
Trunk Player Documentation

Release 0.0.1

Dylan Reinhold

Jan 30, 2022

Contents

1	Installation	3
1.1	System Prerequisites	3
1.2	Assumptions	3
1.3	Getting Trunk Player	4
1.4	Setup Virtual Environment	4
1.5	Activate Virtual Environment	4
1.6	Install Python Packages	4
1.7	Configure for first use	4
1.7.1	Local settings file	4
1.7.2	Configure Postgres Database	5
1.7.3	Initialize the database	5
1.7.4	Create admin account	5
1.7.5	Starting the test web server	6
2	Configuration	7
2.1	Site Title/Name	7
2.2	Media Location	7
2.3	Set Timezone	7
3	Configure - Local Audio Files	9
3.1	System Prerequisites	9
3.2	Nginx	9
3.3	Configure Trunk Player to use local audio files	10
3.4	Static html files	10
3.5	Starting django web services	10
3.6	Adding new transmissions	10
4	Configure - Local Audio Files Multiple Folders	13
4.1	System Prerequisites	13
4.2	Update trunk-player	13
4.3	Update Trunk-Recorder	14
4.4	Running new config	14
5	Configure - Local dynamic pages	15
5.1	Overview	15
5.2	Setup	15

6	Settings - List of all settings	17
6.1	SITE_TITLE	17
6.2	SECRET_KEY	17
6.3	DEBUG	17
6.4	ALLOWED_HOSTS	17
6.5	DATABASES	18
6.6	SITE_EMAIL	18
6.7	DEFAULT_FROM_EMAIL	18
6.8	AUDIO_URL_BASE	18
6.9	ANONYMOUS_TIME	18
6.10	TIME_ZONE	18
6.11	ACCESS_TG_RESTRICT	18
6.12	GOOGLE_ANALYTICS_PROPERTY_ID	18
6.13	TWITTER_ACTIVE	18
6.14	LOGIN_REDIRECT_URL	19
6.15	JS_SETTINGS	19
6.16	JS_SETTINGS	19
6.17	ALLOW_ANONYMOUS	19
6.18	RADIO_DEFAULT_UNIT_AGENCY	19
6.19	AMAZON_ADDS	19
6.20	SOCIALACCOUNT_PROVIDERS	19
6.21	ACCOUNT_AUTHENTICATION_METHOD	19
6.22	ACCOUNT_EMAIL_REQUIRED	20
6.23	BASE_DIR	20
6.24	LOGIN_URL	20
6.25	INSTALLED_APPS	20
6.26	MIDDLEWARE_CLASSES	20
6.27	ROOT_URLCONF	20
6.28	TEMPLATES	20
6.29	WSGI_APPLICATION	20
6.30	AUTHENTICATION_BACKENDS	20
6.31	AUTH_PASSWORD_VALIDATORS	20
6.32	LANGUAGE_CODE	21
6.33	USE_I18N	21
6.34	USE_L10N	21
6.35	USE_TZ	21
6.36	SECURE_PROXY_SSL_HEADER	21
6.37	STATIC_URL	21
6.38	STATIC_ROOT	21
6.39	REST_FRAMEWORK	21
6.40	MEDIA_URL	21
6.41	MEDIA_URL	21
6.42	CHANNEL_LAYERS	22
6.43	CACHES	22
6.44	SITE_ID	22
6.45	USE_RAW_ID_FIELDS	22
7	Supervisor	23
7.1	Install Supervisor	23
7.2	Update Config	23
7.3	Enable the config	23
7.4	Restarting trunk-player	24
8	Upgrades	25

8.1	Upgrade to Trunk Player 0.1.2	25
8.1.1	System Prerequisites	25
8.1.2	Assumptions	25
8.1.3	Upgrading Trunk Player	26
8.1.3.1	Stopping Trunk Player	26
8.1.3.2	Remove Old Virtual Environment	26
8.1.3.3	Setup Virtual Environment	26
8.1.3.4	Activate Virtual Environment	26
8.1.3.5	Re-Install Python Packages	26
8.1.3.6	Backup Postgres Database	27
8.1.3.7	Upgrade the database	27
8.1.3.8	Starting the test web server	27
8.1.3.9	Restarting Trunk Player	27
8.1.4	You Did It!	27
8.1.5	Rollback Trunk-Player Install	28
8.1.5.1	Rollback Trunk-Player Code	28
8.1.5.2	Rollback Postgres Database (Only if needing to rollback)	28
8.1.5.3	Remove Old Virtual Environment	28
8.1.5.4	Setup Virtual Environment	28
8.1.5.5	Activate Virtual Environment	28
8.1.5.6	Re-Install Python Packages	29
8.1.5.7	Restarting Trunk Player	29
8.2	Upgrade to Trunk Player 0.1.3	29
8.2.1	System Prerequisites	29
8.2.2	Assumptions	29
8.2.3	Upgrading Trunk Player	30
8.2.3.1	Stopping Trunk Player	30
8.2.3.2	Restarting Trunk Player	30
8.2.4	You Did It!	30

9 Indices and tables 31

Contents:

CHAPTER 1

Installation

This document will show you how to get up and running with Trunk Player.

Currently install is based on a Debian like system, Ubuntu.

1.1 System Prerequisites

- Linux Machine it might work in windows but has not been tested
- Python 3.5.x
- Virtualenv
- PIP Should be installed with Python 3
- Redis 3.x.x
- PostgreSQL 9.x
- git

Using apt-get

```
$ sudo apt-get install python3-dev virtualenv redis-server python3-pip postgresql_
↳ libpq-dev postgresql-client postgresql-client-common git
```

1.2 Assumptions

- Project directory is /home/radio/trunk-player - You can replace /home/radio with your own home directory.
- Redis is running and listening on default port of : 6379

1.3 Getting Trunk Player

The source is on GitHub, use git to clone the repository. Starting from your home directory of `/home/radio`

```
$ git clone https://github.com/ScanOC/trunk-player.git
```

This will pull down the most current version of Trunk Player.

1.4 Setup Virtual Environment

Setup a new Python 3.x virtual environment in the `env` directory. Set the visual prompt to `(Trunk Player)`.

```
$ cd trunk-player
$ virtualenv -p python3 env --prompt='(Trunk Player)'
```

1.5 Activate Virtual Environment

You will need to re run this step each time you start a new shell or log into your machine.

```
$ source env/bin/activate
```

This will set you into a new python environment any packages you install via `pip` will only live in this area and do not touch your system files. This allowed you to have multiple projects with different dependencies.

You can use the command `deactivate` to exit back to your normal system environment.

1.6 Install Python Packages

Using `pip` install all required packages from the `requirements.txt` file.

```
(Trunk Player)$ pip install -r requirements.txt
```

1.7 Configure for first use

You will need to setup a local version of the `setting.py` file, create and initialize the database, and create a default admin account.

1.7.1 Local settings file

You will need to create a local settings file to override any settings in the `trunk_player/settings.py` file. This will allow you to pull down updates from GitHub without losing your local settings.

Make a copy of the sample local settings file

```
(Trunk Player)$ cp trunk_player/settings_local.py.sample trunk_player/settings_local.
↪py
```

Important You need to set/change the `SECRET_KEY` in the `trunk_player/settings_local.py`. This value is used to protect sensitive data like passwords. If you keep the one from the project a bad actor may be able to compromise your site or worse your server. See the django project about [SECRET_KEY](#).

1.7.2 Configure Postgres Database

You need to create a postgres user that has full access to your database.

Logged into your postgres database as an admin user

```
$ sudo su - postgres
(postgres)$ psql
```

Create your user (trunk_player_user, with pass CHANGE_ME)

```
postgres=# CREATE USER trunk_player_user WITH PASSWORD 'CHANGE_ME';
```

Create your database named trunk_player

```
postgres=# CREATE DATABASE trunk_player;
```

Allow your user full control of the new DB

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE trunk_player TO trunk_player_user;
```

Configure some settings as recommended by [Django](#)

```
ALTER ROLE trunk_player_user SET client_encoding TO 'utf8';
ALTER ROLE trunk_player_user SET default_transaction_isolation TO 'read committed';
ALTER ROLE trunk_player_user SET timezone TO 'UTC';
```

Exit from postgres and back to your user

```
postgres=# \q
(postgres)$ exit
$
```

Edit the `trunk_player/settings_local.py` and configure the `DATABASES` to match your server/username/passwords.

1.7.3 Initialize the database

Using the `django manage.py` command to build the new database.

```
(Trunk Player)$ ./manage.py migrate
```

1.7.4 Create admin account

```
(Trunk Player)$ ./manage.py createsuperuser
Username: test
Email address: test@sample.com
Password: mypassword
Password (again): mypassword
Superuser created successfully.
```

1.7.5 Starting the test web server

First note this is not full production ready server. It can handle a couple users.

Using the `manage.py` command again

```
(Trunk Player)$ ./manage.py runserver
```

This will start the server up listening on the local loopback address on port 8000. Start your web browser and go to <http://localhost:8000>. You should see the main page Visit `/admin/` to log into the admin area.

If you are running this on a remote server you need to have the web server use its' public IP address so you can connect.

```
(Trunk Player)$ ./manage.py runserver 0.0.0.0:8000
```

This will run the server also on port 8000 but will be accessible via the servers IP address or dns name on port 8000 also.

Post install configurations

2.1 Site Title/Name

In `trunk_player/settings_local.py` set the

```
SITE_TITLE = 'Trunk-Player'
```

2.2 Media Location

In `trunk_player/settings_local.py` include the base url for the amazon s3 location of your audio files

```
AUDIO_URL_BASE = '//s3.amazonaws.com/SET-TO-MY-BUCKET/'
```

2.3 Set Timezone

In `trunk_player/settings_local.py` set the

```
TIME_ZONE = 'America/Los_Angeles'
```

Configure - Local Audio Files

This document will show you how to configure Trunk Player using nginx as a frontend reverse proxy and serving the audio files from local storage

Currently install is based on a Debian like system, Ubuntu.

3.1 System Prerequisites

- After installing using *install.rst* as a guide

All paths in the document are based on trunk-player being in /home/radio, change as needed

3.2 Nginx

Using your system packageing tool install the current version of Nginx

Using apt-get in unbuntu

```
$ sudo apt-get install nginx
```

Use the sample nginx config file in the project as a starting point

```
$ cd /home/radio/trunk-player
$ cp trunk_player/trunk_player.nginx.sample trunk_player/trunk_player.nginx
```

Link this config into the nginx config area for active configs, we will also remove the default conig shipped with nginx. *Leave this if you are alreading using nginx with other sites on the same server, you will need to remove the “default” statemnets from our sample if so”*

```
$ sudo ln -s /home/radio/trunk-player/trunk_player/trunk_player.nginx /etc/nginx/
↳ sites-enabled/
$ sudo rm /etc/nginx/sites-enabled/default
```

Restart nginx to pick up the new configs

```
$ sudo systemctl restart nginx
```

3.3 Configure Trunk Player to use local audio files

Edit local settings file `trunk_player/settings_local.py`

Change

```
AUDIO_URL_BASE = '//s3.amazonaws.com/SET-TO-MY-BUCKET/'
```

to

```
AUDIO_URL_BASE = '/audio_files/'
```

If you are not running an SSL (HTTPS) site add this to the end of `trunk_player/settings_local.py`

```
SECURE_PROXY_SSL_HEADER = ()
```

3.4 Static html files

Have django put all the static files javascript, css, and images into a static folder that nginx can server.

You need to re run this command any time any of these files is changed.

Start by starting the python virtual environment (if you have not yet)

```
$ source env/bin/activate
$ ./manage.py collectstatic --noinput
```

3.5 Starting django web services

You need to start both the daphne service and the runworker. You can run multiple runworker instances, the recommended is not to exceed 2 per cpu core.

You can run one on each window to watch the messages while you test.

First window start daphne, make sure you are in the python virtual environment.

```
$ daphne trunk_player.asgi:channel_layer --port 7055 --bind 127.0.0.1
```

Now start atleast 1 runworker

```
$ ./manage.py runworker livecall-scan-default
```

3.6 Adding new transmissions

You can use `utility/trunk-recoder/encode-upload.sh` as a guide. You want to have both the json and an mp3 file in the `audio_files` folder.


```
$ ./manage.py add_transmission 1200-1488505593_8.60713e+08
```

Note: you leave off the extension of the filename

Configure - Local Audio Files Multiple Folders

This document will show you how to configure Trunk Player using multiple folders to store and play back your audio in the same structure as trunk-recorder.

Currently install is based on a Debian like system, Ubuntu.

4.1 System Prerequisites

- After configuring for Local Audio Files
- Trunk-player is in `/home/radio/trunk-player`
- Audio files will be served via a web-server (nginx) from `/home/radio/trunk-player/audio_files/`
- Trunk-recorder is in `/home/radio/trunk-recorder`
- Both `sox` and `lame` installed.

4.2 Update trunk-player

Change directory into the base trunk-player directory

```
$ cd /home/radio/trunk-player
```

Pull down latest version of trunk-player from github

```
$ git pull
```

Activate virtualenv

```
$ source env/bin/activate
```

Migrate any new database schema changes

```
$ ./manage.py migrate
```

4.3 Update Trunk-Recorder

Configure trunk-recorder to save our wav/json files in `/home/radio/trunk-player/audio_files/`

Edit the trunk-recorder config file `/home/radio/trunk-recorder/config.json`

Change the `captureDir` to `/home/radio/trunk-player/audio_files`

In the systems list for each system you have, edit the `uploadScript` to `encode-local-sys-0.sh` change the 0 to the system number you want it to be saved as in trunk-player

Save the config

Copy the sample script `/home/radio/trunk-player/utility/trunk-recorder/encode-local-sys-0.sh` to `/home/radio/trunk-recorder/`

```
$ cp utility/trunk-recorder/encode-local-sys-0.sh /home/radio/trunk-recorder/encode-  
→local-sys-0.sh
```

If you have multiple systems, update the zero (0) in the second part of the cp command for each system number

4.4 Running new config

Re start both trunk-recorder and trunk-player.

Configure - Local dynamic pages

This document will show you how to setup local web pages

5.1 Overview

Using the WebHtml table in admin you can create custom web pages, just like index/about.

The urls to these pages are all prefixed with /page/.

5.2 Setup

To create a custom page named info (<http://example.com/page/info/>), in the WebHtml admin area create a new entey for info.

In the info page put your HTML that makes up your page, then visit the url of your site.

Settings - List of all settings

This document will list all settings used by trunk player. The settings are read from the base settings file `trunk_player/settings.py` and also the local file you can override `trunk_player/settings_local.py`

These include settings from the base Django code. [Complete list from Django](#) These will be prefixed with a [D], Settings for 3rd party django apps will be prefixed with [D3]

6.1 SITE_TITLE

This value will be used for html page titles.

6.2 SECRET_KEY

[D] Django secret key. Used by the system for all encoding and encrypting of values. **Very important you change this to your own random value**

6.3 DEBUG

[D] When DEUG is True Djago will display debug messages for error on the web page. You do not want this set on an open site. This can expose sensitive data to users.

6.4 ALLOWED_HOSTS

[D] This is a list of host names the web page can be accessed in, you need to set this domain name(s) of your site.

6.5 DATABASES

[D] Configuration data for your database connection

6.6 SITE_EMAIL

Email address emails will come from

6.7 DEFAULT_FROM_EMAIL

Email address emails will come from

6.8 AUDIO_URL_BASE

URL where your audio files are accessible at

6.9 ANONYMOUS_TIME

Time in minutes that an anonymous (non logged in) can view into the past

6.10 TIME_ZONE

[D] Timezone settings

[List of timezones](#) You can set to your local timezone.

6.11 ACCESS_TG_RESTRICT

If set to true all talkgroups are restricted and each user needs to be explicitly granted permission in the talkgroupaccess table

6.12 GOOGLE_ANALYTICS_PROPERTY_ID

Google ID used for analytics

6.13 TWITTER_ACTIVE

If set to true it will display a twitter feed to the right TWITTER_LIST_URL the url for the feed

6.14 LOGIN_REDIRECT_URL

[D3] Where the user is redirected after they login

Note: The settings below this point will most likely not need to be edited

6.15 JS_SETTINGS

Used to allow the page javascript visibility into the settings in the list

6.16 JS_SETTINGS

Used to allow the django HTML template engine visibility into the settings in the list

6.17 ALLOW_ANONYMOUS

Not used for anything

6.18 RADIO_DEFAULT_UNIT_AGENCY

DB ID of the default agency when a new unit is added * This is defaulted to 0 which is invalid in mysql

6.19 AMAZON_ADDS

If set to true the site will display amazons ads on the right hand side

Additional settings for this are : AMAZON_AD_TRACKING_ID Your Amazon ID AMAZON_AD_LINK_ID ID for this site AMAZON_AD_EMPHASIZE_CATEGORIES Categories to display AMAZON_AD_FALL_BACK_SEARCH Search terms

6.20 SOCIALACCOUNT_PROVIDERS

[D3] Config settings to allow google authentication

6.21 ACCOUNT_AUTHENTICATION_METHOD

[D3] Default username setup

6.22 ACCOUNT_EMAIL_REQUIRED

[D3] Require email address

6.23 BASE_DIR

[D] Base directory the project is installed in.

6.24 LOGIN_URL

[D] The url django will redirect users who are not logged in when a page requires login.

6.25 INSTALLED_APPS

[D] List of the django apps that make up the project.

6.26 MIDDLEWARE_CLASSES

[D] List of middleware code that is run in the project

6.27 ROOT_URLCONF

[D] Where the primary URL list comes from

6.28 TEMPLATES

[D] Config values for the HTML template system

6.29 WSGI_APPLICATION

[D] Location of the config file for the primary web setup

6.30 AUTHENTICATION_BACKENDS

[D] How users are authenticated to django

6.31 AUTH_PASSWORD_VALIDATORS

[D] How password are validated

6.32 LANGUAGE_CODE

[D] Language used

6.33 USE_I18N

[D] Use Language code

6.34 USE_L10N

[D] localized formatting

6.35 USE_TZ

[D] Use localtime zones

6.36 SECURE_PROXY_SSL_HEADER

[D] Used to make sure HTTPS connections are set

6.37 STATIC_URL

[D] URL Prefix for static content

6.38 STATIC_ROOT

[D] Directory location of static files

6.39 REST_FRAMEWORK

[D] Config for the REST (API) framework

6.40 MEDIA_URL

[D] URL prefix for media data

6.41 MEDIA_URL

[D] Directory location of media files

6.42 CHANNEL_LAYERS

[D] Config of the channel app. This is where the websocket data is accessed from.

6.43 CACHES

[D] Config of the local cache

6.44 SITE_ID

[D] ID of this site

6.45 USE_RAW_ID_FIELDS

[D] For very large systems, the admin page may load better with raw id fields

Using supervisor to run the web services

7.1 Install Supervisor

```
$ sudo apt-get install supervisor
```

7.2 Update Config

Copy `trunk_player/supervisor.conf.sample` to `trunk_player/supervisor.conf`

Edit the `/home/radio/trunk-player` entries in `trunk_player/supervisor.conf` to match your install directory.

Also, Edit the `user=radio` entries in `trunk_player/supervisor.conf` to match your desired user. Keep in mind this user must have read/write permissions to your installation directory.

7.3 Enable the config

First stop and disable any already running `runserver`, `runworker` or `daphne` processes.

Link in the config and refresh supervisor.

```
$ sudo ln -s /home/radio/trunk-player/trunk_player/supervisor.conf /etc/supervisor/  
↪conf.d/trunk_player.conf  
$ sudo supervisorctl reread  
$ sudo supervisorctl update
```

7.4 Restarting trunk-player

If you make any code changes (git pull) you need to restart the web processes.

```
$ sudo supervisorctl restart trunkplayer:
```

8.1 Upgrade to Trunk Player 0.1.2

This document will show you how to upgrade trunk Player 0.0.1 to 0.1.2

NEW USERS DO NOT NEED TO DO THIS

Currently install is based on a Debian like system, Ubuntu.

8.1.1 System Prerequisites

- Linux Machine with current Trunk Player install
- Python >=3.8.x
- Virtualenv
- PIP Should be installed with Python 3
- Redis >=5.x.x
- PostgreSQL 9.x
- git

Upgrade your system

```
$ sudo apt update && sudo apt upgrade
```

8.1.2 Assumptions

- Project directory is /home/radio/trunk-player - You can replace /home/radio with your own home directory.
- Redis is running and listening on default port of : 6379

8.1.3 Upgrading Trunk Player

The source is on GitHub, use git to upgrade your repo. Starting from your home directory of `/home/radio/trunk-player`

```
$ cd /home/radio/trunk-player && git pull
```

This will pull down the most current version of Trunk Player.

8.1.3.1 Stopping Trunk Player

We need to fire the database to upgrade so stop trunk player. **BE SURE TO STOP TRUNK PLAYER**

If using supervisor do the following

```
$ sudo supervisorctl stop trunkplayer:
```

8.1.3.2 Remove Old Virtual Environment

Remove old env director to allow of clean package upgrades

```
$ rm -rf env/
```

8.1.3.3 Setup Virtual Environment

Setup a new Python 3.x virtual environment in the env directory. Set the visual prompt to (Trunk Player).

```
$ virtualenv -p python3 env --prompt='(Trunk Player)'
```

8.1.3.4 Activate Virtual Environment

You will need to re run this step each time you start a new shell or log into your machine.

```
$ source env/bin/activate
```

This will set you into a new python environment any packages you install via `pip` will only live in this area and do not touch your system files. This allowed you to have multiple projects with different dependencies.

You can use the command `deactivate` to exit back to your normal system environment.

8.1.3.5 Re-Install Python Packages

First Upgrade pip

```
(Trunk Player)$ pip install --upgrade pip
```

Using pip install all required packages from the `requirements.txt` file.

```
(Trunk Player)$ pip install -r requirements.txt
```


8.1.3.6 Backup Postgres Database

You need to create a postgres database backup incase you have migration issues

Log into your postgres database as an admin user and dump the database, assuiming database name `trunk_player`

```
$ sudo su - postgres
(postgres)$ pg_dump trunk_player > /tmp/tunk_player.bak.sql
(postgres)$ chmod 744 /tmp/tunk_player.bak.sql
(postgres)$ exit
$ cp /tmp/trunk_player.bak.sql ~/
```

Your backup is at `/home/radio/tunk_player.bak.sql`

8.1.3.7 Upgrade the database

Using the `django manage.py` command to upgrade the database.

```
(Trunk Player)$ ./manage.py migrate
```

8.1.3.8 Starting the test web server

First note this is not full production ready server. It can handle a couple users.

Using the `manage.py` command again

```
(Trunk Player)$ ./manage.py check
(Trunk Player)$ ./manage.py runserver
```

This will start the server up listening on the local loopback address on port 8000. Start your web browser and go to <http://localhost:8000>. You should seen the main page Visit `/admin/` to log into the admin area.

If you are running this on a remote server you need to have the web server us its' public IP adress so you can connect.

```
(Trunk Player)$ ./manage.py runserver 0.0.0.0:8000
```

This will run the server also on port 8000 but will be accessible via the servers IP address or dns name on port 8000 also.

8.1.3.9 Restarting Trunk Player

If using supervisor do the following

```
$ sudo supervisorctl start trunkplayer:
```

8.1.4 You Did It!

You have upgraded trunk player to 0.1.2! If you run into any issues open an issue at <https://github.com/ScanOC/trunk-player/issues>

8.1.5 Rollback Trunk-Player Install

8.1.5.1 Rollback Trunk-Player Code

Reset to Trunk-Player 0.0.1

```
(Trunk Player)$ git reset --hard cb22634f43a4e8ca233847a4d0e2864b3f7d1f2b
```

8.1.5.2 Rollback Postgres Database (Only if needing to rollback)

Log into your postgres database as an admin user and restore the database, assuming database name `trunk_player`

```
$ chmod 744 /home/radio/tunk_player.bak.sql
$ sudo su - postgres
(postgres)$ psql
```

Drop the Database; and recreate

```
postgres=# DROP DATABASE trunk_player;
postgres=# CREATE DATABASE trunk_player;
postgres=# \q;
```

Restore Database

```
(postgres)$ psql trunk_player < /home/radio/tunk_player.bak.sql
(postgres)$ exit
```

8.1.5.3 Remove Old Virtual Environment

Remove old env director to allow of clean package upgrades

```
$ rm -rf env/
```

8.1.5.4 Setup Virtual Environment

Setup a new Python 3.x virtual environment in the `env` direcorey. Set the visual prompt to `(Trunk Player)`.

```
$ virtualenv -p python3 env --prompt='(Trunk Player)'
```

8.1.5.5 Activate Virtual Environment

You will need to re run this step each time you start a new shell or log into your machine.

```
$ source env/bin/activate
```

This will set you into a new python environment any packages you install via `pip` will only live in this area and do not touch your system files. This allowed you to have multiple projects with different dependencies.

You can use the command `deactivate` to exit back to your normal system environment.

8.1.5.6 Re-Install Python Packages

First Upgrade pip

```
(Trunk Player)$ pip install --upgrade pip
```

Using pip install all required packages from the `requirements.txt` file.

```
(Trunk Player)$ pip install -r requirements.txt
```

8.1.5.7 Restarting Trunk Player

If using supervisor do the following

```
$ sudo supervisorctl start trunkplayer:
```

8.2 Upgrade to Trunk Player 0.1.3

MAKE SURE YOU ARE RUNNING 0.1.2 If you are not it will break Trunk Player

This document will show you how to upgrade trunk Player 0.1.2 to 0.1.3

NEW USERS DO NOT NEED TO DO THIS

Currently install is based on a Debian like system, Ubuntu.

8.2.1 System Prerequisites

- Linux Machine with current Trunk Player install
- Python $\geq 3.8.x$
- Virtualenv
- PIP Should be installed with Python 3
- Redis $\geq 5.x.x$
- PostgreSQL 9.x
- git

Upgrade your system

```
$ sudo apt update && sudo apt upgrade
```

8.2.2 Assumptions

- Project directory is `/home/radio/trunk-player` - You can replace `/home/radio` with your own home directory.
- Redis is running and listening on default port of `:6379`

8.2.3 Upgrading Trunk Player

The source is on GitHub, use git to upgrade your repo. Starting from your home directory of `/home/radio/trunk-player`

```
$ cd /home/radio/trunk-player && git pull
```

This will pull down the most current version of Trunk Player.

8.2.3.1 Stopping Trunk Player

We need to fire the database to upgrade so stop trunk player. **BE SURE TO STOP TRUNK PLAYER**

If using supervisor do the following

```
$ sudo supervisorctl stop trunkplayer:
```

8.2.3.2 Restarting Trunk Player

If using supervisor do the following

```
$ sudo supervisorctl start trunkplayer:
```

8.2.4 You Did It!

You have upgraded trunk player to 0.1.3! If you run into any issues open an issue at <https://github.com/ScanOC/trunk-player/issues>

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`