
abandi Documentation

Release 0.0.4

ponty

August 01, 2011

CONTENTS

1	Basic usage	2
2	Installation	3
2.1	General	3
2.2	Ubuntu	3
2.3	Uninstall	3
3	Dependencies	4
3.1	basic	4
3.2	html parsers	4
3.3	supported emulators	4
4	game sources	5
5	Usage	6
5.1	update	6
5.2	install	6
5.3	check database	6
5.4	search database	7
5.5	run game	7
6	command line help	9
6.1	dbdownload	9
6.2	parse	9
6.3	update	9
6.4	info	10
6.5	search	10
6.6	install	10
6.7	run	11
6.8	srun	11
7	system information	12
7.1	versions	12
7.2	plugins	12
8	related projects	13
9	Development	14
9.1	Tools	14
9.2	Install on ubuntu	14

9.3	Tasks	15
10	Indices and tables	16

abandi

Date August 01, 2011

PDF [abandi.pdf](#)

Contents:

Console-based abandonware game installer and runner. Games are downloaded over internet.

source: <https://github.com/ponty/abandi>

documentation: <http://ponty.github.com/abandi>

BASIC USAGE

The selected game will be downloaded, unpacked and started by an emulator.

if you know the id:

```
$ python -m abandi.run --auto-install gb64 3021
```

if you have an up-to-date database:

```
$ python -m abandi.srun --auto-install --name galaga
```

Note: It is only a wrapper,so you have to install unpackers (7zip, unrar,..) and emulators (dosbox, scummvm, stella, vice,..)

INSTALLATION

2.1 General

- install `setuptools`
- install backends for `pyunpack` (optional)
- install supported emulators (optional)
- install the program:

```
# as root
easy_install https://github.com/ponty/abandi/zipball/master
```

2.2 Ubuntu

```
sudo apt-get install python-setuptools
sudo easy_install https://github.com/ponty/abandi/zipball/master
# optional
sudo easy_install http://sourceforge.net/projects/patool/files/0.13/patool-0.13.tar.gz/download
sudo apt-get install unzip unrar p7zip-full
sudo apt-get install dosbox openmsx scummvm stella vice
```

2.3 Uninstall

first install `pip`:

```
# as root
pip uninstall abandi

# database, games
rm -r ~/.abandi
```

DEPENDENCIES

3.1 basic

python		
yapsy		plugin system

3.2 html parsers

lxml	optional	html parser
BeautifulSoup	optional	html parser

3.3 supported emulators

id	name	platforms	url
dosbox	DOSBox	[dos]	http://www.dosbox.com/
openmsx	openMSX	[msx]	http://openmsx.sourceforge.net/
scummvm	ScummVM	[dos, c64]	http://www.scummvm.org/
stella	Stella	[atari2k6]	http://stella.sourceforge.net/
vice	VICE	[c64]	http://www.viceteam.org/

GAME SOURCES

id	name	platforms	url	max id
aban-doneer	Abandoneer	[dos]	http://www.abandoneer.com/	150
aban-donia	Abandonia	[dos, win]	http://www.abandonia.com/	1000
gb64	Gamebase 64	[c64]	http://www.gamebase64.com/	22000
os-comp	Oldschool Computer	[dos, c64, c128, cplus4, msx, megadrive, mastersystem, gb, atari2k6, nes, snes, amiga]	http://oscomp.hu/	1230
osd	Old School DOS	[dos]	http://www.oldschooldos.com/	1021

USAGE

5.1 update

update some games in database from [gamebase64](#)

```
$ python -m abandi.update gb64 3020-3023 --force
force updating gb64/3020... found:"Galaga"... OK
force updating gb64/3021... sleeping 2 sec... found:"Galaga"... OK
force updating gb64/3022... sleeping 2 sec... found:"Galaxia 7"... OK
force updating gb64/3023... sleeping 2 sec... found:"Galaxian"... OK
```

or download the default game database:

```
$ python -m abandi.dbdownload
```

5.2 install

install [galaga](#)

```
$ python -m abandi.install gb64 3020
downloading Galaga... OK
unpacking Galaga... OK
```

5.3 check database

```
$ python -m abandi.info gb64 3020
source:          gb64
id:              3020
name:            Galaga
platform:        c64
game_file_url:    http://gamebase64.hardabasht.com/games/g/GALAGA1_03020_02.zip
release_year:     None
genre:           Shoot'em Up - Space Invaders
programmer:       Henrik Wening
language:         English
musician:         None
publisher:        Henrik Wening
home_url:         http://www.gamebase64.com/game.php?id=3020
music_file_url:   None
```

```
screenshot_url_list: http://www.gb64.com/Screenshots/G/Galaga_v1.png|http://www.gb64.com/Screenshots
zip:                /home/titi/.abandi/cache/gamezip/gb64/http___gamebase64.hardabasht.com_games_g_
dir:                /home/titi/.abandi/games/gb64/Galaga.3020
```

5.4 search database

```
$ python -m abandi.search -n galaga
[gb64 3020 c64] Galaga
[gb64 3021 c64] Galaga
```

5.5 run game

run `galaga` by id:

```
$ python -m abandi.run -a gb64 3020
```



run `galaga` by name:

```
$ python -m abandi.srun -a --name galaga
```



run maniac mansion using vice:

```
$ python -m abandi.run -a gb64 4577 --runner vice
```



run maniac mansion using scummvm:

```
$ python -m abandi.run -a gb64 4577 --runner scummvm
```



COMMAND LINE HELP

6.1 dbdownload

```
$ python -m abandi.dbdownload --help
usage: dbdownload.py [-h] [-u URL] [--debug]
```

downloads and unpacks game database

optional arguments:

-h, --help	show this help message and exit
-u URL, --url URL	packed sqlite file
--debug	set logging level to DEBUG

6.2 parse

```
$ python -m abandi.parse --help
usage: parse.py [-h] [--debug] source id
```

parse game on source and print all information

positional arguments:

source	['gb64',...]
id	['1',...]

optional arguments:

-h, --help	show this help message and exit
--debug	set logging level to DEBUG

6.3 update

```
$ python -m abandi.update --help
usage: update.py [-h] [-f] [--debug] source id
```

parse and update game in database

positional arguments:

source	['all','gb64',...]
id	['all','1','1-5','1,5-8,10',...]

```
optional arguments:
  -h, --help      show this help message and exit
  -f, --force
  --debug         set logging level to DEBUG
```

6.4 info

```
$ python -m abandi.info --help
usage: info.py [-h] [--debug] source id

print all information for game in database

positional arguments:
  source      ['gb64',...]
  id          ['1',...]
```

```
optional arguments:
  -h, --help      show this help message and exit
  --debug         set logging level to DEBUG
```

6.5 search

```
$ python -m abandi.search --help
usage: search.py [-h] [-c COL_FORMAT] [-w WHERE] [-o ORDERBY] [-n NAME]
                [-p PLATFORM] [-s SOURCE] [-r RUNNER] [--debug]

search in game database
```

```
optional arguments:
  -h, --help            show this help message and exit
  -c COL_FORMAT, --col-format COL_FORMAT
  -w WHERE, --where WHERE
                        SQL where, e.g. "id>5 and name like falcon"
  -o ORDERBY, --orderby ORDERBY
  -n NAME, --name NAME  game name like this
  -p PLATFORM, --platform PLATFORM
                        check lsplatform for list
  -s SOURCE, --source SOURCE
                        check lsplugin for list
  -r RUNNER, --runner RUNNER
  --debug               set logging level to DEBUG
```

6.6 install

```
$ python -m abandi.install --help
usage: install.py [-h] [-d] [-u] [-r] [-n] [--debug] source id

download and unpack game found in database

positional arguments:
```

```

source          ['gb64',...]
id              ['1',...]

```

optional arguments:

```

-h, --help          show this help message and exit
-d, --downloadonly
-u, --unpackonly
-r, --removezip
-n, --nocache
--debug            set logging level to DEBUG

```

6.7 run

```
$ python -m abandi.run --help
```

```
usage: run.py [-h] [-r RUNNER] [-a] [--debug] source id
```

start game using selected emulator

positional arguments:

```

source          ['gb64',...]
id              ['1',...]

```

optional arguments:

```

-h, --help          show this help message and exit
-r RUNNER, --runner RUNNER
                    emulator ['auto','dosbox','scummvm',...]
-a, --auto-install
--debug            set logging level to DEBUG

```

6.8 srunch

```
$ python -m abandi.srun --help
```

```
usage: srunch.py [-h] [-c COL_FORMAT] [-w WHERE] [-o ORDERBY] [-n NAME]
                [-p PLATFORM] [-s SOURCE] [-r RUNNER] [-a] [-i INDEX] [--debug]
```

search and run

optional arguments:

```

-h, --help          show this help message and exit
-c COL_FORMAT, --col-format COL_FORMAT
-w WHERE, --where WHERE
-o ORDERBY, --orderby ORDERBY
-n NAME, --name NAME
-p PLATFORM, --platform PLATFORM
-s SOURCE, --source SOURCE
-r RUNNER, --runner RUNNER
-a, --auto-install
-i INDEX, --index INDEX
--debug            set logging level to DEBUG

```

SYSTEM INFORMATION

7.1 versions

```
$ python -m abandi.lsversion
python          2.6.6
abandi          0.0.4
dosbox          0.74
openmsx         0.8.0
scummvm         1.1.1
soup            3.1.0.1
stella          3.1.2
vice            unknown
```

7.2 plugins

```
$ python -m abandi.lsplugin
abandoneer      game_source
abandonia       game_source
cache           downloader
curl            downloader
dosbox          runner
gb64            game_source
lxml            html_parser
openmsx         runner
oscomp          game_source
osd             game_source
scummvm         runner
soup            html_parser
stella          runner
urllib          downloader
vice            runner
```

RELATED PROJECTS

- GameBase: Universal Emulator Frontend (<http://www.bu22.com/>)
- jGameBase: Java port of GameBase (<http://sourceforge.net/projects/jgamebase/>)
- Kamefu: the emulator frontend and collection manager for KDE
- Frontends for DOSBox : (<http://www.dosbox.com/wiki/DOSBoxFrontends>)

DEVELOPMENT

9.1 Tools

1. `setuptools`
2. `Paver`
3. `nose`
4. `ghp-import`
5. `pyflakes`
6. `pychecker`
7. `paved fork`
8. `Sphinx`
9. `sphinxcontrib-programsscreenshot`
10. `sphinxcontrib-paverutils`
11. `autorun` from `sphinx-contrib` (there is no simple method, you have to download/unpack/setup)

9.2 Install on ubuntu

```
sudo apt-get install python-setuptools
sudo apt-get install python-paver
sudo apt-get install python-nose
sudo easy_install ghp-import
sudo apt-get install pyflakes
sudo apt-get install pychecker
sudo easy_install https://github.com/ponty/paved/zipball/master
sudo apt-get install scrot
sudo apt-get install xvfb
sudo apt-get install xserver-xephyr
sudo apt-get install python-imaging
sudo apt-get install python-sphinx
sudo easy_install sphinxcontrib-programsscreenshot
sudo easy_install sphinxcontrib-programoutput
sudo easy_install sphinxcontrib-paverutils
```

9.3 Tasks

[Paver](#) is used for task management, settings are saved in `pavement.py`. [Sphinx](#) is used to generate documentation.

print [paver](#) settings:

```
paver printoptions
```

clean generated files:

```
paver clean
```

generate documentation under *docs/_build/html*:

```
paver cog pdf html
```

upload documentation to [github](#):

```
paver ghpages
```

run unit tests:

```
paver nose
#or
nosetests --verbose
```

check python code:

```
paver pyflakes
paver pychecker
```

generate python distribution:

```
paver sdist
```

upload python distribution to [PyPI](#):

```
paver upload
```

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*