

---

# **MPD Album Art Documentation**

***Release 0.1***

**Jamie Macdonald**

**Oct 06, 2017**



---

## Contents

---

|          |                             |          |
|----------|-----------------------------|----------|
| <b>1</b> | <b>mpd_album_art Module</b> | <b>3</b> |
| <b>2</b> | <b>Indices and tables</b>   | <b>5</b> |
|          | <b>Python Module Index</b>  | <b>7</b> |



Contents:



# CHAPTER 1

---

## mpd\_album\_art Module

---

**copyright** 2013 Jamie Macdonald

**license** GNU General Public License (GPL) version 3

**class** `mpd_album_art.Grabber` (*save\_dir*, *library\_dir=None*, *link\_path=None*)  
Bases: `object`

An object to grab artwork for MPD songs

### Parameters

- **save\_dir** (*str*) – Directory into which Grabber should download new images.
- **library\_dir** (*str*) – Directory MPD is currently playing from.
- **link\_path** (*str*) – Path to symlink to current image file. By default, `os.path.join(save_dir, "current")`

Construct a Grabber with `save_dir = ~/.covers`:

```
>>> import os
>>> home_dir = os.environ['HOME']
>>> cover_path = os.path.join(home_dir, '.covers')
>>> grabber = Grabber(cover_path)
```

specify more stuff:

```
>>> music_path = os.path.join(home_dir, 'Music/Library')
>>> current_cover_link = os.path.join(home_dir, '.current_cover')
>>> grabber = Grabber(save_dir=cover_path, library_dir=music_path,
...                   link_path=current_cover_link)
```

**get\_art** (*song*)

Get artwork from LastFM.

Before connecting to network, if there exists a file whose name begins with `_sanitize(song['artist'] + ' ' + song['album'])`, return that file path. Do not overwrite existing files. Set `os.path.join(save_dir, 'current')` as symlink to image file.

**Parameters** `song` (*dict*) – A dictionary with keys 'album' and 'artist' to correspond with string representations of the album and artist (resp.) of interest. Use `MPDClient.currentsong()` to return such a dictionary.

**Returns** A string representation of the local file path to the image file for `song` or `None` if no results found

**get\_local\_art** (*song*)

Get artwork from `song` folder.

**Parameters** `song` (*dict*) – A dictionary with keys 'album' and 'artist' to correspond with string representations of the album and artist (resp.) of interest. Such a dictionary returns from `MPDClient.currentsong()`

**Returns** A string representation of the local file path to the largest image file for `song` found in `song_folder`, or `None` if no results found

**remove\_current\_link** ()

Delete current artwork symlink.

**set\_current\_link** (*file\_path*)

Set current artwork symlink. :param `file_path`:

Path to artwork to be set as current.



## CHAPTER 2

---

### Indices and tables

---

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



**m**

`mpd_album_art`, 3



## G

`get_art()` (`mpd_album_art.Grabber` method), 3  
`get_local_art()` (`mpd_album_art.Grabber` method), 4  
`Grabber` (class in `mpd_album_art`), 3

## M

`mpd_album_art` (module), 3

## R

`remove_current_link()` (`mpd_album_art.Grabber`  
method), 4

## S

`set_current_link()` (`mpd_album_art.Grabber` method), 4