshold, output will be muted and a full resynchronisation will occur. The default threshold is 2,205 frames, i.e. 50 milliseconds. Specify 0 to disable resynchronisation.

log_verbosity=0;

Use this to specify how much debugging information should be output or logged. "0" means no debug information, "3" means most debug information. The default is "0".

ignore_volume_control="choice";

Set this *choice* to "yes" if you want the volume to be at 100% no matter what the source's volume control is set to. This might be useful if you want to set the volume on the output device, independently of the setting at the source. The default is "no".

"LATENCIES" SETTINGS

There are four default latency settings, chosen automatically. One latency matches the latency used by recent versions of iTunes when playing audio and another matches the latency used by so-called "AirPlay" devices, including iOS devices and iTunes and Quicktime Player when they are playing video. A third latency is used when the audio source is forked-daapd. The fourth latency is the default if no other latency is chosen and is used for older versions of iTunes.

If you want to change latencies to compensate for a delay in the audio output device (which will have the same effect on all sources), instead of changing these individual latencies, consider using the **audio_backend_latency_offset** setting in the **alsa** group (or the appropriate other group if you're not outputting through the alsa backend).

itunes=*latency*;

This is the *latency*, in frames, used for iTunes 10 or later. Default is 99,400.

airplay=*latency*;

This is the *latency*, in frames, used for AirPlay devices, including iOS devices and iTunes and Quicktime Player when they are playing video. Default is 88,200.

forkedDaapd=*latency*;

This is the *latency*, in frames, used for forkedDaapd sources. Default is 99,400.

default=*latency*;

This is the *latency*, in frames, used when the source is unrecognised. Default is 88,200.

"METADATA" SETTINGS

shairport-sync can process metadata provided by the source, such as Track Number, Album Name, cover art, etc. and can provide additional metadata such as volume level, pause/resume, etc. It provides the metadata to a pipe, by default */tmp/shairport-sync-metadata*. To process metadata, shairport-sync must have been compiled with metadata support included. You can check that this is so by running **shairport-sync -V**; the identification string will contain the word **metadata**.

The **metadata** group of settings allow you to enable metadata handling and to control certain aspects of it:

enabled="choice";

Set the *choice* to "yes" to enable shairport-sync to look for metadata from the audio source and to forward it, along with metadata generated by shairport-sync itself, to the metadata pipe. The default is "no".

include_cover_art="choice";

Set the *choice* to "yes" to enable shairport-sync to look for cover art from the audio source and to include it in the feed to the metadata pipe. You must also enable metadata (see above). One reason for not including cover art is that the images can sometimes be very large and may delay transmission of subsequent metadata through the pipe. The default is "no".

pipe_name="filepathname";

Specify the absolute path name of the pipe through which metadata should be sent. The default is */tmp/shairport-sync-metadata*.

"SESSIONCONTROL" SETTINGS

shairport-sync can run programs just before it starts to play an audio stream and just after it finishes. You specify them using the settings **run_this_before_play_begins** and **run_this_after_play_ends**.

run_this_before_play_begins="/path/to/application and args";

Here you can specify a program and its arguments that will be run just before a play session begins. Be careful to include the full path to the application. The application must be marked as executable and, if it is a script, its first line must begin with the standard *#!/bin/...* as appropriate.

run_this_after_play_ends="/path/to/application and args";

Here you can specify a program and its arguments that will be run just after a play session ends. Be careful to include the full path to the application. The application must be marked as executable and, if it is a script, its first line must begin with the standard *#!/bin/...* as appropriate.

wait_for_completion="choice";

Set *choice* to "yes" to make shairport-sync wait until the programs specified in the **run_this_before_play_begins** and **run_this_after_play_ends** have completed execution before continuing. The default is "no".

allow_session_interruption="*choice*";

If **choice** is set to "yes", then another source will be able to interrupt an existing play session and start a new one. When set to "no" (the default), other devices attempting to interrupt a session will fail, receiving a busy signal.

session_timeout=*seconds*;

If a play session has been established and the source disappears without warning (such as a device going out of range of a network) then wait for *seconds* seconds before ending the session. Once the session has terminated, other devices can use it. The default is 120 seconds.

"ALSA" SETTINGS

These settings are for the ALSA back end, used to communicate with audio output devices in the ALSA system. (By the way, you can use tools such as **alsamixer** or **aplay** to discover what devices are available.) Use these settings to select the output device and the mixer control to be used to control the output volume. You can additionally set the desired size of the output buffer and you can adjust overall latency. Here are the **alsa** group settings:

output_device="*output_device*";

Use the output device called *output_device*. The default is the device called "default".

mixer_control_name="*name*";

Specify the *name* of the mixer control to be used by shairport-sync to control the volume. The mixer control must be on the mixer device, which by default is the output device. If you do not specify a mixer control name, shairport-sync will adjust the volume in software.

mixer_type="*mixer_type*";

This setting is deprecated and is ignored. For your information, its functionality has been automatically incorporated in the **mixer_control_name** setting -- if you specify a mixer name with the **mixer_control_name** setting, it is assumed that the mixer is implemented in hardware.

mixer_device="*mixer_device*";

By default, the mixer is assumed to be **output_device**. Use this setting to specify a device other than the output device.

audio_backend_latency_offset=*offset*;

Set this *offset*, in frames, to compensate for a fixed delay in the audio back end. For example, if the output device delays by 100 ms, set this to -4410.

audio_backend_buffer_desired_length=length;

Use this to set the desired number frames to be in the output device's hardware output buffer. The default is 6,615 frames, or 0.15 seconds. If set too small, buffer underflow may occur on low-powered machines. If too large, the response times when using software volume control (i.e. when not using a mixer control to control volume) become annoying, or it may exceed the hardware buffer size. It may need to be larger on low-powered machines that are also performing other tasks, such as processing metadata.

"PIPE" SETTINGS

These settings are for the PIPE backend, used to route audio to a named unix pipe. The audio is in raw CD audio format: PCM 16 bit little endian, 44,100 samples per second, stereo.

Use the *name* setting to set the name and location of the pipe.

There are two further settings affecting timing that might be useful if the pipe reader is, for example, a program to play an audio stream such as **aplay**. The *audio_backend_latency_offset* affects precisely when the first audio packet is sent and the *audio_backend_buffer_desired_length* setting affects the nominal output buffer size.

These are the settings available within the **pipe** group:

name="/path/to/pipe";

Use this to specify the name and location of the pipe. The pipe will be created and opened when shairport-sync starts up and will be closed upon shutdown. Frames of audio will be sent to the pipe in packets of 352 frames and will be discarded if the pipe has not have a reader attached. The sender will wait for up to five seconds for a packet to be written before discarding it.

audio_backend_latency_offset=offset_in_frames;

Packets of audio frames are written to the pipe synchronously -- that is, they are written to at exactly the time they should be played. You can offset the time of initial audio output relative to its nominal time using this setting. For example to send an audio stream to the pipe 100 milliseconds before it is due to be played, set this to -4410. Default setting is 0.

audio_backend_buffer_desired_length=buffer_length_in_frames;

Use this setting, in frames, to set the size of the output buffer. It works by determining how soon the second and subsequent packets of audio frames are sent to the pipe. For example, if you send the first packet of audio exactly when it is due and, using a *audio_backend_buffer_desired_length* setting of 44100, send subsequent packets of audio a second before they are due to be played, they will be

buffered in the pipe reader's buffer, giving it a nominal buffer size of 44,100 frames. Note that if the pipe reader consumes audio packets faster or slower than they are supplied, the buffer will eventually empty or overflow -- shairport-sync performs no stuffing or interpolation when writing to a pipe. Default setting is 44,100 frames.

"STDOUT" SETTINGS

These settings are for the STDOUT backend, used to route audio to standard output ("stdout"). The audio is in raw CD audio format: PCM 16 bit little endian, 44,100 samples per second, stereo.

There are two settings affecting timing that might be useful if the stdout reader is, for example, a program to play an audio stream such as **aplay**. The *audio_backend_latency_offset* affects precisely when the first audio packet is sent and the *audio_backend_buffer_desired_length* setting affects the nominal output buffer size.

These are the settings available within the **stdout** group:

audio_backend_latency_offset=*offset_in_frames*;

Packets of audio frames are written to stdout synchronously -- that is, they are written at exactly the time they should be played. You can offset the time of initial audio output relative to its nominal time using this setting. For example to send an audio stream to stdout 100 milliseconds before it is due to be played, set this to -4410. Default setting is 0.

audio_backend_buffer_desired_length=*buffer_length_in_frames*;

Use this setting, in frames, to set the size of the output buffer. It works by determining how soon the second and subsequent packets of audio frames are sent to stdout. For example, if you send the first packet of audio exactly when it is due and, using a *audio_backend_buffer_desired_length* setting of 44100, send subsequent packets of audio a second before they are due to be played, they will be buffered in the stdout reader's buffer, giving it a nominal buffer size of 44,100 frames. Note that if the stdout reader consumes audio packets faster or slower than they are supplied, the buffer will eventually empty or overflow -- shairport-sync performs no stuffing or interpolation when writing to stdout. Default setting is 44,100 frames.

Options

Note: if you are setting up shairport-sync for the first time or are updating an existing installation, you are encouraged to use the configuration file settings described above. Most of the options described below simply replicate the configuration settings and are retained to provide backward compatibility with older installations of shairport-sync.

Many of the options take sensible default values, so you can normally ignore most of them. See the EXAMPLES section for typical usages.

The command line for shairport-sync can take two kinds of options: regular **program options** and **audio backend options**. Program options are always listed first, followed by any audio backend options, preceded by a `--` symbol.

Program Options

These options are used by shairport-sync itself.

-a *service name* | **--name=***service name*

Use this *service name* to identify this player in iTunes, etc. The default name is "Shairport Sync on <hostname>".

-A | **--AirPlayLatency=***latency*

Use this *latency*, in frames, for audio streamed from an AirPlay device. The default is 88,200 frames, where there are 44,100 frames to the second.

-B *program* | **--on-start=***program*

Execute *program* when playback is about to begin. Specify the full path to the program, e.g. `/usr/bin/logger`. Executable scripts can be used, but they must have `#!/bin/sh` (or whatever is appropriate) in the headline.

If you want shairport-sync to wait until the command has completed before starting to play, select the **-w** option as well.

-c *filename* | **--configfile=***filename*

Read configuration settings from *filename*. The default is to read them from `/etc/shairport-sync.conf`. For information about configuration settings, see the "Configuration File Settings" section above.

-D | **--disconnectFromOutput**

Disconnect the shairport-sync daemon from the output device and exit. (Requires that the daemon has written its PID to an agreed file -- see the **-d** option).

-d | **--daemon**

Instruct shairport-sync to demonise itself. It will write its Process ID (PID) to a file, usually at `/var/run/shairport-sync.pid`, which is used by the **-k**, **-D** and **-R** options to locate the daemon at a later time.

-E *program* | **--on-stop=***program*

Execute *program* when playback has ended. Specify the full path to the program, e.g. `/usr/bin/logger`. Executable scripts can be used, but they must have `#!/bin/sh` (or whatever is appropriate) in the headline.

If you want shairport-sync to wait until the command has completed before

continuing, select the **-w** option as well.

--forkedDaapdLatency=latency

Use this *latency*, in frames, for audio streamed from a forked-daapd based source. The default is 99,400 frames, where there are 44,100 frames to the second.

--get-coverart

This option requires the **--meta-dir** option to be set, and enables shairport-sync to request cover art from the source and to transmit it through the metadata pipe.

Please note that cover art data may be very large, and may place too great a burden on your network.

-h | --help

Print brief help message and exit.

-i | --iTunesLatency=latency

Use this *latency*, in frames, for audio streamed from an iTunes source, where iTunes is Version 10 or later. The default is 99,400 frames, where there are 44,100 frames to the second. If the source is iTunes but is earlier than Version 10, the *default latency* is used (see the **-L** option). Some third party programs masquerade as older versions of iTunes.

-k | --kill

Kill the shairport-sync daemon and exit. (Requires that the daemon has written its PID to an agreed file -- see the **-d** option).

-L | --latency=latency

Use this to set the *default latency*, in frames, for audio coming from an unidentified source or from an iTunes Version 9 or earlier source. The standard value for the *default latency* is 88,200 frames, where there are 44,100 frames to the second.

--meta-dir=directory

Listen for metadata coming from the source and send it, along with metadata from shairport-sync itself, to a pipe called *shairport-sync-metadata* in the *directory* you specify. If you add the **--get-cover-art** then cover art will be sent through the pipe too. See <https://github.com/mikebrady/shairport-sync-metadata-reader> for a sample metadata reader.

-m mdnsbackend | --mdns=mdnsbackend

Force the use of the specified mDNS backend to advertise the player on the network. The default is to try all mDNS backends until one works.

-o outputbackend | --output=outputbackend

Force the use of the specified output backend to play the audio. The default is to try the first one. (This is not used at present.)

-p *port* | **--port=***port*

Listen for play requests on *port*. The default is to use port 5000.

--password=*secret*

Require the password *secret* to be able to connect and stream to the service.

-R | **--reconnectToOutput**

Reconnect the shairport-sync daemon to the output device and exit. It may take a few seconds to synchronise. (Requires that the daemon has written its PID to an agreed file -- see the **-d** option).

-r *threshold* | **--resync=***threshold*

Resynchronise if timings differ by more than *threshold* frames. If the output timing differs from the source timing by more than the threshold, output will be muted and a full resynchronisation will occur. The default threshold is 2,205 frames, i.e. 50 milliseconds. Specify **0** to disable resynchronisation.

--statistics

Print some statistics in the standard output, or in the logfile if in daemon mode.

-S *mode* | **--stuffing=***mode*

Stuff the audio stream using the *mode*. "Stuffing" refers to the process of adding or removing frames of audio to or from the stream sent to the output device to keep it exactly in synchrony with the player. The default mode, **basic**, is normally almost completely inaudible. The alternative mode, **soxr**, is even less obtrusive but requires much more processing power. For this mode, support for libsoxr, the SoX Resampler Library, must be selected when shairport-sync is compiled.

-t *timeout* | **--timeout=***timeout*

Exit play mode if the stream disappears for more than *timeout* seconds.

When shairport-sync plays an audio stream, it starts a play session and will return a busy signal to any other sources that attempt to use it. If the audio stream disappears for longer than *timeout* seconds, the play session will be terminated. If you specify a timeout time of **0**, shairport-sync will never signal that it is busy and will not prevent other sources from "barging in" on an existing play session. The default value is 120 seconds.

--tolerance=*frames*

Allow playback to be up to *frames* out of exact synchronization before attempting to correct it. The default is 88 frames, i.e. 2 ms. The smaller the tolerance, the more

likely it is that overcorrection will occur. Overcorrection is when more corrections (insertions and deletions) are made than are strictly necessary to keep the stream in sync. Use the **--statistics** option to monitor correction levels. Corrections should not greatly exceed net corrections.

-V | --version

Print version information and exit.

-v | --verbose

Print debug information. Repeat up to three times to get more detail.

-w | --wait-cmd

Wait for commands specified using **-B** or **-E** to complete before continuing execution.

Audio Backend Options

These options are passed to the chosen audio backend. (At present, the only backend implemented is for ALSA.) The audio backend options are preceded by a **-** symbol to introduce them and to separate them from any program options. In this way, option letters can be used as program options and also as audio backend options without ambiguity.

In the ALSA backend, audio is sent to an output device which you can specify using the **-d** option. The output level (the "volume") is controlled using a level control associated with a mixer. By default, the mixer is implemented in shairport-sync itself, i.e. the type of the mixer is "software". To use a level control on a mixer on the sound card, you must (a) specify that the mixer's type is "hardware" with the **-t** option; (b) you must specify where the mixer is to be found using the **-m** option and finally (c) you must specify the name of the level control you are using with the **-c** option.

-c *controlname*

Use the level control called *controlname* on the hardware mixer for controlling volume. This is needed if the mixer type is specified, using the **-t** option, to be **hardware**. There is no default.

-d *device*

Use the specified output *device*. You may specify a card, e.g. **hw:0**, in which case the default output device on the card will be chosen. Alternatively, you can specify a specific device on a card, e.g. **hw:0,0**. The default is the device named **default**.

-m *mixer*

Use the specified hardware *mixer* for volume control. Use this to specify where the mixer is to be found. For example, if the mixer is associated with a card, as is often

the case, specify the card, e.g. **hw:0**. If (unusually) the mixer is associated with a specific device on a card, specify the device, e.g. **hw:0,1**. The default is the device named in the **-d** option, if given, or the device named **default**.

-t *devicetype*

This option is deprecated and is ignored. For your information, its functionality has been automatically incorporated in the **-c** option -- if you specify a mixer name with the **-c** option, it is assumed that the mixer is implemented in hardware.

Examples

Here is a slightly contrived but typical example:

```
shairport-sync -d -a "Joe's Stereo" -S soxr -- -d hw:1,0 -m hw:1 -c PCM
```

The program will run in daemon mode (**-d**), will be visible as "Joe's Stereo" (**-a "Joe's Stereo"**) and will use the SoX Resampler Library-based stuffing (**-S soxr**). The audio backend options following the **--** separator specify that the audio will be output on output 0 of soundcard 1 (**-d hw:1,0**) and will take advantage of the same sound card's mixer (**-m hw:1**) using the level control named "PCM" (**-c "PCM"**).

The example above is slightly contrived in order to show the use of the **-m** option. Typically, output 0 is the default output of a card, so the output device could be written **-d hw:1** and then the mixer option would be unnecessary, giving the following, simpler, command:

```
shairport-sync -d -a "Joe's Stereo" -S soxr -- -d hw:1 -c PCM
```

Credits

Mike Brady developed shairport-sync from the original shairport by James Laird.

shairport-sync can be found at <https://github.com/mikebrady/shairport-sync>.

shairport can be found at <https://github.com/abrasive/shairport>.

Comments

This man page was written using [xml2man](#) by Oliver Kurth.