htpc-launcher Documentation

Release 1.1.0

Tim Court

November 24, 2015

Contents

HTPC Launcher is a program for launching full-screen applications. It is designed for use in an environment where only one of several resource-intensive applications should be run at once. HTPC Launcher responds to LIRC [1] key presses by starting a configured application and stopping any others that may still be running.

For example, if HTPC Launcher is configured for XBMC and Steam, then pressing the configured LIRC key for XBMC will launch that program. Next, if the button for Steam is pressed, HTPC Launcher will shut down XBMC and launch Steam.

[1] Linux Infrared Remote Control (http://lirc.org/) is used to read IR signals from your remote control.

CHAPTER 1

Configuration

HTPC Launcher requires configuration before use. It looks first for a file called . htpc-launcher.conf in the user's home directory, then for /etc/htpc-launcher.conf. If none of those can be found the program exits immediately.

The configuration file must be formatted according to the rules for the python Config Parser module. It consists of sections denoted by headers in square brackets (`[]`) which contain a series of name/value pairs for each configuration item.

For example:

```
[startup]
launch = KEY_YELLOW

[KEY_YELLOW]
process = xbmc
search = xbmc.bin

[KEY_BLUE]
process = steam
search = .local/share/Steam/.+/steam$
needsKill = true
```

These are the available sections:

- **startup** – global configuration for HTPC Launcher
- **log** – options for the log file
- **key** – options for a single application bound to an LIRC key code

1.1 Global Configuration

The following options are recognized in the `startup` section.

launch The name of an LIRC key code to activate when HTPC Launcher starts. This must match one of the configured key sections.

1.2 Log Configuration

These options control how the HTPC Launcher log file behaves.

path The path to the application log file, including file name. If not given, the log file will be created in the user's home directory and will be called . htpc-launcher.log.

1.3 Key Configuration

Key sections describe an application that HTPC Launcher will be responsible for running. The section name must match an LIRC key code (see *Find LIRC Key Codes* for help finding the key codes for your remote). Within each section the following configuration items are recognized.

process The process to run when this LIRC key is received.

search When HTPC Launcher needs to stop an application it searches for the appropriate process using this perl-compatible regular expression.

needsKill Set this option to true if the application does not shutdown correctly in response to a SIGTERM signal. This will cause HTPC Launcher to stop the application by sending it SIGKILL instead.

1.4 Tips For Creating the Configuration

1.4.1 Find LIRC Key Codes

Run the **irw** command to see the key names as interpretted by LIRC.

For example, this is output of `irw` after pressing the "1" key on a typical remote control:

```
$ irw
000000037ff07bfe 00 KEY_1 mceusb
000000037ff07bfe 01 KEY_1 mceusb
```

CHAPTER 2

Contributing

2.1 Getting the Source

Get the source for HTPC Launcher from this GitHub project: https://github.com/tctimmeh/htpc-launcher

```
git clone https://github.com/tctimmeh/htpc-launcher.git
```

2.2 Setting Up a Development Environment

Use a python virtualenv to work on HTPC Launcher. The included makefile will create an appropriate virtual environment with all of the necessary third-party libraries installed. Run the following command to create the virtual environment:

```
make venv
```

The virtual environment must then be activated before running tests, or building documentation or installation packages. Activate the virtual environment with by running this command from the project root directory:

```
. venv/bin/activate
```

For reference, the following third-party libraries are required to test and/or build HTPC Launcher:

- pytest
- mock
- sphinx

2.3 Executing the Tests

Run the HTPC Launcher tests by executing this command from the project root directory:

```
make test
```

2.4 Code Coverage

Testing coverage reports are created automatically using the python Coverage module. A coverage report is shown when the tests are run. Run `coverage html` after running the tests to generate the report as HTML.

2.5 Continuous Integration

This project is built and tested automatically by Travis CI after every commit to the main repository. Find the latest build here: https://travis-ci.org/tctimmeh/htpc-launcher. See the Travis CI documentation for information about how to configure the build: http://about.travis-ci.org/docs/.

Travis CI also automatically publishes code coverage reports to Coveralls. Find the coveralls page for this project here: https://coveralls.io/r/tctimmeh/htpc-launcher.

2.6 Building Installation Package

The HTPC Launcher source code includes a python `setup.py` file to create a python installation package. The Distribute python packaging library provides some additional functionality to the standard setup.py file. Here are some common operations that can performed using the setup.py file.

Build a source distibution for testing **`./setup.py sdist`**

Install the Development Working Copy to the Python Virtual Environment **`./setup.py develop`**

Upload a New Version to the Python Package Index **`./setup.py sdist register upload`**

or

`make upload`

2.7 Building This Documentation

Build this documentation by running the following command from the project root directory:

`make doc`

To build only the HTML or man page, run **`make html`** or **`make man`** respectively from the `docs` directory.

2.8 Making a New Release

Follow these steps to release a new version:

1. Increment the product version number

- Increment the release number if only bug fixes were made, the minor number if new features were added, or the major number if changes have broken backwards compatibility.

2. Tag the code with the new version number
3. Update the release notes by changing the 1.x label to the new version number. Create a new 1.x label.
4. Upload the new version to the Python Package Index by running **`make upload`**
5. If any documentation was changed since the previous release, move the `doc-latest` branch to match the latest tagged release

CHAPTER 3

Project History

v1.x - Coming Soon

- New: Searches for config file in /etc if not found in user’s home directory
- New: Configure location of log file

v1.0.1 - 2013-01-27

- Fix: lirc key repeats greater than 9 handled gracefully

v1.0.0 - 2013-01-26

- Initial Release

CHAPTER 4

Installation

Install HTPC Launcher using **`pip`**:

```
$ pip install htpc-launcher
```

Usage

Configure HTPC Launcher by creating a configuration file as described in Configuration. HTPC Launcher will not run without a configuration file.

Run HTPC Launcher by executing:

```
$ htpc-launcher
```

HTPC Launcher will respond to LIRC codes by starting and stopping your configured applications.

Inspect the run-time log file to diagnose any problems. The log file is located in htpc-launcher.log.