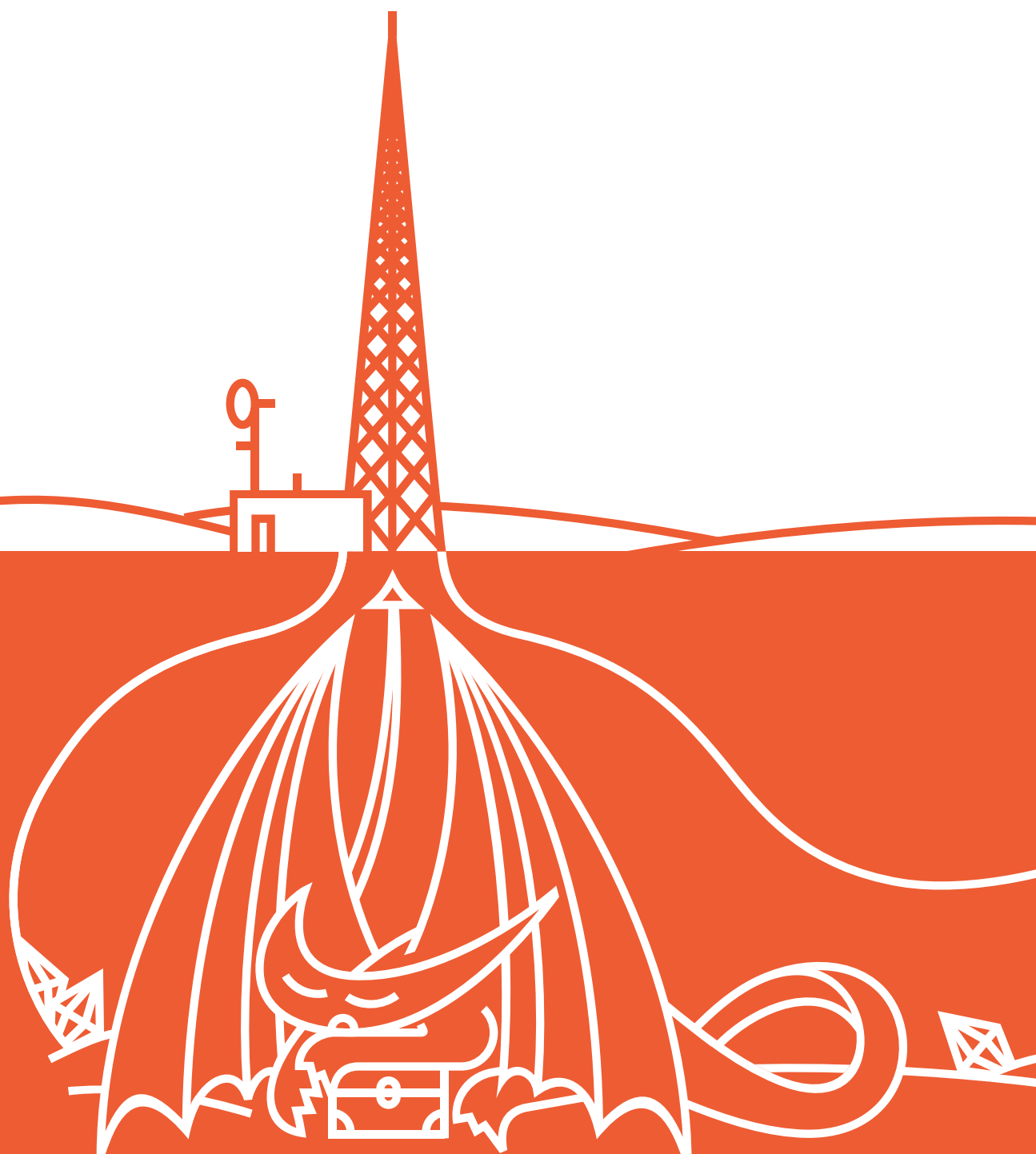# Airtime 2.5
for Broadcasters

The open radio software for scheduling
and remote station management

USER GUIDE

# INTRODUCTION

1. WHAT IS AIRTIME?

Updated for Airtime 2.5.1

Airtime is the open broadcast software for scheduling and remote station management. Web browser access to the station's media library, multi-file upload and automatic metadata import features are coupled with a collaborative on-line scheduling calendar and playlist management. The scheduling calendar is managed through an easy-to-use interface and triggers playout with sub-second precision.
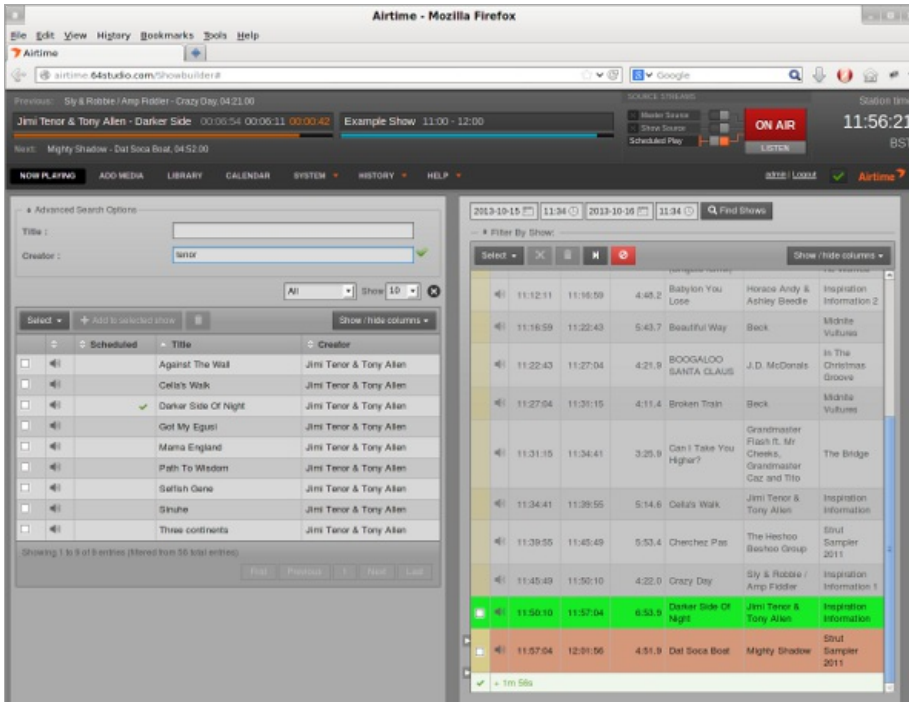

Airtime has been intended to provide a solution for a wide range of broadcast projects, from community to public and commercial stations. The scalability of Airtime allows implementation in a number of scenarios, ranging from an unmanned broadcast unit accessed remotely through the Internet, to a local network of machines accessing a central Airtime storage system. Airtime supports the playout of lossy compressed audio files in both MP3 and AAC formats and the open, royalty-free equivalent Ogg Vorbis. It also supports playout of lossless FLAC and WAV format audio files.

Airtime manages the Liquidsoap stream generator at the heart of the system. Liquidsoap generates streams from files in the Airtime library and any remote input streams that you specify. Available stream output formats include Ogg Vorbis, Ogg Opus, MP3, and AAC. The library is indexed in a PostgreSQL database to enable searching. Live shows can be recorded automatically with Ecasound, using the sound card line input. News editors, DJs and station controllers can use Airtime to build playlists or smart blocks and manage media files (upload, edit metadata, manage advertisements) at the station or via the Internet.

The Airtime administration interface is designed to work with any web browser, on any desktop or mobile platform with a minimum display size of 1280x768 pixels. Airtime looks its best on a high definition display of 1920x1080 pixels. The recommended web browsers are **Mozilla Firefox 25** or **Google Chrome 30** (or later versions). **Apple Safari 6** (or later) is also supported.

International UTF-8 metadata in media files is supported throughout, and the Airtime interface can be localized into any language or dialect using the standard GNU gettext utility. Localizations that are installed by default include Austrian, Brazilian, British, Canadian, Chinese, Czech, French, German, Greek, Hungarian, Italian, Korean, Polish, Russian, Spanish and USA. See the chapter *Interface localization* for details of how to update a translation or add a new localization.
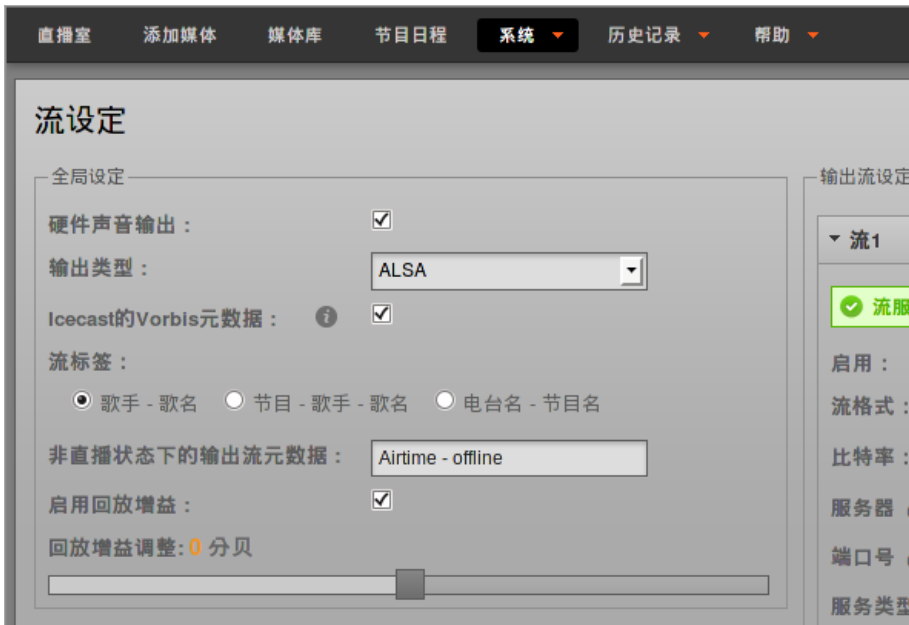
The scheduler in Airtime has a calendar view, organized by months, weeks and days. Here the program editors can schedule playlists and shows for their broadcast station. In some scenarios, the transmitter is situated outside the reach of the broadcaster and all program management has to be maintained through the web interface. Possible reasons for this scenario might be of a pragmatic nature (running many stations from one central office due to limited human resources) or an emergency (running a transmitter in a crisis area without putting staff at risk).

A fully managed Airtime Pro service is available from Sourcefabric. Airtime source code and packages for Debian and Ubuntu are also available for download, under the GNU General Public License version 3. If you're an Airtime Pro user, you can go straight to the *Getting started* chapter of this book and log in. If you wish to install Airtime on your own server, you should start at the *Preparing the server* chapter, and work your way through the installation steps. Alternatively, a demonstration server is available for public use at:

http://airtime-demo.sourcefabric.org/

AIRTIME WORKFLOW

This typical workflow is intended to clarify the difference between the various components that make up a complete Airtime system.

1. There are media files on a storage server, which include metadata in their tags (title, creator, genre and so on). This storage server might be accessed directly via studio workstations on the local network.

2. The Airtime media-monitor keeps track of files being added, renamed, moved or removed from storage, and reads their metadata using the Mutagen library.

3. A PostgreSQL database contains the location of those media files and their metadata. This means you can search for and playlist a set of media files according to the specific metadata that you require, or use a 'smart block' to select the files for you. The database also contains details of specified remote input streams.

4. Pypo, the Python Playout engine, downloads media from the storage up to 24 hours ahead of playout and checks it for average level (with ReplayGain tools) and leading or trailing silence (with Silan). At playout time, the media to be broadcast is sent to Liquidsoap.

5. Liquidsoap takes individual media files and remote input streams, and assembles them into a continuous output stream. This stream can be sent to a sound card (e.g. for a broadcast mixer, on the way to an FM or DAB transmitter) or to a streaming server for IP network distribution, over the LAN, local WiFi or the Internet. You can stream to a sound card and up to three different stream distribution servers with the same Airtime server, if you wish.

6. Icecast (http://www.icecast.org/) is the default stream distribution server, and there is also support for SHOUTcast (http://www.shoutcast.com), but in theory you could stream from Liquidsoap to any online service. If a suitable Liquidsoap output is not available for your streaming service of choice, you can send audio from Liquidsoap to a separate encoding or streaming machine via a sound card or relay stream.

7. Monit monitors the health of pypo, media-monitor and Liquidsoap, and reports the status of these services to Airtime.

8. RabbitMQ pushes messages from Airtime to media-monitor and pypo about changes to media files and the playout schedule.

9. Airtime manages all of these components, and provides an easy, multi-user web interface to the system. It enables your station staff, depending on the permissions you have granted them, to:

a) upload media files to the storage server via the **Add Media** page

b) automatically import file metadata into the PostgreSQL database

c) search for and download media files, and edit the metadata of individual files, if required, on the **Library** page

d) create and edit playlists of media files or create smart blocks of content based on metadata, edit cue points and fades, and audition them. Playlists and smart blocks are also saved in the database, and can be searched for

e) schedule colour-coded broadcast shows (which can contain playlists, smart blocks, pre-recorded complete shows, timed remote input streams, or be live) for specific dates and times on the **Calendar** page. Regular shows can be scheduled by the day of the week or month, and can be linked to share content

f) automatically record live shows at specific times and dates (in 256 kbps Ogg Vorbis format by default) from the sound card input with Ecasound, upload them to the storage server and import them into the database

g) manage presenter, staff and guest access to Airtime, and contact details, via the **Manage Users** page

h) see what is about to be played by Liquidsoap on the **Now Playing** page, with support for last-minute changes to the content

i) upload media files from Airtime to a third-party hosting service, such as SoundCloud

j) audition available output streams from the server using the **Listen** button

k) check the status and resource usage of system components on the **Status** page

l) export the broadcast schedule to external sites via the Schedule API

m) see logs on the **Playout History** page and view graphs on the **Listener Stats** page

n) configure the Airtime system on the **Preferences**, **Media Folders** and **Streams** pages.
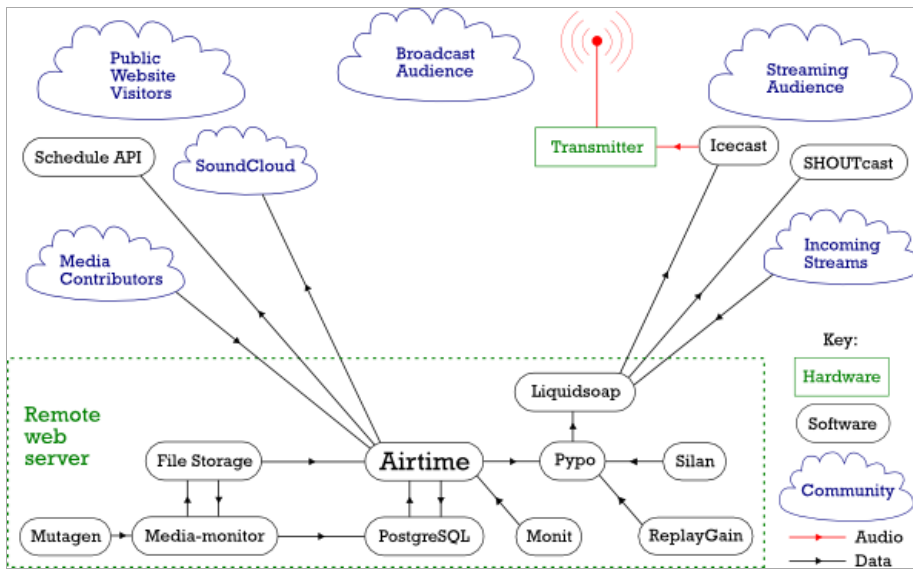
EXAMPLE STUDIO BROADCAST SYSTEM

In the diagram of an FM radio station below, Airtime is hosted on a server connected to the local network, with direct soundcard access. Liquidsoap outputs streams to both the transmitter, via the main studio mixer, and streaming media servers. The machine running Airtime is behind a firewall because it is also connected to the Internet for remote access by media contributors. This enables Airtime to offer password-protected access to the media library and scheduling from both inside and outside the studio building.
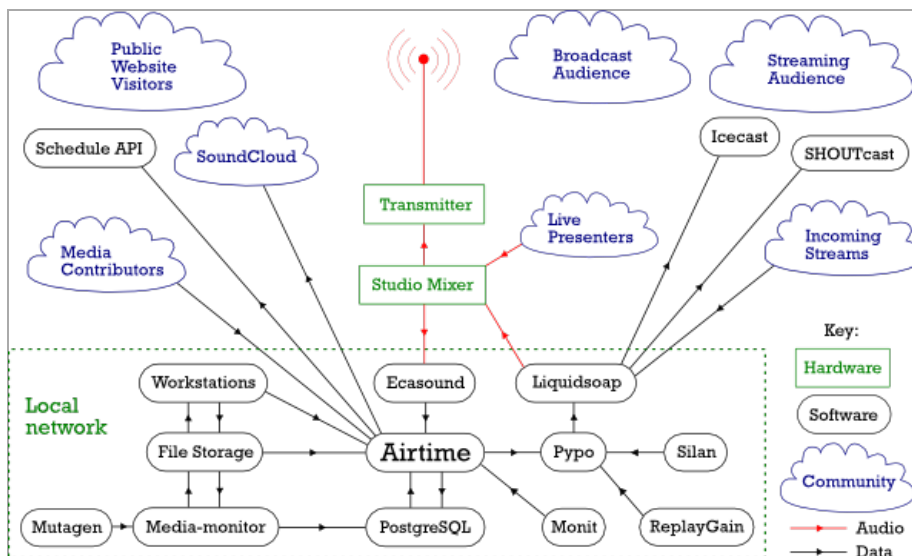
EXAMPLE WEB BROADCAST SYSTEM

In the diagram below, Airtime is hosted on a remote web server, and has no soundcard. There does not need to be a centralised studio, although Airtime can enable remote studios to stream in to Liquidsoap at authorised times. Optionally, the outgoing Icecast stream can be relayed to a transmitter.

Public Website Visitors • Broadcast Audience • Streaming Audience • Schedule API • SoundCloud • Transmitter • Icecast • SHOUTcast • Media Contributors • Incoming Streams • Remote web server • Liquidsoap • File Storage • Airtime • Pypo • Silan • Mutagen • Media-monitor • PostgreSQL • Monit • ReplayGain

Key:
Hardware
Software
Community
→ Audio
→ Data

2. RIGHTS AND ROYALTIES

If you're new to broadcasting, or have not streamed your station online before, reading the following brief explanation of compensation rules for songwriters, musicians and other copyright holders may save you a great deal of trouble later.

Independent music radio on the Internet has faced royalty demands from SoundExchange in the USA, and similar organizations in other territories. These organizations are usually membership societies or government-sanctioned national authorities which are intended to collect money from broadcasters to compensate copyright holders. The royalty collection societies require payment before you can stream just about any music released commercially to the general public — whether you make any money out of streaming, or not. It's not so much the percentage of revenue demanded, but that there are usually annual minimum fees to pay, which can hurt small stations disproportionately.

For example, in the UK, the MCPS-PRS Limited Online Music Licence covers non-commercial music streaming by groups and individuals, as long as their gross revenue is less then £12,500 per year. The cost is on a sliding scale, up to £1,120 plus 20% tax per year for delivering up to 450,000 individual streams or serving 25,000 files; after that, you have to apply for a full MCPS-PRS Online Music Licence. That doesn't sound too bad at first, but 25,000 files per year works out at less than four downloads per hour for a round-the-clock website. This particular licence only covers publishing (songwriter) rights, not recording (record label and musician's performance) rights, so you have to negotiate an additional licence from Phonographic Performance Limited (PPL) to play music online, including digital recordings converted from commercially released CDs, vinyl or tape.

Typically, you have to provide full statistical details to the royalty society of all music streamed or downloaded from your site, which can be onerous. Even if your radio station is mostly speech, there are many limitations in the small print of these music licences. For instance, you can't use music for promotional purposes, and you can't stream a whole opera, without negotiating separate licences. Weirdly, you are not allowed to play a piece of music in a 'derogatory context' to the writer or performers; no drummer jokes allowed, then.

However, the biggest pitfall is that these MCPS-PRS licences for publishing rights only cover listeners in the UK. For recording rights, PPL is a member of the IFPI reciprocal scheme for webcasters, which means its licenses cover listeners in some European countries, Australia, New Zealand and a few other countries, but not listeners in the USA or Canada. So if your Internet station picked up a significant number of listeners in countries not covered by the MCPS-PRS licences or the IFPI reciprocal scheme, you would have to pay for similar music licences in those countries as well. It's no wonder that many not-for-profit radio stations have disappeared from the virtual airwaves over the last few years, since not having the right licences could leave the operator liable to legal action.

If you want to go down the commercial music route, check out the http://www.prsformusic.com and http://www.ppluk.com websites for UK licence details. In the USA, the http://www.soundexchange.com website currently quotes a 500 dollar minimum annual fee for non-commercial webcasters, plus a usage fee above a certain number of listener hours, for the right to stream music recordings to listeners. See the websites of ASCAP, BMI and SESAC for details of music publishing royalties payable by webcasters streaming to the USA.

Free content streaming offers an alternative for DIY Internet radio. Since royalty collection societies like MCPS-PRS and SoundExchange can only represent the interests of their own members, it follows that if you are not a member, you can stream your own self-produced content without paying for their licences. If you state somewhere on your website that the stream is of your own copyrighted material, and is made available to the public under a specific licence, then no-one should misunderstand your intentions.

You might be able to persuade other people to allow you to stream their content too, as long as they do not have a conflicting legal obligation, such as having previously joined one of the many royalty collection societies around the world. You can ask for permission to stream when website visitors upload their own music files to you via a HTML form, much as the likes of SoundCloud do. Or you can collect files licensed under an appropriate Creative Commons licence (http://www.creativecommons.org) or other free content licence.

Explicit permission to stream on your particular server is always going to be the ideal, so think about your own terms and conditions before you accept files from third parties for streaming. How, for example, would you know if someone uploaded a file to your online radio station that unknown to you, had been ripped from a commercially released CD? That's the kind of thing that could get you in trouble with the licensing authorities and copyright holders.
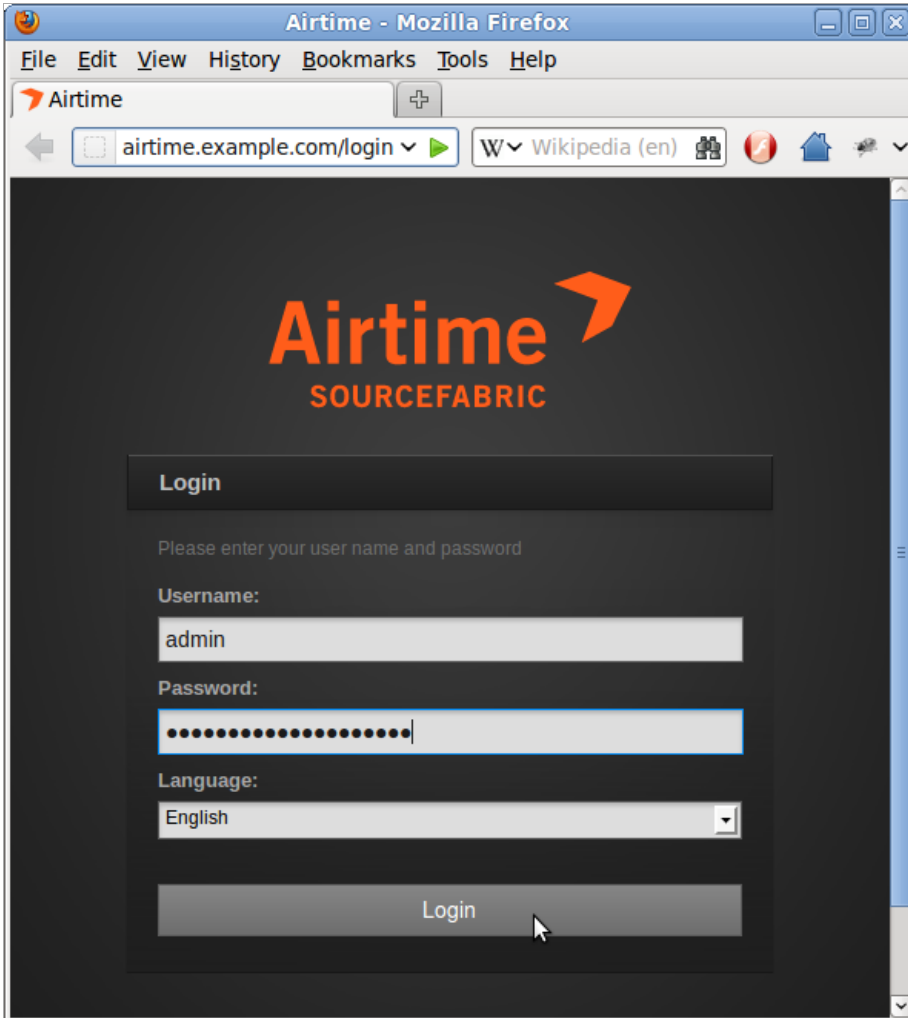
USING AIRTIME

3. ON AIR IN 60 SECONDS!
4. GETTING STARTED
**5. SMARTPHONE JOURNALISM**

3. ON AIR IN 60 SECONDS!

Here's how you can use Airtime to manage your broadcasts. Chapter names in this book are shown in *italics*, to help you find the details of each step (if you need to read more).

1. Log in to your Airtime server with your **Username** and **Password** (*Getting started*).


2. Add your files to the Airtime library by clicking **Add media** on the main menu, then click the **Add files** button. You can drag and drop your files into this window too. Then click the **Start upload** button (*Add media*).
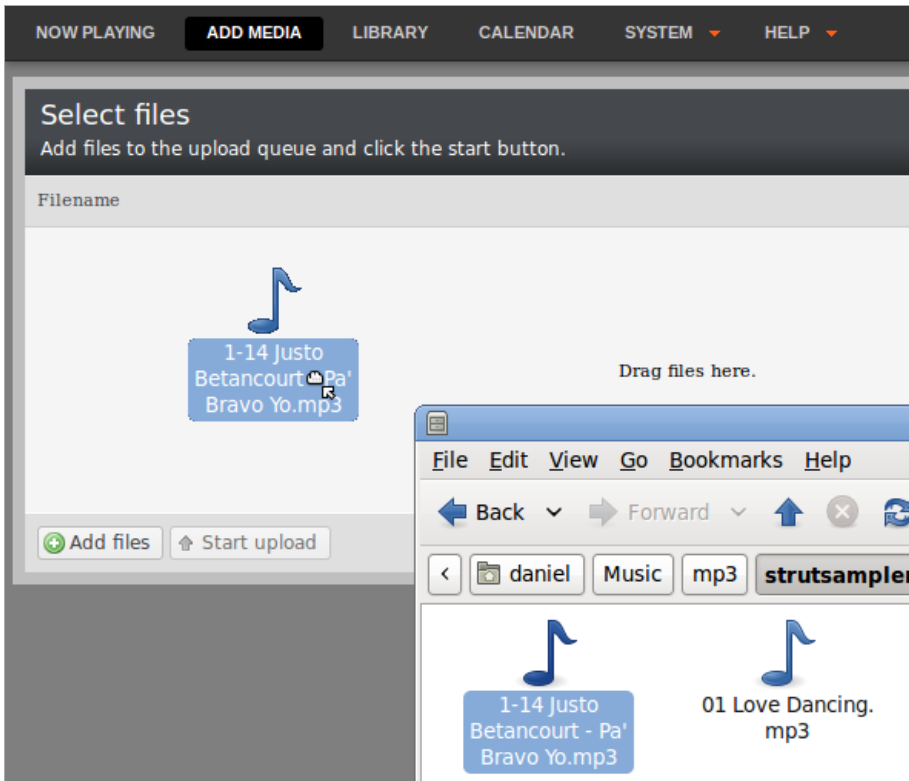
3. Create a show by clicking **Calendar** on the main menu, and then clicking the **+ Show** button (*Calendar*). Only admins and program managers can add shows (Users).
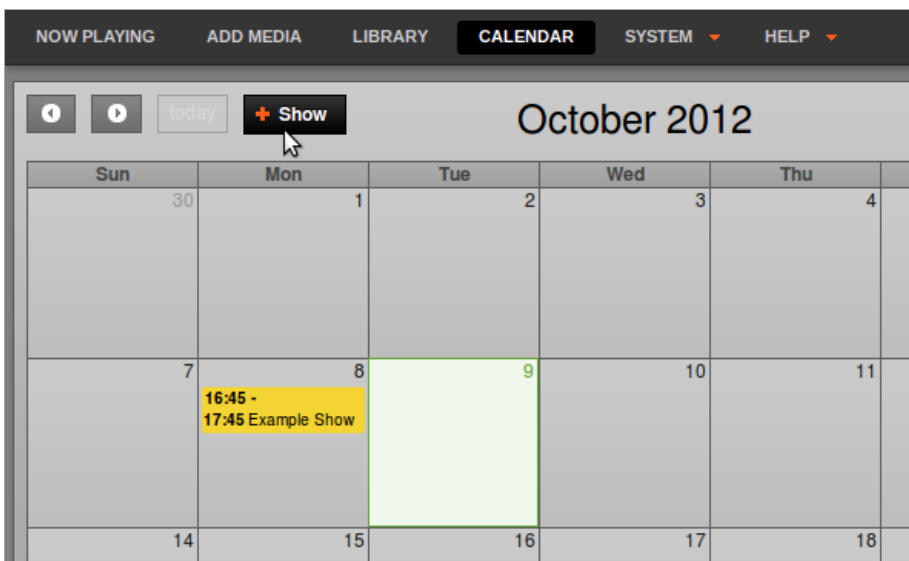

4. Set a name for your show in the **What** section of the box, and a date and time in the **When** section, then click the **+ Add this show** button (*Calendar*).
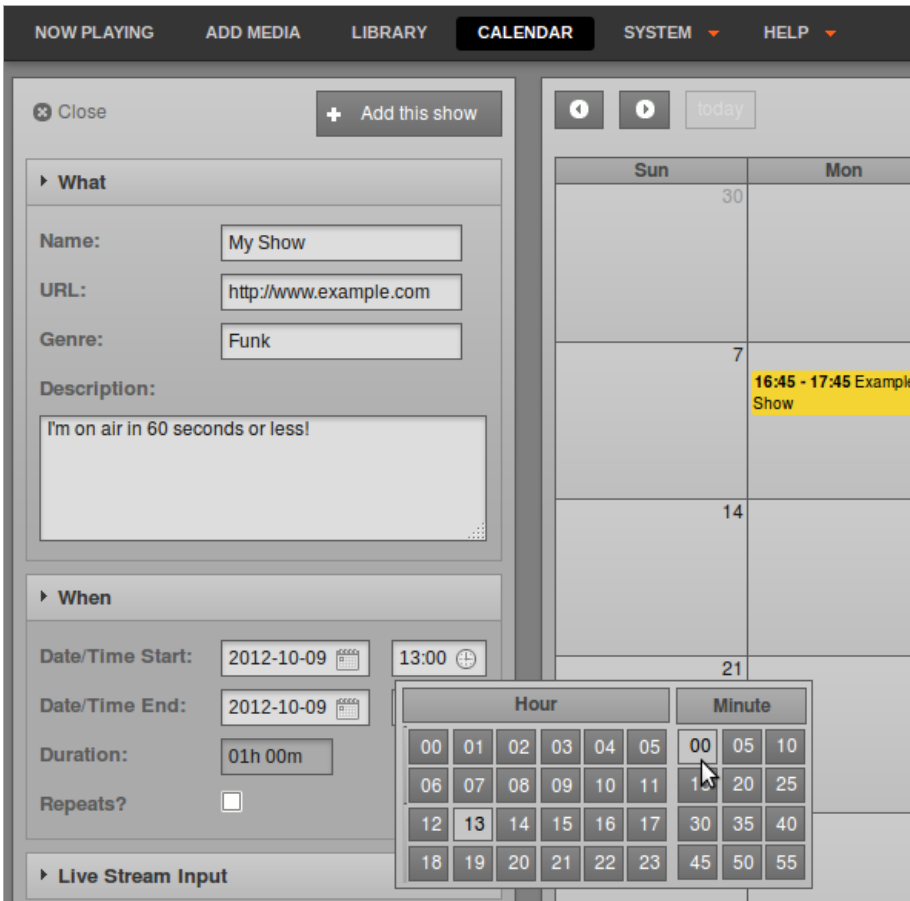
5. Add media to the new show by clicking your show in the **Calendar** and selecting **Add / Remove Content** (*Calendar*). Or if the show has already started, use the **Add / Remove Content** button on the Now Playing page (*Now Playing*).
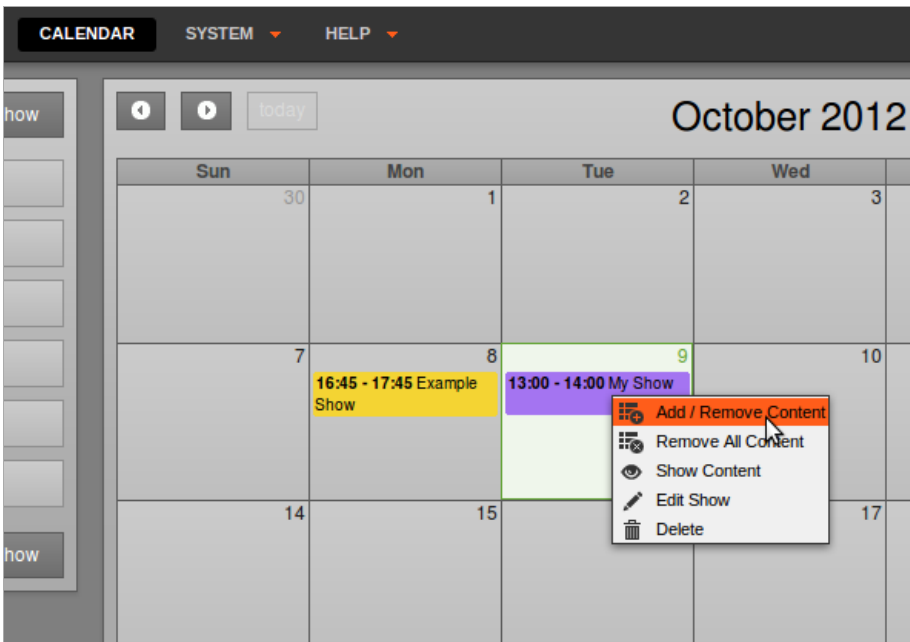
6. Click media in the search results on the left side of the pop-up window which will appear, and drag it into your show on the right side (*Now Playing*).
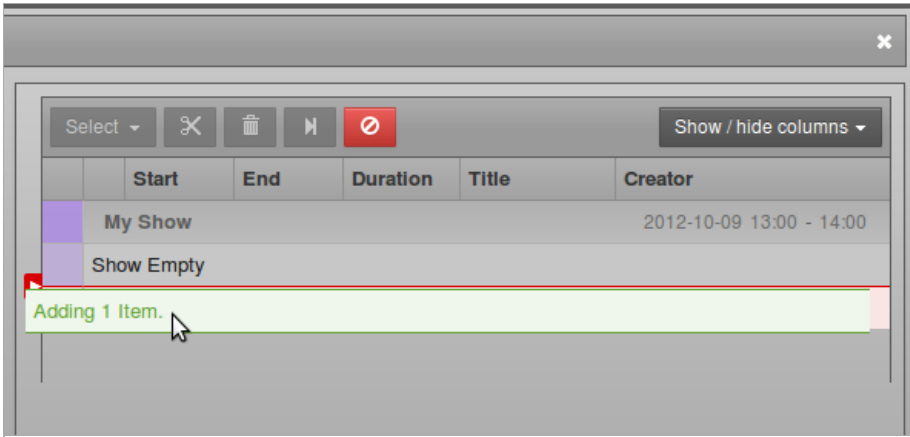
7. You're on air! Click the **Listen** button to hear available streams.
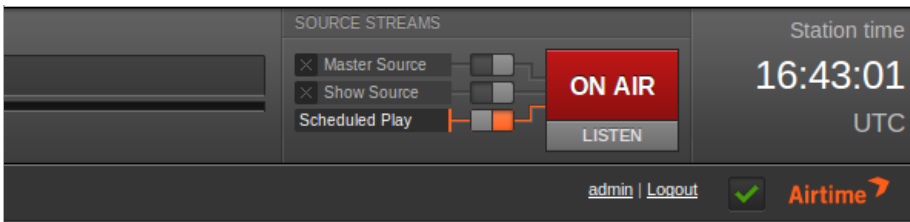
4. GETTING STARTED

If the Airtime server has a web browser installed, you can access the administration interface directly on that server by opening the address:

`http://localhost/`

If you have set up Airtime so that it can be accessed from other computers, you would use a domain name instead. For example:

`http://airtime.example.com/`

Airtime Pro users will have been given an *airtime.pro* domain name when they signed up.

You can log in for the first time with the user name *admin* and the password set during installation. Your browser should automatically focus on the **Username** field.
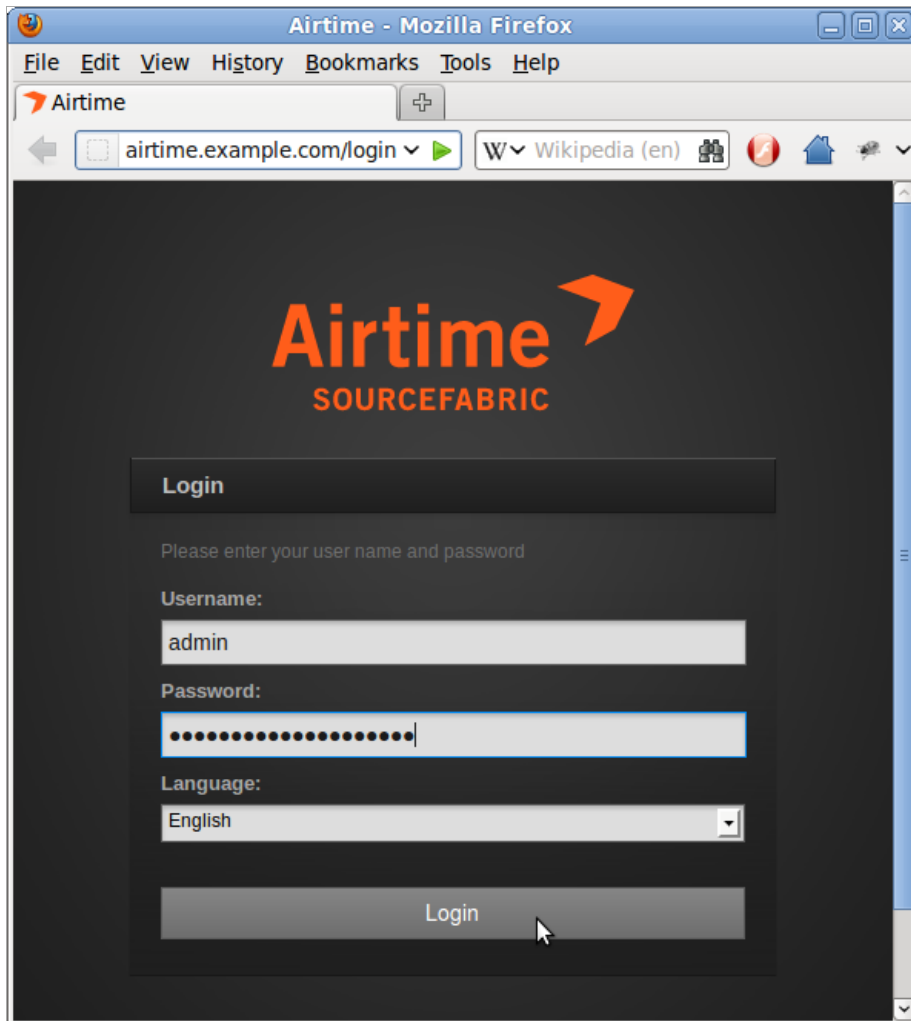
If you enter the password incorrectly three times, you will be presented with a reCAPTCHA challenge to prove that you are a human, and not a password-guessing robot. This feature helps protect your Airtime installation against brute force attacks.

If activated, you will see a link **Reset password** below the password field, which enables users to obtain a password reminder by email. See the chapter *Preferences* for configuration details.

The **E-mail** address you enter must match the address stored in the database for your **Username**.


REGISTER AIRTIME

After you have logged in as *admin* for the first time, a pop-up window will ask if you wish to send technical support data about your server to Sourcefabric.
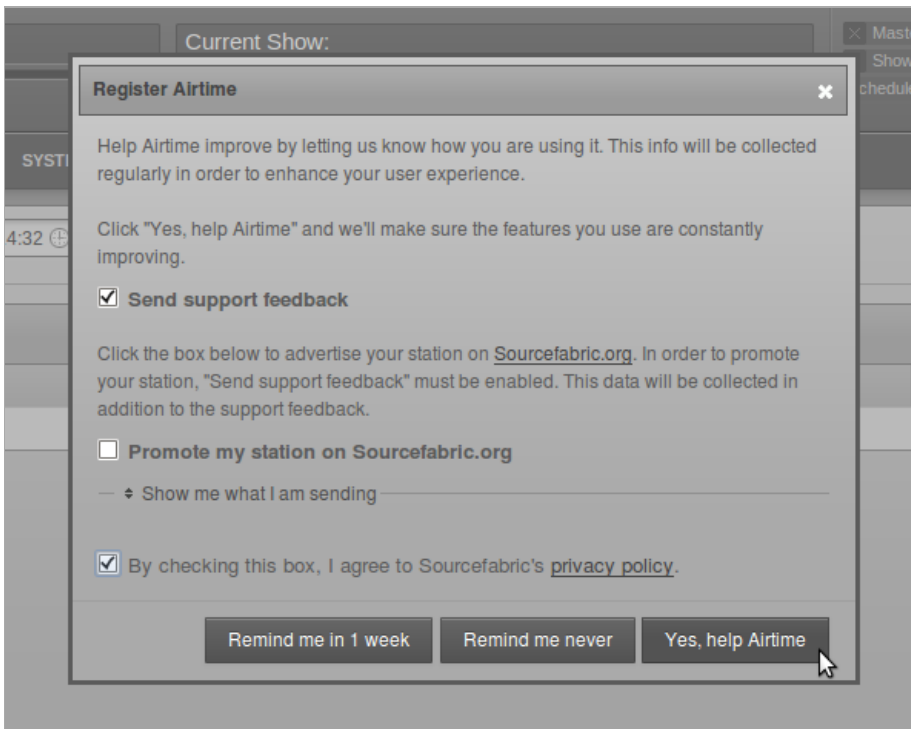
These details can be viewed by clicking on the **Show me what I am sending** link, which expands a scrolling window. The data helps Sourcefabric engineers resolve any problems with your Airtime installation, as well as count the number of installations worldwide.
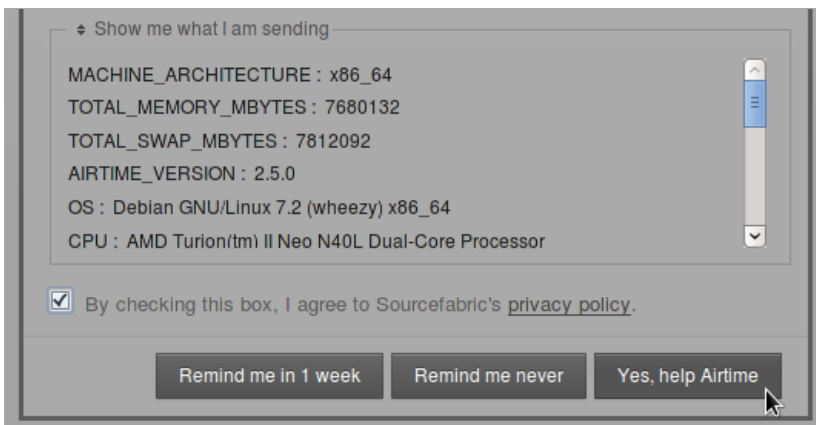
Sourcefabric has a privacy policy regarding data collection, which you can read by clicking the link to http://www.sourcefabric.org/en/about/policy/ further down. After checking the **Send support feedback** and **privacy policy** boxes, you can submit the data by clicking the **Yes, help Airtime** button.

This window also offers the opportunity to **Promote my station on Sourcefabric.org** (on the page http://www.sourcefabric.org/en/airtime/whosusing/) by checking the box. Fill in the form which will appear with some details about your station. The contact details are only requested for verification purposes, and will not be made available to the public. Click the **Browse** button to select a **Station Logo** image from the file manager on your computer.

THE MASTER PANEL

After the pop-up window is closed, you should now see the **Master Panel**, which is present at the top of every page of the Airtime interface. On the left hand side, the Master Panel displays the details of the **Previous** file played out, the current file playing (with an orange progress bar and time elapsed/time remaining), and the details of the **Next** file due to play. It also displays the name and scheduled time of the current show, with a blue progress bar.

Beneath this side of the Master Panel is the main navigation menu, with sub-menus labelled **Now Playing**, **Add Media**, **Library**, **Calendar**, **System**, **History** and **Help**. We'll be looking at the contents of these menus in the following chapters of this book.

On the right hand side of the Master Panel are the switches for the **Source Streams**, which enable you to switch from scheduled play to remote live sources, and back. (See the chapter *Stream settings* for details of this feature). The **On Air** indicator turns from dark grey to red whenever audio is being played out. Underneath this indicator is a **Listen** button, which opens a pop-up player that can be used to audition the available playout streams.

There is also a clock indicating the **Station time** and time zone. Beneath the clock and just to the left, the name of the user currently logged in is displayed, and there is the link for you to **Logout**.

Clicking the username link opens a page in which you can update your Airtime password, contact details, language and time zone preferences. Click the **Save** button once you have made the changes that you require.

On the right of the Logout link, clicking the green check mark opens a pop-up window with information about the version of Airtime installed. If your Airtime installation is not the latest version available, the green check mark changes to a green upgrade arrow. Should your Airtime installation get too far out of date, this arrow will change to a red exclamation mark.

CHECKING AN ICECAST SERVER

If you have installed an Icecast streaming media server, you can check that Icecast is running by opening its default server port of 8000 in your web browser. For example, on the server itself, you can use:

```
http://localhost:8000
```

or from another machine, using the domain name of the Icecast server:

```
http://streaming.example.com:8000
```

You should see the Icecast status page, with details of any connections that your Airtime server has made to this Icecast server. If you have only just installed Airtime, there may not be any media playing out yet.

5. SMARTPHONE JOURNALISM

Using an ordinary smartphone and Airtime, you can upload your audio reports to the studio library or put them straight on the air. Here's how...

The basic voice recorder app on your phone may not have all the features you need for journalism, but there are plenty of more suitable recording apps available for both iPhone and Android. For instance, searching for 'ogg recorder' in the *Play Store* on an Android phone yields plenty of alternatives. The following example uses *RecForge Lite* because of its configurable recording formats. This free app is limited to three minutes per recording, but there is an unrestricted version available at very modest cost.

In the Settings window of the app, set the recording format as Ogg Vorbis, 44.1KHz sample rate, mono. Disable automatic gain control to get a consistent recording level, rather than artificially boosting quiet sounds, which should result in less distortion. Under 'Limit folders browsing' give the app access to all of the SD Card storage of the phone.

Back in the main recording window of RecForge, create a directory such as *recordings*' under the */mnt/sdcard/media/audio/* path, which is normally used by the phone's music application. You can then make recordings using the big red button, watching the input level meter to ensure a good signal to noise ratio. The recorded files are automatically named with date and time stamps, but can be renamed in the app later by selecting the **Edit** feature. The latest recordings are listed in the main recording window; individual files can be auditioned by tapping on them.

Open the web interface of your station's Airtime server in the phone's browser, log in with your username and password, and tap **Add Media** on Airtime's main menu. After tapping the **Add Files** button, the phone will offer a choice of file to upload. By tapping **Select music track** you can browse the */mnt/sdcard/media/audio/* directory on the SD card filesystem, without requiring the installation of a file manager app on the phone.
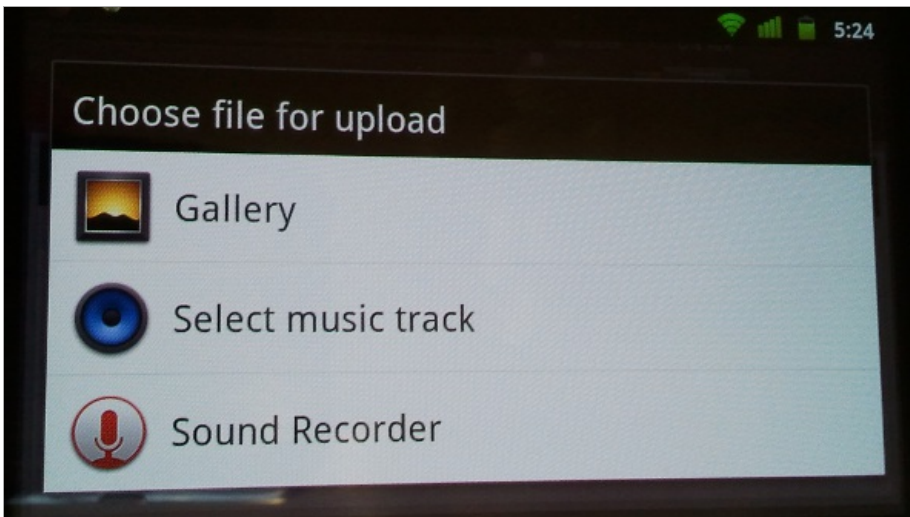
Tap one of the date-and-time-stamped files listed from the *'recordings'* directory to select it, then tap the **OK** button. The files are credited to '*Unknown artist*' because they have not yet been tagged with creator metadata.

21
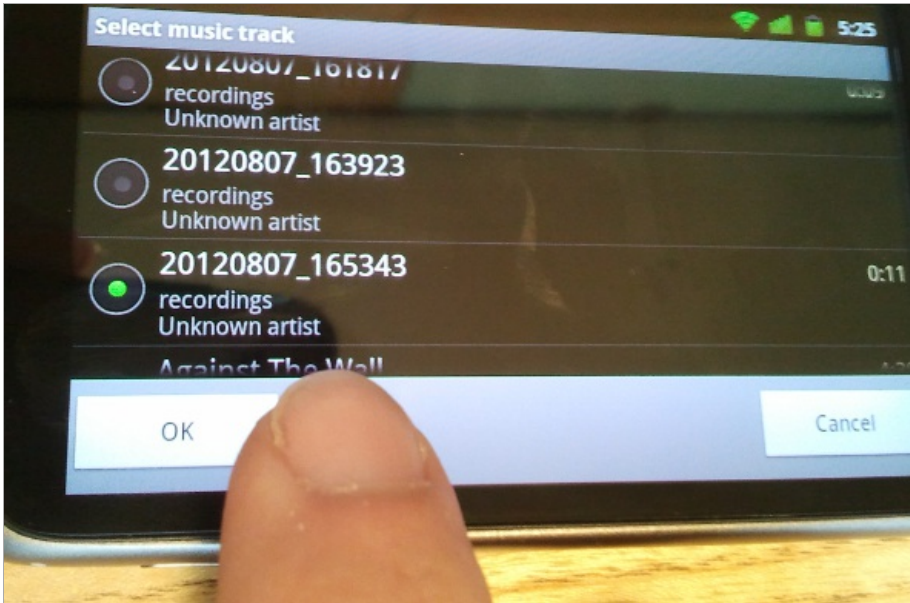
After tapping the **Start upload** button on the Add Media page, the files are now copied into the remote Airtime library.
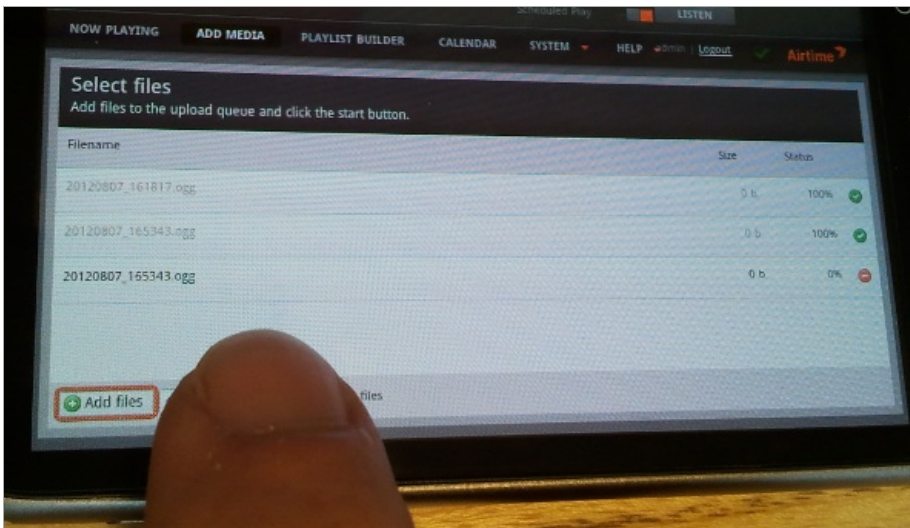
If you prefer, you can upload audio files to the *organize* folder on the Airtime server using an SFTP client, skipping the web browser step. See the chapter *Manage media folders* for details. Suitable client applications include **AndFTP** for Android: http://www.lysesoft.com/products/andftp/index.html

You can now add the uploaded files directly to a forthcoming show, use them to create smart blocks or playlists in the**Library**, or edit their metadata to make them easier to find in searches later. One of the limitations of mobile recording is that it isn't always convenient to enter large amounts of text metadata when you're on the move. So a phone app that tagged audio recordings with creator, location and other relevant metadata would be very useful.

GOING LIVE

Mobile apps which enable you to stream directly into Airtime's live rebroadcast feature include**KoalaSan** for iOS: http://koalasan.net/ and **BroadcastMySelf** for Android: https://play.google.com/store/apps/details?id=sa.broadcastmyself DemoIcesShout for Android also shows promise, although it can only stream a static file, rather than stream from a microphone: http://droidtools.sourceforge.net/content/icecast-client-android

A live workflow does of course depend on a steady network connection between the phone and Airtime server. Because streaming media on a regular basis can use a lot of bandwidth, it is advisable to use WiFi rather than mobile data (3G/4G) whenever possible, especially when roaming across networks.

KoalaSan can stream input from either the built-in microphone or the external jack of the iOS device. An example configuration for the **Settings** tab might be:

```
Name: Airtime
Server: airtime.example.com
Port: 8003
Mountpoint: mastermount
Stream name: Airtime
Description: A live stream to Airtime
Genre: News
```

```
Username: masterstreamer
Password: masterpassword
Codec: mp3
Bitrate: 128
Protocol: Icecast2
Stereo: off
AudioBus: off
Dump: on
Broadcast: on
```

You can start streaming from the iOS device by tapping the start button in KoalaSan's **Stream!** tab.

To configure BroadcastMySelf, tap the gears icon in the top left corner to enter the settings page, then tap **Output Settings**. This action opens the **Icecast Settings** page, where example settings could be:

```
Hostname
airtime.example.com

Port
8003

Path
/mastermount

Username
masterstreamer

Password
masterpassword

MP3 Encoder Bitrate
128 Kbit/s

Metadata Template
_song__artist

Streaming Buffer Size
MP3 buffer size X 8
```

You may also wish to configure the **Tracks Folder** setting, which enables you to stream static files from the storage of the device. Depending on where you keep your audio files, this could be a path such as:

```
/mnt/sdcard/media/audio/
```

To connect the Android device to the Airtime server, tap the square button in the lower middle of the main interface. The connection indicator on the button should turn blue, and the message *connected* should be shown. Push up the fader on the left to increase input level from the internal microphone, or push up the fader on the right and tap one of the named files in the upper right window to stream it. Tap either of the **LOCK** buttons to prevent the stream input level being changed accidentally.

When the connection is made, Airtime output will fade from scheduled play to the stream from your mobile device. See the chapter *Stream settings* for more details of connection preferences.

THE SYSTEM MENU

6. PREFERENCES
7. USERS
8. MEDIA FOLDERS
9. STREAM SETTINGS
10. SUPPORT FEEDBACK
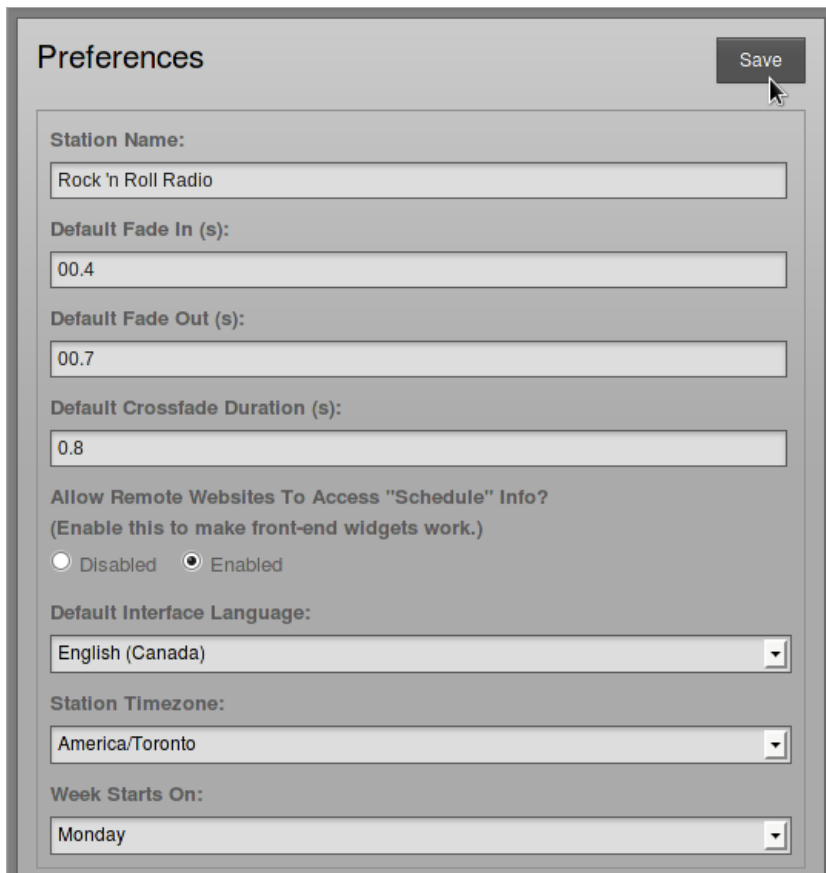**11. STATUS**
12. LISTENER STATS

6. PREFERENCES

On the System menu, click **Preferences** to set your **Station Name**. This text is shown in your web browser's title bar when your station staff are logged into Airtime, and optionally in stream metadata.

Initially, the **Default Fade In** and **Default Fade Out** times for automated fades are set to half a second, and the **Default Crossfade Duration** is set to zero seconds. Custom fade and crossfade times can be set for adjacent items in a playlist or static smart block. See the chapter *Library* for details.

You can enable live, read-only access to the Airtime schedule calendar for your station's public website with the **Allow Remote Websites to Access "Schedule" Info?** option, if you wish. (There is more about this feature in the *Exporting the schedule* chapter, in the *Advanced Configuration* section of this book).

The **Default Interface Language** drop-down menu sets the default localization for your Airtime instance, and the **Station Timezone** drop-down menu can be used to display local time at your station. Airtime stores show times internally in UTC format (similar to *Greenwich Mean Time*), but displays local time for the convenience of your station staff. You can also set the day of the week that you wish to start your station's weekly schedule on, which defaults to Sunday. Then click the **Save** button.

Individual Airtime users can choose another interface localization when they log in, or set personal preferences for localization and time zone by clicking their username on the right side of the menu bar.
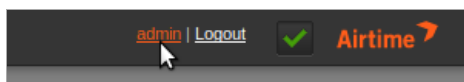
EMAIL / MAIL SERVER SETTINGS

The **Enable System Emails (Password Reset)** box is not checked by default, because although enabling staff users to retrieve their lost password for themselves may be convenient, this option does have security implications. If the email account of the staff user is compromised, an attacker could very easily gain access to your Airtime server by clicking the **Reset Password** link on the login page.

If you wish to enable the password reset option, your Airtime server can have an SMTP mail program installed. (See the chapter *Preparing the server* for details). Alternatively, you can check the box **Configure Mail Server** and specify the details of an external SMTP server, which would usually require authentication if it is outside of your local network.

Whether SMTP runs on the same server as Airtime or on another server, you should enter a valid email address into the **Reset Password 'From' Email** field to help ensure emails are delivered correctly. Then click the **Save** button.

SOUNDCLOUD SETTINGS

If your station has a SoundCloud account (on http://soundcloud.com), you may wish to upload files and show recordings to this service, in order to broaden distribution and balance the bandwidth load with your own servers. Click the small black triangle next to **SoundCloud Settings** to show the options available. Check the **Enable Soundcloud Upload** box, then optionally check the **Automatically Upload Recorded Shows** box. You will also need to enter your SoundCloud login email address and password, and the tag metadata that SoundCloud will use to categorize your show recordings.

Check the box **Automatically Mark Files "Downloadable" on SoundCloud** if you wish to enable this option. You can also set a default genre, track type and copyright license here, including public domain, all rights reserved, or one of the *Creative Commons* licenses (see http://creativecommons.org). Then click the **Save** button again.

SoundCloud Settings

☑ **Enable SoundCloud Upload**

    ☑ **Automatically Upload Recorded Shows**

    ☑ **Automatically Mark Files "Downloadable" on SoundCloud**

SoundCloud Email *(Required)* :

soundcloud@example.com

SoundCloud Password *(Required)* :

••••••••••••••••••••••••••

SoundCloud Tags: (separate tags with spaces)

rock rockabilly roots blues radio

Default Genre:

Rock

Default Track Type:

Recording

Default License:

Creative Commons Attribution Share Alike

Save

Please note that like most online distribution services, SoundCloud terms of service require you to have copyright in, or permission for Internet distribution from the copyright holder of, any media that you upload. Commercially released music files uploaded to SoundCloud are likely to be automatically removed from your station's SoundCloud page.

7. USERS

If your Airtime server is accessible from the public Internet, it will not be secure until you set your own, strong password for the all-powerful *admin* user. Should the password for the *admin* user still be set to *admin* or some other easily guessed word, you should set a new password immediately, via the **Users** page on the System menu. Only users with the **User Type** of *Admin* can see the **System** menu when they log in.

SETTING PASSWORDS AND CONTACT DETAILS

On the left side of the **Users** page, click on *admin* in the table. The details for this user will appear in the box on the right side. To begin with, on a freshly installed Airtime server, the *admin* user only has a **Username**, **Password** and **User Type** set.

To set a new password for the *admin* user, enter a string of letters and numbers (as long and as varied as is practical) into the **Password** field, then again into the **Verify Password** field to guard against typing errors. Keeping this password secret is essential for the smooth running of your station, because it enables access to all scheduling and management features. You can also enter other details for the *admin* user account on this page, including full name and contact details. Then click the **Save** button.

ADDING USER ACCOUNTS

To add further user accounts to the system, one for each of your station staff that need access to Airtime, click the **New User** button with the plus icon. Enter a user name, password and contact details, and then select the **User Type** from the drop down menu, which can be *Admin*, *Program Manager*, *DJ*, or *Guest*. The difference between these user types is:

- An *Admin* (station manager) has read and write access to all the features of Airtime. This role should be reserved for trusted staff members only. If you give the *Admin* role to too many people, there could be arguments at the station!
- A *Program Manager* has write access to the entire broadcast schedule, but cannot see the **System** menu and therefore cannot adjust Airtime preferences, manage user accounts, change media folder or stream settings, check the server status, or see the Listener Stats. A *Program Manager* can view the **History** menu and the Playout History page, but cannot edit History Templates.
- A *DJ* (presenter) only has write access to features related to the specific shows assigned to them by an *Admin* or *Program Manager*. The *DJ* can read data for other shows in the **Calendar**, but not write to them. This read access to other shows helps staff plan their own shows, for instance avoiding the scenario where two DJs play the same music by coincidence. Like a *Program Manager*, a *DJ* cannot see the **System** menu when they log in, and cannot edit History Templates.
- A *Guest* can log in to read the forthcoming schedule or playlists, but has no write permission for any feature. *Guest* users cannot see the **Add Media**, **Library**, **System** or **History** menus when they log in. The group of guest users might include the station accountant, or the show assistants.

EDITING OR DELETING USER ACCOUNTS

New user accounts that you add will be shown in the table on the left side of the **Users** page. If you have a large number of users on the system, you can use the search tool above the table (which has a magnifying glass icon) to identify specific user accounts. Click the white chevrons in the table headings to sort the search results by **Username**, **First Name**, **Last Name** or **User Type**.

To edit a user account, click on that user's row in the table, change the user's details in the box on the right side, and then click the **Save** button. To remove a user account, click the small **x** icon to the right side of its row in the table. You cannot delete your own user account, and usernames cannot be changed once created.

If the Airtime server is running in demo mode, user accounts cannot be created or updated. See the chapter *Host configuration* for details of how to enable demo mode.

UPDATING YOUR OWN ACCOUNT

Users can update their own password, and their contact, language and time zone details, by clicking their username on the right side of the main menu bar, next to the **Logout** link.
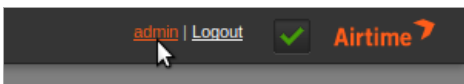
This action opens a separate, individual page which the user can update regardless of their **User Type** and access to the **System** menu. Click the **Save** button to update your account.

8. MEDIA FOLDERS

Airtime's media library is inside the *srv/airtime/stor/* folder on your server, by default. In **Media folders** on the **System** menu, you can change this default location, or add extra folders to be watched by Airtime.
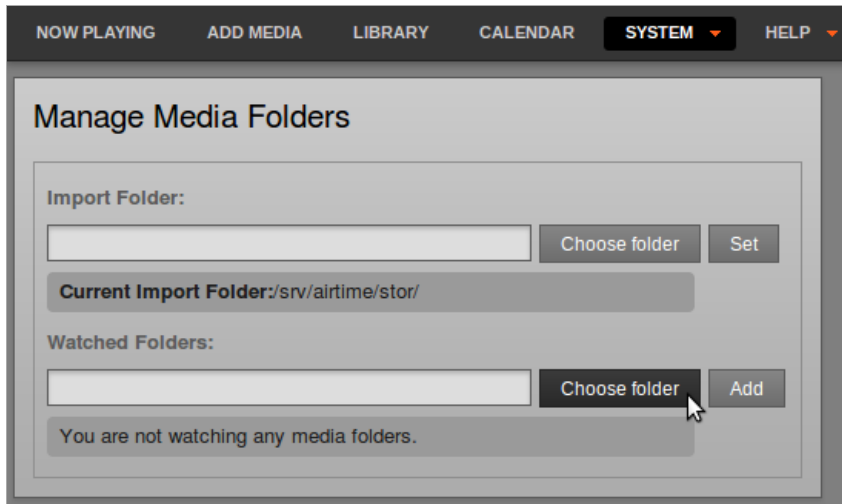
If you specify a network drive as a media folder and that network becomes disconnected for any reason, Airtime's **media monitor** will flag the files as 'missing' in its database until the drive is reconnected. If those files are still missing at the time of the show, the broadcast output could be silenced.

For the same reason, if a removable drive (such as a USB memory stick or MP3 player) is specified as a watched folder, that drive has to be present and powered on until the corresponding show time, so that the Airtime playout engine can download the files needed for the show. If your station staff use removable drives to store media files, it is safer to use the **Add media** page of the Airtime administration interface or the **airtime-import copy** command to copy the files to the main storage server. See the chapters *Add media* and *Using the airtime-import script* for more details.

The metadata for new media files you add to the *organize* folder or a watched folder will be automatically imported into the Airtime database. The *organize* or watched folders can be exported to computers on the local network. This would enable dragging and dropping of media uploads using the file managers on the desktop computers at your studio.
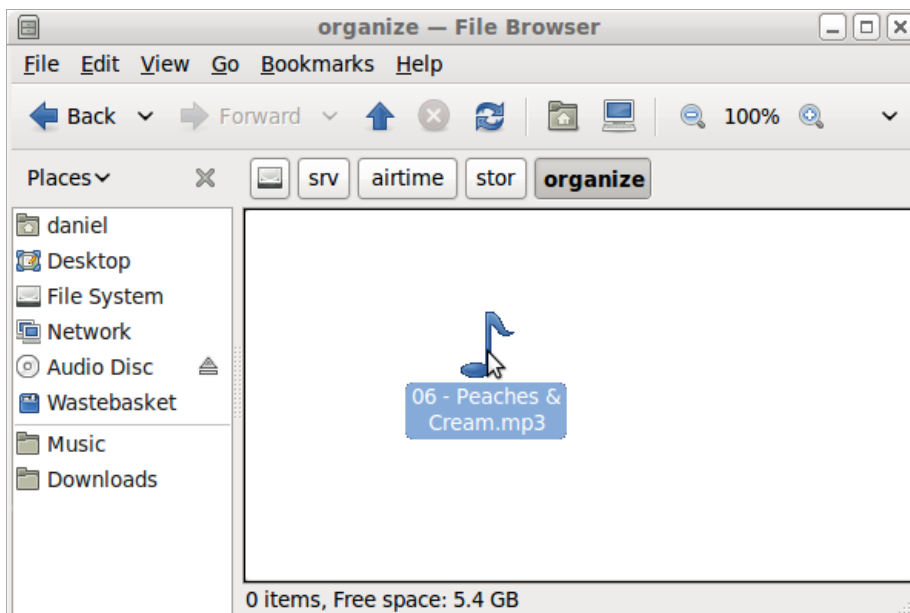
A file dropped into the *organize* folder will appear to vanish as the metadata is read, and the file is moved to the correct location under the *imported* directory, according to its creator and title. This means you can find files for download and editing using a file browser connected to your server, as well as in the **Search** table in the **Library**. This location could be under a filesystem path such as */srv/airtime/stor/imported/Beck/Midnite Vultures/* in the screenshot below.
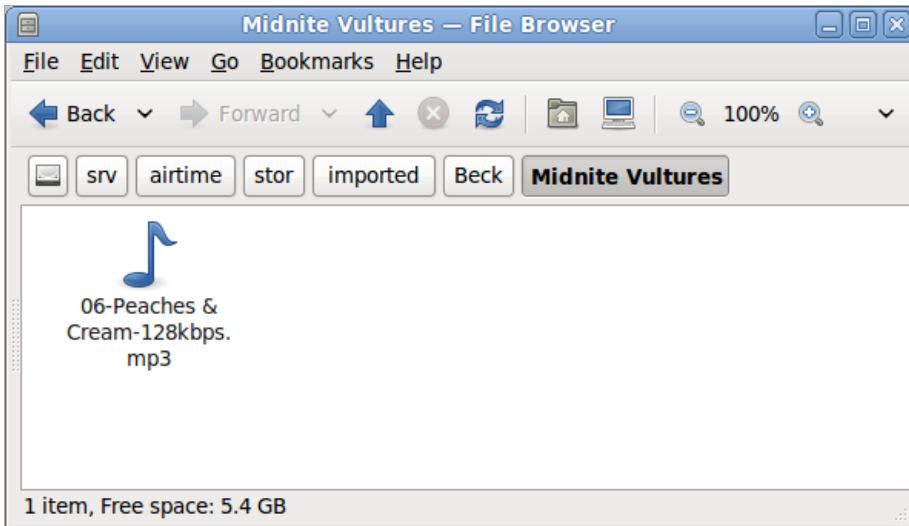
If a media file is corrupted and cannot be played by Liquidsoap, Airtime will move the file to the *problem_files* folder in the storage archive, and its metadata will not be added to the database. This feature ensures that all files are tested for encoding errors or upload corruption before playout time.

Files in watched folders are not moved into the main Airtime storage folder. However, just like in the main storage, files deleted from a watched folder will be automatically flagged as missing in any show that they are part of.

Edits to your stored and watched media files are noticed by Airtime. If you edit any file known to the database and change its duration, Airtime will automatically adjust the duration of playlists, smart blocks and shows that the file is included in.

CHANGING THE STORAGE FOLDER

To change Airtime's storage folder, click the upper **choose folder** button. In the pop-up window that opens, double-click on the folder names to select the folder that you require. Then click the **Open** button to open that folder.
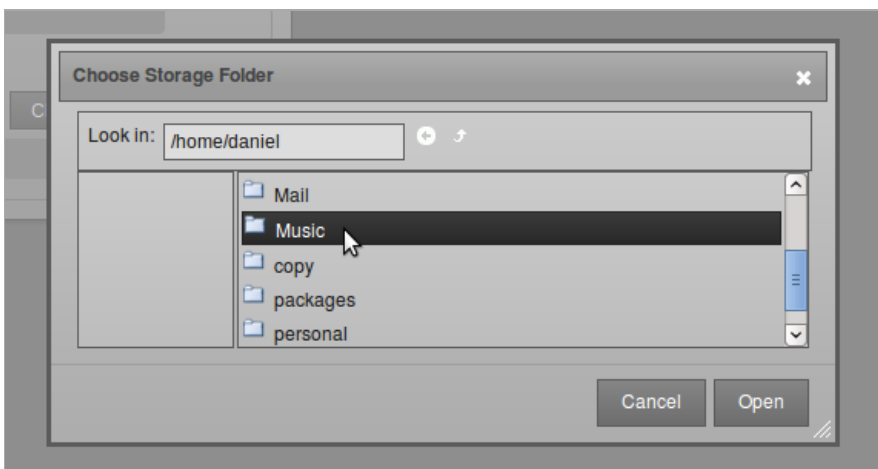
Back on the **Manage Media Folders** page, click the **Set** button to change the storage folder. Airtime will ask if you are sure about this action. Click the **OK** button to confirm your choice.
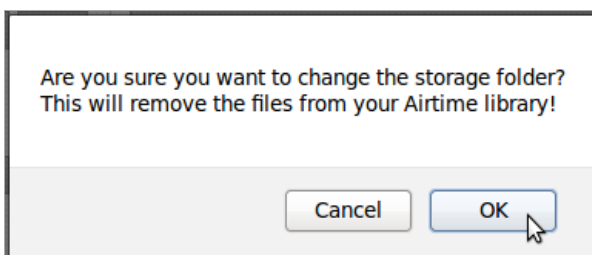
The **Manage Media Folders** page will now display the new storage location.

The storage folder cannot be changed while a file import is in progress. If you attempt to do this, an error message will be displayed.

WATCHING A FOLDER

Under **Watched Folders**, click the lower **choose folder** button, open the folder you require, and then click the **Add** button. You can add as many watched folders as you require.

To remove a watched folder, click the small **x** on the right side of its row in the list. Again, you will be asked to confirm if you are sure about the action.

9. STREAM SETTINGS

You can configure direct Icecast and SHOUTcast streams and sound card output by clicking **Streams** on the **System** menu.

At the top left of the **Stream Settings** page are global settings including **Hardware Audio Output**, which enables playout from the default sound card on the server, if one is fitted. The default **Output Type** of *ALSA* on the drop-down menu will be suitable for most servers with a sound card. If not, you have the option to choose from other Liquidsoap interfaces available, such as *OSS* or *PortAudio*.

The second checkbox under Global Settings enables the sending of **Icecast Vorbis Metadata** with direct streams. This setting is optional, because some media players have a bug which makes them disconnect from Ogg Vorbis streams when an Icecast server notifies the player that a new track is starting.

The **Stream Label** radio button allows you to set the metadata that will be sent with direct streams; *Artist* and *Title*, *Show*, *Artist* and *Title*, or *Station name* and *Show name*.

The **Off Air Metadata** field configures the text that will be sent to any configured streaming servers, and from there on to media players, when Airtime is not streaming any output.

Below this is the **Enable Replay Gain** checkbox which, if checked, automatically adjusts Airtime's output level to an average of -14dBFS, based on prior analysis of the level of individual files in the **Library**. This feature helps guard against unwanted changes in Airtime's output level caused by the typical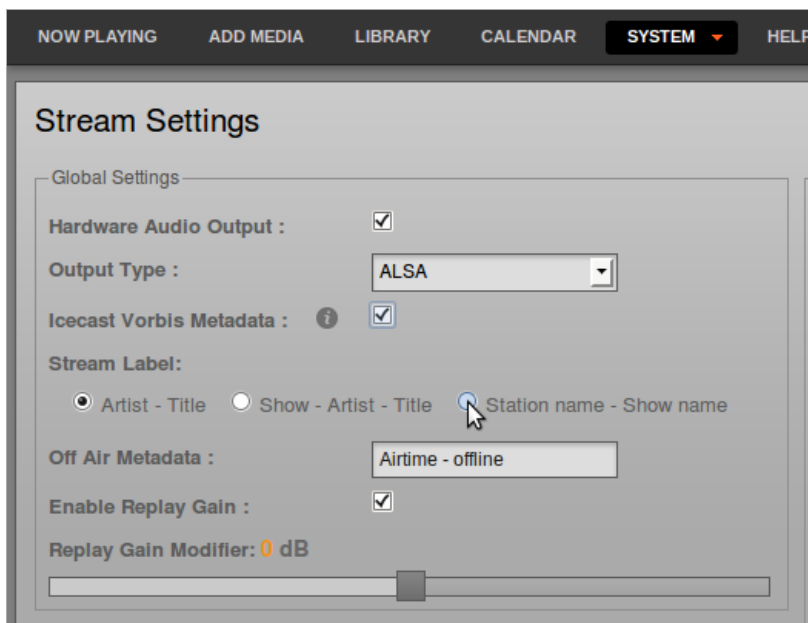 variation in average level of recordings made in different studios at different times. The correction value for each file can be viewed by enabling the ReplayGain column in the **Library**.

If Replay Gain is enabled, output level can be adjusted up to 10dB louder or quieter from the reference level of -14dBFS by clicking and dragging the **Replay Gain Modifier** control. Increasing level too far is likely to result in clipping (distortion) since the output level is adjusted to be greater than that of the original files. This scenario is indicated by amplitude correction factors greater than 1.0 in the Liquidsoap log file.

Click the **Save** button on the right side of the page to save any changes that you have made to global settings. If you are only using the hardware audio output, and will not be streaming to or from Airtime, you can leave the **Stream Settings** page after clicking the **Save** button.

INPUT STREAM SETTINGS

On the lower left side of the Stream Settings page you can configure remote live input streams from DJ programs such as **Mixxx** or **IDJC**, or smartphone applications used by broadcast journalists. Airtime supports two types of live input stream; the **Show Source**, which enables a specific person to stream in during their own show, and the **Master Source**, which can override the Show Source if necessary. If neither type of live input is available, Airtime will fall back to **Scheduled Play** (playlists, smart blocks, remote streams and files scheduled in Airtime, in advance of or during a show).

The **Auto Switch Off** and **Auto Switch On** checkboxes enable playout to be switched automatically to the highest priority source whenever an authenticated input source disconnects from or connects to Airtime, respectively. The field **Switch Transition Fade** sets the length of the audio fade as scheduled playout is switched to a remote input source, and back.

Each type of input stream requires a username and password before the remote broadcaster can connect to Airtime. The **Master Username** and **Master Password** can be set in the Input Stream Settings box, while the authentication for individual Show Sources is set up in Airtime's schedule calendar. See the *Calendar* chapter for details.

Input streams must have a **Port** for the remote broadcaster to connect to, which should be a number in the range from 1024 to 49151. If you have the Icecast or SHOUTcast streaming server running on the same machine as Airtime, you should avoid using port 8000 or 8001 for either type of Airtime input stream. This is because both Icecast and SHOUTcast use port 8000, and SHOUTcast also uses port 8001. If the usernames and passwords were similar, remote broadcasters might accidentally connect to the streaming server directly, bypassing Airtime.

To avoid further confusion, it is also recommended that you set a different **Mount Point** (the specific filename for broadcasters and listeners to connect to) from that used on your public Icecast or SHOUTcast server.

If your Airtime server is behind a firewall, and you wish remote broadcasters to connect input streams to it across the public Internet, you may need to click the **Override** link to set a **Connection URL** which is available from outside your local area network. This URL might be based on the domain name assigned to the router which forwards the appropriate port to your Airtime server. Then click **OK** to save the new connection URL.

SWITCHING INPUT STREAMS

In the Master Panel, available input source streams are shown with an orange line connecting the source to the switch, which you could think of like a patch cable connecting a source to a broadcast mixer. When that switch is active, another orange line connects the switch to the On Air indicator, like a patch cable connecting a mixer to a transmitter.
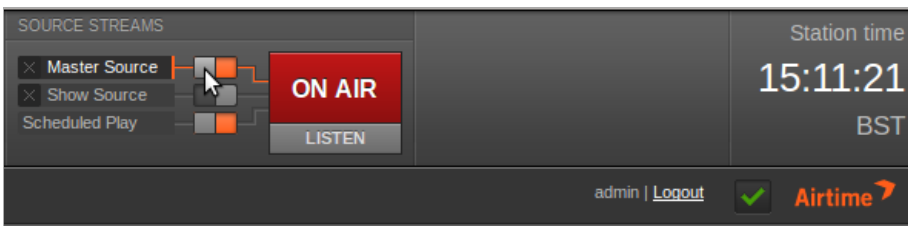
If you have checked the **Auto Switch On** box in the Stream Settings page, the **Master Source** switch will move automatically to the active position, on the left, when an authenticated master source connects to Airtime. Otherwise, you can activate the switches manually by clicking the left side of each switch, or deactivate them by clicking the right side. The switches do not have to be dragged with the mouse, in the way that a switch on a hardware mixer would be pushed sideways. **Show Source** live input streams and **Scheduled Play** can be manually activated or deactivated in the same way.
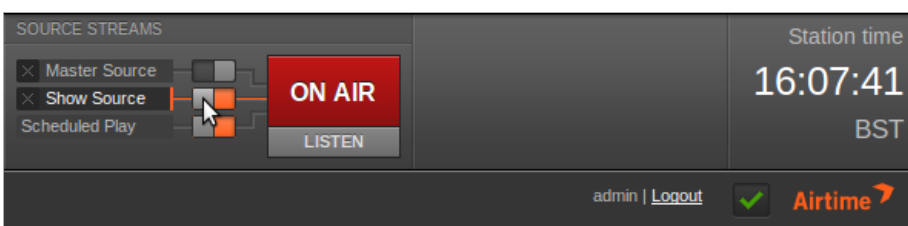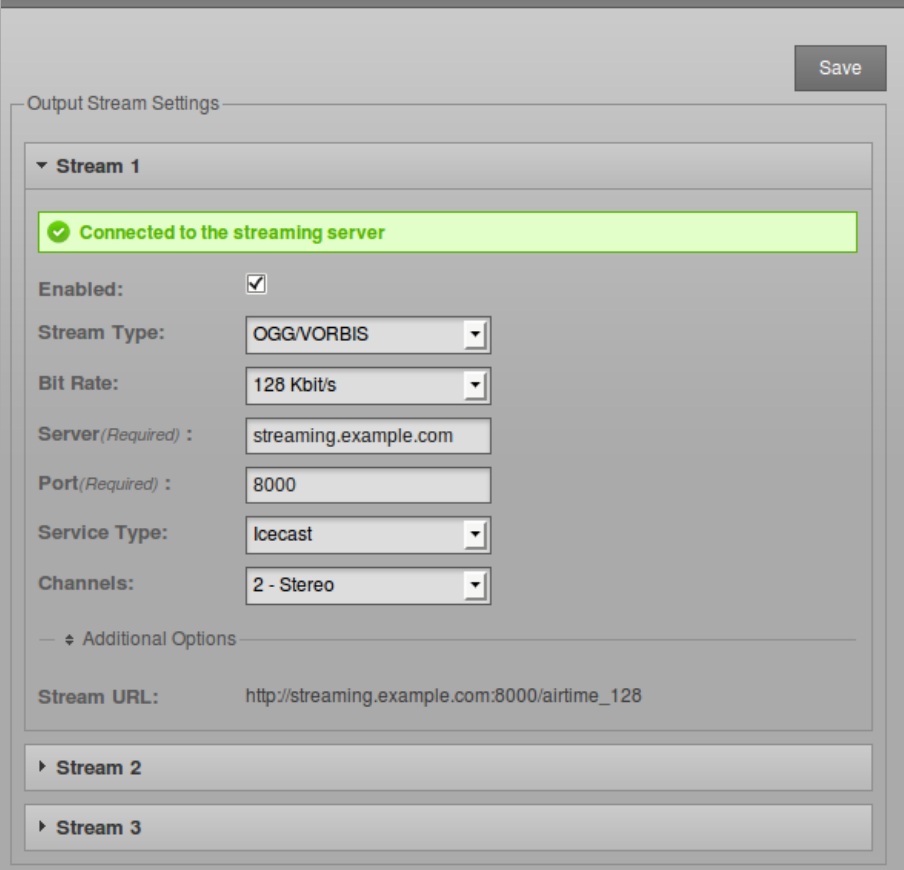
To force disconnection of a live remote source, for example when the remote input source has crashed and is no longer sending audio data, click the **X** icon to the left of the source name.

OUTPUT STREAM SETTINGS

On the right side of the page, you can configure up to three independent output streams with different bit rates or formats, and send these streams to different Icecast or SHOUTcast servers. By default, only Stream 1 is enabled, with this stream being sent the Icecast server configured at installation time.

To configure another stream, click the bar with the stream number to expand its box, and make sure **Enabled** is checked. Enter at least the streaming **Server** IP address or domain name, and **Port** details. The default port for Icecast and SHOUTcast servers is 8000.

Click **Additional Options** to expand a box in which you can enter the usernames, passwords and metadata to send to the streaming server. The default **Username** for Icecast servers is *source*, and if this the name in use on your streaming server, you can leave this field empty. The **Admin User** and **Admin Password** settings are optional, and are used to query the streaming server for audience numbers by the **Listener Stats** page on the **System** menu.

You can also set the specific **Mount Point** that listeners will connect to here. Then click one of the **Save** buttons in the upper or lower right corner of the page to update the Airtime server's settings.

Airtime supports output to Icecast in Ogg Vorbis, Ogg Opus, MP3 and AAC formats. When selecting a SHOUTcast server from the **Service Type** drop-down menu, you are restricted to using MP3 or AAC formats only, so the choice of Ogg Vorbis and Opus formats is greyed out in the **Stream Type** drop-down menu. The SHOUTcast username for stream sources is fixed, so you do not need to enter this value under **Additional Options**, but you will usually have to enter a password.

Any connection problems between Liquidsoap and Icecast or SHOUTcast are shown on the Stream Settings page. For example, if you enter the wrong password, you will see an *Authentication Required* error message. To fix this, enter the correct password in the **Additional Options** box, and click the **Save** button. If the streaming server is down for any reason, or you have entered an incorrect **Server** name or **Port** number, you will see the message *Can not connect to the streaming server.*

▸ Stream 2

❌ **Can not connect to the streaming server**
could not connect to host: connection timeout

Enabled: ☑

Stream Type: MP3

Bit Rate: 128 Kbit/s

Server *(Required)* : shoutcast.example.com
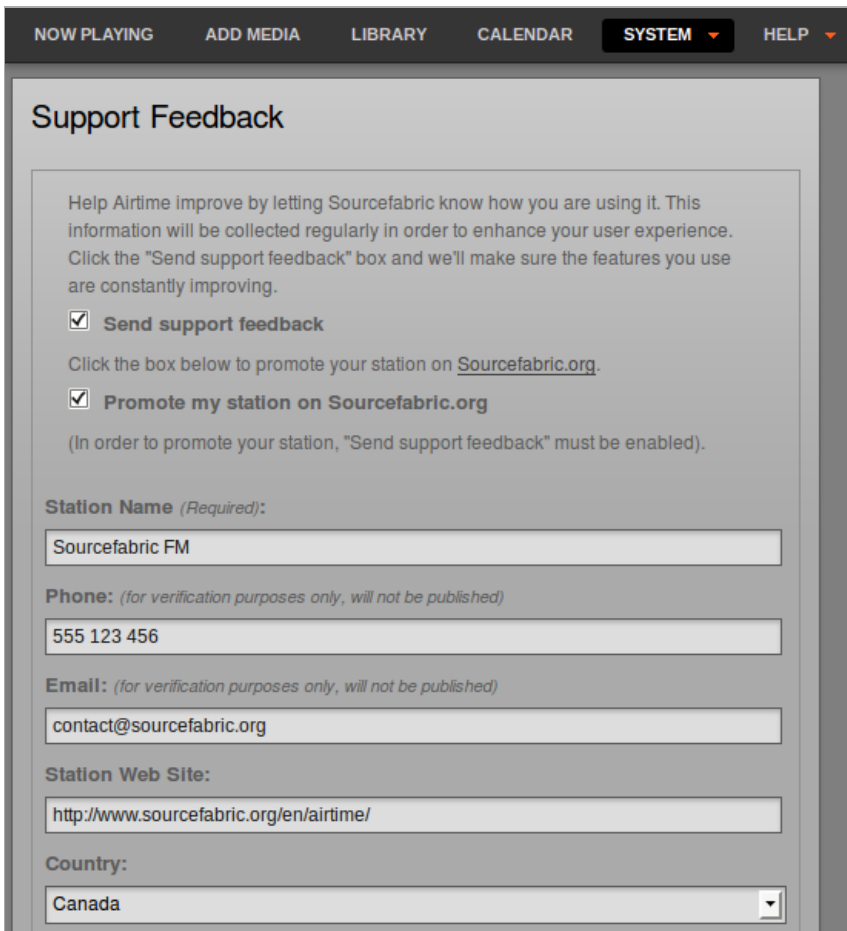
10. SUPPORT FEEDBACK
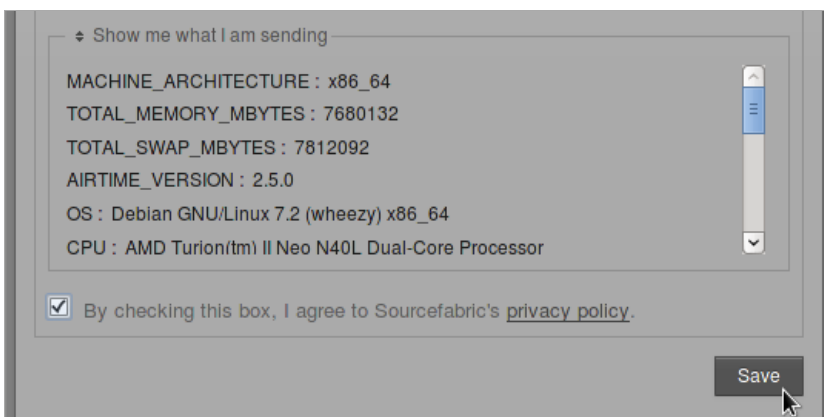
If you did not already register Airtime when you installed it, as shown in the *Getting Started* chapter, you can click **Support Feedback** on the System menu to display Airtime's automated feedback options. Check the **Send support feedback** box in order to post technical details about your Airtime installation to Sourcefabric, over the Internet. These details help Sourcefabric diagnose any problem that you might be having with your Airtime system.

You may also wish to send details of your station to Sourcefabric, so that your station has the opportunity to be promoted with other Airtime users on the http://www.sourcefabric.org website. This feature also helps Sourcefabric target its support services to the countries where they are needed most. Check the box **Promote my station on Sourcefabric.org** and fill in the details of your station. You can upload a station logo file (of up to 600 x 600 pixels) from your desktop computer by clicking the **Browse** button.

Click on the **Show me what I am sending** link to expand a box which displays the technical data being returned to Sourcefabric. The data is collected according to the Sourcefabric privacy policy (http://www.sourcefabric.org/en/about/policy/) which you are required to agree to before you can submit the information. If you have already checked the box to indicate your acceptance of the privacy policy, a link to the policy will be displayed instead.

11. STATUS

On the **System** menu, the **Status** page provides an overview of the health and resource usage of the various services that make up an Airtime system. If all is well, you will only see green check mark icons in the **Status** column. This page also shows how much **Disk Space** you have used on the disk partition containing the main Import folder, as well as any disks or partitions with watched folders.

If any of the check mark icons in the **Status** column have changed to a red warning sign, contact your system administrator for assistance. (The chapter *Troubleshooting* contains some tips). Airtime will do its best to restart any failing services, but sometimes manual intervention may be required; for example, in the case of hardware failure.

If you have run out of storage space, an Airtime user with *admin* privileges could log in and delete media files that are no longer required from the **Library**. Alternatively, you could move some files to a watched folder on another disk, or ask your system administrator to install additional storage capacity.

12. LISTENER STATS

The Listener Stats page on the System menu shows graphs of listener connections to the configured streaming servers for the selected date and time range. On the right side, a green **Status** indicator shows **OK** if the connection to the streaming server is active.
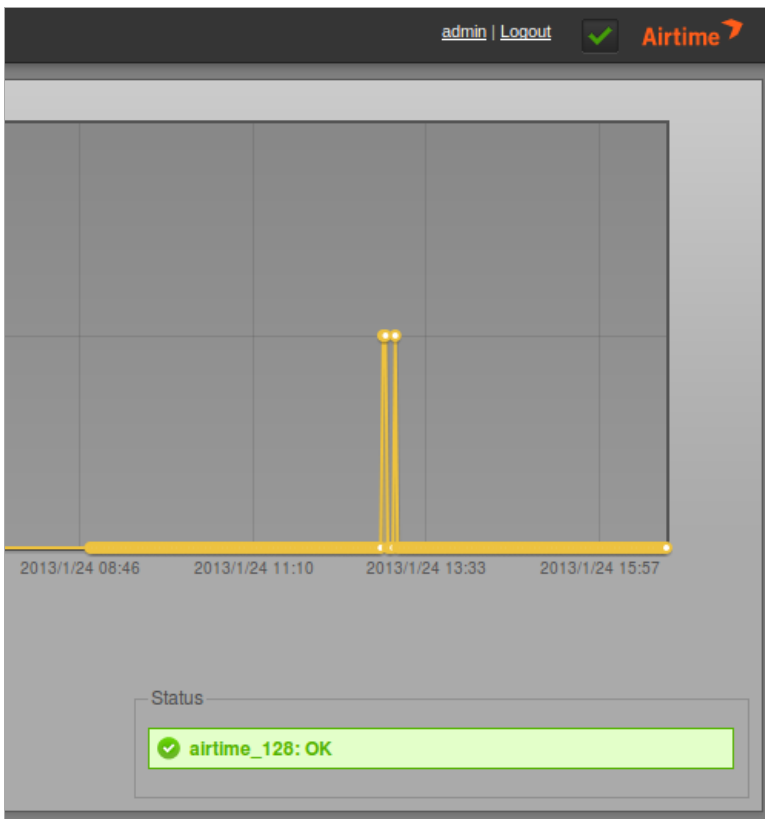
If the status indicator is red, check that the **Admin User** and **Admin Password** settings are correct under **Additional Options** for the named mount point, such as *airtime_128*, on the **Streams** page of the **System** menu.

By default, statistics for the last 24 hours of streaming are shown. To change this date and time range, click the calendar and clock icons in the lower left corner of the page, then click the magnifying glass icon.
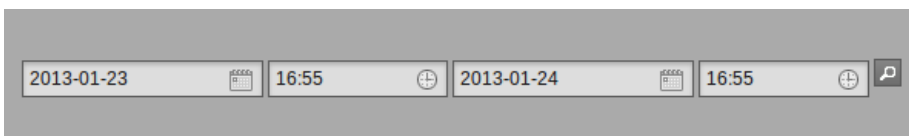
To choose which particular streams should have statistics displayed, click the check boxes for the individual colour-coded mount points, just below the graph.
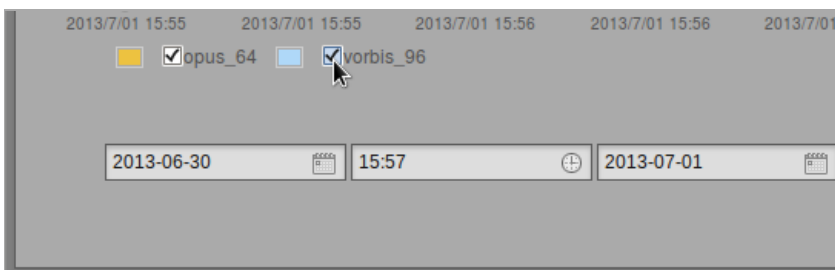
THE MAIN MENUS

13. NOW PLAYING
14. ADD MEDIA
15. LIBRARY
16. CALENDAR
17. HISTORY
18. LISTEN
19. HELP

13. NOW PLAYING

The Now Playing page provides a view of the content your station will play out, or has already played out, which defaults to showing the 24 hours ahead. This page also enables you to make last-minute changes to running shows.

If you've only just installed Airtime, there might not be any content shown yet. Click the calendar and clock icons above the table to change the date and time range, then click the **Find Shows** button (with the magnifying glass icon) to the right.
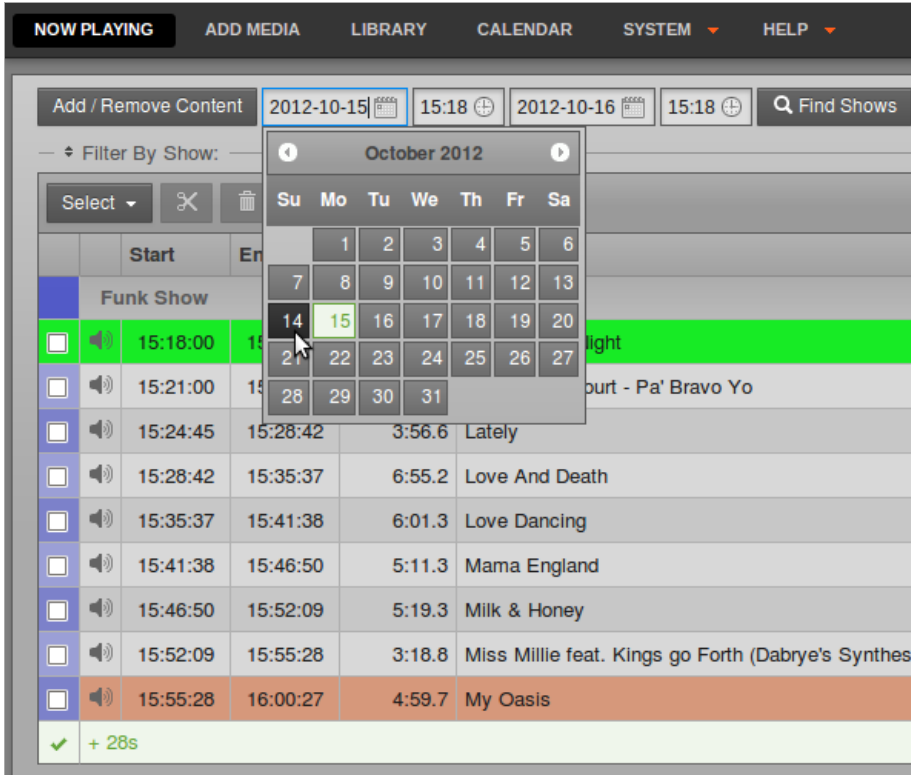
To display the content of a particular show, click **Filter by Show** and select the name of the show from the drop-down menu which will appear.
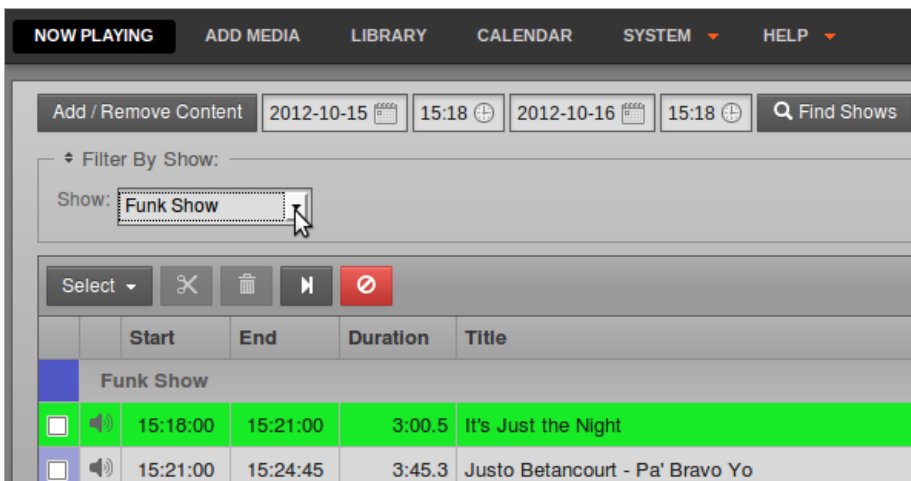
On the left side of the page, the **Start** and **End** times, **Duration** and **Title** of each content item are shown. On the right, **Creator**, **Album**, **Cue** or **Fade** times and **Mime** type (file format) can also be shown. This information can help you to prepare voice tracks for insertion into the show, including time checks if you wish. Putting current time information into voice tracks describing specific content can, of course, limit the re-usability of those voice tracks, unless you always broadcast a particular item at the same time of day.

Click the **Show/hide columns** button on the right to configure the metadata displayed, by checking the boxes in the pop-up window.
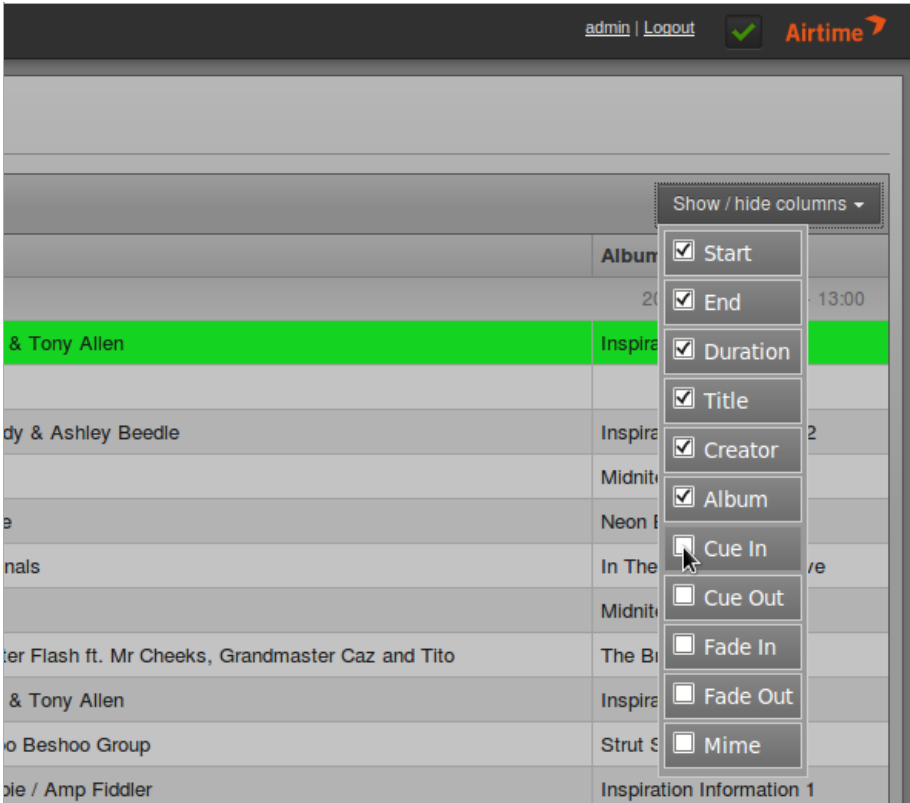
The row for the currently playing item is displayed with a bright green background. Any underbooked shows (shows with insufficient content to fill the time allowed) are displayed with a row indicating the length of the underbooking in minutes and seconds. These rows contain a red exclamation mark in the first column, and have a pink background.

REMOVING CONTENT FROM A RUNNING SHOW

If a show is overbooked, which means the total playout duration is longer than the time allowed for the show, a brown row indicates that the item will be faded out when the show ends. A red row indicates that the item will not be played at all. The length of the overbooking is shown in minutes and seconds in the last row of the show. To remove the extra items from the end of an overbooked show, click the **Scissors** button at the top left of the table.

Alternatively, check the boxes for items which have not yet completed playout, and click the **Trashcan** button, to the right of the Scissors, to remove them from the schedule. If you remove the currently playing item, playout will skip to the next item in the show automatically, so you should make sure you have enough items remaining in the show to avoid dead air.

If you have a long schedule displayed, and you wish to skip to the currently playing item, click the button with the end arrow icon, to the right of the **Trashcan**.

To cancel the current show completely, click the red button to the right again. A pop-up window will ask you to confirm the cancellation, as this action cannot be undone.
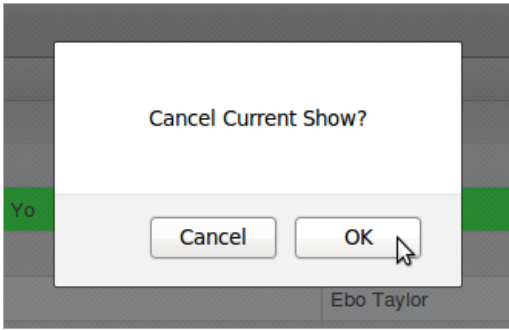
Items which are no longer available have an exclamation mark icon in the second column. This may happen for media files which were part of previous shows, and were removed from Airtime's library (main storage or watched folders) subsequently. Items which are included in forthcoming shows cannot be removed from the Library via the Airtime interface.
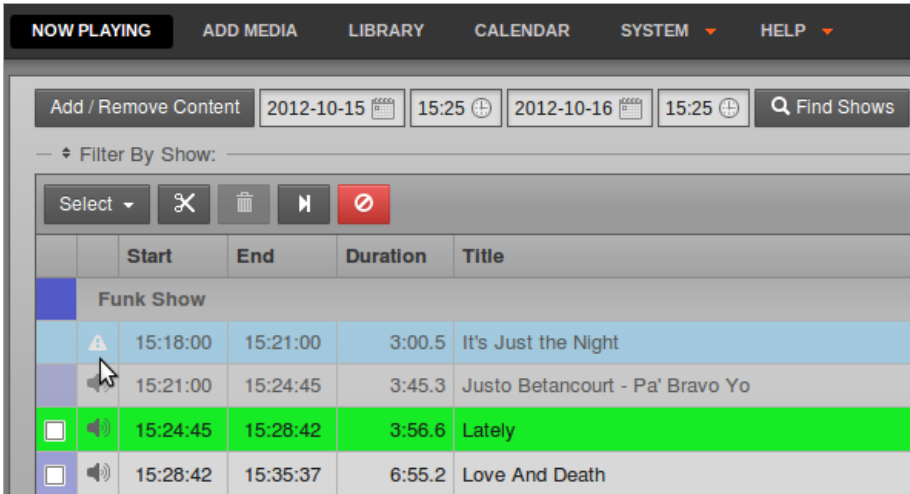
ADVANCED AND SIMPLE LIBRARY SEARCHES

In the top left corner of the page is an **Add / Remove Content** button which enables you to open Airtime's library.
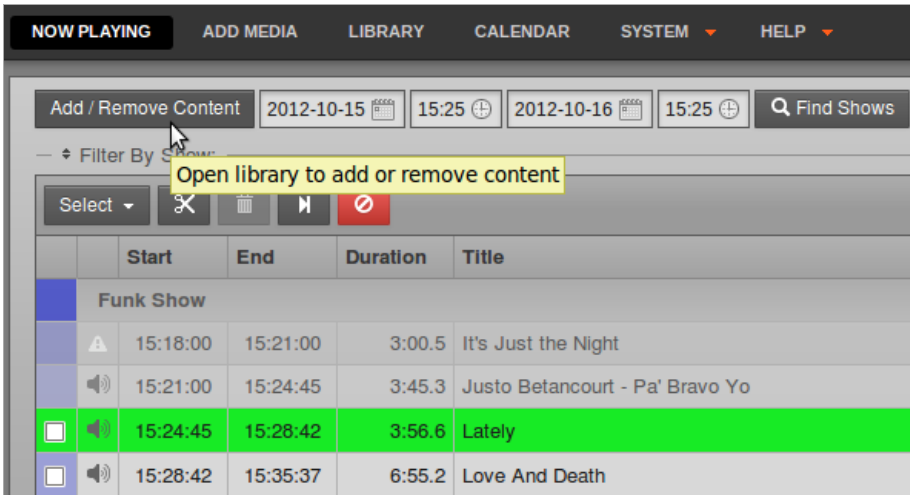
In the library table which opens, the upper section is for the **Advanced Search Options**, which enable you to search within individual fields of the database such as **Title** or **Creator**. Like an Internet search engine, you do not have to type in the correct upper or lower case, press the **Enter** key, or even type the whole of the search term before matches from the Airtime library are displayed.
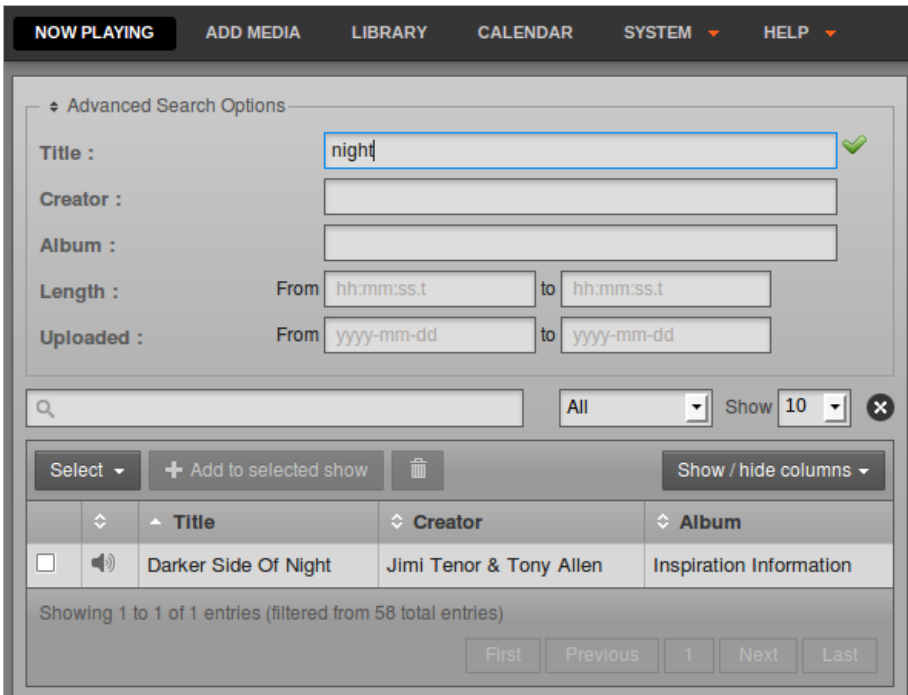
At the bottom of the library table, click the **First**, **Previous**, **Next**, **Last** or individual page number buttons to browse the search results. Right-clicking an item in the search results will display the available metadata for that item in a pop-up window.
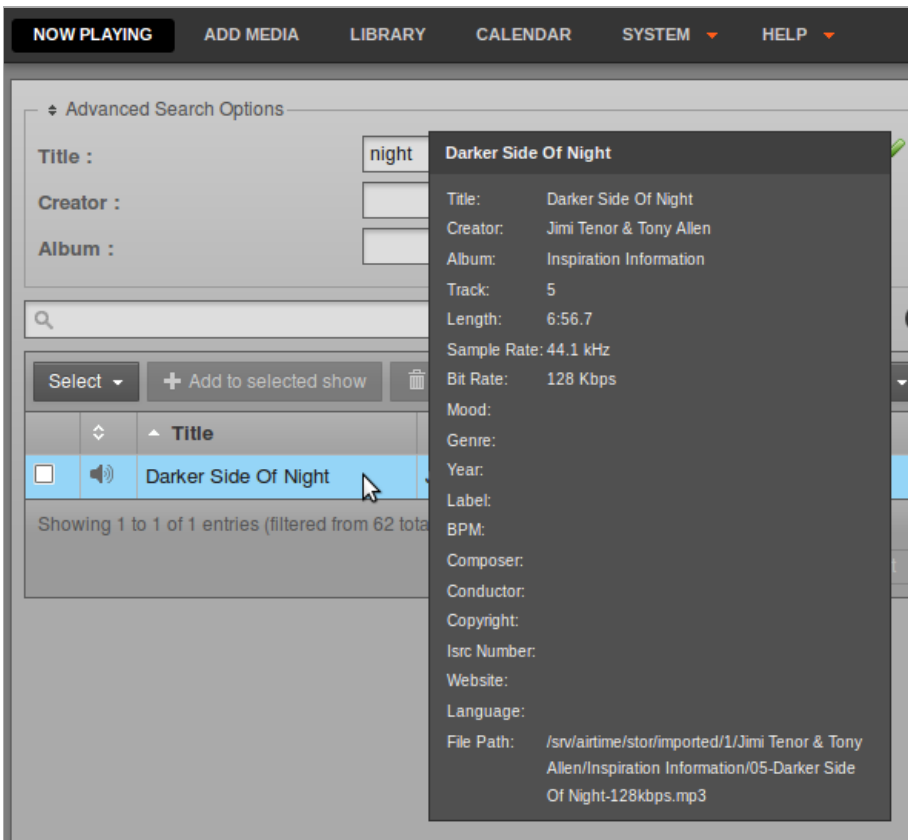
The columns displayed in the search results correspond to the fields available for advanced searches (except for **Scheduled** and **Playlist / Block**, which indicate if a particular item is in use). To change the fields which can be searched, click the **Show / hide columns** button on the right side of the table, just above the search results.
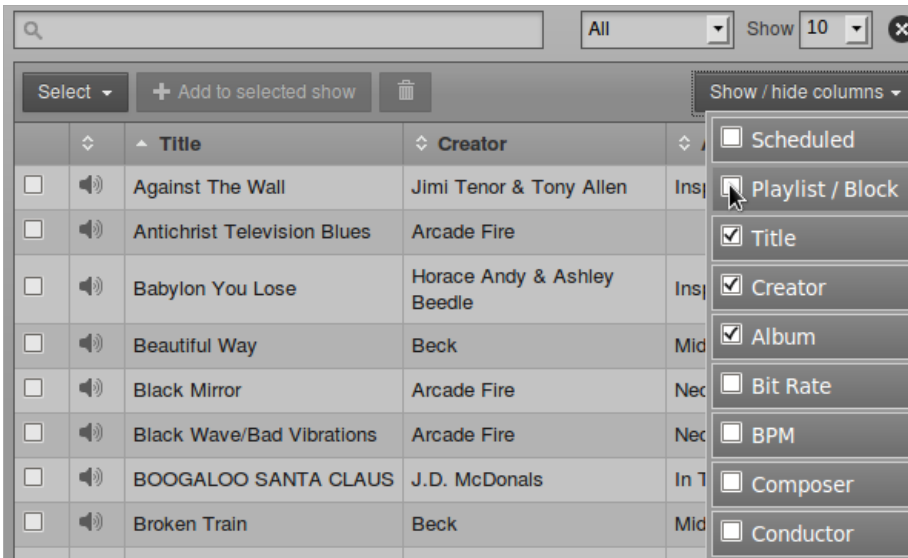
Click the metadata column headings such as **Title**, **Creator**, **Album**, or Genre to sort the entries in ascending or descending order. In the second column, audio files are represented by a loudspeaker icon, while playlists are represented by a document icon. Smart blocks have a think bubble icon, and web streams have an arrow icon.

To find all files uploaded by yourself, or another specific member of the station staff, click the **Show / hide columns** checkbox which enables the **Owner** column. After you have made your search, you can click the header of the **Uploaded** column to find the items that person added to the server most recently.

Advanced search terms are matched using 'and' rather than 'or' logic, to narrow your search. Perhaps you have tagged some jazz-flavoured station idents intended for use in a particular show called 'Jazz Maverick' with the **Album** tag of *Jazz Maverick* and a **Genre** of *Ident*. You could then find exactly the idents you are searching for by typing *maverick* and *ident* into the respective advanced search fields.

Use the drop-down menu at the top of the search results to display **Files**, **Playlists**, **Smart Blocks**, **Web Streams** or **All** items. By default 10 items are listed, but you can click the **Show** drop-down menu to change the list length to between 5 and 100 items.
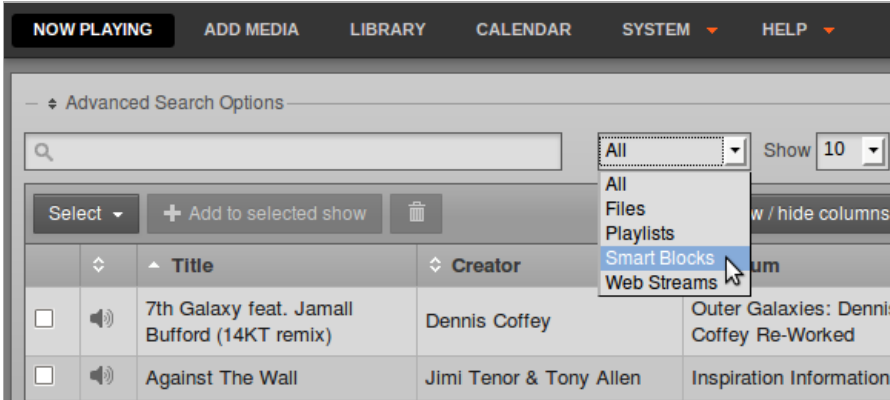
Click the Advanced Search Options link to collapse that area. Beneath the collapsed link is the **Simple Search** field, with a magnifying glass icon, which enables the selection of items based on any tag metadata stored in the Airtime database, whether that metadata column is currently displayed in the table or not. For example, to search for all items between four and five minutes duration, enter *00:04* into the simple search field (for 00 hours and 04 minutes).
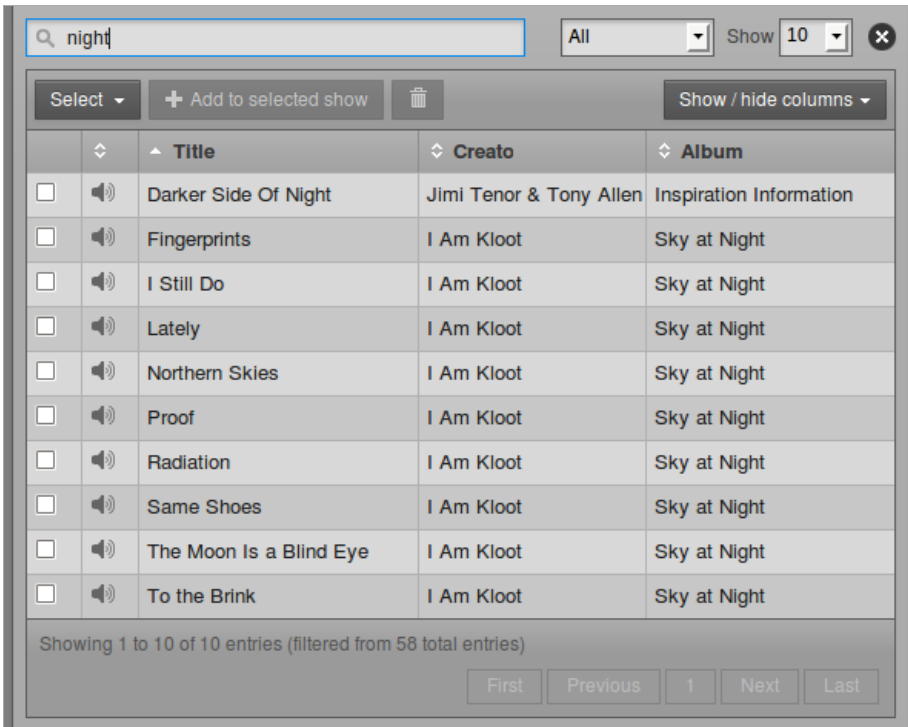
If your jingles are tagged with *Jingle* in the **Genre** or other fields, you can very easily find them by entering *jingle* into the simple search field. The same technique can be used to find advertising and promotional files. (See the chapter *Preparing media for ingest* for tips on batch tagging files with metadata). Unlike advanced searches, simple searches use 'or' logic for multiple search terms, so a simple search for *jingle ident* will return items matching either keyword.

Clicking on an item in the library table opens a pop-up menu which enables you to audition the item, or remove it from the Airtime library. Media types which cannot be auditioned directly in a web browser, such as FLAC, have a lock icon instead of a loudspeaker icon, but these files can be downloaded to your local computer for audition instead. The **Delete** option should be used with caution, because this action cannot be undone. For media files, you can also **Edit Metadata** stored for the file using this menu.
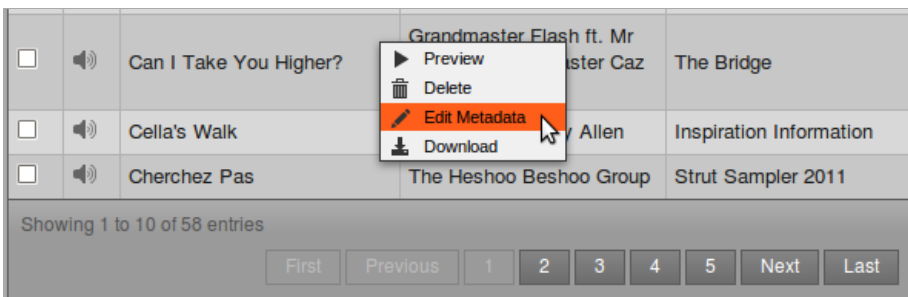
If the **Enable SoundCloud Upload** box on the Preferences page is checked, there will be an additional option on the pop-up menu, **Upload to SoundCloud**. As mentioned previously, you should only upload audio files to SoundCloud with the permission of the copyright holder.
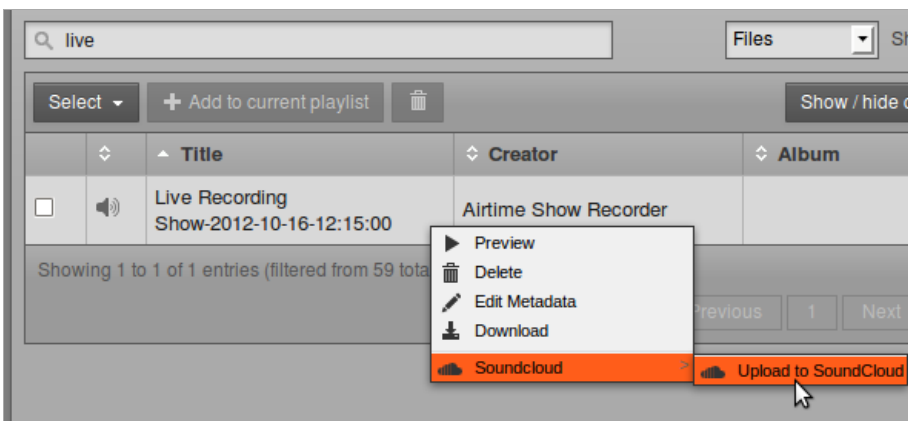
ADDING CONTENT TO A RUNNING SHOW

After you have found the items that you want using the search tools, you can then drag and drop them from the library table on the left side of the page into the shows on the right side, including the current playing show.

If the current show has nothing playing out at the time, the new item will begin playing immediately. This manual triggering of playout can be used as a live assist technique, in which the Airtime server's soundcard output is mixed with other sources such as microphones or telephone hybrids on its way to a transmitter, or a separate stream encoder. For instance, a live show's host may not wish to cut off a studio discussion in order to play music at a fixed time.
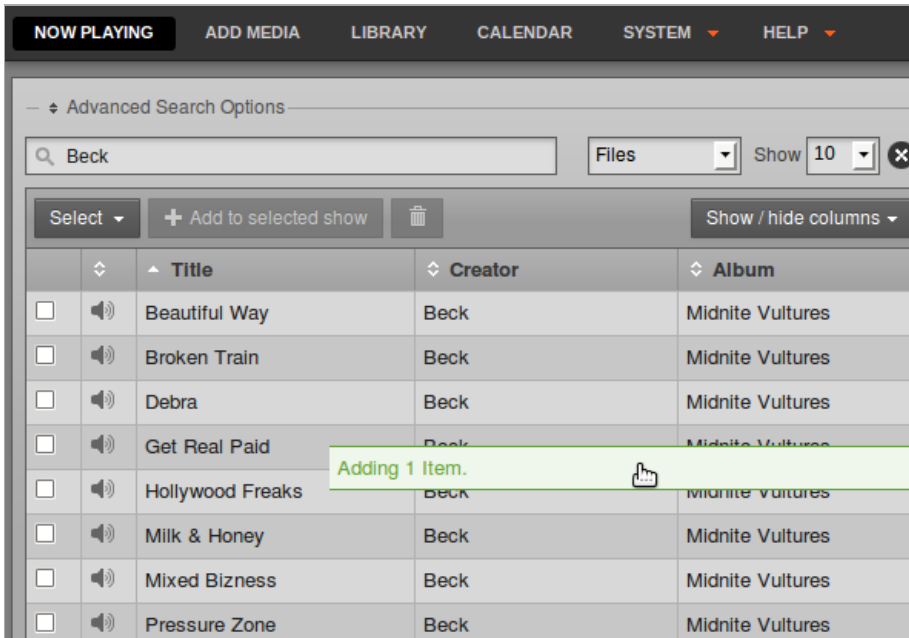
You can also select multiple items using the **Select** menu button, just beneath the simple search field, which has the options to **Select this page** of search results, **Deselect this page** and **Deselect all**. Alternatively, use the checkboxes on the left side of the library table to select specific items. Then drag one of the items into the show to add all of the selected items, or click the **Add to selected show** button, which has a plus icon. If you wish, you can also use the **Trashcan** button to permanently remove items from Airtime's library. Only *admin* users have permission to delete all items.

To insert checkbox selected items at a specific time in the show schedule, click one of the grey and white triangle icons on the left side of the schedule table, which will change to red and white. A red horizontal line will be shown at the insertion point. Then click the **Add to selected show** button in the library table.

To add a single item at the insertion point, double-click on it in the library. There is no need to select or drag the item first.
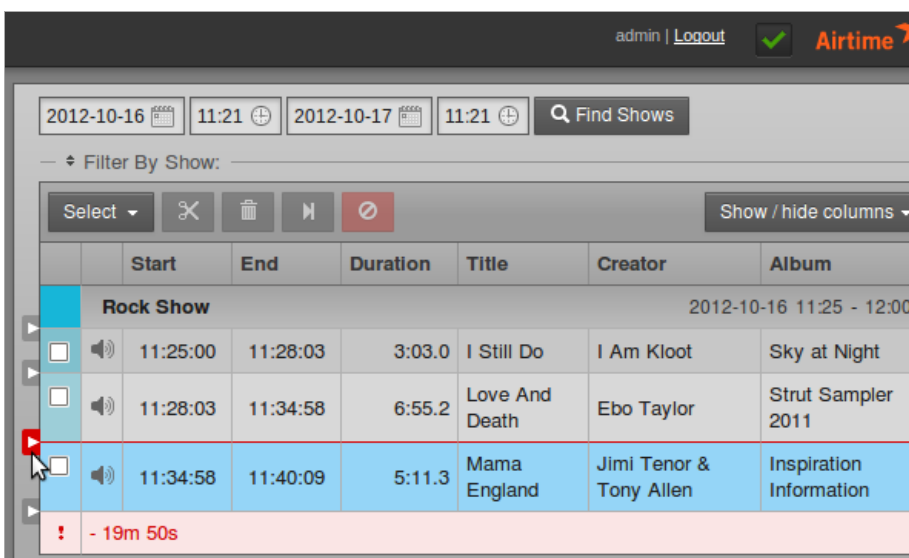
Another way to create an insertion point is to click an item in the show table, then click **Select cursor** on the small pop-up menu that will appear. This pop-up menu also enables you to audition the entire show in advance of playout, or remove the item that was clicked on from the show.

Multiple insertion points can be enabled, so that the same item is inserted into the schedule at different times. For example, you may wish to play a news report every hour, or a station ident after every five music files.

A live show set for recording from the Airtime server's sound card is displayed with a red dot icon in the first column. It would not contain any files or playlists, smart blocks or web streams.

14. ADD MEDIA

If you do not have direct access to the Airtime server, you can add files to the Airtime library using the **Add Media** page of the administration interface. (The **Add Media** page is not visible to *Guest* users). This page includes an upload queue for media files, which supports drag and drop from your computer's file manager if you are using a recent web browser, such as *Mozilla Firefox 16* or later.

Some web browsers may set an upload limit for a single file, between 200MB and 2GB. If you need to upload files larger than 200MB to the Airtime server on a regular basis, you may find it more convenient to perform the upload using SFTP, rather than through the browser. See the chapter *Automated file import* for more details.
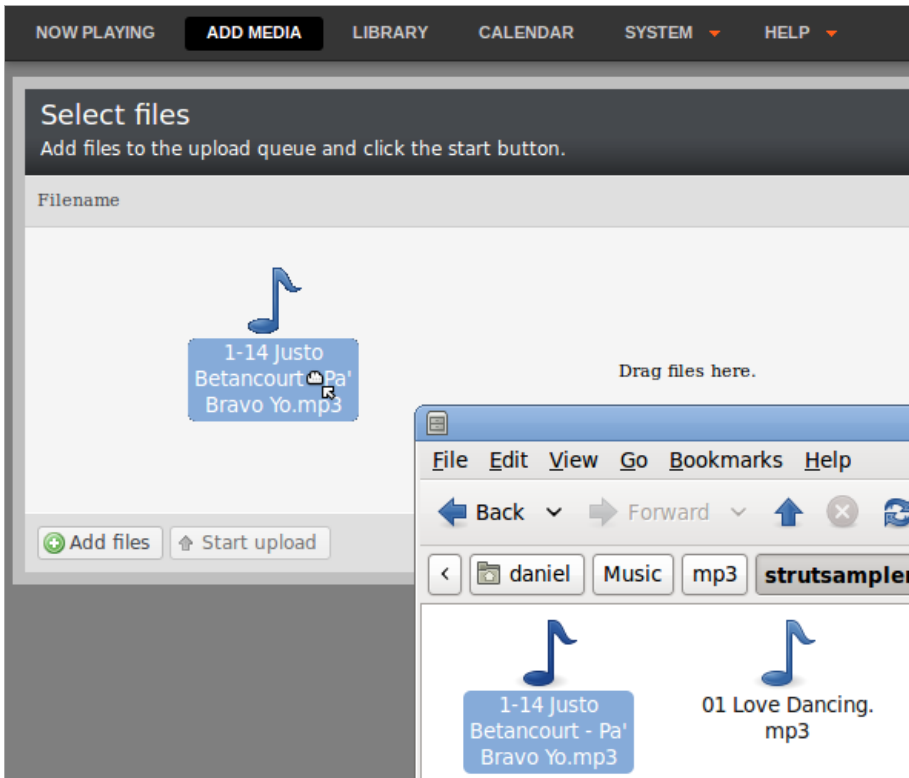
If your web browser does not support drag and drop, you can use the **Add files** button, which has a white plus sign in a green circle icon, to open a file selection window on your computer.
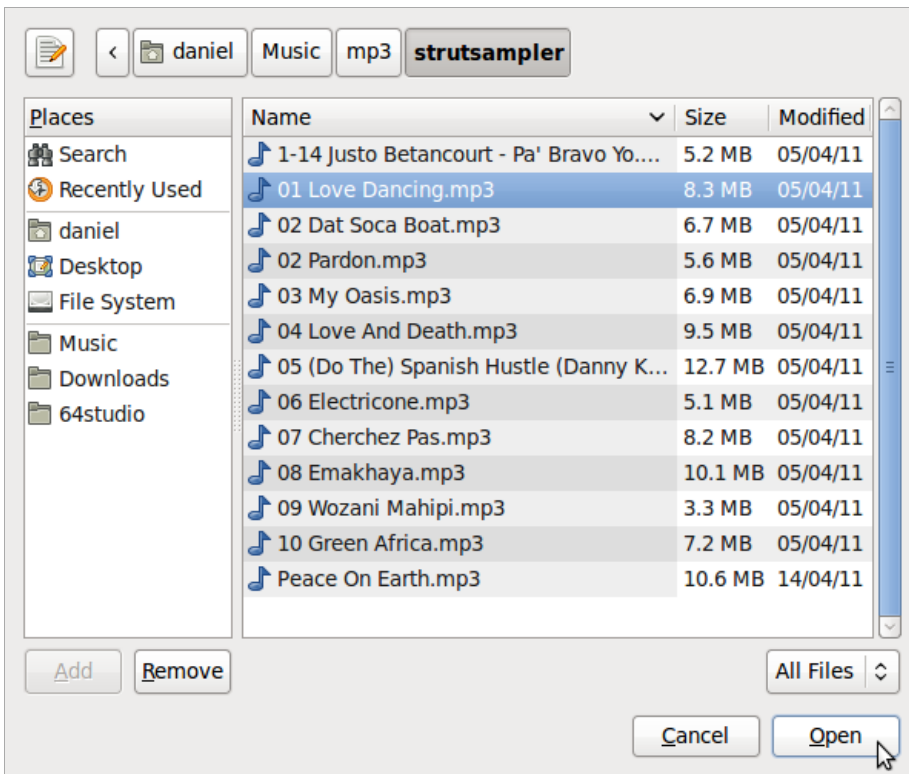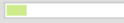
After you have added all the files that you require to the upload queue, click the**Start upload** button, which has a green arrow icon.
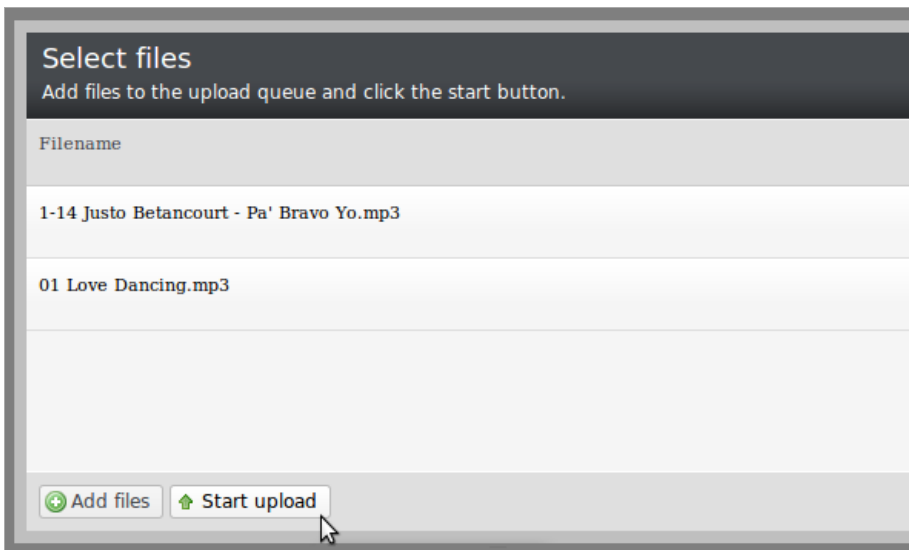
The row of the file currently being uploaded is highlighted in pale green. At the bottom of the upload queue, a progress bar and percentage indicates how much of the upload has taken place so far. The upload speed will depend on the network connection between your computer and the Airtime server.
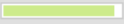
Once it has been uploaded successfully, each file row displays a white check mark in a green circle icon.

Your files are now imported into the Airtime library, ready to be included in your broadcast playlists, smart blocks and shows.

15. LIBRARY

This page of the Airtime interface enables you to search the media library, sort and display the search results by the criteria that you choose, audition library items, and drag and drop those items into a playlist. You can also adjust fade and cue points, create smart blocks (automatically generated playlists), or add incoming web streams to the library. The **Library** page is not visible to *Guest* users.

SEARCHING THE LIBRARY

The search features on the left side of the **Library** page are the same as those available on the **Now Playing** page. Refer to the chapter *Now Playing* for details.

CREATING A NEW PLAYLIST

Once you have found the media that you require using the search tools, you can create a new playlist on the right hand side of the **Playlist Builder** page. Click the **Open Media Builder** button to begin.
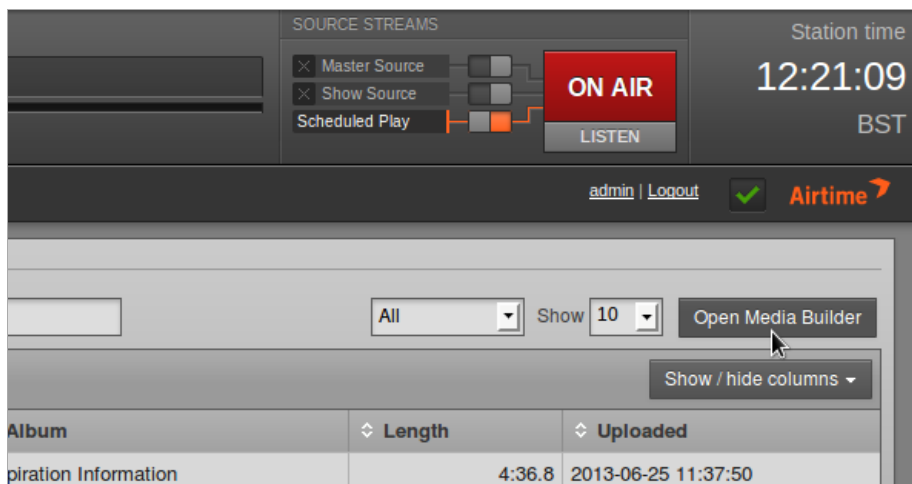
Then click the **New** button and select **New Playlist** from the pop-up menu.
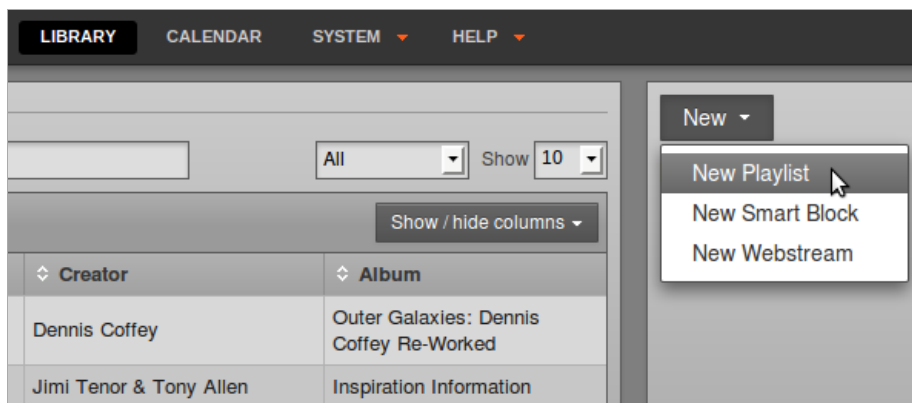
At first, the new playlist will be shown as *Untitled Playlist*. Click the pencil icon on the right to give the playlist a name.
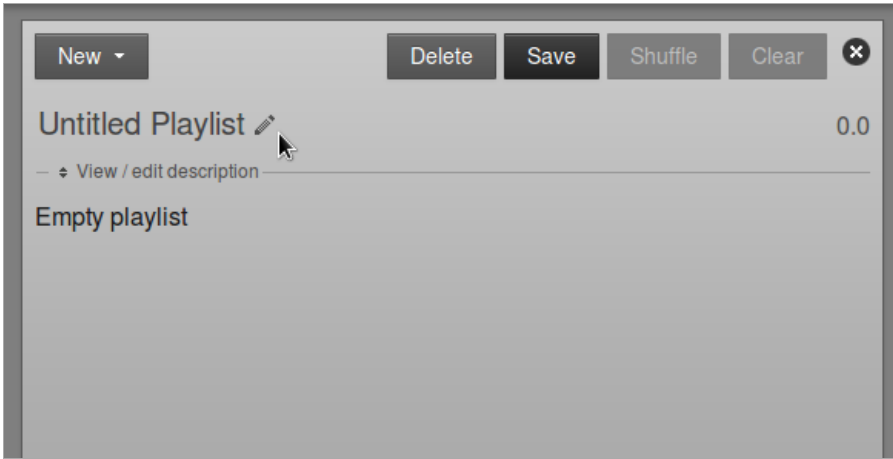
Type the name you have chosen, then press the **Enter** key on your keyboard to save the new name. You can edit the name of the playlist later, by clicking on the pencil icon again.
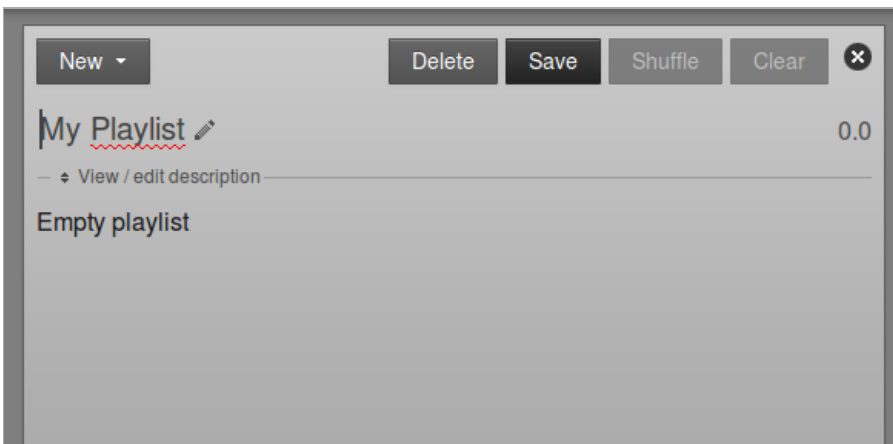
Click the link **View / edit description** to expand a box where you can enter a **Description** for the playlist, then click the **Save** button. Setting good quality metadata here will help you find the playlist using the search box later, so you should be as descriptive as possible.
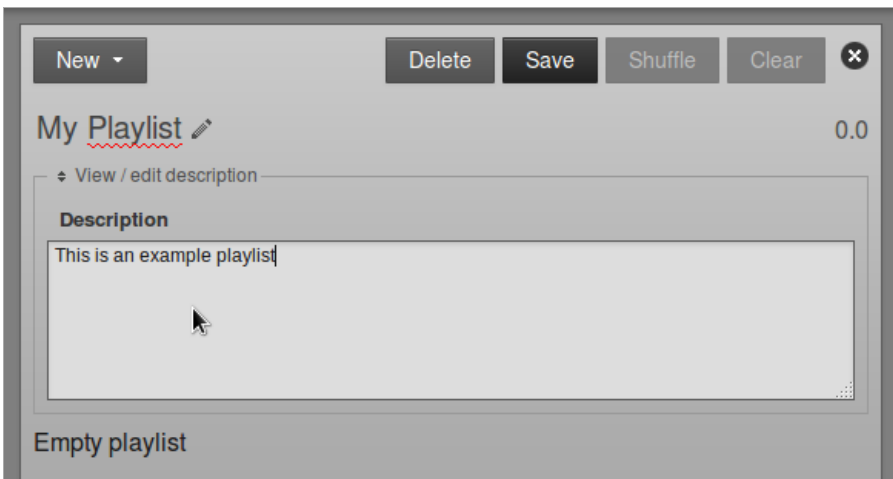
ADDING CONTENT TO A PLAYLIST

With a playlist open, click on an item in the search results and then click **Add to Playlist** on the pop-up menu. Or drag and drop items from the search results on the left into the playlist on the right. Jingles and voice tracks can be added before, after or between music items.
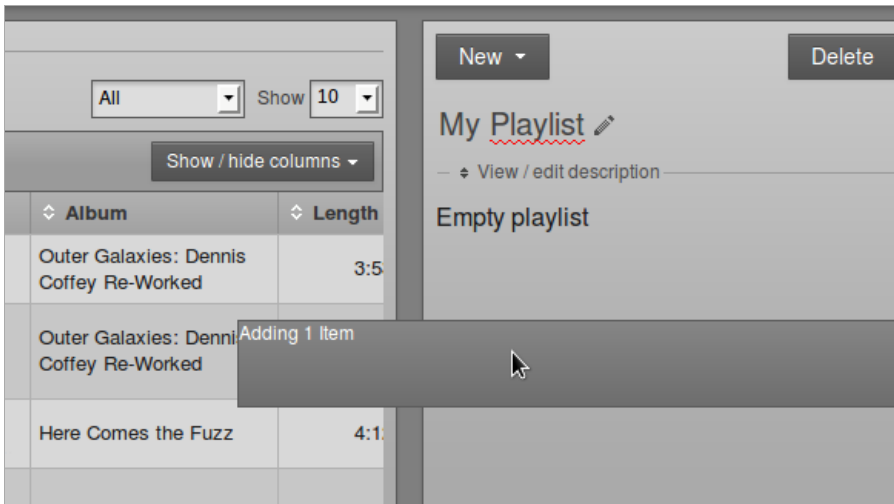
You can also select or deselect a whole page of search results using the **Select** button, just below the simple search field. Alternatively, use the checkboxes in the first column of the search results table to select individual items, then click the **Add to current playlist** button. Items that you own which you no longer require can be deleted from Airtime's library using the **Trashcan** button.
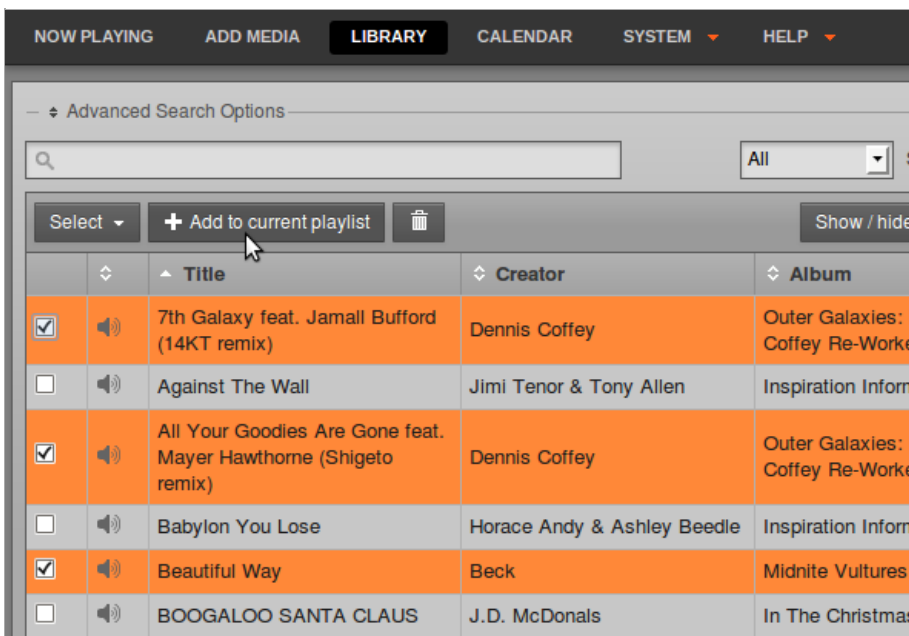
After adding files to the playlist, the total playlist time is displayed in the top right corner. The duration of an individual file is shown in each row of the playlist in a white font, and beneath this figure the time since the beginning of the playlist is displayed in a smaller light grey font. This elapsed time figure can be used as a time check for voice tracks, although this option may limit the re-usability of the voice track.

To audition a playlist file in your web browser, click the white triangle **preview** button on the left side of its row. (If the format of the file is not supported by your browser, the triangle in this button will be greyed out). If preview of the file format is supported, a pop-up audition window will open, with the playlist starting at the file you clicked. Click the small white **x** icon on the right hand side of each row to remove a file from the playlist. You can also drag and drop files to re-order them, or click the **Shuffle** button to re-order files automatically. Click the **Clear** button to remove all content from the playlist, or click the **Delete** button to remove the playlist from the Airtime library altogether.
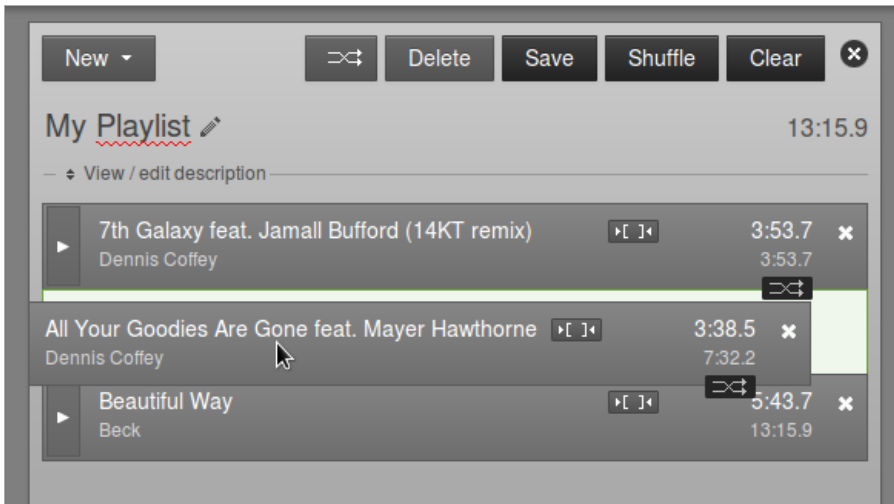
To adjust start and end fades, click the playlist **Fade** button (two horizontal white arrows crossing in a grey rectangle), to the left of the **Delete** and **Save** buttons. This action opens a beige bar in which you can set the **Fade in** duration for the first item in this playlist, and the **Fade out** duration for the last item. This duration figure represents the length of the fade, in seconds and tenths of a second, not the time at which the fade takes place. The default fade duration is set in the **Preferences** page on the **System** menu.
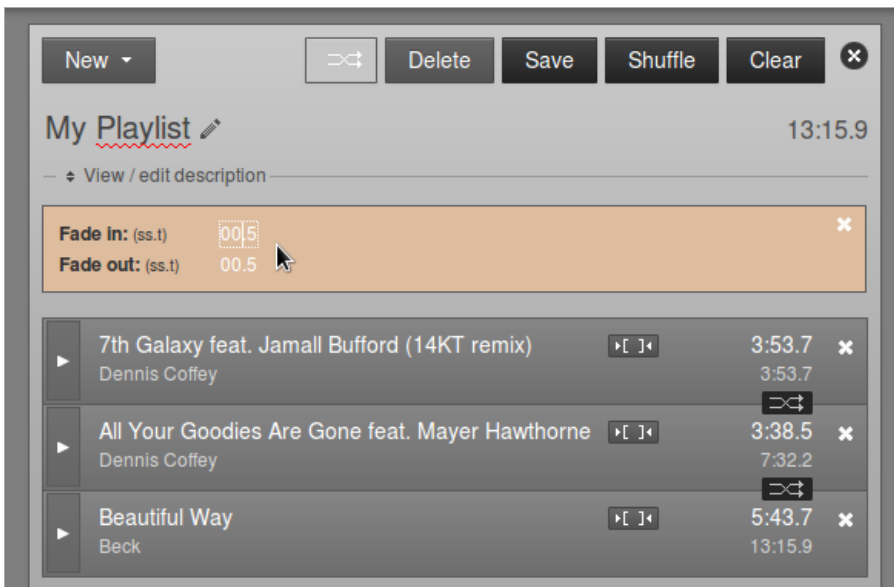
Click any one of the smaller **Fade** buttons between file rows to open another beige bar, which enables you to set **Fade out** and **Fade in** durations between two adjacent files in the playlist. The fade buttons for adjacent files change to an orange background when you click them.
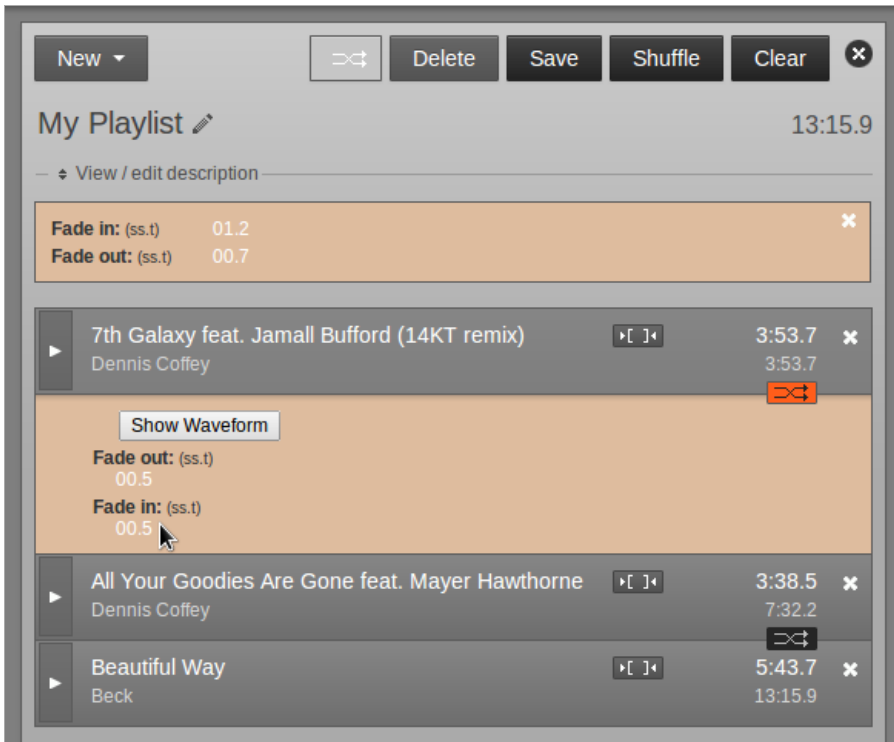
If your web browser supports the Web Audio API, you will see a **Show Waveform** button which enables you to view the waveforms of the adjacent items, adjust the fades and audition them. Click the **Fade Out** or **Fade In** button to change the fade curves by clicking in the waveforms, then click the **Play** button to audition the effect of any adjustment you have made. To audition just one part of the item, use the **Cursor** button to move the play cursor, which appears as a thin red line, on the waveform. The playback progress is shown by an orange colour on the waveform.
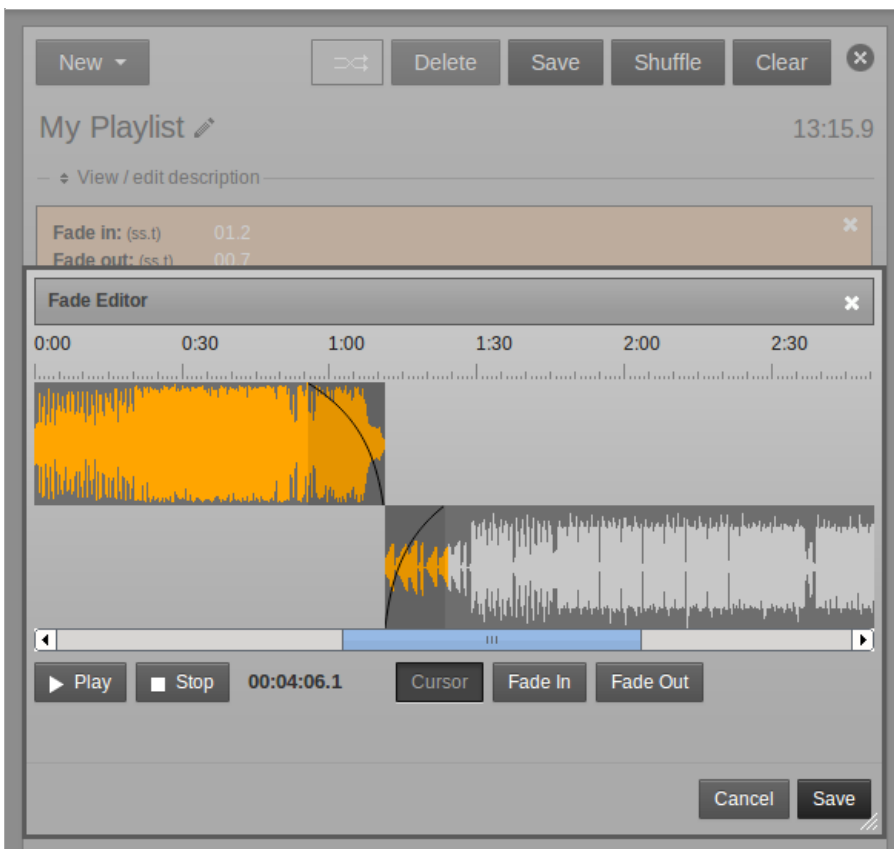
Each file in the playlist also has a button with two square brackets, which enables you to adjust **Cue In** and **Cue Out** times for that particular file. Like the fade button, the cue button changes to an orange background when you click it.

Cue In and **Cue Out** points are set in hours, minutes, seconds and tenths of a second, relative to the start of the item. If a file has leading or trailing silence, corresponding cue points will be set automatically. The duration of the file in the playlist is updated automatically when you adjust a cue point, but the **Original Length** of the file is also displayed for your reference.
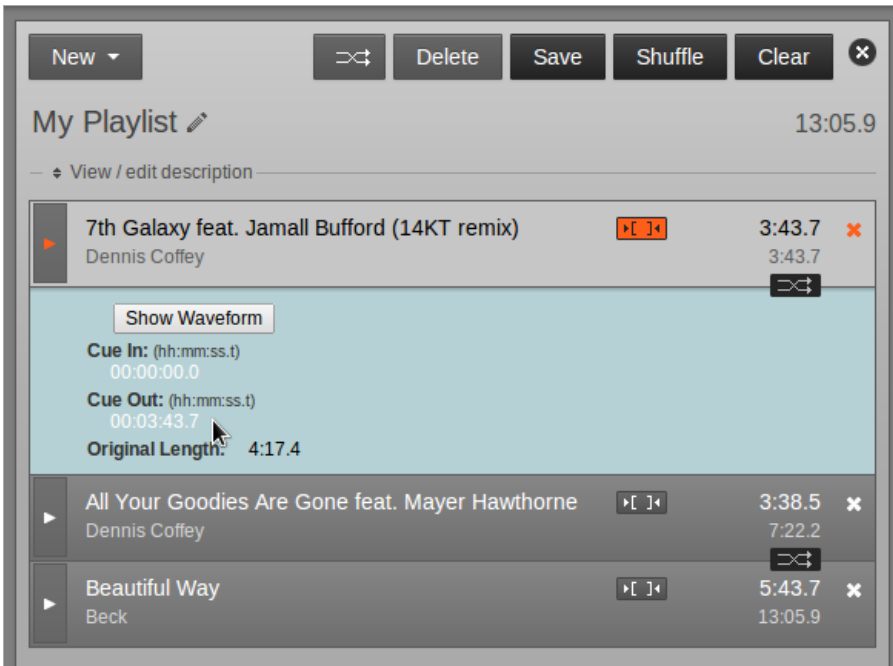
If a **Fade In** and **Cue In** are set on the same item in the playlist, the fade in begins at the **Cue In** point. If a **Fade Out** and **Cue Out** are set on the same item, the fade out ends at the **Cue Out** point. If your web browser supports the Web Audio API, you can click the **Show Waveform** button to adjust and audition cue points.
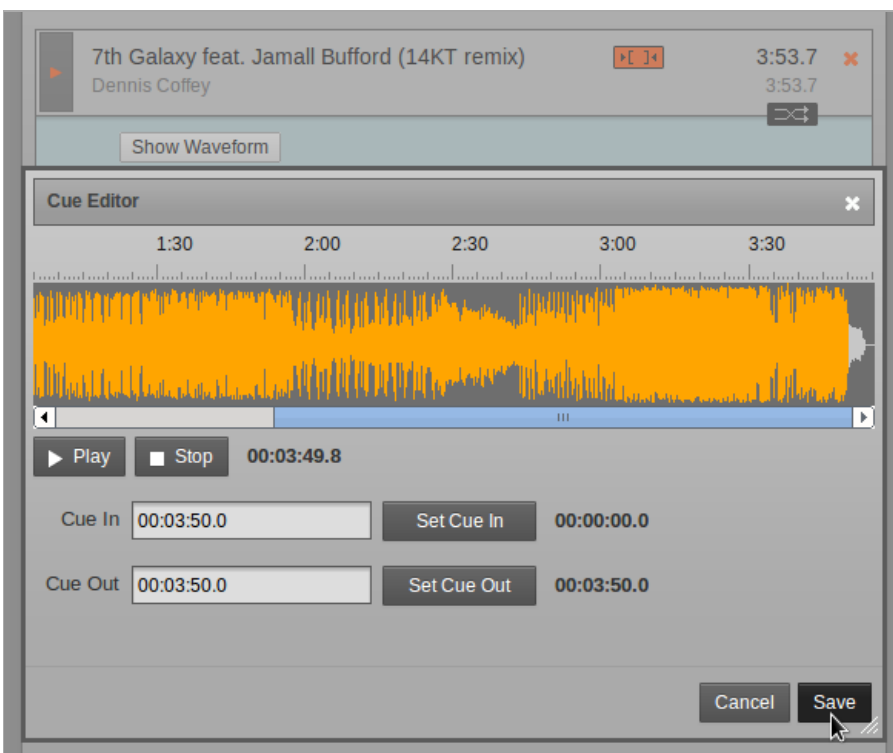
When your playlist is complete, click the **New** button in the top left corner to create another playlist, click the close icon (a white cross in a black circle) in the top right corner, or browse to another page of the Airtime interface.

If you want to edit the playlist content or metadata later, you can find it by **Title**, **Creator**, **Last Modified** date, **Length**, **Owner** or **Year** using one of the search tools on the Library page. Click the playlist in the search results list, and then click **Edit** from the pop-up menu. You can also **Preview** the entire playlist in a pop-up audition window, **Duplicate** or **Delete** one of your playlists from this menu.
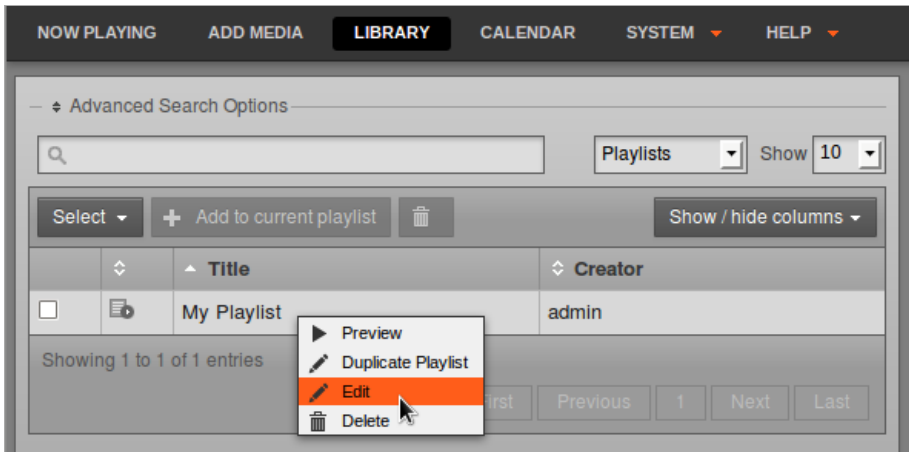
CREATING A SMART BLOCK

Smart blocks are automatically filled with media files from the Airtime library, according to the criteria that you specify. This feature is intended to save staff time, compared to selecting items for a playlist manually.

To create a smart block, click the **New** button on the right side of the Library page, and select **New Smart Block** from the pop-up menu. Like a playlist, smart blocks can have a title and **Description**, which you can edit. This helps you find relevant smart blocks in searches.
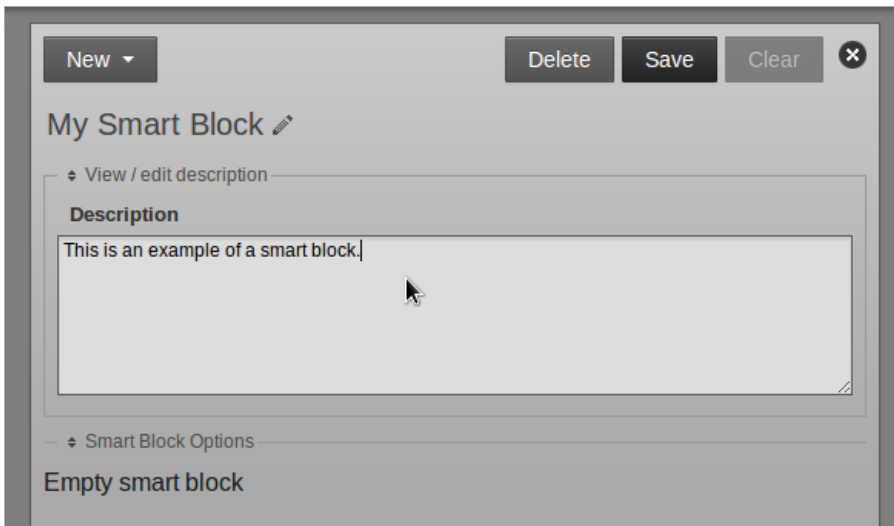
Click the link **Smart Block Options** to display the criteria and modifiers for the smart block. The criteria can be any one of Airtime's metadata categories, such as **Title**, **Creator** or **Genre**. The modifier depends on whether the metadata in question contains letters or numbers. For example, **Title** has modifiers including *contains* and *starts with*, whereas the modifiers for **BPM** include *is greater than* and *is in the range*.

You can also set the **smart block type**. A **Static** smart block will save the criteria and generate the block content immediately. This enables you to edit the contents of the block in the **Library** page before adding it to a show. A **Dynamic** smart block will only save the criteria, and the specific content will be generated at the time the block is added to a show. After that, the content of the show can be changed or re-ordered in the **Now Playing** page.
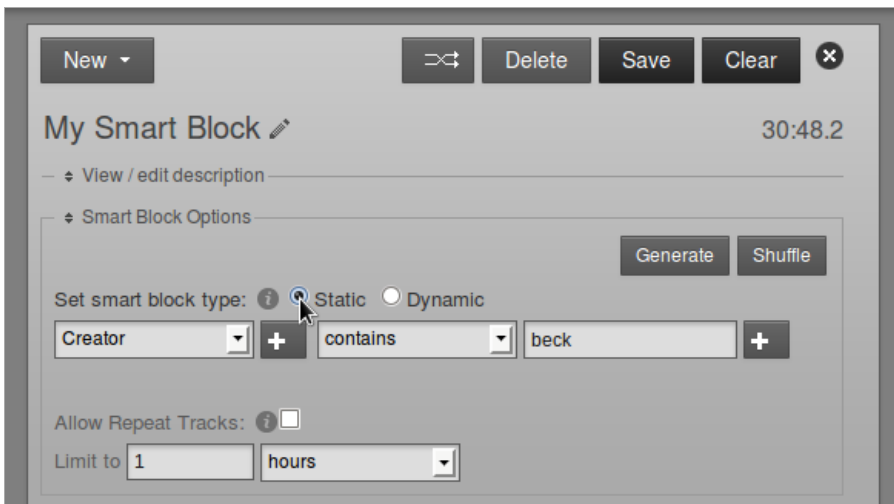
Click the **plus button** on the left to add OR criteria, such as **Creator** containing *beck* OR *jimi*. (The criteria are not case sensitive). For a static smart block, click the **Generate** button to see the results. Dynamic smart blocks do not display the **Generate** or **Shuffle** buttons.
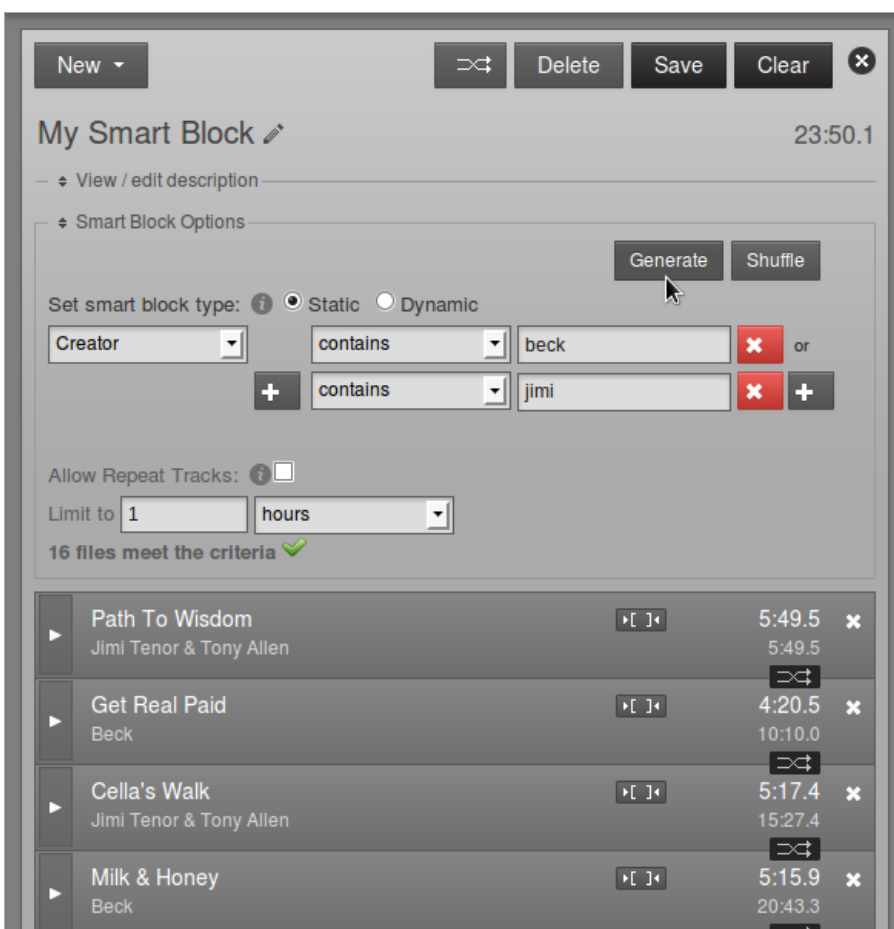
If you don't like the ordering which is generated, click the **Shuffle** button, or drag and drop the smart block contents into the order that you prefer. You can also remove items or add new items manually from the Library. Changes to static smart block contents are saved automatically when you add items, remove or re-order them, or click the **Generate** button. Click the **Save** button in the upper right corner to save any changes to smart block criteria.

To add an AND criteria, such as **Creator** containing *jimi* AND BPM in the range *120* to *130*, click the **plus button** on the right. If you see the message **0 files meet the criteria** in this case, it might mean that the files in the Library have not been tagged with BPM metadata. See the chapter *Preparing media for ingest* for tips on tagging content.
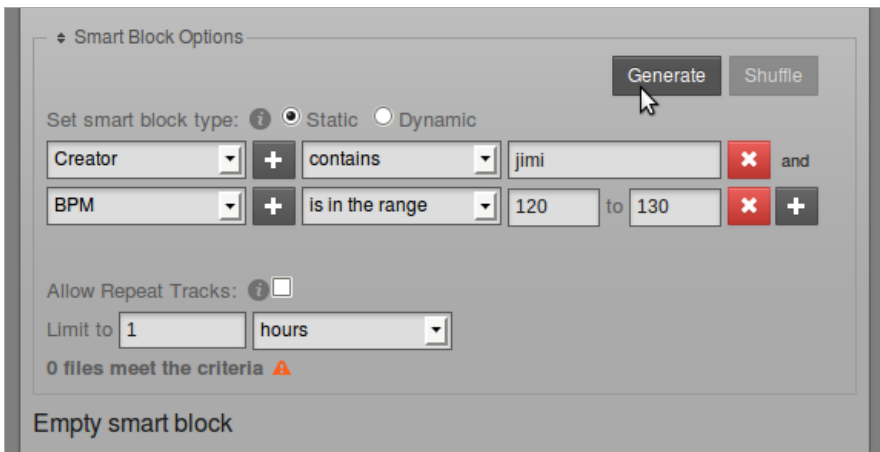
By default, a smart block will not contain repeated items, which will limit the duration of the block if you do not have sufficient items meeting the specified criteria in your **Library**. To override the default behaviour, check the **Allow Repeat Tracks** box.

If you have a large number of files which meet the criteria that you specify, you may wish to limit the duration of the smart block using the **Limit to** field, so that it fits within the show you have in mind. Select **hours**, **minutes** or **items** from the drop-down menu, and click the **Generate** button again, if it is a static smart block. Then click the **Save** button.

Smart blocks can be added to shows in the same way as a manually created playlist is added. Smart blocks can also be added to one or more playlists. In the case of a playlist containing a static smart block, click **Expand Static Block** to view the contents. For a dynamic smart block, you can review the criteria and duration limit by clicking **Expand Dynamic Block**.
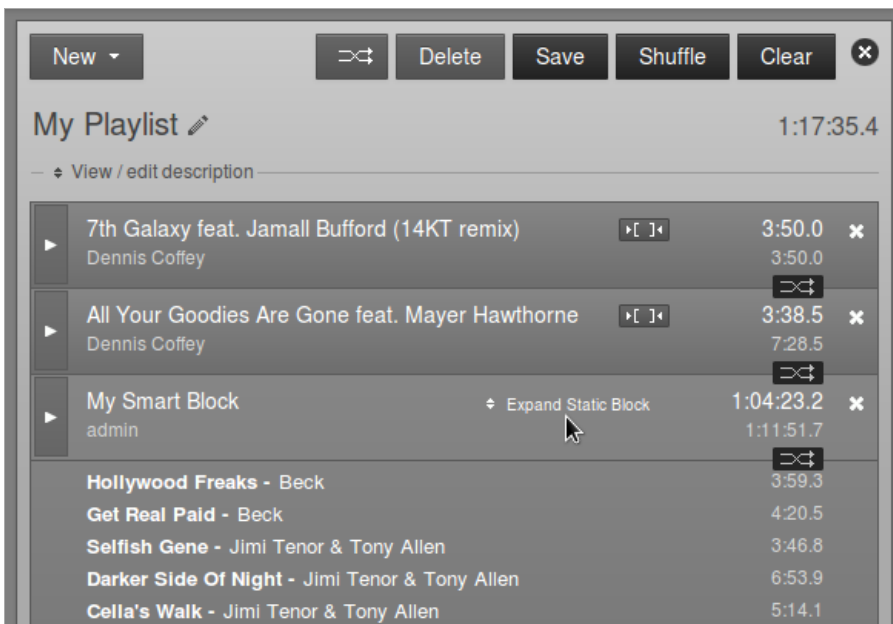
Once created, smart blocks can be found in Library searches and refined at any time. They can be re-opened by clicking on the smart block and selecting **Edit** from the pop-up menu.

ADDING A WEB STREAM

A web stream URL and metadata can be added to the Airtime library, so that a remote stream can be searched for and scheduled to be *pulled* into a show. For example, at the top of the hour your station may pull a news report from journalists working in another studio. This is a different concept from **Master Source** and **Show Source** remote streams which are *pushed* into the Airtime playout schedule.

To add a web stream, click the **New** button on the right side of the Library page, and select **New Webstream** from the pop-up menu. Like a playlist, web streams in the Library can have a title and **Description**, which may help you find them in searches later.

The **Stream URL** setting must include the *port number* (such as **8000**) and *mount point* (such as remote_stream) of the remote stream, in addition to the streaming server name. A **Default Length** for the remote stream can also be set. If the stream is added at the end of a show which becomes overbooked as a result, it will be faded out when the show ends.

16. CALENDAR

The Calendar page of the Airtime administration interface has three views: **day**, **week** and **month**, which can be switched using the grey buttons in the top right corner. By default, the **month** view is shown, with today's date highlighted by a pale green background.
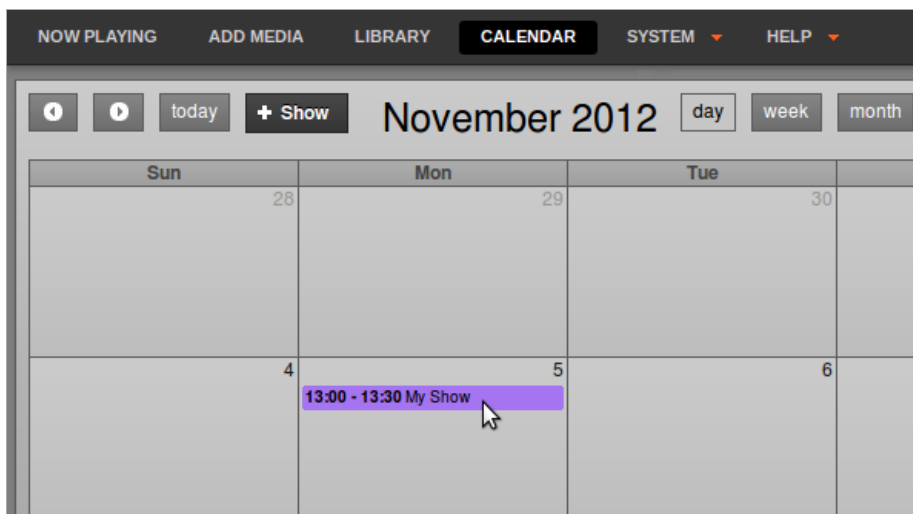
In the top left corner of the page, you can go back or forward through the **Calendar** by clicking on the buttons which have a small grey triangle in a white circle. Click the **today** button to jump to today's date in the current view. (The **today** button will be greyed out if you are already viewing that date). In the **day** or **week** views, there is also a drop-down menu which allows you to set the resolution displayed for the calendar, ranging from one minute per row to sixty minutes per row.
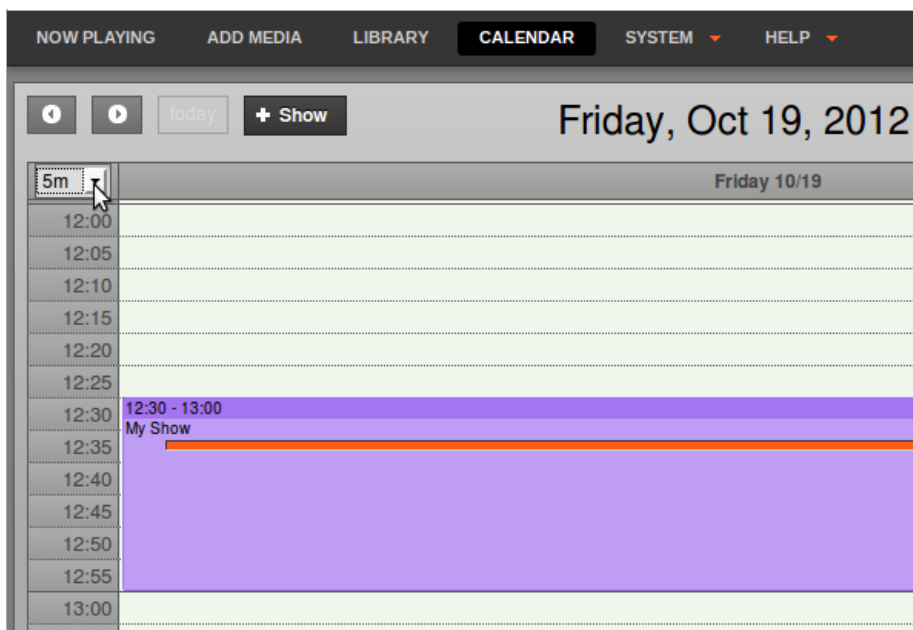
ADDING A SHOW

Only *Admins* and *Program Managers* can use this feature. To add a new show to the Calendar, click the **+ Show** button in the top left corner of the page, or click on any future row or box in the Calendar which is empty. Either of these actions opens the **Add this show** box, which has six sections, arranged vertically: **What**, **When**, **Live Stream Input**, **Record & Rebroadcast**, **Who**, and **Style**. Click the small black triangle to the left of the section name if you wish to minimize or maximize it.

WHAT

In the **What** box, enter the **Name**, public website **URL**, **Genre** and **Description** for the show that you are creating.
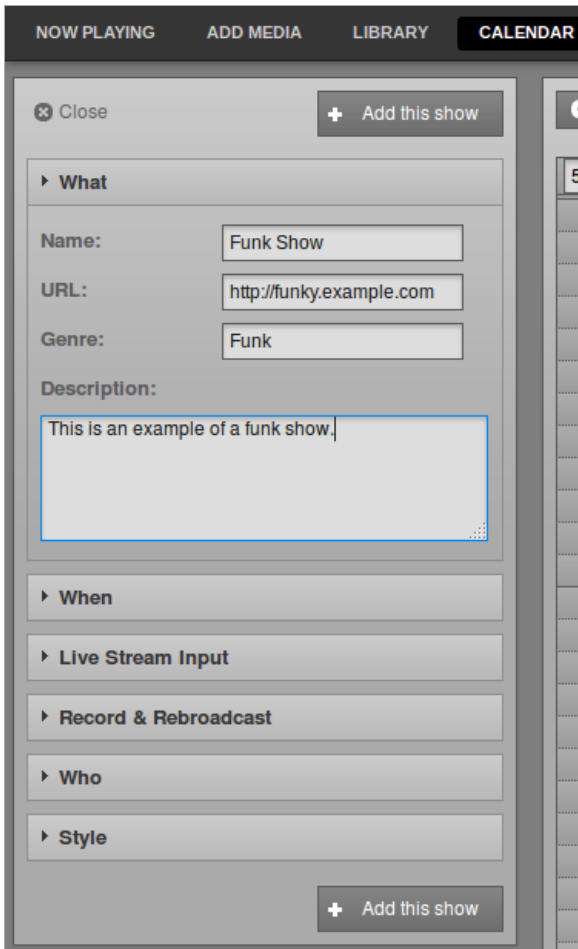
WHEN

Next, in the **When** section, if you clicked on a date in the **Calendar** this should already be entered in the **Date/Time Start** field. To set another date for the show, click on the date in the **Date/Time Start** field and select the date that you require from the small pop-up calendar which will appear. Click on the adjacent time field to set the start time for the show, with the pop-up **Hour** and **Minute** box. The Minute values in the pop-up time boxes are rounded to the nearest five minutes. You can also adjust the times manually by clicking into the fields and typing. Repeat the process to set the **Date/Time End** fields. The **Duration** of the show will be displayed automatically, based on the start and end times you have set.

By default, the **Timezone** of the show start and end times will be the timezone of the station, as set on the Preferences page on the System menu. You can specify an alternative time zone using the drop-down menu, if you wish. The time zone displayed in the Calendar when you log in depends on your personal settings. See 'Updating your own account' in the chapter *Users* for more details.
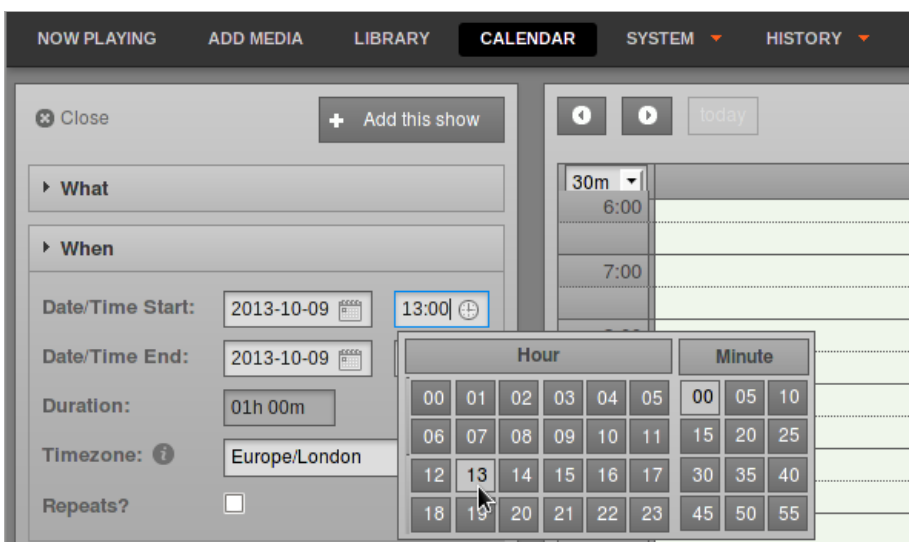
Airtime will only allow you to enter valid times for shows. If you attempt to schedule show times which would be impossible, by ending the show before it begins, Airtime will highlight the error with a pink background.

To schedule a regular show, check the **Repeats?** box, which will make a new section of the dialog appear. Optionally, click the **Link:** box to automatically schedule the same content in the repeated shows as in the original show.

Then select either **weekly**, **every 2 weeks**, **every 3 weeks**, **every 4 weeks** or **monthly** from the **Repeat Type** drop-down menu. If you have selected a weekly option, check the boxes for the days of the week that you want to schedule the regular show on. If you have selected **monthly**, you have the option to **Repeat By: day of the month** (for example the 9th of each month) or **day of the week** (for example the second Wednesday of each month).

Uncheck the **No End?** box and set the **Date End** for the regular show to finish, or leave the **No End?** box checked to schedule the show indefinitely.

LIVE STREAM INPUT

The **Live Stream Input** section can be used to enable live input streams during the show, and also to configure authentication for them. If you check the **Use Airtime Authentication** box, the DJs (presenters) of the show will be able to connect a live stream to Airtime's **Show Source Mount Point** using their login name and password. See the chapter *Stream Settings* for details.

If you check the **Use Custom Authentication** box instead, you can set a one-time username and password pair for live stream input to the show. For example, you may wish to create an authenticated live stream input for a particular remote news reporter, without providing any further access to your Airtime server for that person.
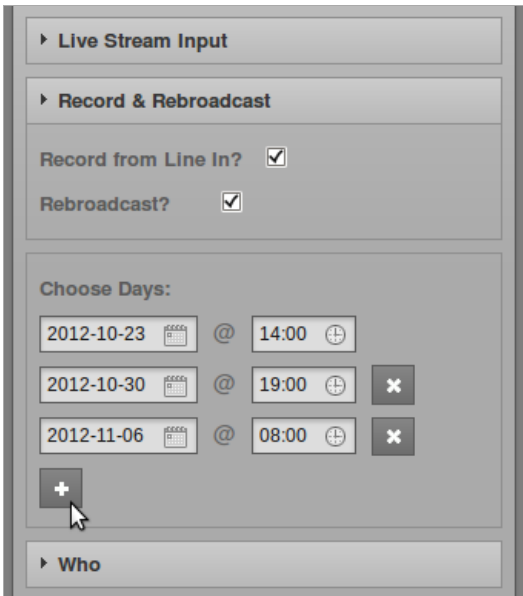
A reminder of the **Connection URL** for the live input stream to use is shown at the end of the **Live Stream Input** section.

RECORD & REBROADCAST

In the **Record & Rebroadcast** section, checking the **Record from Line In?** box enables automatic recording of the soundcard line input, if your Airtime server has one, at the time of the show. Shows set for line-in recording should not also contain files or playlists. The default audio format for live recordings is 256kbps Ogg Vorbis, and the files are saved in the *recorded* folder, under the **Import Folder** path set in the **Media Folders** page on the **System** menu. See the chapter *Host configuration* for details of recorder settings.

If you wish the recording to be played out at a later time, check the **Rebroadcast?** box, and then select up to ten date and time slots in the **Choose Days** box.

Shows set for recording have a small red dot icon in the calendar, while rebroadcast shows have a white loop icon.

WHO

In the **Who** section, type the first few letters of the name of the show's DJ (presenter) in the **Search Users** field to select a name from the Airtime database, or check one of the **DJs** boxes in the vertical list below. This association of a DJ name with a particular show enables that presenter to add playout media to the show, and also to connect a live Show Source input stream (if that has been enabled), so it is important to get the DJ's name right.
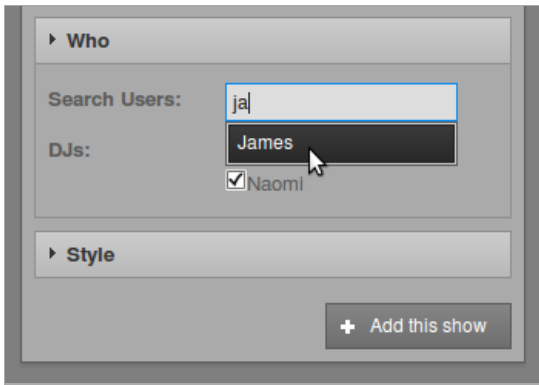
STYLE

Show colours in the Calendar are set automatically, based on the **Name** of the show, so that individual shows can be easily identified in the **Calendar**. If you wish, you can select a **Background Colour** and **Text Colour** manually by clicking the boxes in the **Style** section. Click the coloured circle icon in the lower right corner to close this pop-up window.
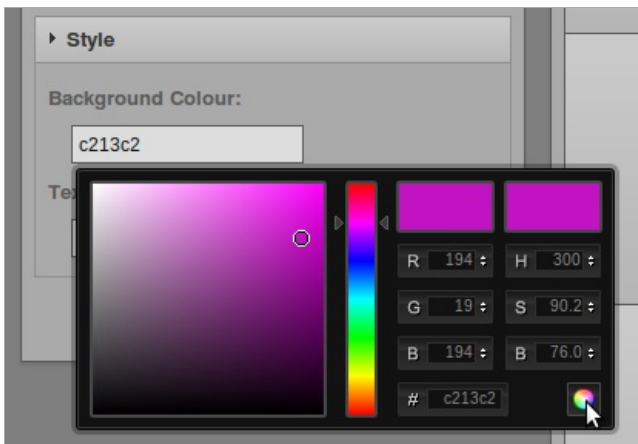
Finally, click the **Add this show** button at the top or bottom of the box. The new show will now be displayed in the **Calendar**, with a regular slot if you have chosen to schedule one.

EDITING A SHOW

Show configuration and metadata can be changed at any time, except for **Date/Time Start** and **Record from Line In?** options, which are fixed after broadcast of that show commences. Click the show in the Calendar, and select **Edit Show** from the pop-up context menu. This opens the **Update Show** box, which is almost exactly the same as the **Add this Show** box. Click the **+ Update show** button at the top or bottom of the box when you are done.
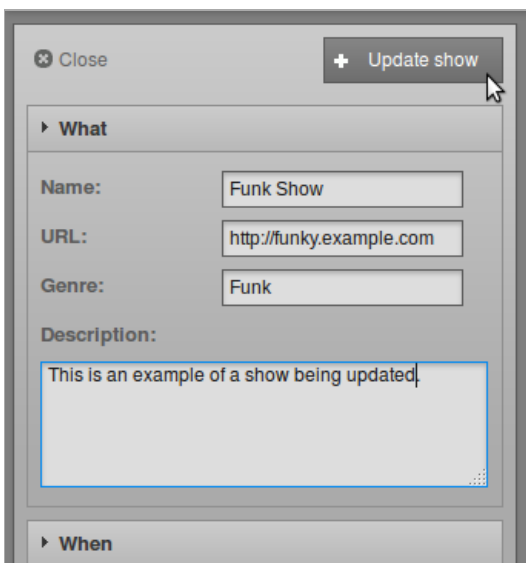
Alternatively, individual shows can be clicked on and dragged to new days and times in the calendar. However, Airtime will not allow you to drag a future show into the past, or drag and drop instances of a repeated show. In the **Day** and **Week** views, show length can be adjusted by clicking on the lower edge of the show box, and dragging the edge of the box upwards or downwards. The new show length is calculated automatically.

ADDING CONTENT TO A SHOW

To add content to a show, click the show in any view on the Calendar, and select **Add/Remove Content** from the pop-up menu. Shows that do not yet contain any scheduled content are marked with a red exclamation mark icon, to the right of the show start and end times in the top bar. Shows partially filled with content have a yellow exclamation mark icon. During playout of the show, a green play icon will also be shown in the top bar.
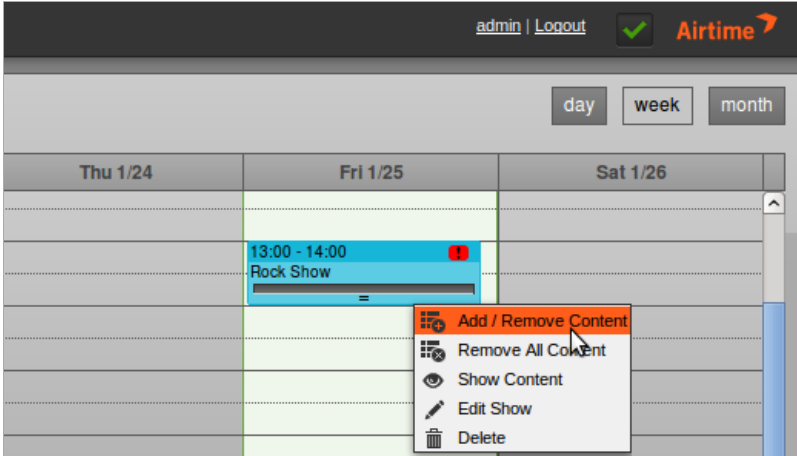
The **Add/Remove Content** action opens a window with the name of the show. Like when using the **Now Playing** page, you can search for content items and add them to the show schedule on the right side of the page. Refer to the *Now Playing* chapter for details.

When your show has all the required content, click the **OK** button in the bottom right corner to close the window. Back in the **Calendar**, click the show and select **Show content** from the pop-up menu to view a list of content now included in the show.
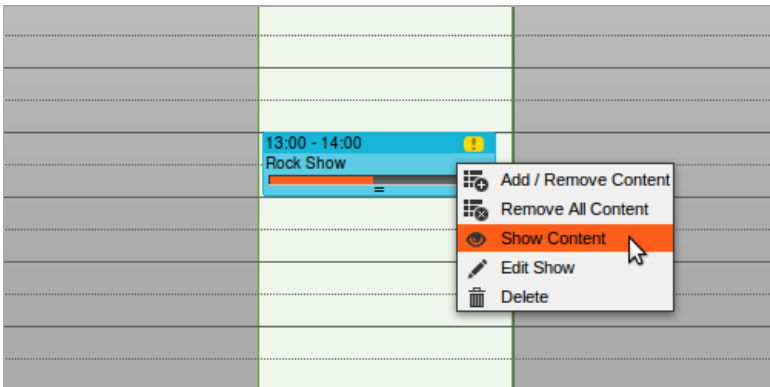
The **Contents of Show** window is a read-only interface featuring an orange bar which indicates how much media has been added to the show. Click the **OK** button in the bottom right corner, or the white **x** icon in the top right corner, to close the window.

REMOVING CONTENT FROM A SHOW

To remove an individual item from a show, click on the show in the **Calendar**, and select **Add/Remove Content** from the pop-up menu. In the window which opens, click any item you wish to remove from the show, then click **Delete** on the pop-up menu, or check the box in the item's row then click the **Trashcan** icon at the top of the table. To remove all files and playlists from a show, click on the show in the **Calendar**, and select **Remove All Content** from the pop-up menu.

DELETING A FORTHCOMING SHOW

To delete one forthcoming instance of a repeating show, click on the show in the **Calendar**, and select **Delete**, then **Delete This Instance** from the pop-up menu. If you wish to delete all future instances of a repeating show, select **Delete This Instance and All Following** from the pop-up menu.
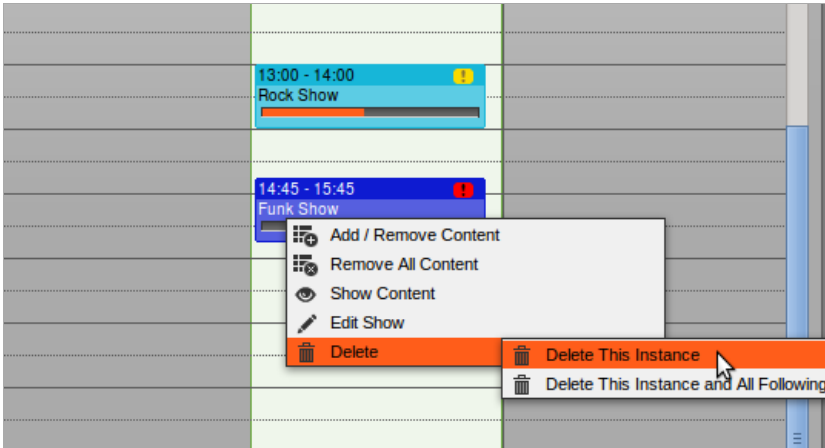
You cannot delete or remove content from shows that have already played out. These shows have only one option on the pop-up menu, which is **Show Content**.

CANCELLING PLAYOUT

If you wish to cancel playout of a show while it is running, click on the show in the **Calendar** and select **Cancel Current Show** from the pop-up menu. Airtime will ask you if you are sure about this action, as it cannot be undone.
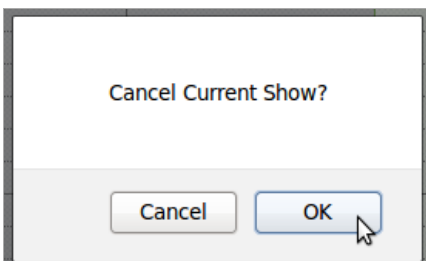
17. HISTORY

On the History menu, the **Playout History** page enables you to view a list of files played within a specific date and time range. This page is designed to help your station prepare reports for music royalty collection societies and regulatory agencies.

Search results can be copied to the clipboard using the **Copy** button, exported as data in **CSV** format (comma separated values), exported as a document in **PDF** format, or displayed in a printer-friendly format using the **Print** button. (Your web browser must have an Adobe Flash plugin installed for these buttons to appear). Press the **Esc** key to return to the Airtime interface once the print job is complete.

This page has three tabs: **Log Sheet**, **File Summary** and **Show Summary**. On any of these tabs, you can select a date and time range by clicking the calendar and clock icons in the upper left corner of the page. Then click the search button, which has a magnifying glass icon, to the right. A list of files played during that date and time range will appear further down the page.

In the **Log Sheet** tab, the playout history is sorted by **Start Time** and **End Time** by default.

The number of times each file was played and the length of the files are shown in the **File Summary** tab. To make optimal use of this feature for royalty reporting purposes, music files must be tagged with **Composer** and **Copyright** metadata. The artist performing a piece of music may not be the original composer of the work, or the copyright holder of the sound recording.

On the **Show Summary** tab, click the name of a show within the search range to expand its row and see its playout details.
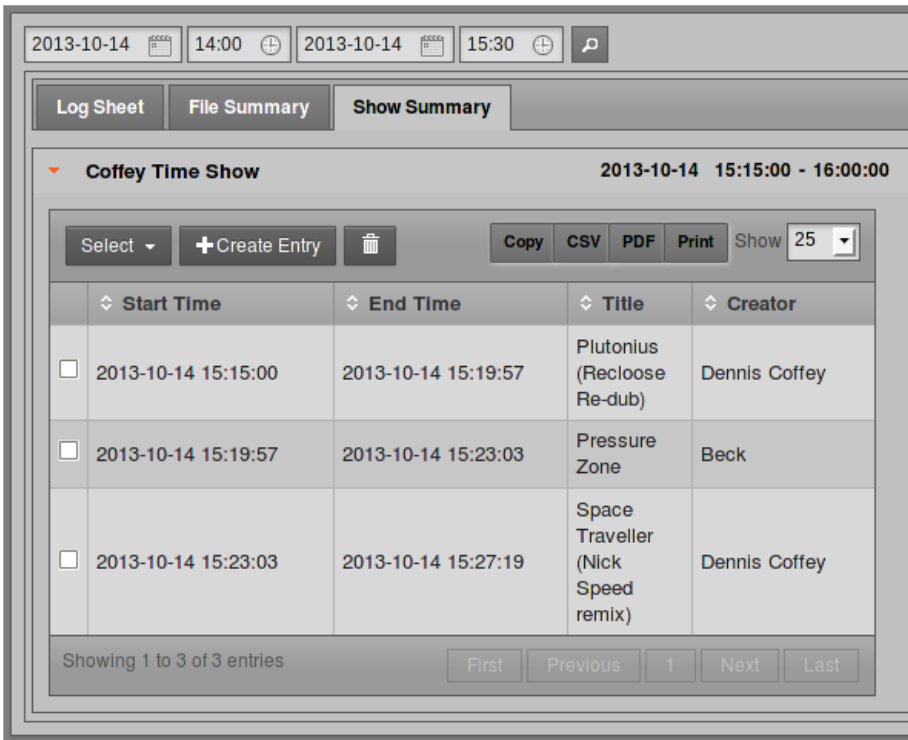
MANUAL LOGGING

If your station features playout from analogue sources such as turntables or microphones, there is no automatic metadata for Airtime to collect from these inputs. To ensure that the playout history is complete, you can add log entries manually by clicking the **+ Create Entry** button. This action opens a pop-up window with default fields of Start Time, End Time, Title and Creator. Click the**Find** button to automatically fill the **Choose Show Instance** menu with the names of shows that took place within the specified time range. Then click the **Save** button to enter the new item into the playout history.
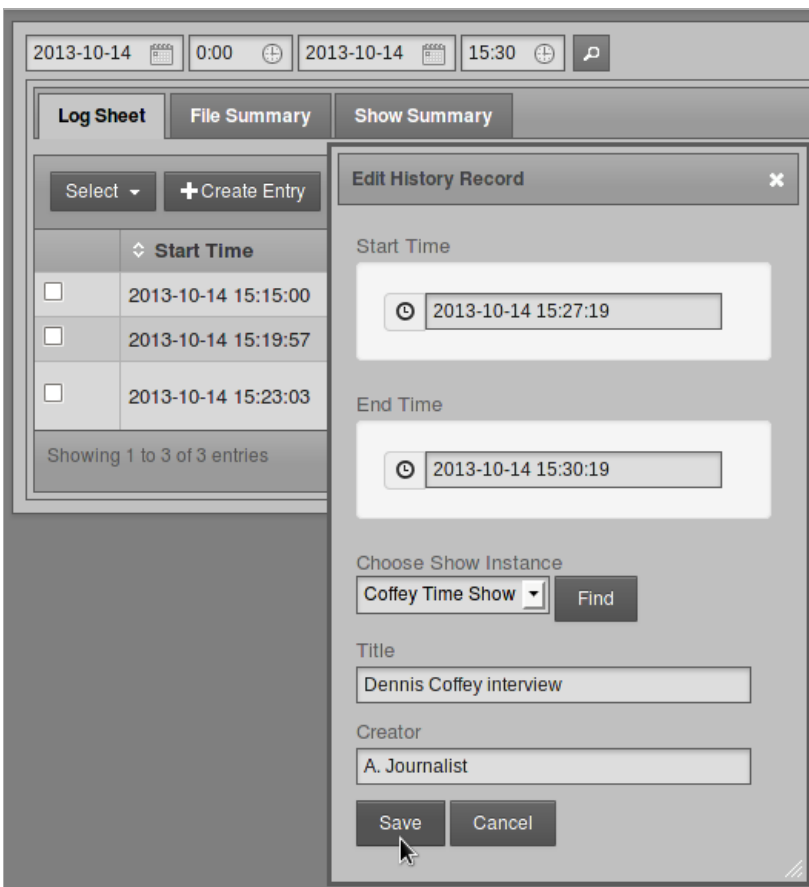
Log entries can also be manually deleted, using the button with the trashcan icon, to the right of the**+ Create Entry** button. Pages of entries can be selected for deletion using the **Select** drop-down menu.

74

HISTORY TEMPLATES

The **History Templates** page on the History menu enables you to prepare reports with the exact content required by regulatory agencies in the territories that you are broadcasting to. You can begin creating a custom template by clicking the button **New Log Sheet Template** or the button **New File Summary Template**.
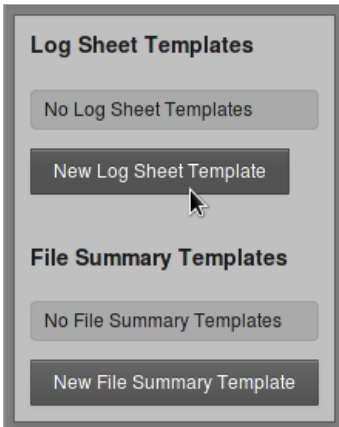
Either of these actions opens a page in which you can name the new template, and add or remove elements from the list on the left. To add a new element from the list on the right, click the plus icon for the item you require. If the element you require is not listed, you can use the **Add New Field** box at the lower end of the right side column. Select *string*, *boolean*, *integer*, or *float*, depending on the type of data that you wish to log, and then click the **+ Add** button.
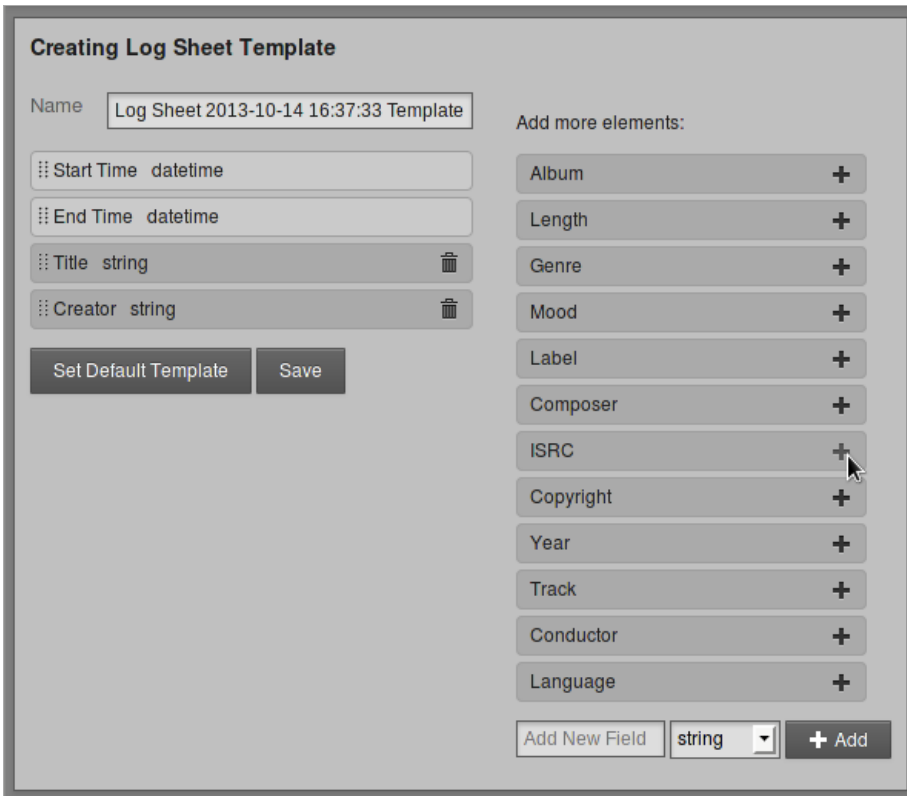
When the template is in the format you require, click the **Save** button, and **Set Default Template** if you wish. The new template will now be listed on the History Templates page. If you have set a new default template, any changes will be visible on the tabs of the Playout History page.

18. LISTEN

In the Master Panel, beneath the **ON AIR** indicator, you will find the **LISTEN** button.
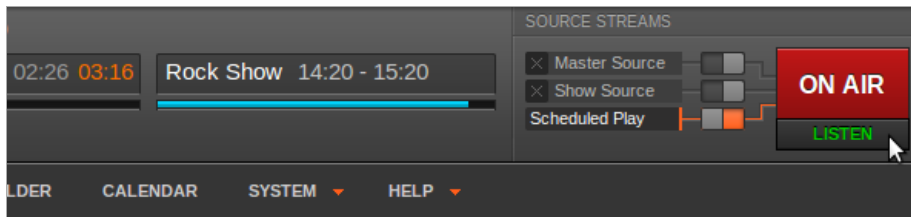
This button opens a pop-up **Live stream** window, which enables you to monitor the streams that have been configured previously in the **Streams** page on the **System** menu. In the **Live stream** window, a drop-down menu enables you to switch between the streams which are currently available. Both the streaming server and name of the stream are shown. Your station logo is shown in the top left corner of the window, if you have uploaded one via the **Support Feedback** page on the **System** menu.
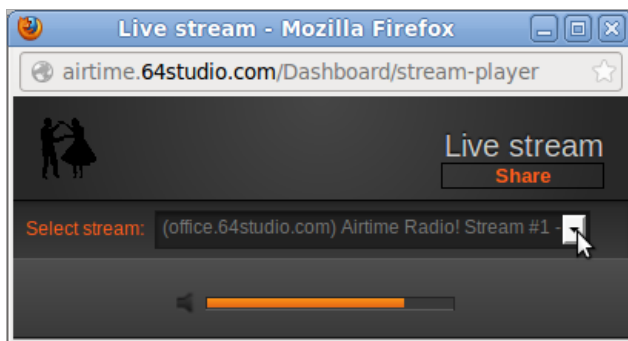
Beneath the drop-down menu for stream selection is an orange volume control bar. This volume control only adjusts the output level of the pop-up **Live Stream** window, not the output level of the Airtime server itself. To adjust output level between muted and maximum, click on the corresponding place in the orange bar, with maximum level on the right side. Click on the left side speaker icon to mute the output.
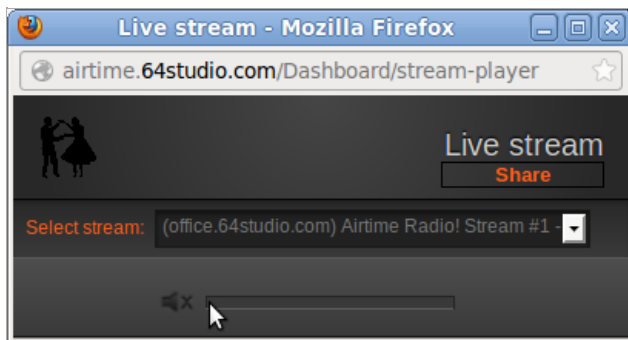
To display the URL of the stream you are monitoring, so that you can copy and paste it into an email or web page, click the **Share** button. Click the **X** icon to the right of the URL to return to the drop-down menu of available streams.
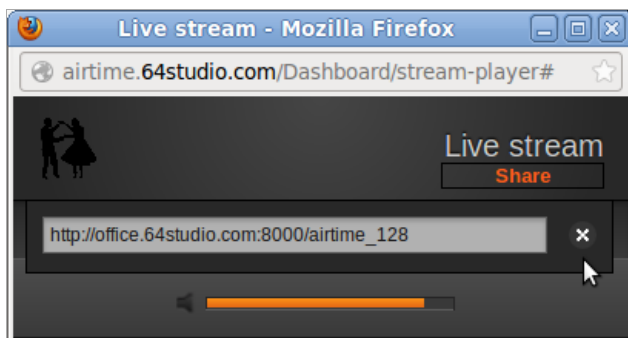
When you have finished monitoring the streams, you can close the pop-up window in the normal way, depending on the browser you are using. In Firefox, you can close the window by clicking the **X** button in the top right corner. This action will not shut down the output from the Airtime server, only the stream monitoring on your desktop computer or laptop.

19. HELP

The first entry on Airtime's **Help** menu offers a **Getting Started** guide for new users. Further down, there is also a link to the online version of this **User Manual** and an **About** page, which displays version and licensing information.
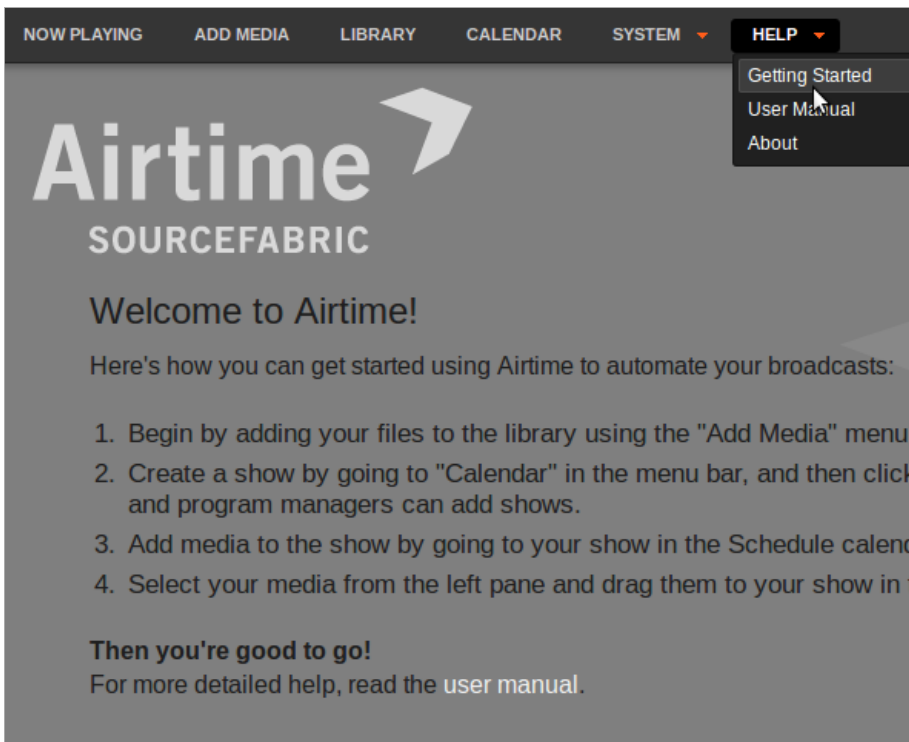
You can visit the Airtime online support forum, and sign up for the mailing list, at http://forum.sourcefabric.org/categories/airtime-support

This forum is mirrored by the mailing list, so posts on the forum appear on the mailing list and vice versa. You can therefore also post a message there by emailing airtime-support@lists.sourcefabric.org

To subscribe to forum updates via email, please register or login to the forum by clicking the appropriate link. Then visit your profile page at http://forum.sourcefabric.org/profile/ and click the link **My Email Subscriptions** on the left side menu. In the pop-up window that opens, check the boxes for the mailing lists you wish to subscribe to, then click the **Save** button.

BUG REPORTING

Airtime needs your input to improve. If you think you've found a bug, please visithttp://dev.sourcefabric.org/ and sign in, using the same login and password that you registered for the Airtime forum. Create a bug report by selecting **Create Issue**, then **Airtime**, and then **Bug**. That way, the Airtime team can keep track of your problem and notify you when it has been fixed. You can also suggest improvements and new features for Airtime on that site.

CONTACT

Finally, when all other avenues have been exhausted, email us directly at contact@sourcefabric.org and we'll try to help!

OTHER HELP

The UNESCO publication *Community Radio - A user's guide to the technology*: http://sourcefabric.booktype.pro/airtime-25-for-broadcasters/help/static/CommunityRadioUserGuide.pdf features a very comprehensive guide to setting up a community radio station. This guide is aimed at people thinking about setting up a radio station in India, but includes lots of practical advice that would be useful in any country.

AIRTIME IN THE STUDIO

20. RECORDING SHOWS

Live show recording from the input of the server's sound card, if one is fitted, can be enabled in the **Add Show** box of Airtime's **Calendar** (see the *Calendar* chapter for details). During a recording, a red light is shown in the Master Panel, and the word **Recording** appears in a red font to the left of the show name.
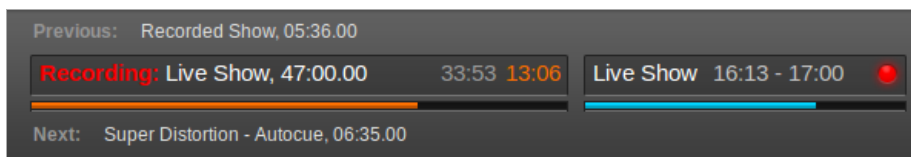
Before the first broadcast show scheduled for recording begins, you should check that the level of recording is sufficient to ensure a good signal to noise ratio, but not so high a level that clipping (distortion) occurs in the recorded file. You can perform this check using the command **alsamixer** on the Airtime server. This command opens a soundcard mixer application in the server console.
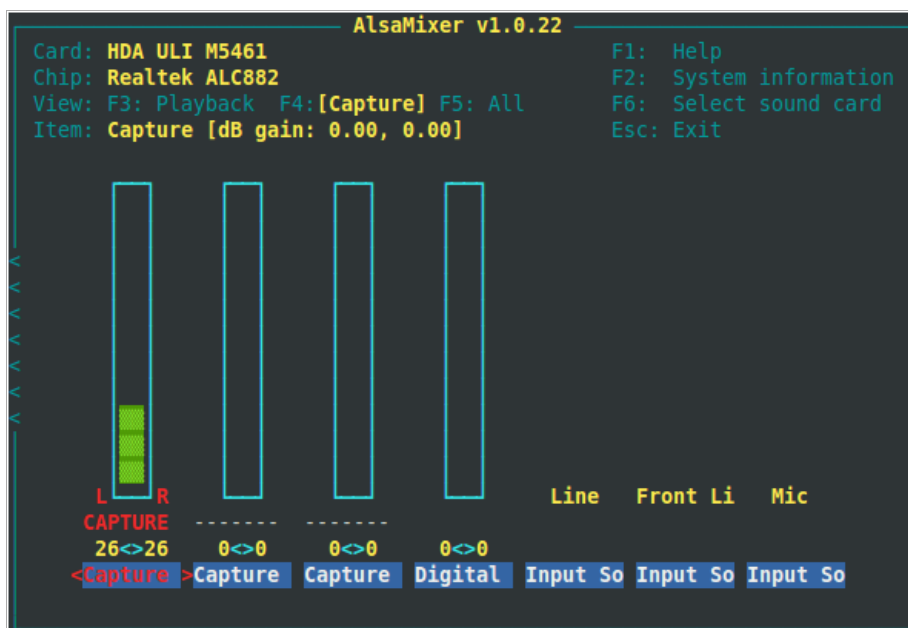
Press the **F4** key on your keyboard to set capture levels. Some experimentation may be required to find the correct control on the mixer, using the **Left** and **Right** arrow keys on your keyboard. Levels are set with the **Up** and **Down** arrows, and *Capture* is toggled with the **Space** bar. *Capture dB gain* should be set to *0.00, 0.00* initially.

If you hear nothing at all in the recording, you may need to set the value of *Input Source* to *Line*, using the **Up** or **Down** arrows. Depending on the particular sound card and sockets you are using, you may have to enable other inputs, such as *Digital* or *S/PDIF*. If a test recording is too quiet, try raising the line output level of your broadcast mixer towards *0dB* before increasing gain above *0dB* on the sound card, in order to achieve the optimal gain structure.

USING RECORDINGS

After the recording has finished, you can find the recorded file in the **Library**, by searching for *recorder*. The file will be labelled with a **Title** containing a date and time stamp, as well as the name of the show.
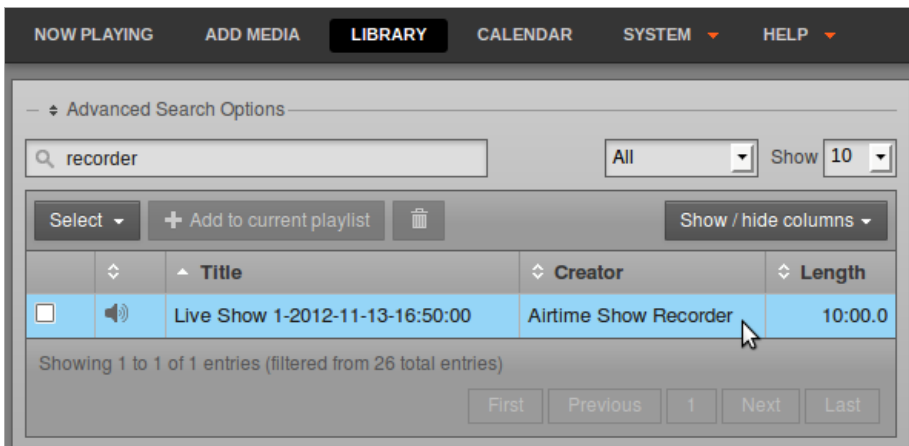
Click a file and select **Edit Metadata** from the pop-up menu to enter additional details of the recording which will help you find it in searches later, such as **Album** or **Language**. Then click the **Save** button.
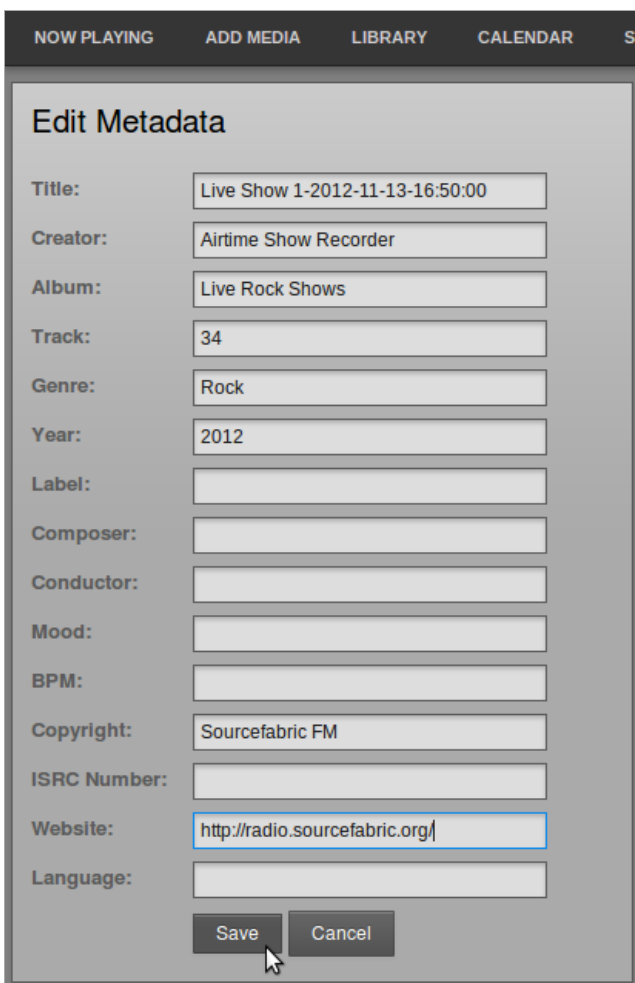
CANCELLING A RECORDING

If you wish to cancel the recording of a live show, click on the show in the**Calendar** and select **Cancel Current Show** from the pop-up menu.

Airtime will ask you if you are sure about this action, as it cannot be undone. The recorded show file in the Airtime library will be truncated if you click the **OK** button.
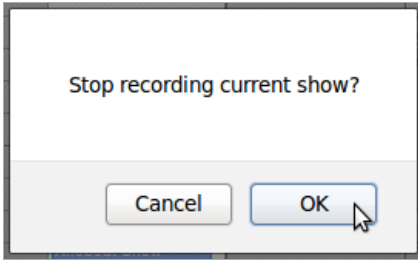
21. LIVE SHOWS WITH MIXXX

Mixxx is a cross-platform Open Source application for DJs, available from http://www.mixxx.org/

Installed on a desktop or laptop computer, Mixxx complements your Airtime server to provide a complete system for both live and scheduled broadcasting. Although Mixxx has many features designed for dance music DJs that require beat matching and pitch independent time stretching, the program can be used for any kind of manually triggered broadcast playout, including live speech shows such as news or current affairs.

Mixxx supports a wide variety of popular hardware control surfaces, which can be connected to your computer using a USB cable. A control surface might replace or augment an analogue mixer in your studio, depending on your live mixing and playout requirements.

SOUND CARDS

If you wish to use Airtime and Mixxx on the same machine, you will need two or more sound cards, as each program requires exclusive access to the sound hardware. Otherwise, you may see an error message that Mixxx cannot access the sound device.
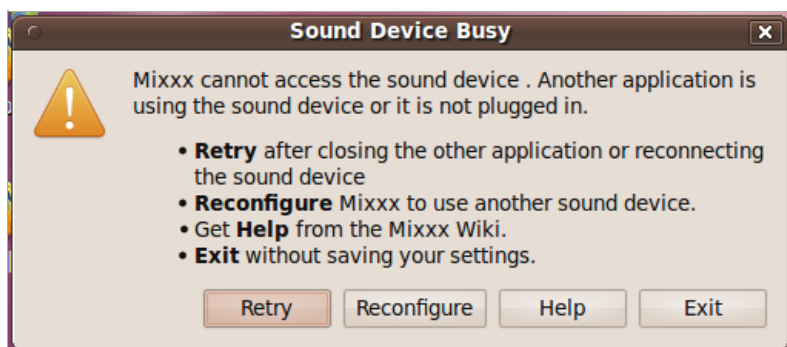
The solution is to configure Mixxx to use additional sound cards by clicking **Options**, **Preferences**, then **Sound Hardware** in the main Mixxx menu. Select devices other than the ALSA default of *hw:0,0* for at least the **Master** and **Headphones** outputs. Then click the **OK** button.
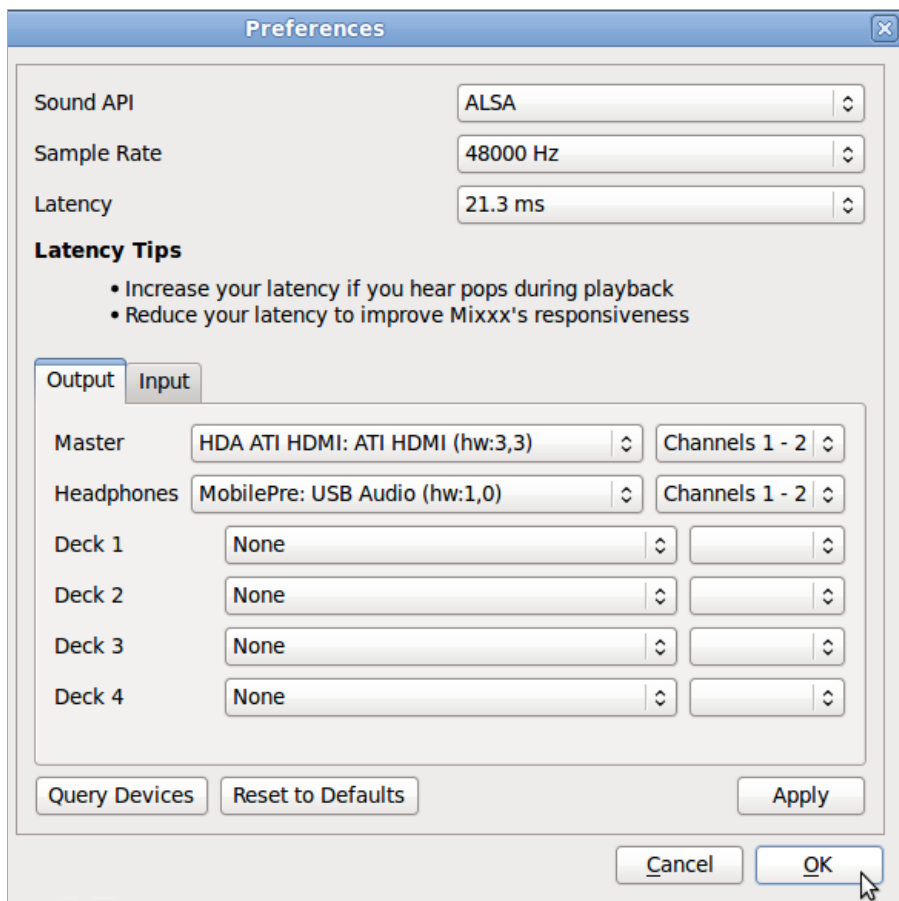
SHARING STORAGE

If you make the Airtime server's storage directory */srv/airtime/stor/* accessible to a desktop machine as a read-only location, Mixxx will accept that location as its default music library when starting up for the first time. (This location can also be configured after installation by clicking **Options**, **Preferences**, then **Library** in the main Mixxx menu).
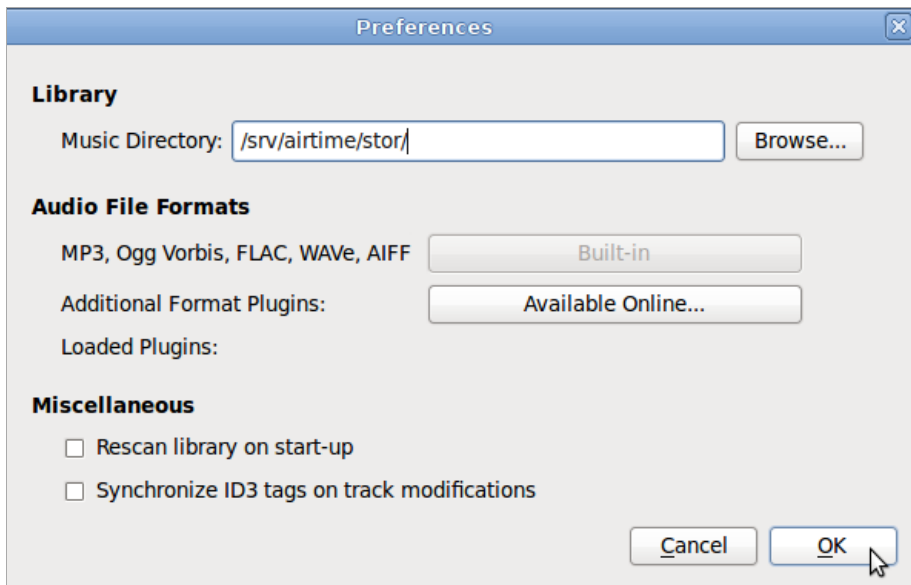
You may need to adjust file and directory permissions so that the storage directory has read access from the desktop user account. Enabling write access directly to the storage server is not recommended, as this would allow desktop users to delete files which might be needed for playout later.

If the filesystem path has been configured correctly, the metadata for the files in the Airtime storage server will be displayed in the main window of the Mixxx interface. Individual files from the Airtime storage server can then be added to either of Mixxx's live players with a right-click on the filename, or by using the appropriate hardware buttons on a control surface. Therefore it is possible to manage the station's storage archive remotely and collaboratively through Airtime, while using Mixxx as the live playout client in multiple, remote studios.

The Airtime storage archive can be exported like any other file server share. The method that you implement would depend on the operating system of your desktop client machines, and whether they were on the same local network as the Airtime server, or remote.

For performance and redundancy reasons it is advisable to cache files required for a particular show on the client machine where Mixxx is installed. For example, for a GNU/Linux client machine, a nightly **rsync** download of new media in the archive would guard against network problems at playout time potentially disrupting a broadcast at a remote studio.

Mixxx users can also record a show, and then upload it through the Airtime web interface on a local or remote server for collaborative or user-generated broadcasts.

STREAMING FROM MIXXX INTO AIRTIME

Mixxx 1.9.0 or later includes a live streaming client which, like Airtime, is compatible with the **Icecast** and **SHOUTcast** media servers. This feature can also be used to stream from Mixxx directly into Airtime, using either the **Show Source** or **Master Source**.

To configure **Mixxx** for streaming into Airtime, click **Options**, **Preferences**, then **Live Broadcasting** on the main Mixxx menu. For server **Type**, select the default of **Icecast 2**. For **Host**, **Mount**, **Port**, **Login** and **Password**, use the **Input Stream Settings** configured in the Airtime **Streams** page, on Airtime's **System** menu. See the chapter *Stream settings* for remote input connection details.

AIRTIME SKINS FOR MIXXX

Airtime-themed skins for Mixxx, designed with broadcast users in mind, are available for download from https://github.com/Airtime/MixxxSkins

These skins provide a simplified interface for live broadcasting which do away with EQ, flange effect, looping and other features required by dance music DJs. Instead, the emphasis is on a clear and uncluttered interface which does not require large mouse movements to operate the most important controls. There are versions available both with and without pitch/tempo controls for beat matching.
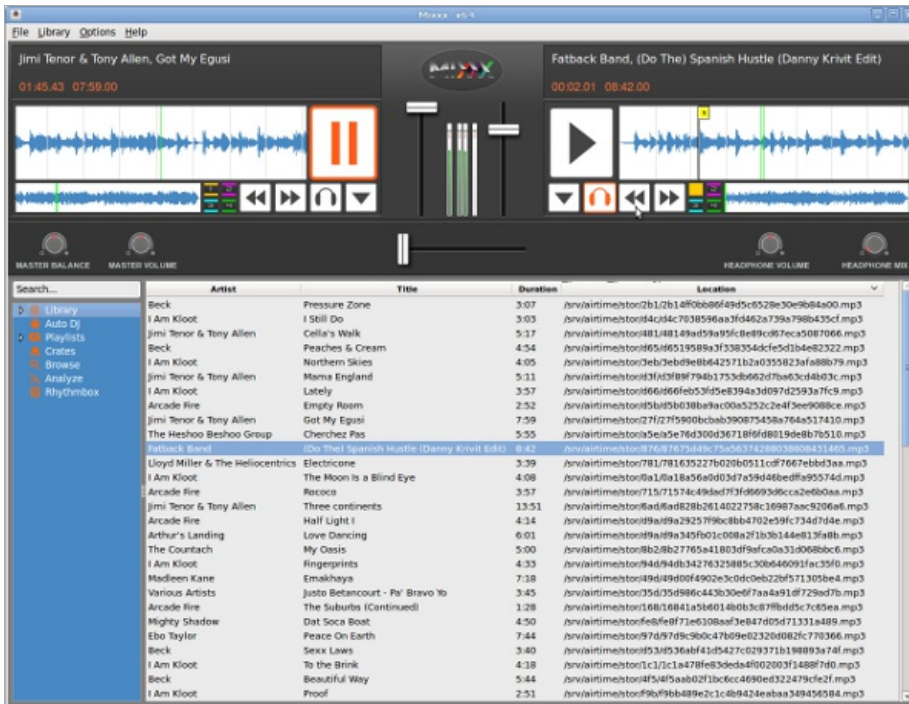
After downloading one of the skins, extract the zip file and copy it to the *skins* directory on the computer where Mixxx is installed. For example, on Debian or Ubuntu:

```
unzip Airtime1280x1024_skin_for_Mixxx.zip
sudo cp -r Airtime1280x1024 /usr/share/mixxx/skins/
```

Then, start Mixxx and select the Airtime skin by clicking **Options**, **Preferences**, then **Interface** in the Mixxx main menu.
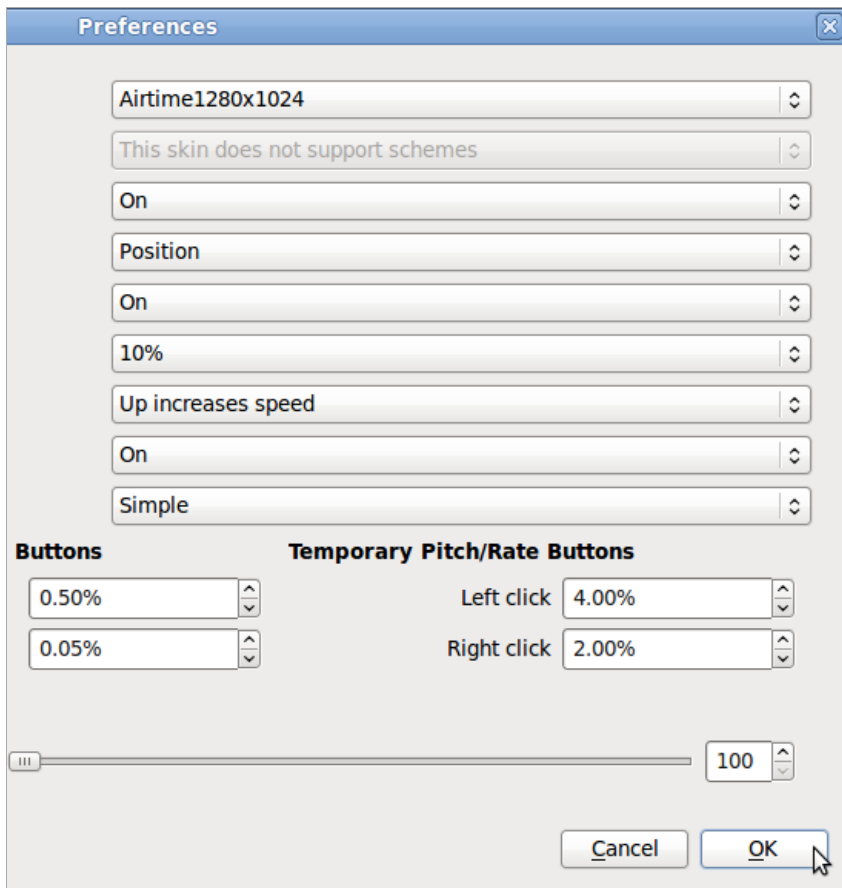
22. ICECAST AND SHOUTCAST

Airtime supports direct connection to two popular streaming media servers, the open source**Icecast** (http://www.icecast.org) and the proprietary **SHOUTcast** (http://www.shoutcast.com). Apart from the software license, the main difference between these two servers is that Icecast supports simultaneous MP3, AAC, Ogg Vorbis or Ogg Opus streaming from Airtime, whereas SHOUTcast supports MP3 and AAC streams but not Ogg Vorbis or Opus. The royalty-free Ogg Vorbis format has the advantage of better sound quality than MP3 at lower bitrates, which has a direct impact on the amount of bandwidth that your station will require to serve the same number of listeners. Ogg Opus also benefits from good sound quality at low bitrates, with the added advantage of lower latency than other streaming formats. Opus is now an IETF standard (http://tools.ietf.org/html/rfc6716) and requires Icecast 2.4 or later to be installed on the streaming server.

Ogg Vorbis playback is supported in **Mozilla Firefox**, **Google Chrome** and **Opera** browsers, via **jPlayer** (http://jplayer.org/), and is also supported in several popular media players, including VideoLAN Client, also known as VLC (http://www.videolan.org/vlc/). (See the chapter *Stream player for your website* on how to deliver **jPlayer** to your audience). Ogg Opus is relatively new and is supported natively in the very latest browsers, such as Mozilla Firefox 25.0, and media players including VLC 2.0.4 or later.

Streaming MP3 below a bitrate of 128kbps is not recommended for music, because of a perceptible loss of high audio frequencies in the broadcast playout. A 96kbps or 64kbps MP3 stream may be acceptable for voice broadcasts if there is a requirement for compatibility with legacy hardware playback devices which do not support Ogg Vorbis or Opus streams.

Because Airtime supports simultaneous streaming in multiple formats, it is possible to offer one or more streams via your website, and another independent stream for direct connection from hardware players. You can test whether Ogg streams sound better at low bitrates for yourself, by using the **LISTEN** button in Airtime's **Master Panel** to switch between streaming formats.

Conversely, you may have a music station which wants to stream at 160kbps or 192kbps to offer a quality advantage over stations streaming at 128kbps or less. Since Ogg, AAC and MP3 formats use lossy compression, listeners will only hear the benefit of higher streaming bitrates if the media files in the Airtime storage server are encoded at an equivalent bitrate, or higher.

UTF-8 METADATA IN ICECAST MP3 STREAMS

When sending metadata about your stream to an Icecast server in non-Latin alphabets, you may find that Icecast does not display the characters correctly for an MP3 stream, even though they are displayed correctly for an Ogg Vorbis stream. In the following screenshot, Russian characters are being displayed incorrectly in the *Current Song* field for the MP3 stream:

Mount Point /airtime.mp3

Stream Title:	airtime.mp3
Stream Description:	Airtime Radio!
Content Type:	audio/mpeg
Mount started:	Sun, 18 Sep 2011 18:59:23 -0400
Bitrate:	128
Current Listeners:	0
Peak Listeners:	0
Stream Genre:	genre
Stream URL:	**http://airtime.sourcefabric.org**
Current Song:	Ð ÐµÐ°Ð»ÑÐ½ÑÐ¹ ÑÐ°Ð·Ð³Ð¾Ð²Ð¾Ñ - Anthrax - I'm

Mount Point /airtime.ogg

Stream Title:	airtime.ogg
Stream Description:	Airtime Radio!
Content Type:	application/ogg
Mount started:	Sun, 18 Sep 2011 18:59:23 -0400
Quality:	0.3
Current Listeners:	0
Peak Listeners:	0
Stream Genre:	genre
Stream URL:	**http://airtime.sourcefabric.org**
Current Song:	Реальный разговор - Anthrax - I'm The Man

The solution is to specify that the metadata for the MP3 mount point you are using should be interpreted using UTF-8 encoding. You can do this by adding the following stanza to the */etc/icecast2/icecast.xml* file, where *airtime.mp3* is the name of your mount point:

```
<mount>
     <mount-name>/airtime.mp3</mount-name>
     <charset>UTF-8</charset>
</mount>
```

After saving the */etc/icecast2/icecast.xml* file, you should restart the Icecast server:

```
sudo invoke-rc.d icecast2 restart
Restarting icecast2: Starting icecast2
Detaching from the console
icecast2.
```

INSTALLATION

23. PREPARING THE SERVER

The following instructions assume that you have root access (**sudo** on Ubuntu) to a GNU/Linux server, and are familiar with basic command line tasks. Experienced system administrators may prefer to skip to the *Expert install* chapter in the appendix of this book after preparing the server as shown in the steps below.

The recommended Airtime server platform is Debian 7.2 'wheezy'. Ubuntu 12.04 LTS 'Precise Pangolin' is also supported. Users of other GNU/Linux distributions may be able to adapt these instructions to suit their needs.

The server should have at least a 1GHz processor and 512MB of RAM, preferably 1GB RAM or more. If you are using a desktop environment and web browser directly on the server you should install at least 2GB RAM, to avoid swapping to disk.

The Airtime installation does not use much disk space, but you should allow plenty of storage capacity for the Airtime library. A hot-swap RAID array is recommended for media storage, in case of disk failure. You should also consider a UPS or other battery-powered system to offer some protection against short-term power failures.

SOUND CARDS

If your Airtime machine will only be used to stream directly to an **Icecast** or **SHOUTcast** streaming media server, you do not require a sound card to be installed on the Airtime server side. This option is suitable for Airtime installations at your ISP's data centre, remote from any transmitter. However, you will not be able to take advantage of Airtime's live show recording feature (yet).

If you intend that your Airtime server will have a direct audio output to a broadcast transmitter or a separate stream encoder, your server machine must have a sound card supported by an **ALSA** driver. Almost all standard sound cards have ALSA drivers built into the Linux kernel, which do not need to be installed separately. If in doubt about driver support for your sound card, check the ALSA support matrix at: http://www.alsa-project.org/main/index.php/Matrix:Main

USB AUDIO DEVICE INDEX

Some server motherboards do not have a default ALSA device (index number zero, shown as hw:0,0 in ALSA notation), and a USB sound card is often prevented from getting index zero by the GNU/Linux distribution's configuration. This setting may be in a file such as */etc/modprobe.d/alsa-base.conf* in Ubuntu, which can be edited with **nano**:

```
sudo nano /etc/modprobe.d/alsa-base.conf
```

Comment out the lines beginning *options snd-usb-* to fix the problem:

```
# Prevent abnormal drivers from grabbing index 0

# options snd-usb-audio index=-2
# options snd-usb-us122l index=-2
# options snd-usb-usx2y index=-2
# options snd-usb-caiaq index=-2
```

Save the file with **Ctrl+O** and close **nano** with **Ctrl+X**. Then remove and re-insert the cable connecting the USB sound card to the server (making sure any mixer or amplifier connected is faded down, to avoid a 'thump' on the output). The command **aplay -l** should now confirm that the *USB Audio* device has index 0 (shown as card 0, device 0):

```
aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: MobilePre [MobilePre], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

When using a USB sound card with ALSA, some how-to documents advocate setting the *nrpacks=1* option, but this is not recommended for Airtime because it can increase CPU load significantly.

INTEL HDA MIXER LABELS

If you have an **Intel HDA** sound card, as built in to many contemporary motherboards, you may discover that the recording controls in **alsamixer** have incorrect labels. This problem can make it difficult to adjust mixer levels except by trial and error. To fix these labels, you can pass a specific *model=* parameter to the *snd-hda-intel* module of the Linux kernel. On Debian or Ubuntu GNU/Linux, you can do this by first identifying the model you have, with the command:

```
cat /proc/asound/card0/codec* | grep Codec
```

The server should respond with a line such as:

```
Codec: Realtek ALC882
```

Referring to the appendix *HD Audio Models* in this book, find the matching codec and model. In this example, the codec is *ALC882* and the motherboard has six analogue jacks and two S/PDIF sockets, so the model is *6stack-dig*.

```
ALC882/883/885/888/889
=====================
  3stack-dig 3-jack with SPDIF I/O
  6stack-dig 6-jack digital with SPDIF I/O
```

Edit the file */etc/modprobe.d/alsa-base.conf* with **nano** as follows:

```
sudo nano /etc/modprobe.d/alsa-base.conf
```

Add an appropriate line for your soundcard model to the end of the file, such as:

```
# Realtek ALC882
options snd-hda-intel model=6stack-dig
```

Save the file with **Ctrl+O** and close **nano** with **Ctrl+X**. Then reboot the server. After the reboot, you should now see that the mixer controls are correctly labelled.

DISABLE DESKTOP AND OTHER SOUNDS

If you are installing Airtime on a desktop computer, make sure you disable or remove any programs that could send unintended audio to a sound card you are using for broadcast output. This includes alert sounds which play when the computer is ready for use, or when a user logs in. On Ubuntu, these sounds are configured using *System -> Preferences -> Sound* on the main desktop menu. (This configuration tool only works when the PulseAudio sound server is installed).

You may prefer to remove all system sound files from the computer, in case they could be played unintentionally via the sound card. For example, on Ubuntu you may wish to remove the **ubuntu-sounds** package, with the following command:

```
sudo apt-get purge ubuntu-sounds
```

PULSEAUDIO

The PulseAudio sound server is not recommended for Airtime sound card output, but is installed by default on Ubuntu. To remove PulseAudio from an Ubuntu machine, type the following command:

```
sudo apt-get purge pulseaudio
```

Removing the **pulseaudio** package on a desktop Ubuntu machine may force the removal of the **ubuntu-desktop** metapackage. However, this metapackage is only installed on the system for managing upgrades; removing it does not remove the entire desktop.

After removing PulseAudio, if your Airtime machine has a desktop, you can install a mixer applet which can control the ALSA soundcard driver directly, such as **gnome-alsamixer**:

```
sudo apt-get install gnome-alsamixer
```

On a server with a sound card but without a desktop, you can control sound input and output levels using the command line program **alsamixer**:

This program should already be installed on an Ubuntu or Debian machine with a sound card. If not, you can install it with the command:

```
sudo apt-get install alsa-utils
```

THE PHP CACHE

This dependency, introduced in Airtime 2.5.0, can improve the responsiveness of the Airtime administration interface and reduce server load. Various PHP cache software is available, but the current cache software required by Airtime is **APC** (http://pecl.php.net/package/APC). You should uninstall or disable any other PHP cache that may be present on the server, such as **XCache**, before installing Airtime and APC. This is because having multiple caches enabled can cause problems for PHP.

You can remove the XCache package **php5-xcache** from a Debian or Ubuntu server with the command:

```
sudo apt-get remove php5-xcache
```

APC is available in Debian or Ubuntu as the **php-apc** package. If you already have Apache installed on the server you are going to use for Airtime, you can install APC and get it working with the following commands:

```
sudo apt-get install php-apc
sudo invoke-rc.d apache2 restart
```

APC has an administration page, which is not installed by default. If you wish, you can install and configure it with the following commands:

```
sudo cp /usr/share/doc/php-apc/apc.php /usr/share/airtime/public/
sudo nano /usr/share/airtime/public/apc.php
```

You should set an admin password for the *apc* user on line 42 of the file, such as *mynewpassword* in this example:

```
 defaults('ADMIN_PASSWORD','mynewpassword');
```

Save the *apc.php* file with **Ctrl+O** and close it with **Ctrl+X**. Then open a web browser at the *apc.php* page of your Airtime server, such as:

```
http://airtime.example.com/apc.php
```

You will then be able to confirm that APC is working, and view some statistics for the cache.
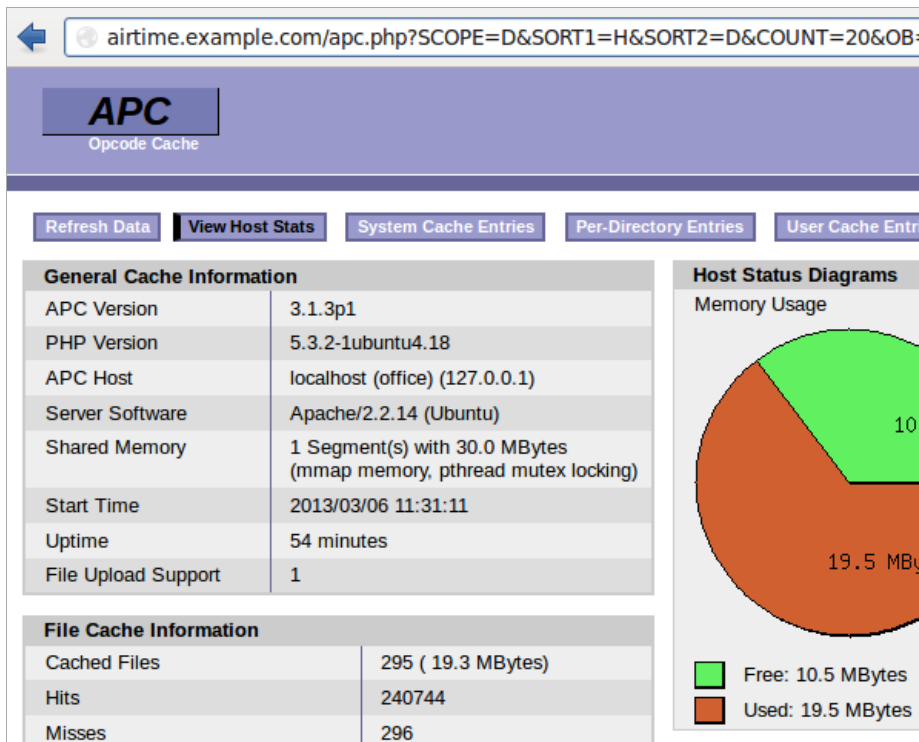
REMOVE WEBMIN, IF INSTALLED

The **webmin** control panel (http://www.webmin.com) has been known to remove Apache and PHP packages on Debian and Ubuntu systems, which can cause the Airtime package to be removed in turn. This problem is easily reversed by re-installation of the affected packages, but it has the potential to disrupt your broadcast playout from Airtime. Webmin is not likely to be installed on your server unless your system administrator has installed it manually. This is because webmin was removed from official Debian and Ubuntu package repositories some years ago.

RABBITMQ HOSTNAME

RabbitMQ requires a fixed and resolvable hostname (see http://www.rabbitmq.com/ec2.html#issues-hostname), which is normal for a server. For a desktop or laptop machine where the hostname changes frequently or is not resolvable, this issue may prevent RabbitMQ from starting. When using a desktop or laptop computer with a dynamic IP address, such as an address obtained from a wireless network, the **rabbitmq-server** daemon must not start up before the **NetworkManager** service.

RABBITMQ ON DEBIAN SQUEEZE

In Debian 6.0 'squeeze' the **rabbitmq-server** daemon does not start automatically after a reboot. This should be fixed before installing Airtime, to prevent problems at playout time. If the rabbitmq-server package was installed before the last reboot, you will need to run:

```
invoke-rc.d rabbitmq-server start
```

as the root user before the installation of Airtime. If it is not already installed, run the following command as root:

```
apt-get install rabbitmq-server
```

After a fresh installation, rabbitmq-server will start automatically, so there is no need to run the **invoke-rc.d** command mentioned above.

In either case, you should then edit lines 13 and 14 of the file*/etc/init.d/rabbitmq-server* (as root) to show:

```
# Default-Start:    2 3 4 5
# Default-Stop:     0 1 6
```

and then run the command (as root):

```
update-rc.d rabbitmq-server defaults
```

This should make sure that **rabbitmq-server** starts after the next reboot.

LIBAV ON DEBIAN SQUEEZE AND .M4A FILES

The version of libav shipped with Debian 6.0 'squeeze' is known to cause problems when Airtime analyses .m4a files containing the AAC codec for silence, using Silan (see https://github.com/x42/silan/issues/2). It is recommended that Debian servers are upgraded to the current stable 'wheezy' 7.2 release. As a short-term measure, updated libav packages from the squeeze-backports repository can be used to work around this problem. This work-around is only required if you do not wish to upgrade to 'wheezy' yet, but have AAC codec .m4a files in your Airtime library.

MAIL SERVER

If you wish Airtime to be able to send password reminder emails to your staff users, the server can have an SMTP program installed, such as the **postfix** package in Debian or Ubuntu. For configuration options, refer to the documentation for your mail server of choice. Alternatively, you can specify an external SMTP server in the **Preferences** page on Airtime's **System** menu.

PROCEED TO INSTALLATION

Now your server should be ready for Airtime to be installed. Depending on your requirements, you should now proceed to either the chapter *Easy setup*, the chapter *Automated installation,* the chapter *Manual installation* or the *Expert install* appendix.

24. EASY SETUP

The **airtime-easy-setup** package sets up a typical Airtime configuration without prompting you for any settings. You can then install the latest Airtime package from the Sourcefabric repository with a few clicks.

After that, the Airtime administration interface can be accessed at the default domain name of the computer (such as *http://airtime.example.com*). If you have not yet set a fully qualified domain name for the computer, you can use a URL such as *http://ubuntu/* on the localhost for testing purposes. Whichever domain name you use, the Icecast administration interface will be at port 8000 of the same URL (such as *http://airtime.example.com:8000*).

You can download the **airtime-easy-setup** package from http://apt.sourcefabric.org/misc/airtime-easy-setup.deb which is a link to the latest version of the package. You should stay connected to the Internet throughout the installation, as a number of dependency packages have to be installed from online repositories.

On Ubuntu 10.04 'lucid', Debian 6.0 'squeeze' or Debian 7.2 'wheezy', you can run the**airtime-easy-setup** package from your browser using the program **GDebi**.
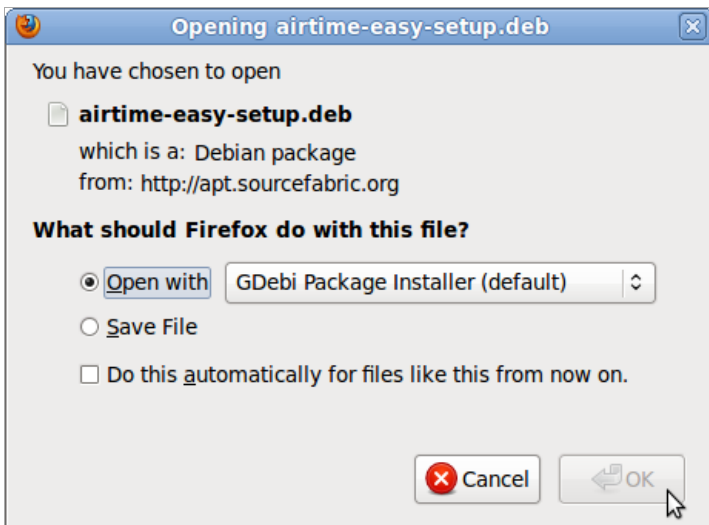
If you have chosen to save the package to your computer instead, in the desktop file manager, right-click on the **airtime-easy-setup** package and select **Open with GDebi Package Installer**:
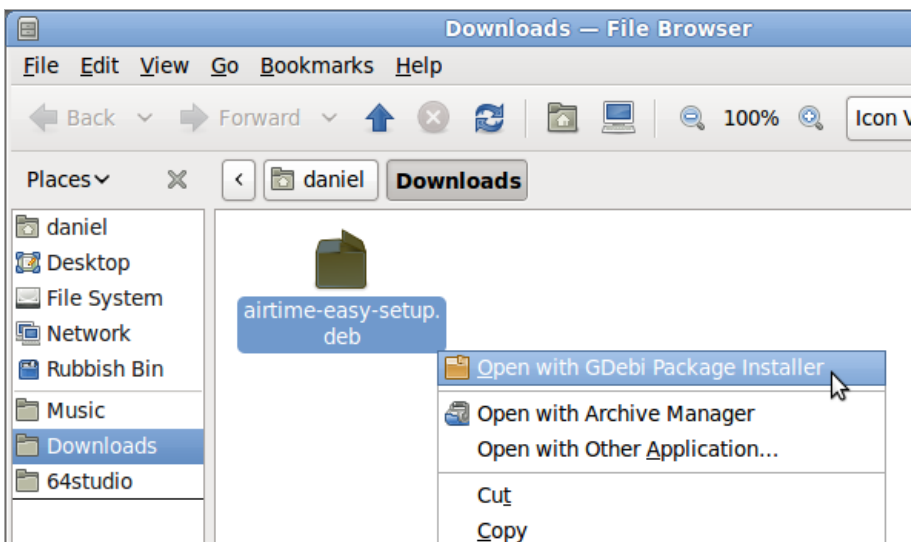
Or for an Ubuntu Lucid or Debian server without a desktop, you can use**gdebi** on the command line:

```
sudo apt-get update
sudo gdebi airtime-easy-setup.deb
```

If gdebi is not installed, you can run this command first:

```
sudo apt-get install gdebi
```

Later versions of Ubuntu have replaced GDebi with the program **software-center**. On a desktop installation, right-click on the package and select **Open with Ubuntu Software Centre**. This desktop program can also be run from the command line, for example:

```
software-center airtime-easy-setup.deb
```
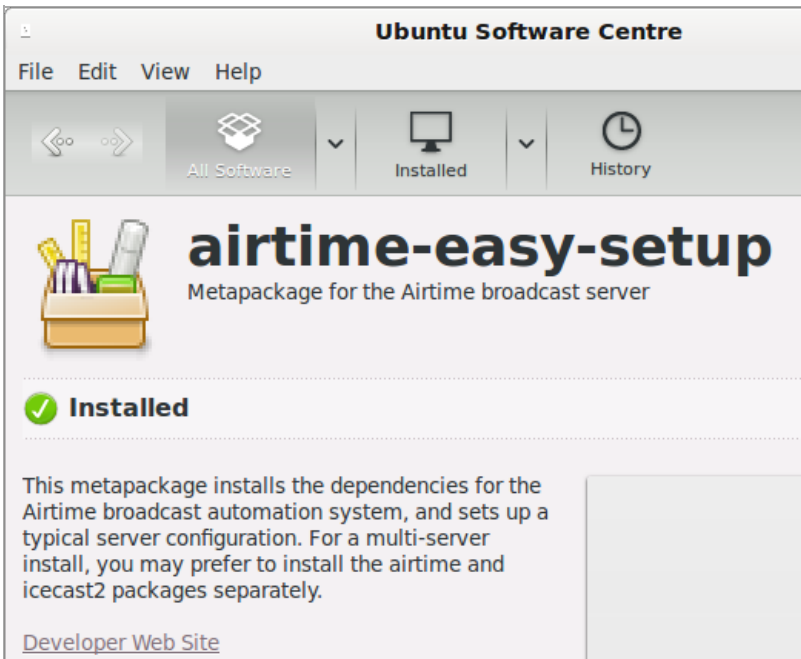

Any of the methods above should resolve package dependencies automatically.

INSTALL AIRTIME

Once the installation of **airtime-easy-setup** has completed, you can install the latest Airtime package on a desktop system using your usual package manager, such as **Ubuntu Software Centre**, or **Synaptic** on Debian.
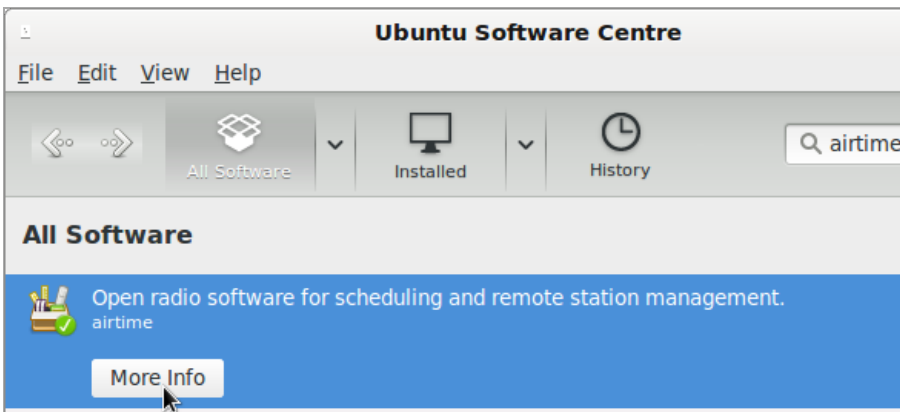
On a server, you can use the command:

```
sudo airtime-easy-setup
```

CONFIGURE FOR PRODUCTION

To convert a test installation into a production installation, you can run the command:

```
sudo dpkg-reconfigure airtime
```

The **dkpg-reconfigure** command will run through the configuration steps shown in the *Automated installation* chapter, so that you can set the correct hostnames and passwords for your production Airtime server.

25. AUTOMATED INSTALLATION

This installation method is intended for computers running Ubuntu or Debian GNU/Linux, and is the recommended method for production Airtime systems. If you have previously performed a manual installation of Airtime on the server, you should run the *airtime-uninstall* script to remove it before setting up the server for automated installation.

SET UP REPOSITORIES FOR UBUNTU

When installing on an Ubuntu server, a few of the packages that Airtime relies on are in the Ubuntu *universe* or *multiverse* repositories. If either of these repositories is disabled, you can enable them in the */etc/apt/sources.list* file, by opening the *nano* editor in your server's console. The *nano* editor should be installed by default, but if not, you can install it with the command:

```
sudo apt-get install nano
```

Then open the *sources.list* file with the command:

```
sudo nano /etc/apt/sources.list
```

For an Ubuntu 10.04 Lucid server, find the lines which begin *deb* and end with *lucid universe* or *lucid-updates universe*, adding *multiverse* to the end of these lines, if it is not there already. Other versions of Ubuntu will have similar lines. The *multiverse* repository is required for the *libmp3lame0* library, which is an MP3 encoder. Later versions of Ubuntu, such as 12.04 Precise, have *libmp3lame0* in the *universe* repository.

```
 GNU nano 2.2.2        File: /etc/apt/sources.list

# deb cdrom:[Ubuntu 10.04 LTS _Lucid Lynx_ - Release amd64 (20100429)]/ l$
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.

deb http://gb.archive.ubuntu.com/ubuntu/ lucid main

## Major bug fix updates produced after the final release of the
## distribution.
deb http://gb.archive.ubuntu.com/ubuntu/ lucid-updates main

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubun$
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://gb.archive.ubuntu.com/ubuntu/ lucid universe multiverse
deb http://gb.archive.ubuntu.com/ubuntu/ lucid-updates universe multiverse



^G Get Help  ^O WriteOut  ^R Read File^Y Prev Page^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page^U UnCut Tex^T To Spell
```

The exact repository mirror URLs in your *sources.list* file will differ from the above screenshot, depending on your location.

The Sourcefabric repository contains packages for Airtime, and any other packages which Airtime requires. To add the Sourcefabric repository to an Ubuntu Lucid server, scroll to the end of the *sources.list* file and add the following line:

```
deb http://apt.sourcefabric.org/ lucid main
```

For Ubuntu 12.04 Precise, 12.10 Quantal, 13.04 Raring or 13.10 Saucy substitute *precise*, *quantal*, *raring* or *saucy* in place of *lucid* in the line above.

```
  GNU nano 2.2.2         File: /etc/apt/sources.list

## Also, please note that software in backports WILL NOT receive any revi$
## or updates from the Ubuntu security team.
# deb http://gb.archive.ubuntu.com/ubuntu/ lucid-backports main restricte$
# deb-src http://gb.archive.ubuntu.com/ubuntu/ lucid-backports main restr$

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and t$
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu lucid partner
# deb-src http://archive.canonical.com/ubuntu lucid partner

deb http://security.ubuntu.com/ubuntu lucid-security main
deb http://security.ubuntu.com/ubuntu lucid-security universe

deb http://apt.sourcefabric.org/ lucid main█



^G Get Help  ^O WriteOut  ^R Read File^Y Prev Page^K Cut Text ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page^U UnCut Tex^T To Spell
```

Press Ctrl+O (the Ctrl key and the letter O together) to save the file, then Ctrl+X to exit the nano editor.

SET UP REPOSITORIES FOR DEBIAN

On a Debian squeeze or wheezy server, you can edit the */etc/apt/sources.list* file as root with the command:

`nano /etc/apt/sources.list`

Packages for MP3 encoding are not included in the Debian squeeze repositories. If your server runs squeeze, you can obtain the necessary *libmp3lame0* package by adding the following repository to the end of the file:

`deb http://backports.debian.org/debian-backports squeeze-backports main`

To add the Sourcefabric repository to a Debian squeeze server, add the following line to the end of the file:

`deb http://apt.sourcefabric.org/ squeeze main`

or for Debian wheezy:

`deb http://apt.sourcefabric.org/ wheezy main`

Press Ctrl+O (the Ctrl key and the letter O together) to save the file, then Ctrl+X to exit the *nano* editor.

INSTALL THE SOURCEFABRIC SIGNING KEY

Reload the system's package list with:

`sudo apt-get update`

You will see an error message about a missing public key.

```
Hit http://gb.archive.ubuntu.com lucid Release
Hit http://gb.archive.ubuntu.com lucid-updates Release
Hit http://gb.archive.ubuntu.com lucid/main Packages
Hit http://gb.archive.ubuntu.com lucid/universe Packages
Hit http://gb.archive.ubuntu.com lucid/multiverse Packages
Hit http://gb.archive.ubuntu.com lucid-updates/main Packages
Hit http://gb.archive.ubuntu.com lucid-updates/universe Packages
Hit http://gb.archive.ubuntu.com lucid-updates/multiverse Packages
Fetched 491B in 2s (218B/s)
Reading package lists... Done
W: GPG error: http://apt.sourcefabric.org lucid Release: The following si
gnatures couldn't be verified because the public key is not available: NO
_PUBKEY 0888FE5B174C1854
daniel@office:~$ █
```

To fix this system error, you need to install the *sourcefabric-keyring* package, which contains the package signing key. This encryption key is a security measure which helps ensure that the Airtime packages you will be downloading in future have not been tampered with by a third party. You can install the key with the command:

`sudo apt-get install sourcefabric-keyring`

When prompted, press the *y* key on your keyboard to install the *sourcefabric-keyring* package without verification. If you wish to verify the authenticity of the package signing key, please contact Sourcefabric for assistance.

```
daniel@office:~$ sudo apt-get install sourcefabric-keyring
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed
  sourcefabric-keyring
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.
Need to get 3,610B of archives.
After this operation, 49.2kB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
  sourcefabric-keyring
Install these packages without verification [y/N]? y
Get: 1 http://apt.sourcefabric.org/ lucid/main sourcefabric-keyring 0.1 [3
,610B]
Fetched 3,610B in 0s (12.3kB/s)
Selecting previously deselected package sourcefabric-keyring.
(Reading database ... 348822 files and directories currently installed.)
Unpacking sourcefabric-keyring (from .../sourcefabric-keyring_0.1_all.deb)
 ...
Setting up sourcefabric-keyring (0.1) ...

daniel@office:~$ █
```

Update your computer's software sources again, to confirm that you are now using a trusted software repository:

```
sudo apt-get update
```

You should no longer see the error message about the missing public key.

INSTALL THE DATABASE MANAGEMENT SYSTEM

Airtime uses a PostgreSQL database to keep track of media assets and associated metadata in its storage server. Depending on the scale of your Airtime installation and the hardware available, you may wish to install PostgreSQL on a separate server. If you only have one server, you can install the *postgresql* package on the same machine as Airtime with the command:

```
sudo apt-get install postgresql
```

INSTALL A STREAMING SERVER

Optionally, you may wish to stream directly from Airtime to an **Icecast** media distribution server, without requiring a soundcard or mixer in the broadcast chain. This option is particularly suitable for fully automated stations, in which all shows are played out using Airtime. You can install the **icecast2** package on your server with the command:

```
sudo apt-get install icecast2
```

In some scenarios, you may wish to stream from Airtime to a remote Icecast server, for example when Icecast is installed on a server in a data center with greater bandwidth available than an Airtime server located at your broadcast studio has. This separation may become necessary if your stream becomes popular, because each additional listener which connects to the Icecast stream uses additional bandwidth. In this case, you do not need to install the **icecast2** package on the same server as Airtime.

Before running Icecast on a remote server for the first time, you should change the default *<source-password>*, *<relay-password>* and *<admin-password>* values from *hackme* to something more secure. On more recent GNU/Linux distributions, such as Ubuntu 12.04 'Precise Pangolin' or later, you may be prompted for these settings during the installation of Icecast. If not, you can edit the Icecast configuration manually, with the command:

```
sudo nano /etc/icecast2/icecast.xml
```

```
GNU nano 2.2.2        File: /etc/icecast2/icecast.xml

        change from the default 64k. Applies to all mountpoints  -->
     <burst-size>65535</burst-size>
</limits>

<authentication>
     <!-- Sources log in with username 'source' -->
     <source-password>hackme</source-password>
     <!-- Relays log in username 'relay' -->
     <relay-password>hackme</relay-password>

     <!-- Admin logs in with the username given below -->
     <admin-user>admin</admin-user>
     <admin-password>hackme</admin-password>
</authentication>

<!-- set the mountpoint for a shoutcast source to use, the default if$
     specified is /stream but you can change it here if an alternativ$
     wanted or an extension is required

^G Get Help  ^O WriteOut  ^R Read File^Y Prev Page^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page^U UnCut Tex^T To Spell
```

You should also set the value of *<hostname>* to the domain name of the Icecast server, for example:

<hostname>icecast.example.com</hostname>

This step ensures that stream URLs are generated correctly by the Icecast server.

Save and close the *icecast.xml* file with Ctrl+O and Ctrl+X. Then check the Icecast server is set to start automatically when the server boots in the */etc/default/icecast2* file:

sudo nano /etc/default/icecast2

The value of *ENABLE* should be set to *true* on the last line of that file:

ENABLE=true

Save and close this file with Ctrl+O and Ctrl+X, then start Icecast:

sudo service icecast2 start

The server should respond:

```
Starting icecast2: Starting icecast2
Detaching from the console
icecast2.
```

INSTALL AIRTIME

You can now install the Airtime package with:

sudo apt-get install airtime

This command will install all of the Airtime components, plus any other packages that Airtime requires in order to run.

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  airtime-audio-samples
The following NEW packages will be installed
  airtime
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 8,446kB of archives.
After this operation, 33.3MB of additional disk space will be used.
Get: 1 http://apt.sourcefabric.org/ precise/main airtime 2.5.0-1 [8,446kB]
Fetched 8,446kB in 8s (1,015kB/s)
Preconfiguring packages ...
Selecting previously deselected package airtime.
(Reading database ... 657254 files and directories currently installed.)
Unpacking airtime (from .../airtime 2.5.0-1 all.deb) ...
```

Once all the packages have been downloaded and installed, you will be asked a few questions about the configuration of the Airtime system. You can accept the default settings by pressing the **Enter** key, or use the **Tab** key to change your selection.

Firstly, you will asked if you wish to create an **Apache** virtual host web server configuration automatically.

```
┤ Configuring airtime ├
This setup script can perform apache web-server configuration so
that you can connect to airtime directly after this installation.

Production systems should choose "dedicated v-host". This option
will ask for a Server host-name (FQDN) and will create a minimal
apache vhost configuration that you can adapt.

"no, thanks": no problem. You're welcome to set it up however you
like. Note that the files in /etc/airtime/ may come in handy doing
so.

Create apache2 config:

                    dedicated v-host
                    no thanks


                         <Ok>
```

If so, you will need to enter the domain name that your station staff will use to access Airtime. This would normally be a fully qualified domain name (FQDN) which is already set up in DNS, or dynamic DNS, to point to the IP address of the Airtime server. For a test server that will only be accessed directly from the same machine, you can use the default setting of *localhost* here.

```
┤ Configuring airtime ├
Enter the main hostname of the web server. The DNS of this name
must resolve to the apache server running on this machine.

e.g. "example.com" or "www.example.com" (without the quotes)

You can customize /etc/apache2/sites-enabled/airtime.vhost
afterward and add ServerAliases and further custom configuration.

more info soon..

FQDN - apache-vhost ServerName:

localhost


                         <Ok>
```

Next, set the contact email address of the server administrator, which is good practice in case of any server problems. For a test server, using an address at the *localhost* is acceptable.

```
┤ Configuring airtime ├
required for dedicated vhost-setup.

more info soon..

Email of the vhost-admin - ServerAdmin:

root@localhost


          <Ok>
```

Debian and Ubuntu servers are set up with a default Apache configuration, which might block station staff from accessing Airtime. If you wish, this default configuration can be removed automatically for you.

```
                    ┤ Configuring airtime ├
  By default the apache webserver comes with configuration to send
  all v-hosts to /var/www.

  This option will invoke `sudo a2dissite default`

  and is recommended when using airtime as vhost.

  remove OOO-default apache config

                        remove default
                        no change


                            <Ok>
```
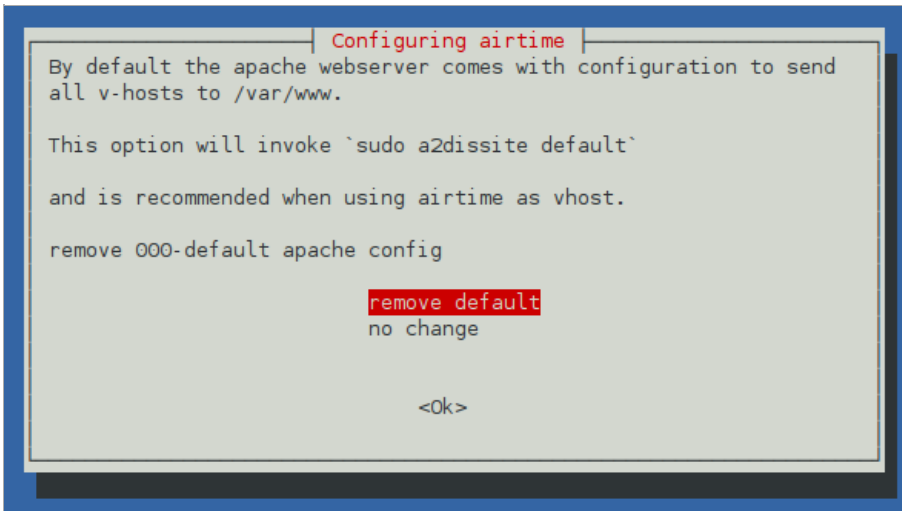
If you are installing Airtime and Icecast on the same machine, you can set the hostname and passwords for the Icecast server during the Airtime installation.

```
Package configuration

                        ┤ Configuring airtime ├
  This option enables icecast2 on this server, and configures
  passwords for both the icecast server and airtime.

  Note: these settings are here for convenience only. Strictly
  speaking they should be done during icecast installation - not
  airtime installation.

  PS. If you have already manually tweaked icecast configuration it
  may not work as expected.

  Set up Icecast2 for live HTTP streaming?

            <Yes>                        <No>
```

First, set the hostname of the Icecast server. For test installs in which you have Icecast installed on the same machine as Airtime, you can use *localhost* here.

```
Package configuration




                        ┤ Configuring airtime ├
   Specify the hostname of the Icecast server. Depending on your
   setup, this might be the same as the Airtime ServerName. For
   testing purposes, you can use the default of 'localhost'.

   Icecast2 hostname:

   icecast.example.com

                        <Ok>
```

The security of your streaming server depends on the strength of the passwords that you choose. You should set strong passwords for *source*, *relay* and *admin* access.
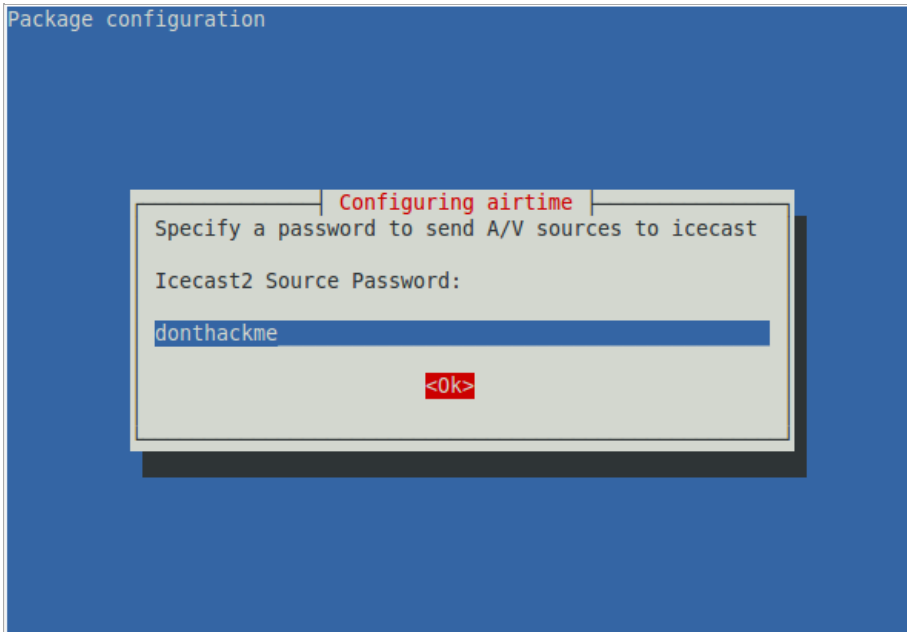
By default, Airtime stores media files under the path */srv/airtime/stor/* but you can specify an alternative path if you wish. If you change this setting, the new path must end with a slash. This main storage path is separate from the *watched folders* which are configured after installation, in the Airtime administration interface. See the chapter *Media Folders* for details of how to add watched folders.
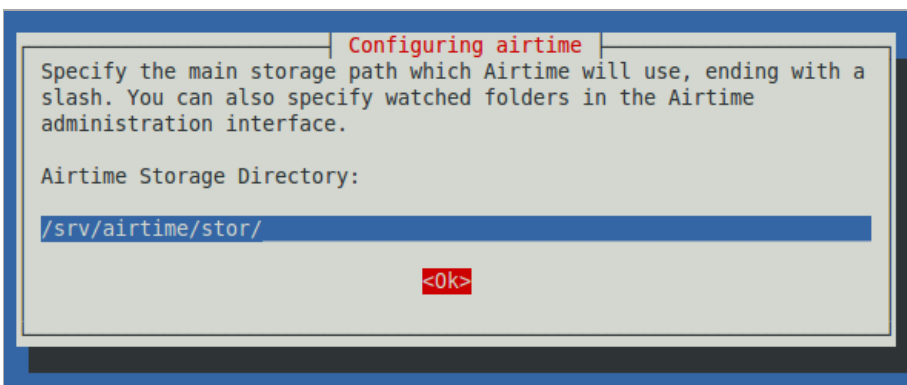
Finally, you have the option to set a more secure password for Airtime's *admin* user than the default of *admin*. You could change the password for the *admin* user later, in the Airtime administration interface, but this would mean that your installation would be insecure until you carried out that task.
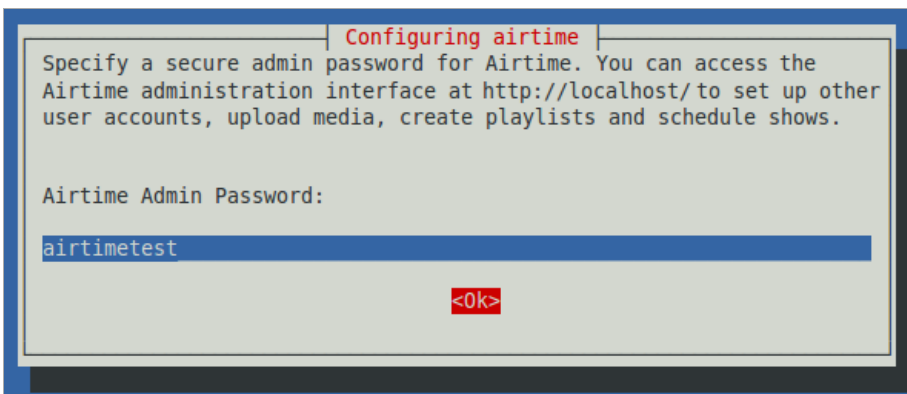
On some GNU/Linux distributions, you may be warned about upgrading the **rabbitmq-server** package, even if you have never installed this package before. If RabbitMQ is only being used by Airtime on your server, it is safe to press the **Enter** key to continue.

The Airtime installer will then run a script to check that your server environment is set up correctly.

```
*** Verifying your system environment, running airtime-check-system ***
AIRTIME_STATUS_URL          = http://airtime.example.com:80/api/status/format/json/api_key/%%api_key%%
AIRTIME_SERVER_RESPONDING   = OK
KERNEL_VERSION              = 3.2.0-4-amd64
MACHINE_ARCHITECTURE        = x86_64
TOTAL_MEMORY_MBYTES         = 2963688
TOTAL_SWAP_MBYTES           = 7812092
AIRTIME_VERSION             = 2.5.1
OS                          = Debian GNU/Linux 7.1 (wheezy) x86_64
CPU                         = AMD Turion(tm) II Neo N40L Dual-Core Processor
WEB_SERVER                  = Apache/2.2.22 (Debian)
PLAYOUT_ENGINE_PROCESS_ID   = 4446
PLAYOUT_ENGINE_RUNNING_SECONDS = 55
PLAYOUT_ENGINE_MEM_PERC     = 0.5%
PLAYOUT_ENGINE_CPU_PERC     = 0.4%
LIQUIDSOAP_PROCESS_ID       = 4685
LIQUIDSOAP_RUNNING_SECONDS  = 49
LIQUIDSOAP_MEM_PERC         = 0.7%
LIQUIDSOAP_CPU_PERC         = 7.4%
MEDIA_MONITOR_PROCESS_ID    = 4410
MEDIA_MONITOR_RUNNING_SECONDS = 55
MEDIA_MONITOR_MEM_PERC      = 0.5%
MEDIA_MONITOR_CPU_PERC      = 0.0%
-- Your installation of Airtime looks OK!

************ Install Complete ************
```

You should now be able to log in to the Airtime administration interface, as shown in the *Getting started* chapter. See the *Host configuration* chapter for advanced settings.

26. MANUAL INSTALLATION

You do not normally need to install Airtime manually, unless you are testing a development version of the software. Versions of Airtime recommended for production use are available for download and upgrade via *secure apt*, as shown in the *Automated installation* chapter.

DEPENDENCIES PROVIDED BY SOURCEFABRIC

The http://apt.sourcefabric.org/ repository contains up-to-date Debian and Ubuntu packages of Airtime dependencies such as **liquidsoap** and **silan** which you may find convenient to install, even if you are installing Airtime manually. Refer to the chapter *Automated installation* for repository setup details.

Airtime requires a version of **python-virtualenv** later than 1.4.8, but Ubuntu Lucid includes the older version 1.4.5 of this package. Before performing a manual installation on Lucid, you should update **python-virtualenv** using the backported package available from the http://apt.sourcefabric.org/ repository. This step is not necessary when performing an automated installation, in which dependencies are resolved automatically.

FULL INSTALL

The **airtime-full-install** script is designed to configure your server for you, using typical default settings.

1. In the server terminal or console, download Airtime from https://github.com/sourcefabric/Airtime/releases with **wget**. For example, to download version 2.5.1, you could use the command:

```
wget https://github.com/sourcefabric/Airtime/archive/airtime-2.5.1-ga.tar.gz
```

Then unzip the downloaded file in your home directory. This action will create a subdirectory called *airtime-2.5.1*:

```
tar -xvzf airtime-2.5.1-ga.tar.gz -C ~/
```

Alternatively, clone the git repository if you wish to work on the latest Airtime source code:

```
git clone https://github.com/sourcefabric/Airtime.git
```

In order to return your code improvements to Airtime, you will need to create your own fork of the Airtime git repository and send pull requests. See the GitHub help page https://help.github.com/articles/fork-a-repo for details.

2. Run the **airtime-full-install** script, for example on Ubuntu:

```
sudo ~/airtime-2.5.1/install full/ubuntu/airtime-full-install
```

The installation script will indicate which files are being installed on your system, and the directories they are being unpacked into. Finally, it will run the **airtime-check-system** script to confirm that your server environment is set up correctly.

```
*** Verifying your system environment, running airtime-check-system ***
AIRTIME_STATUS_URL            = http://airtime.example.com:80/api/status/format/json/api_key/%%api_key%%
AIRTIME_SERVER_RESPONDING     = OK
KERNEL_VERSION                = 3.2.0-4-amd64
MACHINE_ARCHITECTURE          = x86_64
TOTAL_MEMORY_MBYTES           = 2963688
TOTAL_SWAP_MBYTES             = 7812092
AIRTIME_VERSION               = 2.5.1
OS                            = Debian GNU/Linux 7.1 (wheezy) x86_64
CPU                           = AMD Turion(tm) II Neo N40L Dual-Core Processor
WEB_SERVER                    = Apache/2.2.22 (Debian)
PLAYOUT_ENGINE_PROCESS_ID     = 4446
PLAYOUT_ENGINE_RUNNING_SECONDS = 55
PLAYOUT_ENGINE_MEM_PERC       = 0.5%
PLAYOUT_ENGINE_CPU_PERC       = 0.4%
LIQUIDSOAP_PROCESS_ID         = 4685
LIQUIDSOAP_RUNNING_SECONDS    = 49
LIQUIDSOAP_MEM_PERC           = 0.7%
LIQUIDSOAP_CPU_PERC           = 7.4%
MEDIA_MONITOR_PROCESS_ID      = 4410
MEDIA_MONITOR_RUNNING_SECONDS = 55
MEDIA_MONITOR_MEM_PERC        = 0.5%
MEDIA_MONITOR_CPU_PERC        = 0.0%
-- Your installation of Airtime looks OK!

************ Install Complete ************
```

You are now ready to proceed to the *Configuration* chapter.

1. In the server terminal or console, install the list of dependencies. For example, on Ubuntu you could enter the command:

```
sudo apt-get install postgresql python-virtualenv apache2 coreutils \
curl ecasound flac gzip libapache2-mod-php5 libcamomile-ocaml-data \
liquidsoap locales lsof monit mp3gain multitail patch php5-cli \
php5-curl php5-gd php5-json php5-pgsql php-apc php-pear pwgen \
python rabbitmq-server silan sudo sysv-rc tar unzip \
vorbisgain vorbis-tools libzend-framework-php
```

On Debian, install *zendframework* instead of the *libzend-framework-php* package.

2. Check that the Apache web server modules that Airtime requires are enabled:

```
sudo a2enmod php5 rewrite
```

The server should respond:

```
Module php5 already enabled
Module rewrite already enabled
```

3. Create a directory to contain the Airtime web interface:

```
sudo mkdir -p /usr/share/airtime/public
```

4. Next, create the Airtime virtual host configuration file for Apache:

```
sudo nano /etc/apache2/sites-available/airtime
```

and enter the information below, substituting your server's hostname for *airtime.example.com* and your system administrator's email address for *admin@example.com*. Make sure you set the *DocumentRoot* and *Directory* paths correctly. This should normally match the *public* directory that the installer will unpack the web interface into, which by default is the */usr/share/airtime/public/* directory. From Airtime 2.3.0 onwards, the web interface can be installed in a subdirectory of the *DocumentRoot* if you require it to be.

```
<VirtualHost *:80>
   ServerName airtime.example.com
   ServerAdmin admin@example.com
   DocumentRoot /usr/share/airtime/public
   php_admin_value upload_tmp_dir /tmp

  <Directory /usr/share/airtime/public>
      DirectoryIndex index.php
      AllowOverride all
      Order allow,deny
      Allow from all
   </Directory>
</VirtualHost>
```

Press Ctrl+O to save the file, then Ctrl+X to exit the **nano** editor.

5. Create the PHP configuration file */etc/airtime/airtime.ini* in nano:

```
sudo nano /etc/airtime/airtime.ini
```

with the following contents:

```
[PHP]
memory_limit = 512M
magic_quotes_gpc = Off
file_uploads = On
upload_tmp_dir = /tmp
apc.write_lock = 1
apc.slam_defense = 0
```

Save and exit nano, then link this file to the system's PHP configuration with the command:

```
sudo ln -s /etc/airtime/airtime.ini /etc/php5/conf.d/airtime.ini
```

6. Enable the new configuration by entering the command:

```
sudo a2ensite airtime
```

The server should respond:

```
Enabling site airtime.
Run '/etc/init.d/apache2 reload' to activate new configuration!
```

You may also need to disable the default site configuration, which may otherwise interfere with your Airtime installation:

```
sudo a2dissite default
```

As suggested by the output of the command above, reload the web server configuration.

```
sudo /etc/init.d/apache2 reload
```

The server should respond:

```
 * Reloading web server config apache2
```

7. Download Airtime from https://github.com/sourcefabric/Airtime/releases with **wget**, and unzip the downloaded file in your home directory. This action will create a subdirectory called *airtime-2.5.1*:

```
wget https://github.com/sourcefabric/Airtime/archive/airtime-2.5.1-ga.tar.gz
tar -xvzf airtime-2.5.1-ga.tar.gz -C ~/
```

Alternatively, clone the Airtime git repository as shown above.

8. Monit is a utility which Airtime uses to make sure that the system runs smoothly. Enable it by opening the*/etc/default/monit* file in the **nano** editor:

```
sudo nano /etc/default/monit
```

Find the line that begins with *START* and change the value to *yes*:

START=yes

Save the file with Ctrl+O and close nano with Ctrl+X. Now copy the Monit configuration from the Airtime install directory to the */etc/monit/conf.d/* directory:

```
sudo cp ~/airtime-2.5.1/python_apps/monit/airtime-monit.cfg /etc/monit/conf.d/
```

Open the */etc/monit/monitrc* file in **nano**:

```
sudo nano /etc/monit/monitrc
```

At the end of the file, add the line:

```
include /etc/monit/conf.d/*
```

Save the file with Ctrl+O and close nano with Ctrl+X. Then start Monit with:

```
sudo invoke-rc.d monit start
```

More information about monit is available in the chapter *Using Monit*.

9. On Debian squeeze, make sure the rabbitmq-server daemon has started:

```
sudo invoke-rc.d rabbitmq-server start
```

10. Finally, run the minimal **airtime-install** script:

```
sudo ~/airtime-2.5.1/install_minimal/airtime-install
```

Once the **airtime-check-system** script confirms that the install has been successful, you should now be able to log in to the Airtime administration interface, as shown in the *Getting started* chapter, with the username *admin* and the password *admin*. See the *Configuration* chapter for advanced settings.

INSTALL SCRIPT OPTIONS

By default, the **airtime-install** script preserves any existing configuration or installation that it finds on the server. However, it is also possible to dictate the behaviour of the script with a command line option, as follows:

```
--help|-h          Displays usage information.
--overwrite|-o     Overwrite any existing config files.
--preserve|-p      Keep any existing config files.
--no-db|-n         Turn off database install.
--reinstall|-r     Force a fresh install of this Airtime version.
```

MANUAL UNINSTALL

To manually uninstall Airtime from the server, run the **airtime-uninstall** script from the minimal installation directory, for example:

```
sudo ~/airtime-2.5.1/install minimal/airtime-uninstall
```

Optionally, you can also delete the Airtime storage and configuration folders, if you have backups and are not going to need the data on this particular server again. The **rm** command should be used with caution, because it has no undo feature.

```
sudo rm -r /srv/airtime
sudo rm -r /etc/airtime
```

27. SETTING THE SERVER TIME

Accurate time keeping on your server is vital for Airtime performance. You can confirm that the date and time of your server are set correctly with the **date** command:

```
date
```

The server should respond with the date, time, time zone and year in a format similar to the following example:

```
Tue Jul  2 15:08:57 BST 2013
```

If the time on your server is wrong, it is recommended that you take Airtime off-air until the problem is fixed.

CONFIGURING NTP

Although it is possible to set the date and time of the server manually, this is not recommended because the server clock can drift over time, compromising the accuracy of your broadcast schedule. If your Airtime server is permanently connected to the Internet, you can synchronize your server to a time server with the **ntp** program. If **ntp** is not yet installed, you can enter the following command on Debian or Ubuntu:

```
sudo apt-get install ntp
```

Optionally, open the **ntp** configuration file in the **nano** editor to add further time server names:

```
sudo nano /etc/ntp.conf
```

On Ubuntu GNU/Linux, the default time server is *ntp.ubuntu.com*, but there are many other time servers available on the public Internet, including the group of servers listed at http://www.pool.ntp.org/ for each country. Using a variety of NTP servers located closely to your Airtime server should produce the most accurate results. For example, for a server in the United Kingdom you could use the following list:

```
# You do need to talk to an NTP server or two (or three).
server ntp.ubuntu.com
server 0.uk.pool.ntp.org
server 1.uk.pool.ntp.org
server 2.uk.pool.ntp.org
server 3.uk.pool.ntp.org
```

Enter the server names you require, press **Ctrl+O** to write out the */etc/ntp.conf* file, then **Ctrl+X** to exit **nano**. Restart the **ntp** service with:

```
sudo invoke-rc.d ntp restart
```

The server should respond:

```
* Stopping NTP server ntpd                          [ OK ]
* Starting NTP server ntpd                          [ OK ]
```

Then use the **ntpq -p** command to confirm that **ntp** is working. This command should produce output similar to the following:

```
ntpq -p
  remote      refid       st t  when poll reach  delay offset jitter
===============================================================
europium. 193.79.237.14  2 u   28   64    3  39.571 12.600  3.590
norb.v4.c 46.227.200.72  3 u   28   64    3  47.856 -6.908 10.028
82.113.15 193.62.22.82   2 u   29   64    3  11.458 -0.513  2.629
ntppub.le 158.43.192.66  2 u   91   64    2 122.781 44.864  0.001
dns0.rmpl 195.66.241.3   2 u   27   64    3  22.171  1.464  4.242
```

ADJUSTING THE SERVER TIME ZONE

The data centre which hosts your Airtime server could be located anywhere in the world. Some servers are set to *Coordinated Universal Time* or UTC (similar to *Greenwich Mean Time* or GMT), regardless of their location. Airtime uses UTC time in its database for scheduling purposes, independent of the server time zone.

If the server time zone is not appropriate for integration with your station's other systems, on a Debian or Ubuntu server you can reconfigure the **tzdata** (time zone data) package with the command:

```
sudo dpkg-reconfigure tzdata
```

This command opens a menu in which you can select the continent that you require, by pressing the Enter key.
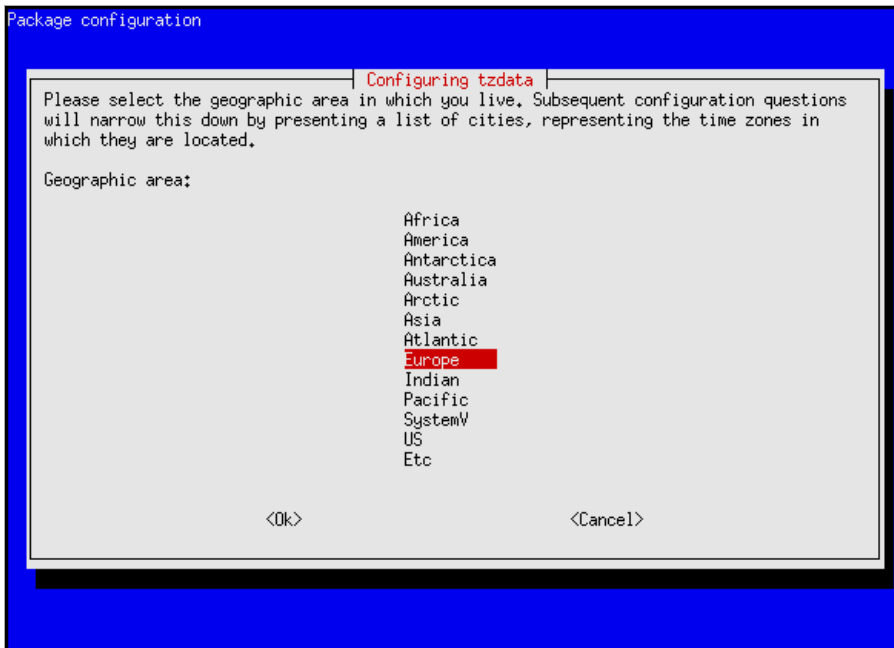
The next step is to select your nearest city, again by pressing the Enter key. The appropriate time zone is selected according to the information that you have entered.
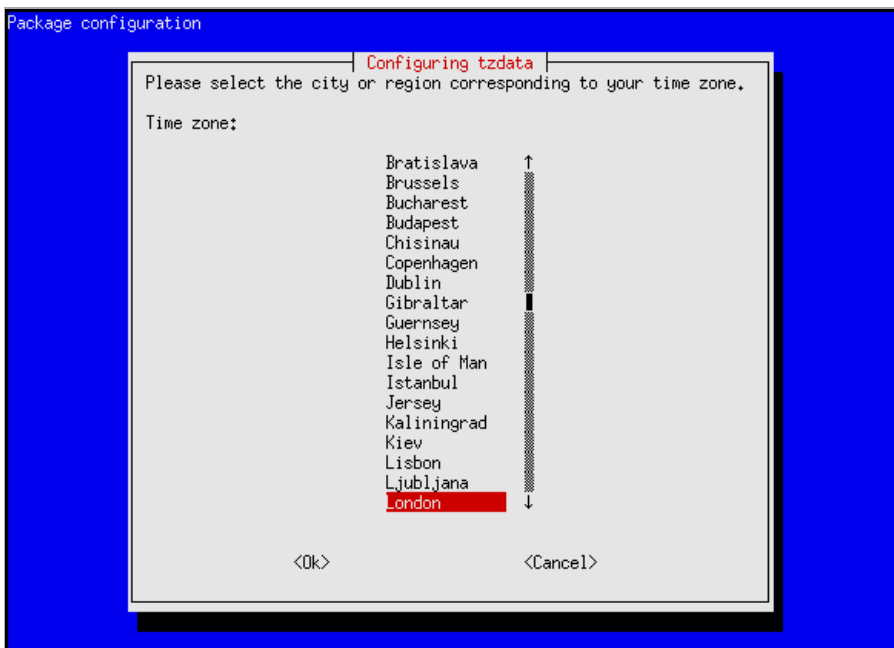
The console output from the **dpkg-reconfigure tzdata** command will confirm the new setting:

```
Current default time zone: 'Europe/London'
Local time is now:      Tue Jul  2 15:18:01 BST 2013.
Universal Time is now:  Tue Jul  2 14:18:01 UTC 2013.
```

ADMINISTRATION

28. PREPARING MEDIA FOR INGEST

Before uploading media to an Airtime server, there are a number of factors which should be considered. Getting your ingest workflow right will save you a lot of time later.

METADATA QUALITY

Airtime automatically imports any metadata that is in the files' ID3 tags. If these tags are incorrect or are missing information, you will have to either edit the metadata manually, or suffer the consequences. For example, if the files have creator or genre metadata missing, it will be impossible to search for, create playlists or generate smart blocks according to these criteria until you add it.

There are a number of programs available which can be used to correct mistakes or incomplete information in ID3 tags. On GNU/Linux, the program **Ex Falso** (http://code.google.com/p/quodlibet/) can be useful for batch setting and editing ID3 tags before importing files into your Airtime server. On a Debian or Ubuntu desktop machine, you can install this program with the command:

```
sudo apt-get install exfalso
```

After installation, you can run this program with the command:

```
exfalso
```

or from the desktop menu. The *Tags From Path* feature of this program is a particularly useful time saver if you have a large archive of untagged files. Sometimes there is useful creator or title information in the file name or directory path structure, which can be converted into an ID3 tag automatically.
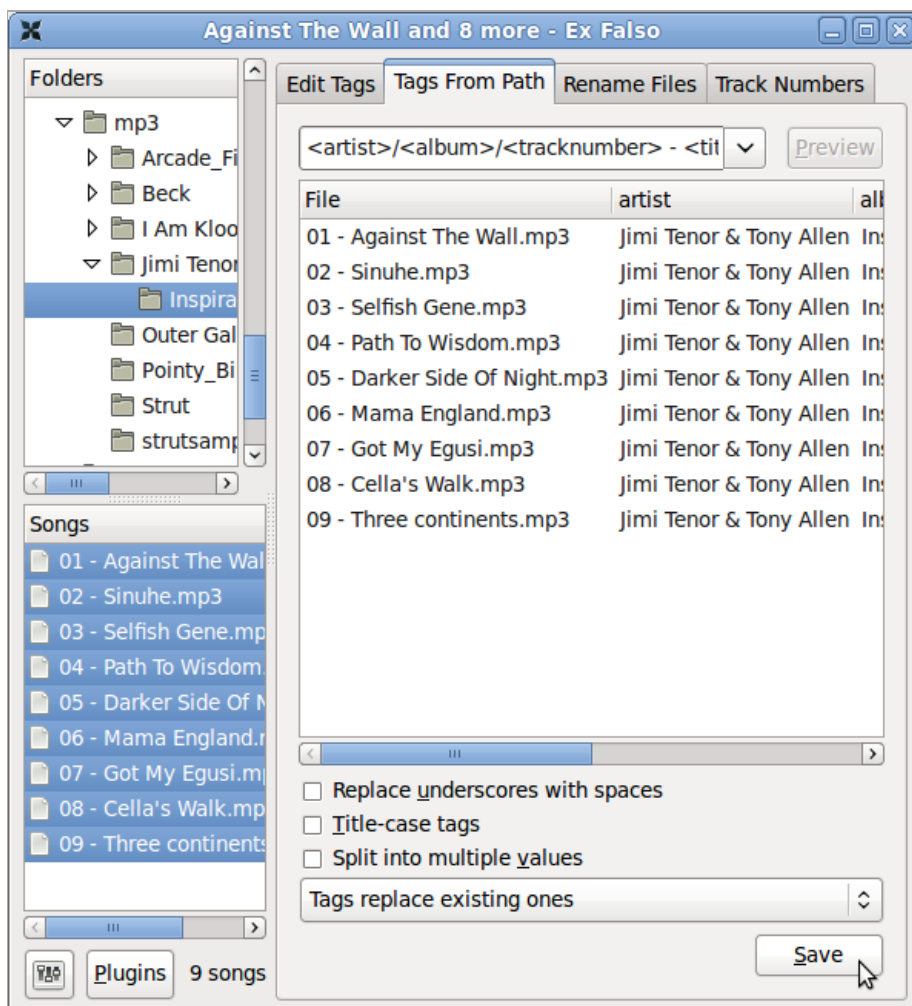
METADATA IN LEGACY CHARACTER SETS

Airtime expects file tag metadata to be stored in the international *UTF-8* character set. Programs such as **Ex Falso** (described above) encode metadata in UTF-8 by default. If you have an archive of files encoded with metadata in a legacy character set, such as the Cyrillic encoding *Windows-1251*, you should convert these files before import.

The program **mid3iconv** (part of the **python-mutagen** package in Debian and Ubuntu) can be used to batch convert the metadata character set of files on the command line. You can install **python-mutagen** with the command:

```
sudo apt-get install python-mutagen
```

For example, to preview the conversion of tags from Windows-1251 (CP1251) character set to UTF-8 for a whole archive of MP3 files, you could use the command:

```
find . -name "*.mp3" -print0 | xargs -0 mid3iconv -e CP1251 -d -p
```

in the base directory of the archive. The **-d** option specifies that the new tag should be printed to the server console (debug mode), and the **-p** option specifies a preview run. This preview will enable you to confirm that the metadata is being read and converted correctly before writing the new tags.

To actually convert all of the tags and strip any legacy ID3v1 tag present from each file at the same time, you could use the command:

```
find . -name "*.mp3" -print0 | xargs -0 mid3iconv -e CP1251 --remove-v1
```

The name of the original character set follows the **-e** option. Other legacy character sets that mid3iconv can convert to UTF-8 include:

KOI8-R: Russian
KOI8-U: Ukrainian

GBK: Traditional Chinese
GB2312: Simplified Chinese

EUC-KR: Korean
EUC-JP: Japanese

CP1253: Greek
CP1254: Turkish
CP1255: Hebrew
CP1256: Arabic

AUDIO LOUDNESS

On file ingest, Airtime analyzes each Ogg Vorbis, MP3, AAC or FLAC file's loudness, and stores a *ReplayGain* value for that file in its database. At playout time, the ReplayGain value is provided to Liquidsoap so that gain can be automatically adjusted to provide an average output of -14 dBFS loudness (14 decibels below full scale). See http://www.replaygain.org/ for more details of ReplayGain.

Because of this automatic gain adjustment, any files with average loudness higher than -14 dBFS will not sound louder than quieter files at playout time, but the lower crest factor in the louder files (their relatively low peak-to-average ratio) may be apparent in the output, making those files sound less dynamic. This may be an issue for contemporary popular music, which can average at -9 dBFS or louder before ReplayGain adjustment. (See http://www.soundonsound.com/sos/sep11/articles/loudness.htm for a detailed analysis of the problem).

Your station's producers should therefore aim for 14dB between peak and average loudness to maintain the crest factor of their prepared material (also known as *DR14* on some dynamic range meters, such as the command-line DR14 T.meter available from http://sourceforge.net/projects/dr14tmeter/). If the producers are working to a different loudness standard, the ReplayGain modifier in Airtime's Stream Settings page can be adjusted to suit their material.

Large transient peaks in otherwise quiet files should be avoided, to guard against the need for peak limiting when ReplayGain is applied to those quieter files.

The **vorbisgain** command-line tool, available in the **vorbisgain** package in Debian/Ubuntu, can be used to indicate the ReplayGain of an individual Ogg Vorbis file before ingest into Airtime. (A similar tool for MP3 files is available in the **mp3gain** package in Debian/Ubuntu).

Here is an example of a very quiet file where the use of ReplayGain would make the output more than 17dB louder:

```
$ vorbisgain -d Peter_Lawson-Three_Gymn.ogg
Analyzing files...

   Gain    | Peak | Scale | New Peak | Track
----------+------+-------+----------+------
+17.39 dB | 4536 | 7.40  |    33585 | Peter_Lawson-Three_Gymn.ogg
```

And here is an example of a very loud file, with lower crest factor, where the output will be more than 7dB quieter with ReplayGain applied:

```
$ vorbisgain -d Snoop_Dogg-Doggfather.ogg
Analyzing files...

   Gain   | Peak  | Scale | New Peak | Track
----------+-------+-------+----------+------
 -7.86 dB | 36592 | 0.40  |    14804 | Snoop Dogg-Doggfather.ogg
```

In the output from vorbisgain, *Peak* is the maximum sample value of the file before any ReplayGain has been applied, where a value of 32,767 represents full scale when decoding to signed 16 bit samples. Note that lossy compressed files can have peaks greater than full scale, due to encoding artifacts. The *New Peak* value for the Snoop Dogg file may be relatively low due to the hard limiting used in the mastering of that piece of music.

SILENCE IN MEDIA FILES

Before importing media, it is good practice to check for any silent sections in the media files. While Airtime compensates for leading and trailing silence with the use of automatic cue-in and cue-out points, it may be preferable to trim these files to the intended length before upload. This is because media in the Airtime library could potentially be re-used in many different systems. **Audacity** is a cross-platform editor suitable for the task of trimming audio files, available from http://audacity.sourceforge.net/

Very quiet introductions or over-long fades can also lead to apparent gaps in your broadcast playout. Some audio CDs feature a 'hidden track' at the end, which in fact uses a long period of silence within the final track, rather than an actual separate track on the disc. This means that CD encoding programs will often encode both the hidden material and the silence in the media file. For example, the track *Debra* from the CD *Midnite Vultures* by *Beck* includes hidden material preceded by seven minutes of silence, as shown in the screen shot from Audacity below.
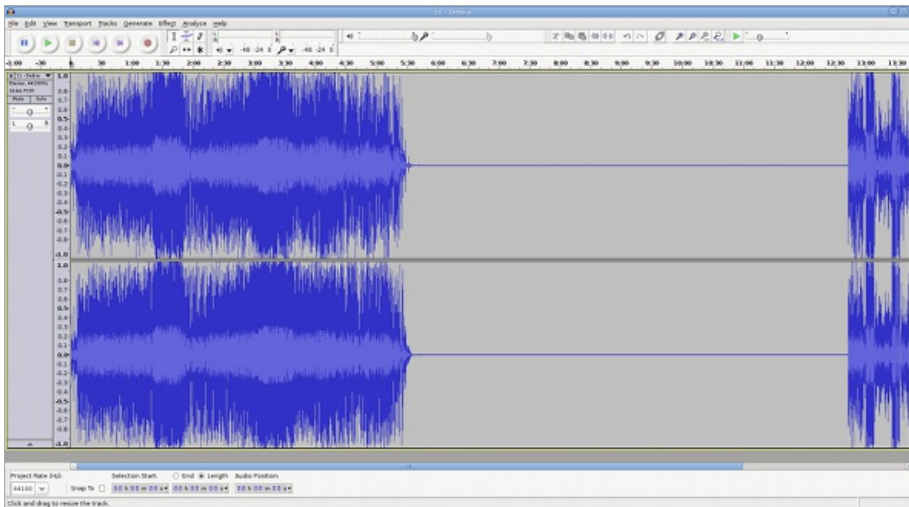
29. USING THE IMPORT SCRIPT

If you have a large number of files in your media library, importing these files one at a time into a broadcast automation system would be time-consuming and tedious. That's why Airtime includes a script that can import an entire directory of files in one go.

COPYING VERSUS WATCHING

There are three main options when using the **airtime-import** script: Either to **copy** or **move** files into Airtime's main storage folder, or to **watch** files elsewhere. Each option has advantages and disadvantages, so you will have to think about how the files will be used in your station in the long term.

- For files that are *copied* or *moved*, you may run into problems with hard disk space if you do not remove files from the storage server periodically.
- For files that are watched, you must make sure that the files are actually going to be there at playout time, otherwise they will be skipped. For this reason, it is not recommended to use the **watch** option for easily-removable storage (like MP3 players or USB memory sticks). There's a strong likelihood that the storage might be unplugged and taken away before playout time. It is also important to make sure that any external hard disk with watched files is powered on, and stays on.

The **airtime-import** script works from the command line, so you must first log into a console on the Airtime server. In recent versions of Airtime, you no longer need to specify the full file system path to the media files that you wish to import. Copying is specified with the **copy** sub-command:

```
airtime-import --copy <directory>
```

where *<directory>* is the base directory of the media files that you wish to import.

Alternatively, you may wish to delete the files from the original location after copying them to the storage archive, which is accomplished with the **move** sub-command:

```
airtime-import --move <directory>
```

If the new files are exact duplicates of files that have already been imported, the **airtime-import** script will detect this.

Any users logged into the administration interface will be warned if a **File import** is **in progress**... Users assigned the *admin* privilege should not attempt to change Airtime's default storage directory while an import is taking place.
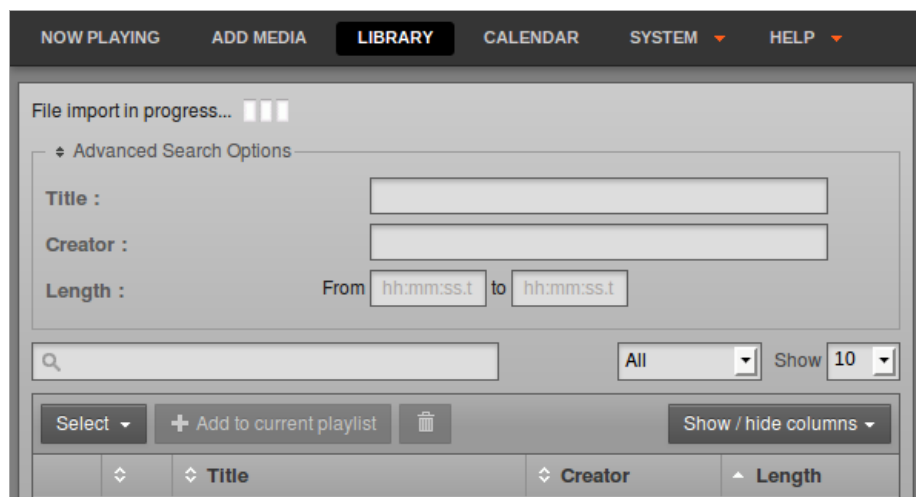
Watching a new folder on the command line is specified by using the **watch add** sub-command:

```
airtime-import --watch-add <directory>
```

The script will report the name of the folder now being watched, for example:

```
airtime-import --watch-add /home/Music/
/home/Music/ added to watched folder list successfully
```

The watched folder will also be listed on the **Media Folders** page on the **System** menu of the Airtime administration interface. (See the chapter *Media Folders* for making changes to watched folders via the administration interface).

To obtain a list of currently watched folders in the server console, you can use the **watch list** sub-command:

```
airtime-import --watch-list
/home/Music/
```

Any new files copied to a watched folder (for example, using your desktop computer's file manager) will automatically appear in the Airtime database, and are then ready for playout. In the same way, if you delete a media file using your file manager, it will be automatically removed from the Airtime database, and will no longer be available for playout.

If you wish to no longer watch a particular folder, you can use the **watch remove** sub-command:

```
airtime-import --watch-remove /home/Music/
/home/Music/ removed from watched folder list successfully
```

GET OR SET THE STORAGE FOLDER PATH

See the chapter *Media Folders* for making changes to the storage folder path via the administration interface. The **airtime-import** script also enables you to read or write Airtime's storage path configuration. You can find out the current setting with the **storage-dir get** sub-command:

```
airtime-import --storage-dir-get
/srv/airtime/stor
```

A new storage folder path can be configured using the **storage-dir set** sub-command:

```
airtime-import --storage-dir-set /home/Music/
Successfully set storage folder to /home/Music/
```

The **storage-dir set** sub-command should be used with caution on a production server, because moving the storage folder during scheduled programming has the potential to disrupt your broadcast playout. The Airtime playout engine caches files up to 24 hours ahead, but any last-minute additions to the schedule during storage migration could cause problems.

30. THE AIRTIME-LOG COMMAND

The airtime-log command provides convenient access to the logging output from the services which make up the Airtime system: *media-monitor*, *recorder*, *playout*, *liquidsoap* and *web*.

Using this command requires root privileges (**sudo** on Ubuntu). Entering the command without any options returns a list of options that you can specify:

```
sudo airtime-log

Usage: airtime-log [options]

--view|-v <string> Display log file
        media-monitor|recorder|playout|liquidsoap|web

--dump|-d <string> Collect all log files and compress into a tarball
        media-monitor|recorder|playout|liquidsoap|web (ALL by default)

--tail|-t <string> View any new entries appended to log files in real-time
        media-monitor|recorder|playout|liquidsoap|web (ALL by default)
```

For example, to view the media-monitor log, you could use the command:

```
sudo airtime-log -v media-monitor
```

Use the **PageUp** and **PageDown** keys on your keyboard to navigate through the log file, or press the **q** key to quit the viewer.

To dump all log files and compress them into a tarball placed in the working directory, you could add the -d switch to the command:

```
sudo airtime-log -d

Creating Airtime logs tgz file at /root/logs/airtime-log-all-2012-11-14-16-22-02.tgz
```

To view just the Liquidsoap log output in real-time, you could enter the command:

```
sudo airtime-log -t liquidsoap

Tail liquidsoap log 2012/11/14 15:47:20 [server:3] New client: localhost.
2012/11/14 15:47:20 [server:3] Client localhost disconnected.
2012/11/14 15:47:20 [server:3] New client: localhost.
2012/11/14 15:47:20 [lang:3] dynamic_source.get_id
2012/11/14 15:47:20 [server:3] Client localhost disconnected.
2012/11/14 16:17:20 [server:3] New client: localhost.
2012/11/14 16:17:20 [server:3] Client localhost disconnected.
2012/11/14 16:17:20 [server:3] New client: localhost.
2012/11/14 16:17:20 [lang:3] dynamic_source.get_id
2012/11/14 16:17:20 [server:3] Client localhost disconnected.
```

Press the **Ctrl+C** keys to interrupt the real-time log output and return to the server console.

31. BACKING UP THE SERVER

The following shell commands can be used for database backup and restore on a running *PostgreSQL* server in an Airtime system.

You can dump the entire database to a zipped file with the combination of the **pg_dumpall** command and **gzip**. The **pg_dumpall** command is executed as the user *postgres*, by using the **sudo** command and the **-u** switch. It is separated from the **gzip** command with the pipe symbol.

```
sudo -u postgres pg_dumpall | gzip -c > airtime-backup.gz
```

This command can be automated to run on a regular basis using the standard **cron** tool on your server.

When restoring a production database on a cleanly installed Airtime system, it may be necessary to drop the empty database that was created during the new installation, by using the **dropdb** command. Again, this command is executed with **sudo** as the user *postgres*:

```
sudo -u postgres dropdb airtime
```

This **dropdb** command above is necessary to avoid 'already exists' errors on table creation when overwriting an empty Airtime database in the next step. These errors might prevent some data from being restored, such as user account data.

To restore, first unzip the backup file with **gunzip**, then use the **psql** command as the *postgres* user:

```
gunzip airtime-backup.gz
sudo -u postgres psql -f airtime-backup
```

You should now be able to log in to the Airtime web interface in the usual way.

For safety reasons, your regular database backups should be kept in a directory which is backed up by your storage backup tool of choice; for example, the */srv/airtime/database_backups* directory. This should ensure that a storage restore can be made along with a matching and complete version of the Airtime database from the day that the storage backup was made.

STORAGE BACKUP

Backing up the Airtime database with **pg_dumpall** will not back up the Airtime media storage server, which is likely to need a great deal more backup space. Creating a compressed file from hundreds of gigabytes of storage server contents is likely to take a very long time, and may have little benefit for the amount of CPU power used, if the media files are already stored in a highly compressed format. It is also impractical to copy very large backup files across the network on a daily basis.

Instead, it is preferable to use an incremental backup technique to synchronize the production Airtime server storage with a backup server each day or night. If the backup server also contains an Airtime installation, it should be possible to switch playout to this second machine relatively quickly, in case of a hardware failure or other emergency on the production server.

A standard incremental backup tool on GNU/Linux servers is *rsync* (http://rsync.samba.org/) which can be installed using the package manager of your GNU/Linux distribution. However, incremental backup alone cannot help in the scenario where a file which later proves to be important has been deleted by an administrator. For backups that can be rolled back to restore from an earlier date than the current backup, the tool *rdiff-backup* (http://www.nongnu.org/rdiff-backup/) can be deployed.

32. UPGRADING

Airtime 2.5.x versions support upgrading from version 2.3.0 and above. If you are running a production server with a version of Airtime prior to 2.3.0, you should upgrade it to version 2.3.0 before continuing.

Before upgrading a production Airtime server, you should back up both the PostgreSQL database and the storage server used by Airtime. This is especially important if you have not already set up a regular back up routine. This extra back up is a safety measure in case of accidental data loss during the upgrade, for example due to the wrong command being entered when moving files. See the chapter *Backing up the server* in this book for details of how to perform these back ups.

If you have deployed Airtime using the method shown in the *Automated installation* chapter, you can upgrade in the same way. A new Airtime package available in the Sourcefabric repository can be installed with:

```
sudo apt-get update
sudo apt-get upgrade
```

If you have used the method shown in the *Manual installation* chapter, you should repeat the installation steps of downloading and unpacking the tarball to an installation directory, or pulling from the git repository, and running the **airtime-install** or **airtime-full-install** script. The installation script will detect an existing Airtime deployment and back up any configuration files that it finds.

After the upgrade has completed, you may need to clear your web browser's cache before logging into the new version of the Airtime administration interface. If the playout engine starts up and detects that a show should be playing at the current time, it will skip to the correct point in the current item and start playing.

In Airtime 1.9.0 onwards, the concept of *linked files* was replaced with the concept of *watched folders*. If you are upgrading from a version of Airtime earlier than 1.9.0 and you have previously linked files, the folders they are in will not be watched until you add them to your watched folder list. See the chapter *Media Folders* for more details.

UPGRADING THE SERVER DISTRIBUTION

After your Airtime server has been deployed for a few years, you may need to upgrade the GNU/Linux distribution that it runs in order to maintain security update support. If the upgrade does not go smoothly, it may cause significant downtime, so you should always have a fallback system available during the upgrade to ensure broadcast continuity.

After upgrading a server from Ubuntu Lucid 10.04 LTS to Ubuntu Precise 12.04 LTS, you may experience problems with Monit failing to start, due to a change in the format of its configuration files. If so, a double equals sign can be changed to single equals in line 39 of /etc/init.d/monit so that it reads:

```
  if [ "$1" = "start" ]
```

Also, the file /etc/default/monit should contain the line:

```
START=yes
```

instead of how it was configured in Ubuntu Lucid:

```
startup=1
```

This modification is sometimes necessary because during distribution upgrade it is normal to keep any locally modified configuration files.

33. TROUBLESHOOTING

If your Airtime server is not working as expected, individual components of the system can be started, stopped, restarted or checked in the server console using the invok**e-rc.d** command:

```
sudo invoke-rc.d airtime-liquidsoap      start|stop|restart|status
sudo invoke-rc.d airtime-playout         start|stop|restart|status
sudo invoke-rc.d airtime-media-monitor   start|stop|restart|status
sudo invoke-rc.d apache2                 start|stop|restart|status
sudo invoke-rc.d rabbitmq-server         start|stop|restart|status
```

For example, to restart the Airtime playout engine, you could enter the command:

```
sudo invoke-rc.d airtime-playout restart
```

The server should respond:

```
Restarting Airtime Playout: Done.
```

The **status** option for **airtime-playout** and **airtime-media-monitor** runs the **airtime-check-system** script to confirm that all of Airtime's dependencies are installed and running correctly.

LOG FILES

Airtime stores log files under the directory path */var/log/airtime/* which can be useful for diagnosing the cause of any problems. Copies of these log files may be requested by Sourcefabric engineers while they are providing technical support for your Airtime deployment. See the chapter *The airtime-log command* for more details.

TEST TONES

Liquidsoap output can be tested using two commands provided by Airtime. The **airtime-test-soundcard** command enables you to send a test tone to the default sound card on the system, so you can check that your audio equipment is working. Press **Ctrl+C** on your keyboard to stop the tone.

```
airtime-test-soundcard [-v]
                [-o alsa | ao | oss | portaudio | pulseaudio ]
                [-h]
Where:
    -v verbose mode
    -o Linux Sound API (default: alsa)
    -h show help menu
```

The **airtime-test-stream** command enables you to send a test tone to a local or remote streaming media server. Press **Ctrl+C** on your keyboard to stop the tone being streamed.

```
airtime-test-stream [-v]
                [-o icecast | shoutcast ] [-H hostname] [-P port]
                [-u username] [-p password] [-m mount]
                [-h]
Where:
    -v verbose mode
    -o stream server type (default: icecast)
    -H hostname (default: localhost)
    -P port (default: 8000)
    -u user (default: source)
    -p password (default: hackme)
    -m mount (default: test)
    -h show help menu
```

34. USING MONIT

Monit is a utility which can be used to manage and monitor processes, files, directories and filesystems on your Airtime server. It is installed by default when you install Airtime.

To view the Monit web interface, open port 2812 of your server in a web browser. For example, on the localhost you can use:

`http://localhost:2812/`

You will have to log in to see the Monit web interface. The username to use is *guest* and the default password is *airtime*.
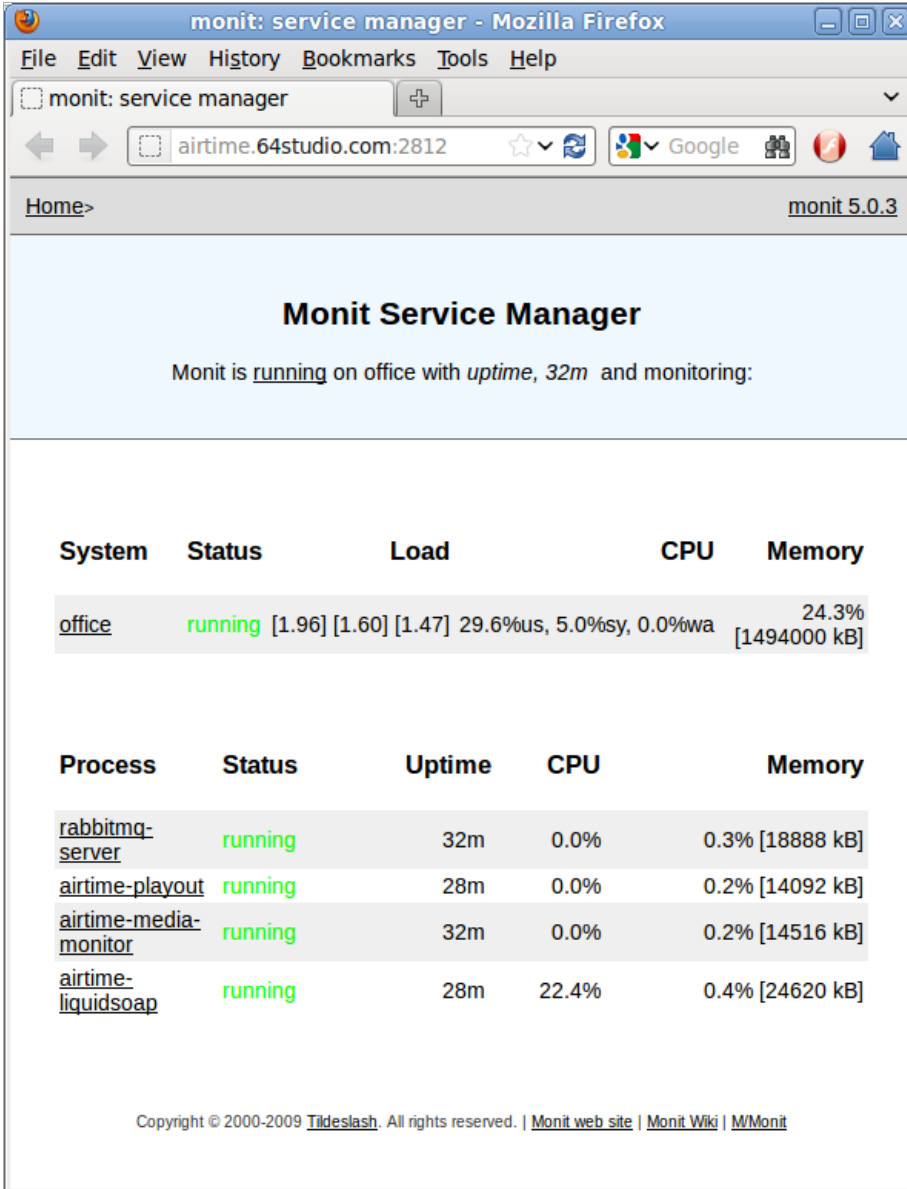
Clicking on the name of a service in the Process column, such as *airtime-playout*, opens another page with details of that service.

By default, the Monit *guest* login is configured for read-only access, which means you cannot restart services through its web interface. To log in as the *admin* user, you will need the randomly generated password set by Airtime in the */etc/monit/conf.d/monit-airtime-generic.cfg* file. You can see this with the command:

```
sudo less /etc/monit/conf.d/monit-airtime-generic.cfg
```

The random password for the admin user should follow on the line *allow admin:* such as the this example of *LHR32YP7H1*:

```
set daemon 10 # Poll at 10 second intervals
set logfile /var/log/monit.log

set httpd port 2812
    allow admin:LHR32YP7H1
    allow guest:airtime read-only
```

Logging in as the *admin* user, you will now see buttons for controlling a service at the end of each service page:

Pid	If changed 1 times within 1 cycle(s) then alert
Ppid	If changed 1 times within 1 cycle(s) then alert

Start service   Stop service   Restart service   Disable monitoring

Monit does not have a logout button, so if you have already logged in as *guest*, you may have to clear the active login from your web browser before logging in as *admin*. In *Firefox* on Ubuntu, this is accomplished using *Clear Recent History* on the *History* menu.

LOCKING DOWN REMOTE ACCESS

To make remote access more secure, you can edit the file */etc/monit/conf.d/monit-airtime-generic.cfg* with **nano**:

```
sudo nano /etc/monit/conf.d/monit-airtime-generic.cfg
```

If you wish to enable access from the localhost only, append an *and use address* statement to the end of the line which sets the server port:

```
 set httpd port 2812 and use address localhost
```

To enable access from other servers, comment out the part of the line *use address localhost* and then append an '*allow*' line with the IP address and netmask of the machine to connect to Monit from, such as a local network address of *10.0.1.2* and netmask of *255.255.255.0*:

```
    allow 10.0.1.2/255.255.255.0
```

If you enable remote *admin* access, be sure to change the randomly generated password in the line beginning *allow admin:* on a regular basis, like you would for any important password. The password for the read-only *guest* access should remain as *airtime*, because Airtime uses this password to collect information for the **Status** page of the administration interface.

```
 set httpd port 2812 and # use address localhost
    allow localhost
    allow 10.0.1.2/255.255.255.0
    allow admin:LGW12WB9J7
    allow guest:airtime read-only
```

Press Ctrl+O to save the file, then Ctrl+X to exit **nano**. Then restart Monit with:

```
sudo invoke-rc.d monit restart
```

SENDING EMAIL ALERTS

To configure Monit to send email alerts, edit the file */etc/monit/monitrc* to uncomment the *set mailserver* parameter. Change this line to show the name of the SMTP server on your Airtime server's network, as provided by your system administrator.

```
sudo nano /etc/monit/monitrc
```

```
  GNU nano 2.2.2           File: /etc/monit/monitrc

#
# set statefile /var/.monit.state
#
## Set the list of mail servers for alert delivery. Multiple servers may b
## specified using comma separator. By default monit uses port 25 - this
## is possible to override with the PORT option.
#
# set mailserver mail.bar.baz,                # primary mailserver
#                backup.bar.baz port 10025,   # backup mailserver on port 1
#                localhost                     # fallback relay
#
#
## By default monit will drop alert events if no mail servers are availabl
## If you want to keep the alerts for a later delivery retry, you can use
## EVENTQUEUE statement. The base directory where undelivered alerts will
## stored is specified by the BASEDIR option. You can limit the maximal qu
## size using the SLOTS option (if omitted, the queue is limited by space
## available in the back end filesystem).
#

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Po
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spe
```

Further down the configuration file, you can set the *From:* and *To:* addresses for the alert emails using the *set mail-format* and *set alert* parameters. Then uncomment these lines for the parameters to be read by Monit.

```
  GNU nano 2.2.2           File: /etc/monit/monitrc

## are expanded at runtime. For example, to override the sender:
#
# set mail-format { from: monit@foo.bar }
#
#
## You can set alert recipients here whom will receive alerts if/when a
## service defined in this file has errors. Alerts may be restricted on
## events by using a filter as in the second example below.
#
# set alert sysadm@foo.bar                          # receive all alerts
# set alert manager@foo.bar only on { timeout }  # receive just service-
#                                                  # timeout alert
#
#
## Monit has an embedded web server which can be used to view status of
## services monitored, the current configuration, actual services paramete
## and manage services from a web interface.
#
# set httpd port 2812 and

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Po
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spe
```

Press Ctrl+O to save the file, then Ctrl+X to exit **nano**. Then restart Monit with:

```
sudo invoke-rc.d monit restart
```

More information about using Monit is available at http://mmonit.com/monit/documentation/

ADVANCED CONFIGURATION

35. AUTOMATED FILE IMPORT

The *airtime-import* script can be combined with the standard *SFTP* (secure FTP) program and *cron* daemon on a GNU/Linux server to enable automated file import from multiple remote computers. This could save time for your station staff when using distributed production methods, or content syndication.

Traditional FTP servers use plain text passwords (without encryption) and are therefore not recommended for upload accounts on Airtime servers accessible from the public Internet. SFTP is a cross-platform protocol which works with many desktop programs including **gFTP** for GNU/Linux (http://www.gftp.org/). This program can be installed on Debian or Ubuntu desktop computers with the command:

```
 sudo apt-get install gftp
```

Other popular SFTP clients include **FileZilla** for Windows (http://filezilla-project.org/) and **Cyberduck** for Mac and Windows (http://cyberduck.ch/).

To enable SFTP uploads, first invoke the **adduser** command to create the *uploads* account on the server. For security reasons this user account is restricted to using SFTP only; it cannot be used for executing other commands in a login shell.

```
sudo adduser --home /srv/airtime/uploads --shell /usr/lib/sftp-server uploads
```

The server will then invite you to type in the password for the new *uploads* user, and once again for confirmation. The security of your Airtime server depends on the strength of the password that you set, so be sure to use a long and complex password with upper case, lower case and numerical characters. It is not necessary to set a full name or other details for this account.

```
Adding user `uploads' ...
Adding new group `uploads' (1003) ...
Adding new user `uploads' (1002) with group `uploads' ...
Creating home directory `/srv/airtime/uploads' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for uploads
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
```

 Next, create a folder to contain the incoming files:

```
 sudo mkdir /srv/airtime/uploads/incoming/
```

Then create a script to run once per hour:

```
 sudo nano /etc/cron.hourly/airtime-upload
```

The script should import the newly uploaded files from the incoming folder specified, using the *copy* option, and then remove the original uploaded files. This step, rather than simply using the *watch* option on the */srv/airtime/uploads/incoming/* folder, ensures that the *uploads* SFTP account does not have direct write access to the Airtime storage archive. That could be a security risk if the password was compromised.

```
#!/bin/sh

# Run the import script on fresh uploads

airtime-import copy /srv/airtime/uploads/incoming/

# Clean the incoming directory to save disk space

rm -r /srv/airtime/uploads/incoming/*
```

Finally, the script should be made executable so that the cron daemon can run it.

```
sudo chmod +x /etc/cron.hourly/airtime-upload
```

By default, Debian and Ubuntu GNU/Linux run *cron.hourly* tasks at 17 minutes past each hour. This value can be adjusted in the file */etc/crontab* on the server, if required.

Remote users should connect to the Airtime server using their client software of choice, making sure that they specify an SFTP rather than FTP connection. The remote directory for the clients to use would be */srv/airtime/uploads/incoming/* as configured above.
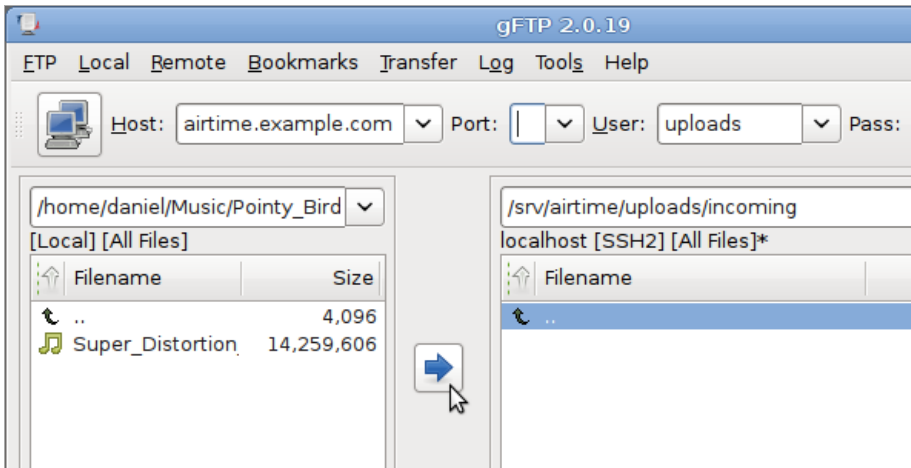
For additional security, you could configure your Airtime server to use an encryption key pair for the *uploads* account, instead of a password. See https://help.ubuntu.com/community/SSH/OpenSSH/Keys for details of how to do this on an Ubuntu server.

36. ICECAST HANDOVER

In a typical radio station configuration, the live output from the broadcast studio and the scheduled output from Airtime are mixed together before being sent further along the broadcast chain, to a transmitter or streaming media server on the Internet. (This may not be the case if your Airtime server is remote from the studio, and you are using the **Show Source Mount Point** or **Master Source Mount Point** to mix live and scheduled content. See the *Stream Settings* chapter for details).

If your **Icecast** server is hosted in a remote data centre, you may not have the option to handover the streaming media source manually, because you have no physical access to connect a broadcast mixer to the server. Disconnecting the stream and beginning another is less than ideal, because the audience's media players will also be disconnected when that happens.

The **Icecast** server has a *fallback-mount* feature which can be used to move clients (media players used by listeners or viewers) from one source to another, as new sources become available. This makes it possible to handover from Airtime output to a show from another source, and handover to Airtime again once the other show has ended.

To enable fallback mounts, edit the main Icecast configuration file to define the mount points you will use, and the relationship between them.

```
sudo nano /etc/icecast2/icecast.xml
```

The example *<mount>* section provided in the *icecast.xml* file is commented out by default. Before or after the commented section, add three mount point definitions. The default mount point used by Airtime is */airtime_128* which is shown in the */etc/airtime/liquidsoap.cfg* file. You must also define a mount point for the live source (called */live.ogg* in this example) and a mount point for the public to connect to (called */stream.ogg* in this example).

```
<mount>
      <mount-name>/airtime_128</mount-name>
      <hidden>0</hidden>
</mount>

<mount>
      <mount-name>/live.ogg</mount-name>
      <fallback-mount>/airtime_128</fallback-mount>
      <fallback-override>1</fallback-override>
      <hidden>0</hidden>
</mount>

<mount>
      <mount-name>/stream.ogg</mount-name>
      <fallback-mount>/live.ogg</fallback-mount>
      <fallback-override>1</fallback-override>
      <hidden>0</hidden>
</mount>
```

These mount point definitions mean that a client connecting to a URL such as *http://icecast.example.com:8000/stream.ogg* will first fall back to the */live.ogg* mount point if it is available. If not, the client will fall back in turn to the */airtime_128* mount point for Airtime playout.

Setting the value of *<fallback-override>* to 1 (enabled) means that when the */live.ogg* mount point becomes available again, the client will be re-connected to it. If you wish to hide the */airtime_128* and */live.ogg* mount points from the public Icecast web interface, set the value of *<hidden>* in each of these definitions to 1.

SOURCE CONFIGURATION

Connect the other source to the Icecast server with the same parameters defined in the */etc/airtime/liquidsoap.cfg* file, except for the mount point. This should one of the mount points you have defined in the */etc/icecast2/icecast.xml* file, such as */live.ogg* in the example above.

To configure **Mixxx** for streaming to Icecast, click *Options*, *Preferences*, then *Live Broadcasting*. For server *Type*, select the default of *Icecast 2* when streaming to Debian or Ubuntu servers, as this is the current version of Icecast supplied with those GNU/Linux distributions.

By default, Icecast streams are buffered to guard against network problems, which causes latency for remote listeners. When monitoring the stream from a remote location, you may have to begin the live stream a few seconds before the previous stream ends to enable a smooth transition.

37. PROMOTING YOUR STATION

If you have an Icecast server, you can put a link to the Icecast status page (by default at port 8000) on your station's homepage, to provide an overview of available streams. See the chapter *Interface customization* for tips on theming the Icecast status page. You can also use Now Playing widgets (see the chapter *Exporting the schedule*) or HTML5 stream players (see the chapter *Stream player for your website*) to help grow your audience.

On an Icecast server, you can uncomment the *<directory>* section in the */etc/icecast2/icecast.xml* file to have your station automatically listed on the Icecast directory website http://dir.xiph.org which could help you pick up more listeners.

```
<!-- Uncomment this if you want directory listings -->

<directory>
    <yp-url-timeout>15</yp-url-timeout>
    <yp-url>http://dir.xiph.org/cgi-bin/yp-cgi</yp-url>
</directory>
```

The Indymedia stream directory at http://radio.indymedia.org/en/yp links to grassroots independent radio projects around the world. You can add your station to their list with an additional *<directory>* section, as follows:

```
<directory>
     <yp-url-timeout>15</yp-url-timeout>
     <yp-url>http://radio.indymedia.org/cgi-bin/yp-cgi</yp-url>
</directory>
```

Another stream directory service is provided by the Liquidsoap Flows! sitehttp://flows.liquidsoap.fm/. The following section can be added to the file */usr/lib/airtime/pypo/bin/liquidsoap_scripts/ls_script.liq* after *add_skip_command(s)* on line 174, for a stream named '*ourstation*':

```
ourstation = register_flow(
  radio="Rock 'n Roll Radio",
  website="http://radio.example.com/",
  description="Canada's most rockin' radio!",
  genre="Rock",
  user="",
  password="",
  streams=[("ogg/128k","http://streaming.example.com/airtime_128")],
  ourstation)
```

For the time being, a stream can be registered on the Liquidsoap Flows! site with any username and password. Authenticated services may be offered in future.

38. STREAM PLAYER FOR YOUR WEBSITE

If you are using Airtime with Icecast or SHOUTcast for web streaming, you can embed a player applet into your website. **jPlayer** is a player applet (available under the GNU GPL from http://jplayer.org/) which uses the *<audio>* tag feature of HTML5 to play your streams. If the listener's browser does not support HTML5, the applet falls back to using Adobe Flash for MP3 format streams.

Proprietary browsers, such as Internet Explorer on Windows or Safari on Apple OS X, may not support Ogg Vorbis or Opus streams. Listeners who do not wish to install a browser with Ogg support (such as **Mozilla Firefox**, **Google Chrome** or **Opera**) can listen to these streams using a separate media player such as **VideoLAN Client**, also known as VLC (http://www.videolan.org/vlc/).

EXAMPLE CODE

You can download the example code for this chapter from:

http://sourcefabric.booktype.pro/airtime-25-for-broadcasters/help/static/jPlayer_demo4.zip

Unzip this file on your computer, and then open the file *jplayer-demo.html* in your editor:

```
nano jplayer-demo.html
```

We'll focus on the two areas that you need to concern yourself with. Firstly, in the *<head>* tag of the document, you'll see some code like this:

```
$("#jquery_jplayer_1").jPlayer({
    ready: function (event) {
        $(this).jPlayer("setMedia", {
            oga:"http://sourcefabric.out.airtime.pro:8000/sourcefabric_a"
        });
    },
    swfPath: "js",
    supplied: "oga",
    wmode: "window",
    smoothPlayBar: true,
    keyEnabled: true
});
```

This code loads jPlayer, and specifies the source of an Ogg Vorbis stream. When testing on your own server, you should change this setting to the IP address or domain name, port number and mount point of the Icecast server you are using, like so:

```
oga: "http://streaming.example.com:8000/airtime_128"
```

If your stream is in MP3 format, you will need to prefix the stream setting with mp3, like this:

```
mp3: "http://streaming.example.com:8000/airtime 128"
```

and change the *supplied* line from:

```
    supplied: "oga"
```

to:

```
    supplied: "mp3"
```

As soon as jPlayer has finished loading, it will automatically begin to play the stream. The parameters *ready*, *ended*, *swfPath* and *supplied* are arguments passed to jPlayer. A full list of constructor arguments is available in the *jPlayer Developer Guide* at http://www.jplayer.org/latest/developer-guide/

JPLAYER CONTROLS

Secondly, the *<body>* tag of the file *jplayer-demo.html* defines the controls displayed by jPlayer. These controls can be as simple as just one Play/Pause button, or a fully-fledged playback interface with a playlist, progress bar and volume control. This example code uses one of the simpler skins available for jPlayer, *Blue Monday*.

A stop button or progress bar is not useful when we are streaming continuously from Icecast or SHOUTcast, so we can remove these tags from the demo code, along with the tags specifying the playback time and file duration. The simplified HTML creates a play/pause button and a mute button, and looks as follows:

```
<ul class="jp-controls">
  <li><a href="javascript:;" class="jp-play" tabindex="1">play</a></li>
  <li><a href="javascript:;" class="jp-pause" tabindex="1">pause</a></li>
  <li><a href="javascript:;" class="jp-mute" tabindex="1" title="mute">mute</a></li>
  <li><a href="javascript:;" class="jp-unmute" tabindex="1" title="unmute">unmute</a></li>
  <li><a href="javascript:;" class="jp-volume-max" tabindex="1" title="max volume">max volume</a></li>
</ul>
```

When you open the HTML file *jplayer-demo.html* in a web browser, your player should appear as follows:
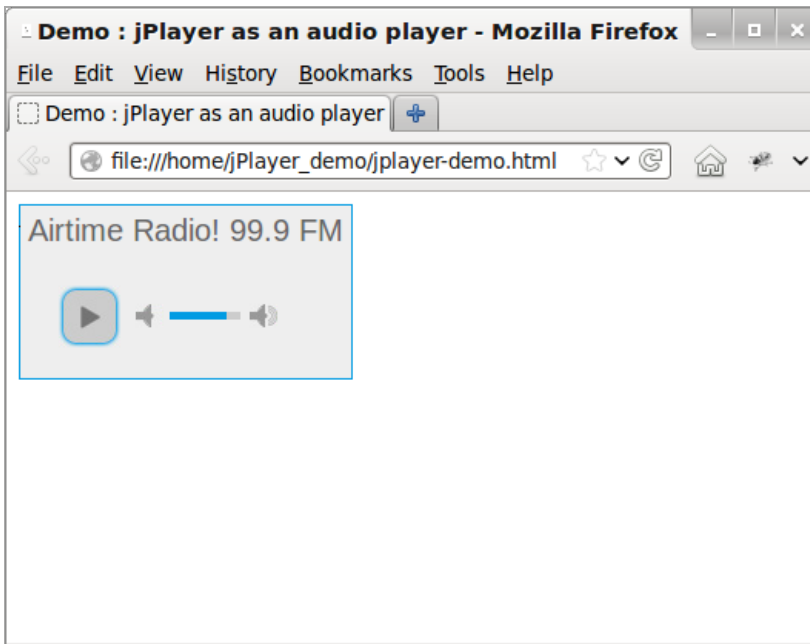
The original version of this demo code and skin are available from http://jplayer.org/download/ for further experimentation.

39. EXPORTING THE SCHEDULE

Airtime has a feature which enables your station's show and schedule information to be displayed on remote websites. This feature is included in Airtime because you would not usually invite the general public to access your Airtime server directly. If you had very large numbers of people requesting data from the Airtime server at once, the burst of network traffic might overload the server, potentially disrupting your broadcasts. If carried out maliciously, this network overload is known as a *denial of service attack*.

Instead, your public-facing web server can retrieve the schedule information from the Airtime API. This information can then be displayed on your broadcast station or affiliate websites by a content management system, such as Sourcefabric's **Newscoop** (http://newscoop.sourcefabric.org/). It can be presented using Javascript widgets and styled with CSS, in any format that you require. The **Broadcaster** theme for Newscoop (https://github.com/newscoop/theme-Broadcaster) integrates these widgets with ready-to-use styles.

There are two kinds of information that can be retrieved remotely from the Airtime API without authentication; the metadata for the current show plus the following show (live-info), or the schedule for the current week and the week ahead (week-info). The week-info metadata includes show names, times, and individual show URLs on your public website. That way, the audience for your station can click through from the schedule information to find out more about a particular show, or download a previous show recording that you might have made available.

If your Airtime server was accessible at https://airtime.example.com the live show information could be retrieved by your web server using this URL:

```
https://airtime.example.com/api/live-info/?callback
```

The comma-separated text metadata returned to your web server might be something like this:

```
({"env":"production",
"schedulerTime":"2013-07-03 14:07:31",

"previous":{"name":"Earth, Wind &amp; Fire - Boogie Wonderland",
"starts":"2013-07-01 22:53:00",
"ends":"2013-07-01 22:55:00",
"type":"track"},
"current":{"name":"Deodato - Also Sprach Zarathustra",
"starts":"2013-07-03 13:07:06",
"ends":"2013-07-03 13:16:05",
"media_item_played":true,
"record":0,
"type":"track"},

"next":{"id":8,
"instance_id":10,
"name":"Rock Show",
"url":"https:\/\/rock.example.com\/",
"start_timestamp":"2013-07-03 14:00:00",
"end_timestamp":"2013-07-03 15:00:00",
"starts":"2013-07-03 14:00:00",
"ends":"2013-07-03 15:00:00",
"record":0,
"type":"show"},

"currentShow":[{"start_timestamp":"2013-07-03 14:07:00",
"0":"2013-07-03 13:07:00",
"end_timestamp":"2013-07-03 15:00:00",
"1":"2013-07-03 14:00:00",
"name":"Funk Show",
"2":"Funk Show",
"id":7,
"3":7,
"instance_id":9,"4":9,
"record":0,
"5":0,
"url":"https:\/\/funk.example.com\/",
"6":"",
"starts":"2013-07-03 14:07:00",
"7":"2013-07-03 13:07:00",
"ends":"2013-07-03 15:00:00",
"8":"2013-07-03 14:00:00"}],

"nextShow":[{"id":8,
"instance_id":10,
"name":"Rock Show",
"url":"https:\/\/rock.example.com\/",
"start_timestamp":"2013-07-03 15:00:00",
"end_timestamp":"2013-07-03 16:00:00",
"starts":"2013-07-03 15:00:00",
"ends":"2013-07-03 16:00:00",
"record":0,
"type":"show"}],

"timezone":"BST",
"timezoneOffset":"3600",
"AIRTIME_API_VERSION":"1.1"})
```

The information for the current week's schedule could be retrieved using the URL:

```
https://airtime.example.com/api/week-info/?callback
```

In this case, the metadata returned would be in a different format from the above example, something like the following. To keep the example short, this particular schedule export only contains four shows on a Monday. A full weekly schedule export would contain a great deal more text.

```
({"monday":[

{"start_timestamp":"2013-07-01 12:05:00",
"end_timestamp":"2013-07-01 13:00:00",
"name":"Elvis Show",
"id":2,
"instance_id":2,
"record":0,
"url":"https:\/\/elvis.example.com\/",
"starts":"2013-07-01 12:05:00",
"ends":"2013-07-01 13:00:00"},

{"start_timestamp":"2013-07-01 13:00:00",
"end_timestamp":"2013-07-01 14:00:00",
"name":"News",
"id":3,
"instance_id":4,
"record":0,
"url":"https:\/\/news.example.com\/",
"starts":"2013-07-01 13:00:00",
"ends":"2013-07-01 14:00:00"},

{"start_timestamp":"2013-07-01 14:00:00",
"end_timestamp":"2013-07-01 15:00:00",
"name":"Funk Show",
"id":4,
"instance_id":6,
"record":0,
"url":"https:\/\/funk.example.com\/",
"starts":"2013-07-01 14:00:00",
"ends":"2013-07-01 15:00:00"},

{"start_timestamp":"2013-07-01 15:00:00",
"end_timestamp":"2013-07-01 17:30:00",
"name":"Rock Show",
"id":5,
"instance_id":7,
"record":0,
"url":"https:\/\/rock.example.com\/",
"starts":"2013-07-01 15:00:00",
"ends":"2013-07-01 17:30:00"},

],

"tuesday":[],
"wednesday":[],
"thursday":[],
"friday":[],
"saturday":[],
"sunday":[],
"AIRTIME_API_VERSION":"1.1"})
```

If you see the message *You are not allowed to access this resource* when attempting to display schedule information in your web browser, log in to the Airtime administration interface, click *System* in the main menu, then *Preferences*. Set **Allow Remote Websites To Access "Schedule" Info?** to **Enabled**, click the **Save** button, then refresh the browser window opened on the schedule export URL. If you do not wish to make schedule information available to the public, set this option to **Disabled** instead.
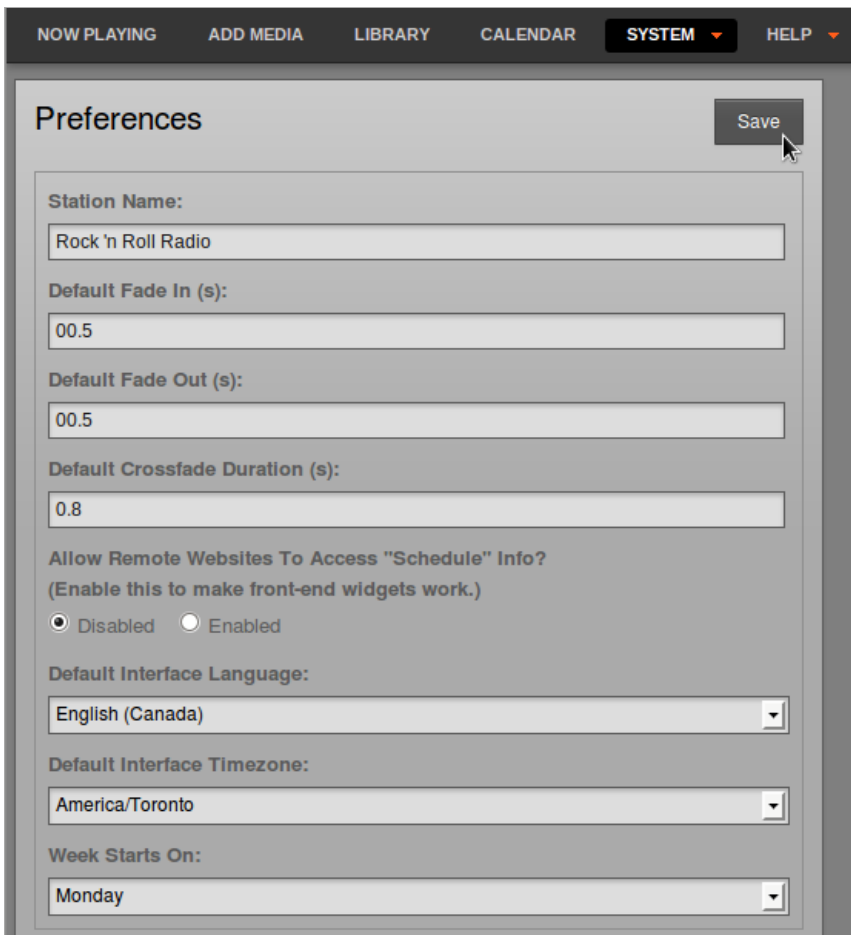
CACHING SCHEDULE INFORMATION

If the Airtime server is behind a firewall, or you want to protect the Airtime server from large numbers of schedule requests, you may wish to cache the schedule information on a public-facing or intermediate server. You can then create a firewall rule that only allows the schedule server to connect to the Airtime server, in addition to any remote users of the Airtime web interface.

Your system administrator can set up schedule caching on a standard Apache and PHP enabled web server with the *curl* program installed, using the following steps:

1. Create a shell script on the schedule server (schedule.example.com) that polls the remote Airtime server (airtime.example.com), and writes the metadata returned into a pair of local temporary files:

```
sudo nano /usr/local/bin/airtime-schedule.sh
```

The content of this file should be like the following script, replacing airtime.example.com with the name of your Airtime server:

```
#!/bin/sh

curl -s "https://airtime.example.com/api/live-info/?callback=***" > /tmp/live-info

curl -s "https://airtime.example.com/api/week-info/?callback=***" > /tmp/week-info
```

2. Make the script executable:

```
sudo chmod +x /usr/local/bin/airtime-schedule.sh
```

3. Create an Apache VirtualHost configuration for the schedule server:

```
sudo nano /etc/apache2/sites-available/schedule
```

containing a definition like the following, replacing *schedule.example.com* with the name of your schedule server:

```
<VirtualHost *:80>
    ServerName schedule.example.com
    DocumentRoot /var/www/schedule/
</VirtualHost>
```

4. In the schedule server's DocumentRoot folder, create the folders *api/live-info/* and *api/week-info/*

```
sudo mkdir -p /var/www/schedule/api/live-info/
sudo mkdir -p /var/www/schedule/api/week-info/
```

5. Create an index.php file in the *api/live-info/* folder:

```
sudo nano /var/www/schedule/api/live-info/index.php
```

containing the following code:

```php
<?php
$filename = '/tmp/live-info';  // define here the path and name of uploaded live-info file

header('Content-Type: text/javascript');
header("Expires: Thu, 01 Jan 1970 00:00:00 GMT");
header("Cache-Control: no-store, no-cache, must-revalidate");

$callback = empty($_GET['callback']) ? null : $_GET['callback'];
$content = file_get_contents($filename);
$content = str_replace('***', $callback, $content);
echo $content;
?>
```

6. Create an index.php file in the *api/week-info/* folder:

```
sudo nano /var/www/schedule/api/week-info/index.php
```

containing the following code:

```php
<?php
$filename = '/tmp/week-info';  // define here the path and name of uploaded week-info file

header('Content-Type: text/javascript');
header("Expires: Thu, 01 Jan 1970 00:00:00 GMT");
header("Cache-Control: no-store, no-cache, must-revalidate");

$callback = empty($_GET['callback']) ? null : $_GET['callback'];
$content = file_get_contents($filename);
$content = str_replace('***', $callback, $content);
echo $content;
?>
```

7. Enable the new configuration and reload the Apache web server:

```
sudo a2ensite schedule
sudo /etc/init.d/apache2 reload
```

8. Create a cron job to run the shell script each minute:

```
sudo nano /etc/cron.d/airtime-schedule
```

containing the line:

```
* * * * * www-data /usr/local/bin/airtime-schedule.sh
```

The schedule server will now be serving the same show information as the Airtime server, with a cache lifetime of one minute. You can adjust the cache lifetime by altering the frequency of the cron job that polls the Airtime server.

PUSHING SCHEDULE INFORMATION VIA FTP OR SSH

If there is no inbound access to the Airtime server at all, an FTP script can be used to push cached schedule data from Airtime to an external web server. The standard ftp command should be available on the Airtime server and the external web server should have a suitably restricted FTP account set up. After following steps 1 and 2 above to export schedule data to a pair of temporary files on the Airtime server, create a new script on the Airtime server to automatically make the upload:

```
sudo nano /usr/local/bin/upload-schedule-data.sh
```

Replace host, user and password values with appropriate values for your external web server:

```sh
#!/bin/sh
HOST='website.example.com'
USER='ftp_user'
PASSWD='ftp_password'

ftp -n -v $HOST << EOT
user $USER $PASSWD
ascii
prompt
put /tmp/airtime-week-info
put /tmp/airtime-live-info
bye
EOT
```

Then make the new script executable and create a cron job to launch it every minute, as in step 8 above. Steps 3 to 7 above should be carried out on the external web server so that it can convert the two temporary files uploaded via FTP into public schedule data.

If you have secure shell access (SSH) to the remote web server, you could write a script to use the secure copy command (scp) instead of ftp.

WEBSITE WIDGETS

Example HTML, Javascript and CSS code for your public website are provided in the *widgets* folder of the Airtime installation tarball, or on GitHub: https://github.com/sourcefabric/Airtime/tree/master/widgets

If you have performed an automated installation on Debian or Ubuntu, the widgets can be found in the */usr/share/doc/airtime/examples/* directory.

For the widgets to work on a typical web server, links to the Javascript and CSS code have to be included in the HTML page <head> element, like the following example:

```
<head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Airtime widgets</title>
   <script src="js/jquery-1.6.1.min.js" type="text/javascript">
    </script>
    <script src="js/jquery-ui-1.8.10.custom.min.js" type="text/javascript">
    </script>
    <script src="js/jquery.showinfo.js" type="text/javascript">
    </script>
  <link href="css/airtime-widgets.css" rel="stylesheet" type="text/css" />
</head>
```

A full example is shown in the *widgets/sample_page.html* file in the Airtime installation tarball, on GitHub, or in the */usr/share/doc/airtime/examples/* directory if you have installed the Debian/Ubuntu package of Airtime.

The following code is for a small *airtimeLiveInfo* widget that displays information about the current show (show time elapsed, and show time remaining), as well as some information about the next show (start time and end time). In this example, the label text for *onAirNow* is translated into French for local language support:

```
<script>
    $(document).ready(function() {
        $("#headerLiveHolder").airtimeLiveInfo({
            sourceDomain: "https://schedule.example.com/",
            text: {onAirNow:"Sur Les Antennes", offline:"Offline", current:"Current", next:"Next"},
            updatePeriod: 20 //seconds
        });
    });
</script>
```

The <script> element above would typically be added to the page's <head> element. You would then add a <div> element to the <body> element of the page to hold the widget, such as:

```
<div id="headerLiveHolder" style="border: 1px solid #999999; padding: 10px;"></div>
```

On the public website, this widget can be made to look like the following screenshot:

The CSS properties *color:* and *text-transform:uppercase* have been used to style the *onAirNow* label. There is a full example CSS file *widgets/css/airtime-widgets.css* in the Airtime installation tarball, GitHub repository, or */usr/share/doc/airtime/examples/* directory on Debian/Ubuntu.

A variation on this widget displays the item being played out, rather than the show names:

```
<script>
    $(document).ready(function() {
        $("#headerLiveTrackHolder").airtimeLiveTrackInfo({
            sourceDomain: "https://schedule.example.com/",
            text: {onAirNow:"On Air Now", offline:"Offline", current:"Current", next:"Next"},
            updatePeriod: 20 //seconds
        });
    });
</script>
```

The id of the <div> element in the page body would have to be updated to match:

```
<div id="headerLiveTrackHolder" style="border: 1px solid #999999; padding: 10px;"></div>
```

The next widget *airtimeShowSchedule* is medium sized, and displays the upcoming show schedule for that day.

```
<script>
    $(document).ready(function() {
        $("#onAirToday").airtimeShowSchedule({
            sourceDomain: "https://schedule.example.com/",
            text: {onAirToday:"On air today"},
            updatePeriod: 60 //seconds
        });
    });
</script>
```

Again, the id of the <div> element has to match the script for the widget to appear:

```
<div id="onAirToday"></div>
```

The output from the widget script above can be styled to look like this screen shot:

Finally, the following code creates a large widget *airtimeWeekSchedule* that enables site visitors to browse through the show schedule for that week. In this example, all widget labels have been translated into French:

```
<script>
$(document).ready(function() {

 $("#scheduleTabs").airtimeWeekSchedule({
    sourceDomain:"https://schedule.example.com/",
    dowText:{monday:"Lundi", tuesday:"Mardi", wednesday:"Mercredi", thursday:"Jeudi", friday:"Vendredi",
saturday:"Samedi", sunday:"Dimanche"},
    miscText:{time:"Temps", programName:"Nom du Programme", details:"Détails", readMore:"Lire La Suite"},
    updatePeriod: 600 //seconds
 });

    var d = new Date().getDay();
    $('#scheduleTabs').tabs({selected: d === 0 ? 6 : d-1, fx: { opacity: 'toggle' }});

});
</script>
```

The <div> element for this widget would be:

```
<div id="scheduleTabs"></div>
```

Using the code above and the CSS and image files provided with Airtime, the weekly schedule can be styled to look like this:

Optionally, you can display the schedule for the week ahead, as well as the current week. This requires the Javascript file *jquery.showinfo.js* supplied with Airtime 2.5.0 or later, where the value of the variable *dow* (for days of the week) includes fourteen rather than seven days:

```
var dow = ["monday", "tuesday", "wednesday", "thursday", "friday",
          "saturday", "sunday", "nextmonday", "nexttuesday",
          "nextwednesday","nextthursday", "nextfriday",
          "nextsaturday", "nextsunday"];
```

In that case, the labels for the days of the week could be configured as follows:

```
dowText: {monday:"Monday", tuesday:"Tuesday", wednesday:"Wednesday",
          thursday:"Thursday", friday:"Friday", saturday:"Saturday",
          sunday:"Sunday", nextmonday:"Next Monday",
          nexttuesday:"Next Tuesday", nextwednesday:"Next Wednesday",
          nextthursday:"Next Thursday",nextfriday:"Next Friday",
          nextsaturday:"Next Saturday", nextsunday:"Next Sunday"},
```

The value of **sourceDomain** in the code examples above should match the URL that you wish to serve schedule information to the public from. If you have used the *Caching schedule information* method detailed above, this would be the URL of your schedule server, not the Airtime server directly.

40. INTERFACE CUSTOMIZATION

The Airtime administration interface, as a web application, is fully customizable using the same methods that you might use to modify a website. For instance, you may wish to increase certain font sizes or change the colours in the Airtime interface to better suit staff users with impaired vision. To do this, open one of the CSS files in the */public/css/* directory under the Airtime *DocumentRoot* directory in an editor such as **nano**:

```
sudo nano /usr/share/airtime/public/css/styles.css
```

To change the background colour of the administration interface from dark gray to white, the *background:* property of the body tag could be changed to *#ffffff* as follows:

```
body {
    font-size: 62.5%;
    font-family:Arial, Helvetica, sans-serif;
    background: #ffffff;
    margin: 0;
    padding: 0;
}
```

Save the file with **Ctrl+O**, then refresh your browser to see the change to the interface background colour.

Any custom changes that you make to the administration interface should be backed up before upgrading Airtime to a newer version, otherwise they could be overwritten. If you have made improvements that you think might be useful to other Airtime users, please contact Sourcefabric and tell us about them.

MODIFYING THE ICECAST INTERFACE

If you have installed Icecast, in the directory */etc/icecast2/web/* you will find several XSLT and other files which are used to generate the Icecast web interface. If you are familiar with HTML you should be able to modify these pages, as they are well commented. You do have to be careful with syntax, because something as simple as a missing bracket can cause the Icecast web interface to break down.

For example, you could change the *status.xsl* page:

```
sudo nano /etc/icecast2/web/status.xsl
```

Modifying the *status.xsl* page is a good place to start, because this is the default page that site visitors see when they browse port 8000 on your Icecast server. The most obvious change to make in the XSLT pages is the content of the *<title>* and *<h2>* tags, to announce the name of your station. You can also modify the *style.css* file in this directory to change colour and layout options.

After saving the file with Ctrl+O, refresh your web browser, and the new look should now be visible.

41. INTERFACE LOCALIZATION AND GIT

The Airtime administration interface can be localized using the standard GNU **gettext** method. Using GitHub for this task means you don't have to tackle the whole of a localization yourself; just as much as you can manage.
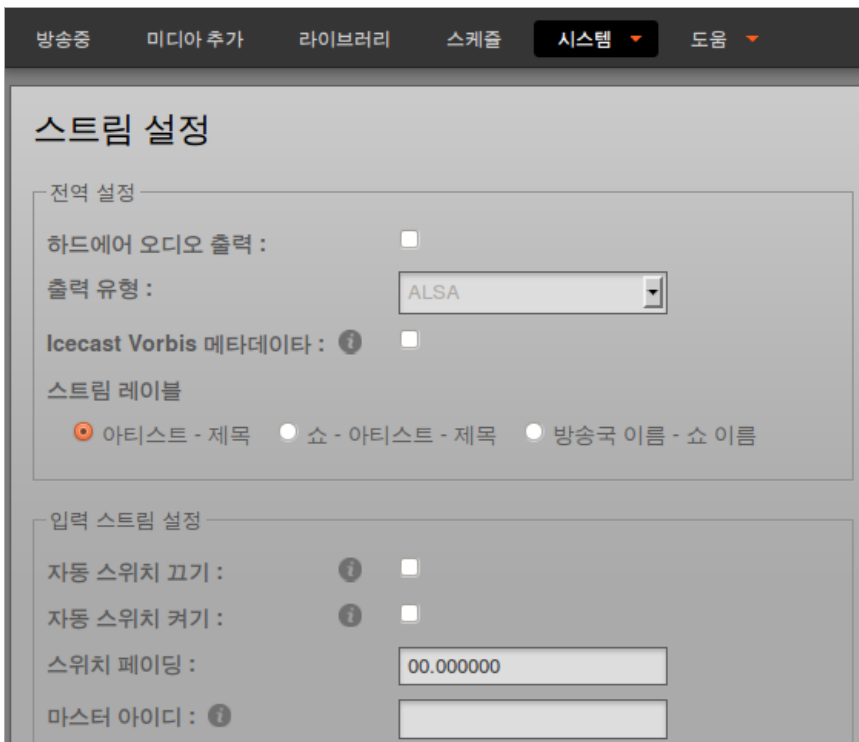
First, you should check if a localization is already under way for your locale of choice. The best way to do this is to take a look at the 'master' branch in the GitHub repository for Airtime at https://github.com/sourcefabric/Airtime/tree/master. You can also ask in the Airtime development forum at http://forum.sourcefabric.org/categories/airtime-development, where you might find community members who can help you with the translation.

GNU **gettext** means using a .po file for each language or dialect, a specially formatted plain text file with groups of three or more lines, like this example from Airtime's Korean localization:

```
#: airtime_mvc/application/configs/navigation.php:57
msgid "Media Folders"
msgstr "때뤫
```

The first of these three lines starts with the hash symbol, and references where this string of text is found in the source code by its file name and line number. If this string is found more than once in the source code, you will see other reference lines here. The second line contains the **msgid**, which is the original version of the string. The third line contains the **msgstr**, which is the translation of that string for the localization that this particular .po file relates to.

If you use the cross-platform program **Poedit** (http://www.poedit.net/) to edit the .po file, this formatting of the text is hidden by an easy-to-use GUI. The *poedit* package can be installed on most GNU/Linux distributions using the standard software installer. Versions of Poedit for Mac and Windows are available for free download from the project's homepage.

Before manually translating strings in Poedit from scratch, you should take a look at the online translation services available, such as Lingohub (https://lingohub.com) or Google's Translator Toolkit (http://translate.google.com/toolkit/), which both support gettext .po files. If using automatic translation, you can then use Poedit to fine-tune the localization and fix any formatting errors.

If you don't already have a GitHub account, you can sign up at https://github.com/signup/free. Once you have a GitHub account, you can fork a copy (https://help.github.com/articles/fork-a-repo) of the Airtime project. Work for the next major version of the software is done in the **master** branch of each project, so that's the branch to **checkout** after you have made the initial **git clone**.

In the locale code *de_CH*, for example, *de* represents the German language and the suffix *_CH* indicates the dialect spoken in Switzerland. Some languages have a wide variety of dialect localizations, which can be differentiated with a suffix in this way. You should update the header information in the .po file, which includes the language code and a country code, using one of the existing .po files as a guide.

After forking the Airtime git repository, make sure you're in the **master** branch:

```
git branch
  devel
* master
```

Create a new locale directory (e.g. *airtime_mvc/locale/de_CH/LC_MESSAGES/* for German as spoken in Switzerland):

```
mkdir -p airtime mvc/locale/de CH/LC MESSAGES/
```

Copy the template *airtime.po* file into the directory you just created:

```
cp airtime mvc locale/template/airtime.po airtime mvc/locale/de CH/LC MESSAGES
```

and update the header information in the new copy of the *airtime.po* file using the **nano** editor:

```
nano airtime_mvc/locale/de_CH/LC_MESSAGES/airtime.po
```

For the example of Swiss German, the header of the file should now look like this:

```
# SWISS GERMAN (de_CH) translation for Airtime.
# Copyright (C) 2013 Sourcefabric
# This file is distributed under the same license as the Airtime package.
# Sourcefabric <contact@sourcefabric.org>, 2013.
#
msgid ""
msgstr ""
"Project-Id-Version: Airtime 2.5\n"
"Report-Msgid-Bugs-To: http://forum.sourcefabric.org/\n"
```

After using an online translation tool to begin a new localization, you can load the exported .po file into Poedit and complete your translation there. Enter the localization team's contact information and language into Poedit's **Edit -> Preferences** and **Catalog -> Settings** dialogs, which will be added to the .po file. When you save a .po file in Poedit, the corresponding binary .mo file will be compiled automatically.

Finally, **git add**, **git commit** and **git push** these new .mo and .po files to your GitHub fork of the project, and send a git pull request (https://help.github.com/articles/using-pull-requests) to the Airtime developers. The localization can then be added to a forthcoming Airtime release.

If you don't want to work with git, that's no problem - download a copy of the .po template file, edit the header, run it through an automatic translator and check it with Poedit. Then email your contribution to the Airtime team as an attachment - it will be very welcome! However, learning to use git is a good idea, because it means you can work directly on the current source code, share the localization work with the Airtime community, and avoid duplicated effort.

42. HOST CONFIGURATION

The streaming host configuration for Airtime is shown in the file */etc/airtime/liquidsoap.cfg* which is automatically generated by the **Streams** page, found on the **System** menu of the Airtime administration interface. For this reason, you would not normally edit the streaming configuration manually, as any changes are likely to be overwritten by the administration interface.

DATABASE AND RABBITMQ HOSTS

Optionally, you may wish to edit the file */etc/airtime/airtime.conf* to set the PostgreSQL database host, and the username and password to connect to the database with:

```
sudo nano /etc/airtime/airtime.conf
```

You can also set options for RabbitMQ messaging, the Airtime server and SoundCloud uploads in this file, although you should not normally need to adjust the defaults unless you are running a large Airtime system distributed across multiple servers. To run the Airtime server in demo mode, which changes the greeting on the login page and prevents user accounts from being created or modified, set the value of *demo* to 1.

```
[database]
host = localhost
dbname = airtime
dbuser = airtime
dbpass = airtime

[rabbitmq]
host = 127.0.0.1
port = 5672
user = airtime
password = XXXXXXXXXXXXXXXXXXX
vhost = /airtime

[general]
api_key = XXXXXXXXXXXXXXXXXXX
web_server_user = www-data
airtime_dir = /usr/share/airtime
base_url = airtime.example.com
base_port = 80
base_dir = /
cache_ahead_hours = 1

[monit]
monit_user = guest
monit_password = airtime

[soundcloud]
connection_retries = 3
time_between_retries = 60

[demo]
demo = 0
```

Save and close the file with **Ctrl+O** and **Ctrl+X**. If you have changed the database settings, you should now run the command:

```
sudo airtime-update-db-settings
```

to make sure all of Airtime's database configuration files are updated. This command should output the following text to the server console:

```
Airtime root folder found at /usr/share/airtime
Updating /usr/share/airtime/application/configs/application.ini
Updating /usr/share/airtime/build/build.properties
Updating /usr/share/airtime/build/runtime-conf.xml
Success!
```

API CLIENT CONFIGURATION

If you have changed the *base_url*, *base_port* or *base_dir* setting in */etc/airtime/airtime.conf* from the defaults, you will probably also have to update the *Hostname* settings in the file */etc/airtime/api_client.cfg* accordingly.

```
bin_dir = /usr/lib/airtime/api_clients
api_key = 'XXXXXXXXXXXXXXXXXXX'
api_base = api
host = airtime.example.com
base_port = 80
base_dir = /
```

MEDIA MONITOR SETTINGS

Airtime's media monitor has its own configuration file in */etc/airtime/media-monitor.cfg* which contains RabbitMQ settings and other preferences.

```
api_client = airtime

# where the binary files live
bin_dir = /usr/lib/airtime/media-monitor

# where the logging files live
log_dir = /var/log/airtime/media-monitor


########################################
# RabbitMQ settings                    #
########################################
rabbitmq_host = localhost
rabbitmq_user = airtime
rabbitmq_password = XXXXXXXXXXXXXXXXXXX
rabbitmq_vhost = /airtime


########################################
# Media-Monitor preferences            #
########################################

#how long to queue up events performed on the files themselves.
check_filesystem_events = 5

#how long to queue metadata input from airtime.
check_airtime_events = 30

# MM2 only:
touch_interval = 5
chunking_number = 450
request_max_wait = 3.0
rmq_event_wait = 0.1
logpath = /var/log/airtime/media-monitor/media-monitor.log
index_path = /var/tmp/airtime/media-monitor/last_index
```

PLAYOUT AND RECORDER SETTINGS

Settings for pypo, the playout and recording engine used by Airtime, are found in the file */etc/airtime/pypo.cfg*. After making changes to this file, you will have to issue the command:

```
sudo /etc/init.d/airtime-playout restart
```

for the changes to take effect.

```
########################################
# pypo - configuration                 #
########################################
# Set the type of client you are using.
# Currently supported types:
# 1) "obp" = Open Broadcast Platform
# 2) "airtime"
#
api_client = airtime


########################################
# Cache Directories                    #
# *include* trailing slash !!          #
########################################
cache_dir = /var/tmp/airtime/pypo/cache/
file_dir = /var/tmp/airtime/pypo/files/
tmp_dir = /var/tmp/airtime/pypo/tmp/


########################################
# Setup Directories                    #
# Do *not* include trailing slash !!   #
########################################
cache_base_dir = /var/tmp/airtime/pypo
bin_dir = /usr/lib/airtime/pypo
log_base_dir = /var/log/airtime
pypo_log_dir = /var/log/airtime/pypo
liquidsoap_log_dir = /var/log/airtime/pypo-liquidsoap


########################################
# Liquidsoap settings                  #
########################################
ls_host = 127.0.0.1
ls_port = 1234


########################################
# RabbitMQ settings                    #
########################################
rabbitmq_host = localhost
rabbitmq_user = airtime
rabbitmq_password = XXXXXXXXXXXXXXXXXXX
```

```
rabbitmq_vhost = /airtime

###########################################
# pypo preferences                        #
###########################################
# Poll interval in seconds.
#
# This will rarely need to be changed because any schedule changes are
# automatically sent to pypo immediately.
#
# This is how often the poll script downloads new schedules and files from the
# server in the event that no changes are made to the schedule.
#
poll_interval = 3600# in seconds.

# Push interval in seconds.
#
# This is how often the push script checks whether it has something new to
# push to liquidsoap.
#
# It's hard to imagine a situation where this should be more than 1 second.
#
push_interval = 1# in seconds

# 'pre' or 'otf'. 'pre' cues while playlist preparation
# while 'otf' (on the fly) cues while loading into ls
# (needs the post_processor patch)
cue_style = pre

###########################################
# Recorded Audio settings                 #
###########################################
record_bitrate = 256
record_samplerate = 44100
record_channels = 2
record_sample_size = 16

#can be either ogg|mp3, mp3 recording requires installation of the package "lame"
record_file_type = ogg

# base path to store recorded shows at
base_recorded_files = /var/tmp/airtime/show-recorder/
```

43. AIRTIME API AUTHENTICATION

The Airtime API enables many types of information about the broadcast schedule and configuration to be retrieved from the Airtime server. Other than the live-info and week-info data fetched by website widgets (see the chapter *Exporting the schedule*), all API requests must be authenticated using the secret API key stored in the file */etc/airtime/api_client.cfg* on the Airtime server. This key is autogenerated during Airtime installation and should be unique for each server.

If you intend to use the Airtime API across a public network, for security reasons it is highly recommended that all API requests are sent over encrypted https: and that the web server is configured to accept requests to the api/ directory from specific host names or IP addresses only.

The format of API requests is:

```
https://airtime.example.com/api/api-action/format/json/api key/XXXXXX
```

where api-action is the type of request and XXXXXX is the secret API key. Available actions include:

- on-air-light - return true if the station is on air
- status - get the status of Airtime components and resource usage
- version - returns the version of Airtime installed
- get-files-without-silan-value - list files for which silence detection has not yet been performed
- get-stream-setting - gets the settings of Airtime output streams
- get-stream-parameters - gets the parameters of Airtime output streams

For example, using the action *get-stream-setting* returns the following output for the first configured stream:

```
{"keyname":"s1_type","value":"ogg","type":"string"},

{"keyname":"s1_host","value":"streaming.example.com","type":"string"},

{"keyname":"s1_port","value":"8000","type":"integer"},

{"keyname":"s1_mount","value":"airtime_128","type":"string"},

{"keyname":"s1_url","value":"http:\/\/airtime.sourcefabric.org","type":"string"},

{"keyname":"s1_description","value":"Airtime Radio! Stream
#1","type":"string"},

{"keyname":"s1_genre","value":"Screamo","type":"string"},
```

which is enough information to construct a player widget dynamically. (s1_url is the station's homepage, not the stream URL). The same information is provided with an s2_ prefix for the second stream, and s3_ prefix for the third stream.

Some API requests require the directory ID number to be specified as *dir_id* including:

- list-all-files - list files in the specified directory
- get-files-without-replay-gain - list files in the specified directory for which ReplayGain has not been calculated yet

For example, using a request such as:

```
 http://airtime.example.com/api/list-all-files/format/json/api_key/XXXXXX/dir_id/1/
```

returns the full path to each media file in the Airtime storage directory:

```
{"files":[
"imported\/1\/Mark Ronson feat. Saigon\/Here Comes the Fuzz\/7-Diduntdidunt-unknown.flac",
"imported\/1\/Jimi Tenor & Tony Allen\/Inspiration Information\/3-Selfish Gene-128kbps.mp3",
"]}
```

44. SECURE LOGIN WITH SSL OR TLS

To increase the security of your server, you can enable encrypted access to the Airtime administration interface, and direct your users towards this more secure login page. The main advantage of using this encryption is that your remote users' login names and passwords are not sent in plain text across the public Internet or untrusted local networks, such as shared Wi-Fi access points.

The padlock icon in a web browser's address bar depends on the browser's recognition of an encryption certificate. Because the users of your Airtime server will often be known to you personally, it is feasible to use a self-signed certificate for this purpose. Alternatively, you can pay a Certificate Authority to sign the certificate for you. Airtime Pro servers are pre-configured with a certificate signed by a Certificate Authority which is automatically recognised by all popular browsers.

DEPLOYING A SELF-SIGNED CERTIFICATE

The Debian/Ubuntu package *ssl-cert* creates a *snakeoil* certificate and key based on your server's hostname. This gratis certificate and key pair created under the */etc/ssl/certs/* and */etc/ssl/private/* directories will not be recognised by users' browsers without manual intervention. You can install the *ssl-cert* package with the command:

```
sudo apt-get install ssl-cert
```

You should enable additional Apache modules for page redirections, custom headers and secure access:

```
sudo a2enmod alias headers ssl
```

Next, edit the virtual host configuration for your Airtime server to include a stanza for the https:// interface on port 443 and a redirect for logins from port 80:

```
sudo nano /etc/apache2/sites-available/airtime-vhost.conf
```

Using the following configuration as a guide, replace *airtime.example.com* with the name of your server and *admin@example.com* with your email address. The SSLv2 protocol should be disabled, as it is generally believed to be insecure. You may wish to create a *ServerAlias* for users to access the administration interface over https:// if required.

On port 80, Apache's *alias* module is used to set a *Redirect permanent* for the login page. Access should be denied to all sites except *localhost* and any other Airtime servers on your network, so that unencrypted communication between Airtime components can continue.

```
<VirtualHost *:443>
      SSLEngine on
      SSLProtocol all -SSLv2
      SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
      SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
      Header always set Strict-Transport-Security "max-age=31536000"

      ServerName airtime.example.com
      #ServerAlias www.example.com

      ServerAdmin admin@example.com

      DocumentRoot /usr/share/airtime/public
      DirectoryIndex index.php

      <Directory /usr/share/airtime/public>
              Options -Indexes FollowSymLinks MultiViews
              AllowOverride all
              Order allow,deny
              Allow from all
      </Directory>
</VirtualHost>

<VirtualHost *:80>
      ServerName airtime.example.com

      ServerAdmin admin@example.com

      DocumentRoot /usr/share/airtime/public
      Redirect permanent /login https://airtime.example.com/login

      SetEnv APPLICATION_ENV "production"

      <Directory /usr/share/airtime/public>
              Options -Indexes FollowSymLinks MultiViews
              AllowOverride All
              Order deny,allow
              Deny from all
              Allow from localhost
      </Directory>
</VirtualHost>
```

Save the file with **Ctrl+O** and exit the **nano** editor with **Ctrl+X**. Then restart Apache with the command:

```
sudo service apache restart
```

When attempting to log into your server via http:// in future, you should be redirected to https:// automatically.

IMPORTING A SELF-SIGNED CERTIFICATE INTO THE BROWSER

The first time you access an Airtime server with a self-signed certificate over https:// your browser will block the login page and display a security warning. In **Mozilla Firefox**, you can click **Technical Details** to confirm that the warning is due to the certificate being self-signed before clicking the **Add Exception** button. In **Google Chrome**, the button to click on the security warning page is **Proceed Anyway**.
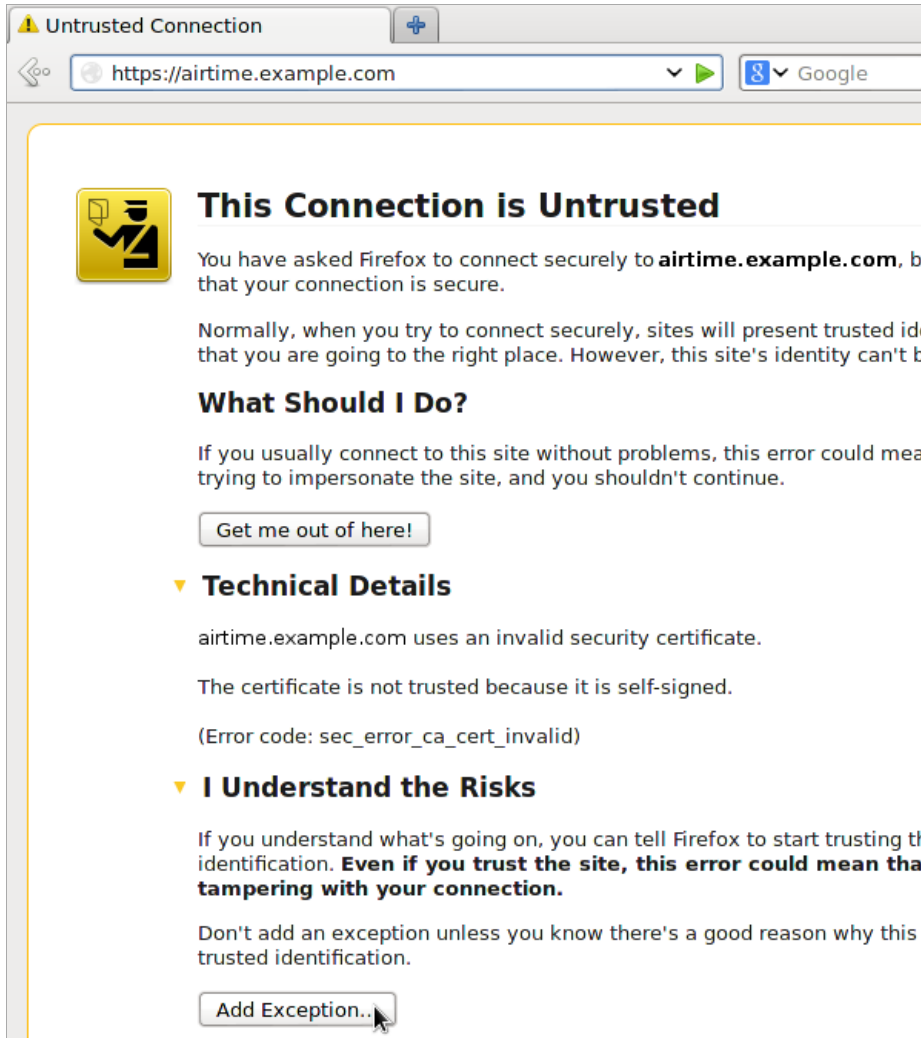
On the next page in Firefox, click the **Get Certificate** button to inspect the details of the self-signed certificate. If all is well, click the **Confirm Security Exception** button. You should now be able to proceed to the https:// login page.

Whether your certificate is self-signed or not, you will see browser security warnings whenever a https:// page is delivering unencrypted content, such as the stream from an Icecast server. In Firefox, an exclamation mark icon is displayed in the address bar of the **Listen** pop-up.
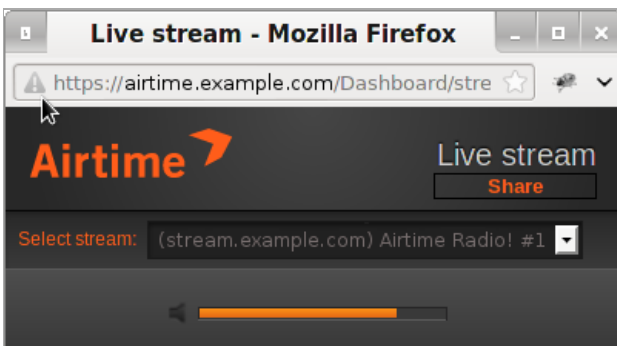
45. ICECAST STATISTICS WITH PIWIK

Piwik (http://piwik.org/) is an open source web analytics platform which supports Icecast's log format, in Piwik version 2.0 or later. For your convenience, packages of Piwik for Debian and Ubuntu are provided in the http://apt.sourcefabric.org repository. If you have already configured your Airtime server for this repository, you can install the **piwik** package and its dependencies with the command:

```
sudo apt-get install piwik php5-geoip php5-cli mysql-server
```

For security reasons, you should set a strong root password for the MySQL server. If you have not set a MySQL root password already, you should be promoted to do this during the installation of the **mysql-server** package.

DATABASE AND WEB SERVER CONFIGURATION

Piwik uses a MySQL database which must be created manually before you can use the Piwik web interface.

1. Log into the database management system with the root password that you set for the MySQL server:

```
mysql -uroot -p
Enter password:
```

The server should respond with:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2. At the mysql command prompt, create a database named *piwik* with the command:

```
CREATE DATABASE `piwik` /*!40100 DEFAULT CHARACTER SET utf8 */;
```

3. Create a MySQL user and a strong password with access to the database previously created:

```
CREATE USER piwik@localhost IDENTIFIED BY 'my-strong-password';
GRANT ALL PRIVILEGES ON piwik.* to piwik@localhost;
GRANT FILE ON *.* TO piwik@localhost;
QUIT
```

4. Create a virtual host for the piwik web interface in *etc/apache2/sites-available/piwik-vhost.conf* setting the *ServerName* as appropriate for your server:

```
<VirtualHost *:80>
   ServerName stats.example.com
   DocumentRoot /usr/share/piwik
   Include "conf.d/piwik.conf"
</VirtualHost>
```

5. Enable the virtual host and reload the web server:

```
sudo a2ensite pwiki-vhost.conf
sudo invoke-rc.d apache2 reload
```

6. Open the *ServerName* that you set for Piwik in your browser. You should see the Piwik setup pages. Provide the MySQL database and Piwik *Super User* details when requested. If you intended to use Piwik for Icecast statistics only, you can skip the step *JavaScript Tracking Code*.

7. Uncomment the last line in the crontab file */etc/cron.d/piwik* and set your Pwik *ServerName* to enable automatic archiving every five minutes:

```
5 * * * * www-data [ -e /usr/share/piwik/misc/cron/archive.php ]
&& [ -x /usr/bin/php ]
&& /usr/bin/php /usr/share/piwik/misc/cron/archive.php
-- "url=http://stats.example.com/" >/dev/null 2>&1
```

See http://piwik.org/docs/ for more details of Piwik installation and configuration.

CONFIGURE PIWIK FOR GEOLOCATION

By default, Piwik uses browser language settings to guess the countries in which your station's audience is physically located. In order to satisfy broadcast regulators and music royalty collection societies, a more accurate method is required, based on the location of individual IP addresses.

1. Edit the file */etc/php5/apache2/conf.d/20-geoip.ini* to set the path to your Piwik installation:

```
extension=geoip.so
geoip.custom directory=/usr/share/piwik/misc
```

Then restart the web server for the change to take effect:

```
sudo invoke-rc.d apache2 restart
```

2. Download GeoLite data from MaxMind's website http://dev.maxmind.com/geoip/legacy/geolite/ under the Creative Commons Attribution-ShareAlike 3.0 Unported License, for example:

```
wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
```

2. Unzip and set permissions on the data file, then move it to the *misc* directory under the Piwik install directory and rename it to *GeoIPCity.dat* :

```
gunzip GeoLiteCity.dat.gz
chmod 644 GeoLiteCity.dat
sudo mv GeoLiteCity.dat /usr/share/piwik/misc/GeoIPCity.dat
```

4. In the Piwik web interface click **Settings** in the top right corner, then **Geolocation** in the left side menu. Change the setting from *Default* to *GeoIP (PECL)* .

5. Optionally, configure *Setup automatic updates of GeoIP databases* at the end of the Geolocation page.

IMPORTING AN ICECAST LOG FILE

In addition to the JavaScript tracking code used for web analytics, Piwik includes a Python script which can import a server log file directly. This script accepts the parameter *--log-format-name=icecast2* which enables support for the connection duration field logged by Icecast when a client disconnects. It is this field which enables stations to track the duration of individual audience connections, and calculate both the average connection duration and the 'aggregate tuning hours' figure required by some broadcast regulators.

For an Icecast log file at */var/log/icecast2/access.log* you can run the script as follows:

```
python /usr/share/piwik/misc/log-analytics/import_logs.py --show-progress
--url=http://stats.example.com --idsite=1 --recorders=8 --enable-http-errors
--log-format-name=icecast2 --strip-query-string /var/log/icecast2/access.log
```

where the parameters are:

```
--url=http://stats.example.com
```

The ServerName of your Piwik installation.

```
--idsite=1
```

Number of the default site configured in Piwik.

```
--recorders=8
```

How many threads to use while parsing the log file. To begin with, set this value to the number of CPU cores you can spare on the Piwik server.

```
--enable-http-errors
```

Collect statistics for errors, such as stream links not found.

```
--strip-query-string
```

Used because any characters requested by the client after the Icecast mounpoint URL can confuse the statistics.

If the Piwik server is remote, you can use an additional parameter *--token_auth=* to authenticate the request. You can find the token to use for your installation by logging in to Piwik and clicking **API** in the main navigation menu at the top of the page. Further details of the import script are shown at http://piwik.org/docs/log-analytics-tool-how-to/

The script will report the number of records processed and the time that has been required to parse the Icecast log file. If the import script is taking too long to execute, there are performance and configuration tips in the https://github.com/piwik/piwik/blob/master/misc/log-analytics/README.md file.

Open the Piwik dashboard in your browser, and you should now see a summary of the imported data, including the most popular streams (listed under *Entry Pages*), *Visits by Server Time*, *Referrer Websites* and a *Visitor Map*.

Visitor Map

201 visits
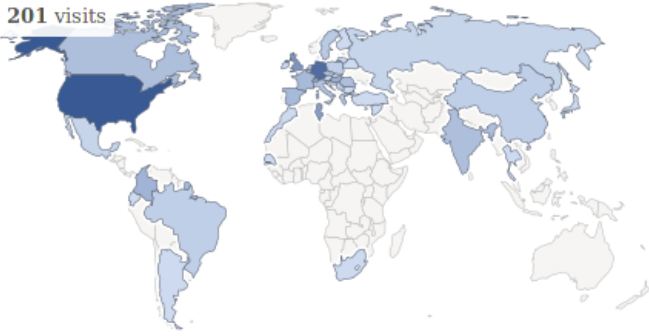
Countries | World-Wic ▼ | Visits ▼

Visitor Browser

BROWSER	VISITS ▼
Unknown	78
Chrome	64
Firefox	30
Internet Explorer	15
Safari	8

1-5 of 8   NEXT ›

5 ▼

Related report: Browser version

APPENDIX

46. EXPERT INSTALL
47. HD AUDIO MODELS
48. ABOUT THIS MANUAL

46. EXPERT INSTALL

These quick install steps are suitable for experienced GNU/Linux system administrators who have already followed the steps shown in the chapter *Preparing the server* earlier in this book. For a more detailed explanation of the steps below, please read the chapter *Automated installation*.

1. Edit the repositories file for your server:

```
sudo nano /etc/apt/sources.list
```

For Ubuntu Precise [or Quantal, Raring, Saucy] servers, use the Sourcefabric repository:

```
deb http://apt.sourcefabric.org/ precise main
```

substituting *precise* if appropriate. Make sure you have enabled the multiverse repository for MP3 encoding support:

```
deb http://archive.ubuntu.com/ubuntu/ precise multiverse
```

For Debian wheezy [or squeeze] servers, use the Sourcefabric repository:

```
deb http://apt.sourcefabric.org/ wheezy main
```

If using Debian squeeze, also enable the backports repository for MP3 encoding support:

```
deb http://backports.debian.org/debian-backports squeeze-backports main
```

2. Install the Sourcefabric package signing key, then update again:

```
sudo apt-get update
sudo apt-get install sourcefabric-keyring
sudo apt-get update
```

3. Install the database management system (for a single server configuration):

```
sudo apt-get install postgresql
```

4. Install the streaming media server (optional, it may be remote):

```
sudo apt-get install icecast2
```

5. Remove PulseAudio, if installed:

```
sudo apt-get purge pulseaudio
```

6. Install Airtime:

```
sudo apt-get install airtime
```

Refer to the *Configuration* chapter for configuration options. Now you should be able to log in to the Airtime administration interface, as shown in the *Getting started* chapter.

47. HD AUDIO MODELS

This listing is provided to help ensure that the correct model parameter is passed to the ALSA kernel module for an Intel HDA soundcard, if one is fitted to your Airtime server. See the chapter *Preparing the server* in this book for more details.

```
  Model name Description
  ---------- -----------
ALC880
======
  3stack 3-jack in back and a headphone out
  3stack-digout 3-jack in back, a HP out and a SPDIF out
  5stack 5-jack in back, 2-jack in front
  5stack-digout 5-jack in back, 2-jack in front, a SPDIF out
  6stack 6-jack in back, 2-jack in front
  6stack-digout 6-jack with a SPDIF out

ALC260
======
  N/A

ALC262
======
  inv-dmic Inverted internal mic workaround

ALC267/268
==========
  inv-dmic Inverted internal mic workaround

ALC269/270/275/276/28x/29x
======
  laptop-amic  Laptops with analog-mic input
  laptop-dmic  Laptops with digital-mic input
  alc269-dmic  Enable ALC269(VA) digital mic workaround
  alc271-dmic  Enable ALC271X digital mic workaround
  inv-dmic  Inverted internal mic workaround
  headset-mic  Indicates a combined headset (headphone+mic) jack
  lenovo-dock     Enables docking station I/O for some Lenovos
  dell-headset-multi Headset jack, which can also be used as mic-in
  dell-headset-dock Headset jack (without mic-in), and also dock I/O

ALC66x/67x/892
==============
  mario    Chromebook mario model fixup
  asus-mode1  ASUS
  asus-mode2  ASUS
  asus-mode3  ASUS
  asus-mode4  ASUS
  asus-mode5  ASUS
  asus-mode6  ASUS
  asus-mode7  ASUS
  asus-mode8  ASUS
  inv-dmic  Inverted internal mic workaround
  dell-headset-multi Headset jack, which can also be used as mic-in

ALC680
======
  N/A

ALC88x/898/1150
====================
  acer-aspire-4930g Acer Aspire 4930G/5930G/6530G/6930G/7730G
  acer-aspire-8930g Acer Aspire 8330G/6935G
  acer-aspire  Acer Aspire others
  inv-dmic  Inverted internal mic workaround
  no-primary-hp  VAIO Z/VGC-LN51JGB workaround (for fixed speaker DAC)

ALC861/660
==========
  N/A

ALC861VD/660VD
==============
  N/A

CMI9880
=======
  minimal 3-jack in back
  min_fp 3-jack in back, 2-jack in front
  full  6-jack in back, 2-jack in front
  full_dig 6-jack in back, 2-jack in front, SPDIF I/O
  allout 5-jack in back, 2-jack in front, SPDIF out
  auto  auto-config reading BIOS (default)

AD1882 / AD1882A
================
  3stack 3-stack mode
  3stack-automute 3-stack with automute front HP (default)
```

```
  6stack 6-stack mode

AD1884A / AD1883 / AD1984A / AD1984B
====================================
  desktop 3-stack desktop (default)
  laptop laptop with HP jack sensing
  mobile mobile devices with HP jack sensing
  thinkpad Lenovo Thinkpad X300
  touchsmart HP Touchsmart

AD1884
======
  N/A

AD1981
======
  basic  3-jack (default)
  hp   HP nx6320
  thinkpad Lenovo Thinkpad T60/X60/Z60
  toshiba Toshiba U205

AD1983
======
  N/A

AD1984
======
  basic  default configuration
  thinkpad Lenovo Thinkpad T61/X61
  dell_desktop Dell T3400

AD1986A
=======
  6stack 6-jack, separate surrounds (default)
  3stack 3-stack, shared surrounds
  laptop 2-channel only (FSC V2060, Samsung M50)
  laptop-eapd 2-channel with EAPD (ASUS A6J)
  laptop-automute 2-channel with EAPD and HP-automute (Lenovo N100)
  ultra  2-channel with EAPD (Samsung Ultra tablet PC)
  samsung 2-channel with EAPD (Samsung R65)
  samsung-p50 2-channel with HP-automute (Samsung P50)

AD1988/AD1988B/AD1989A/AD1989B
==============================
  6stack 6-jack
  6stack-dig ditto with SPDIF
  3stack 3-jack
  3stack-dig ditto with SPDIF
  laptop 3-jack with hp-jack automute
  laptop-dig ditto with SPDIF
  auto  auto-config reading BIOS (default)

Conexant 5045
=============
  laptop-hpsense    Laptop with HP sense (old model laptop)
  laptop-micsense   Laptop with Mic sense (old model fujitsu)
  laptop-hpmicsense Laptop with HP and Mic senses
  benq  Benq R55E
  laptop-hp530 HP 530 laptop
  test  for testing/debugging purpose, almost all controls
  can be adjusted.  Appearing only when compiled with
  $CONFIG_SND_DEBUG=y

Conexant 5047
=============
  laptop Basic Laptop config
  laptop-hp Laptop config for some HP models (subdevice 30A5)
  laptop-eapd Laptop config with EAPD support
  test  for testing/debugging purpose, almost all controls
  can be adjusted.  Appearing only when compiled with
  $CONFIG_SND_DEBUG=y

Conexant 5051
=============
  laptop Basic Laptop config (default)
  hp   HP Spartan laptop
  hp-dv6736 HP dv6736
  hp-f700 HP Compaq Presario F700
  ideapad Lenovo IdeaPad laptop
  toshiba Toshiba Satellite M300

Conexant 5066
=============
  laptop Basic Laptop config (default)
  hp-laptop HP laptops, e g G60
  asus  Asus K52JU, Lenovo G560
  dell-laptop Dell laptops
  dell-vostro Dell Vostro
  olpc-xo-1_5 OLPC XO 1.5
  ideapad       Lenovo IdeaPad U150
  thinkpad Lenovo Thinkpad

STAC9200
```

```
========
  ref  Reference board
  oqo  OQO Model 2
  dell-d21 Dell (unknown)
  dell-d22 Dell (unknown)
  dell-d23 Dell (unknown)
  dell-m21 Dell Inspiron 630m, Dell Inspiron 640m
  dell-m22 Dell Latitude D620, Dell Latitude D820
  dell-m23 Dell XPS M1710, Dell Precision M90
  dell-m24 Dell Latitude 120L
  dell-m25 Dell Inspiron E1505n
  dell-m26 Dell Inspiron 1501
  dell-m27 Dell Inspiron E1705/9400
  gateway-m4 Gateway laptops with EAPD control
  gateway-m4-2 Gateway laptops with EAPD control
  panasonic Panasonic CF-74
  auto  BIOS setup (default)

STAC9205/9254
=============
  ref  Reference board
  dell-m42 Dell (unknown)
  dell-m43 Dell Precision
  dell-m44 Dell Inspiron
  eapd  Keep EAPD on (e.g. Gateway T1616)
  auto  BIOS setup (default)

STAC9220/9221
=============
  ref  Reference board
  3stack D945 3stack
  5stack D945 5stack + SPDIF
  intel-mac-v1 Intel Mac Type 1
  intel-mac-v2 Intel Mac Type 2
  intel-mac-v3 Intel Mac Type 3
  intel-mac-v4 Intel Mac Type 4
  intel-mac-v5 Intel Mac Type 5
  intel-mac-auto Intel Mac (detect type according to subsystem id)
  macmini Intel Mac Mini (equivalent with type 3)
  macbook Intel Mac Book (eq. type 5)
  macbook-pro-v1 Intel Mac Book Pro 1st generation (eq. type 3)
  macbook-pro Intel Mac Book Pro 2nd generation (eq. type 3)
  imac-intel Intel iMac (eq. type 2)
  imac-intel-20 Intel iMac (newer version) (eq. type 3)
  ecs202 ECS/PC chips
  dell-d81 Dell (unknown)
  dell-d82 Dell (unknown)
  dell-m81 Dell (unknown)
  dell-m82 Dell XPS M1210
  auto  BIOS setup (default)

STAC9202/9250/9251
==================
  ref  Reference board, base config
  m1   Some Gateway MX series laptops (NX560XL)
  m1-2  Some Gateway MX series laptops (MX6453)
  m2   Some Gateway MX series laptops (M255)
  m2-2  Some Gateway MX series laptops
  m3   Some Gateway MX series laptops
  m5   Some Gateway MX series laptops (MP6954)
  m6   Some Gateway NX series laptops
  auto  BIOS setup (default)

STAC9227/9228/9229/927x
=======================
  ref  Reference board
  ref-no-jd Reference board without HP/Mic jack detection
  3stack D965 3stack
  5stack D965 5stack + SPDIF
  5stack-no-fp D965 5stack without front panel
  dell-3stack Dell Dimension E520
  dell-bios Fixes with Dell BIOS setup
  dell-bios-amic Fixes with Dell BIOS setup including analog mic
  volknob Fixes with volume-knob widget 0x24
  auto  BIOS setup (default)

STAC92HD71B*
============
  ref  Reference board
  dell-m4-1 Dell desktops
  dell-m4-2 Dell desktops
  dell-m4-3 Dell desktops
  hp-m4  HP mini 1000
  hp-dv5 HP dv series
  hp-hdx HP HDX series
  hp-dv4-1222nr HP dv4-1222nr (with LED support)
  auto  BIOS setup (default)

STAC92HD73*
===========
  ref  Reference board
  no-jd  BIOS setup but without jack-detection
```

```
  intel  Intel DG45* mobos
  dell-m6-amic Dell desktops/laptops with analog mics
  dell-m6-dmic Dell desktops/laptops with digital mics
  dell-m6 Dell desktops/laptops with both type of mics
  dell-eq Dell desktops/laptops
  alienware Alienware M17x
  auto  BIOS setup (default)

STAC92HD83*
===========
  ref  Reference board
  mic-ref Reference board with power management for ports
  dell-s14 Dell laptop
  dell-vostro-3500 Dell Vostro 3500 laptop
  hp-dv7-4000 HP dv-7 4000
  hp_cNB11_intquad HP CNB models with 4 speakers
  hp-zephyr HP Zephyr
  hp-led HP with broken BIOS for mute LED
  hp-inv-led HP with broken BIOS for inverted mute LED
  auto  BIOS setup (default)

STAC9872
========
  vaio  VAIO laptop without SPDIF
  auto  BIOS setup (default)

Cirrus Logic CS4206/4207
========================
  mbp55  MacBook Pro 5,5
  imac27 IMac 27 Inch
  auto  BIOS setup (default)

Cirrus Logic CS4208
===================
  mba6  MacBook Air 6,1 and 6,2
  gpio0  Enable GPIO 0 amp
  auto  BIOS setup (default)

VIA VT17xx/VT18xx/VT20xx
========================
  auto  BIOS setup (default)
```

48. ABOUT THIS MANUAL

Copyright © 2011-2013 Sourcefabric o.p.s

This manual was written for Sourcefabric by Daniel James, using Booktype Pro. Please add your comments and contributions at:
http://sourcefabric.booktype.pro/airtime-25-for-broadcasters/

LICENSE

All chapters in this manual are licensed with the **GNU General Public License version 3**.

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS