

---

# **PyParadox Documentation**

***Release 0.3.10***

**Carmen Bianca Bakker**

November 17, 2016



<b>1</b>	<b>PyParadox</b>	<b>3</b>
1.1	Requirements . . . . .	3
1.2	Features . . . . .	3
1.3	Installation . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Arch Linux . . . . .	5
2.2	openSUSE and Fedora . . . . .	5
2.3	Linux . . . . .	5
2.4	OS X . . . . .	5
2.5	Windows . . . . .	6
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	10
<b>5</b>	<b>Credits</b>	<b>11</b>
5.1	Development Lead . . . . .	11
5.2	Contributors . . . . .	11
<b>6</b>	<b>History</b>	<b>13</b>
6.1	0.3.10 (2016-11-17) . . . . .	13
6.2	0.3.9 (2016-05-13) . . . . .	13
6.3	0.3.8 (2016-05-12) . . . . .	13
6.4	0.3.7 (2016-02-13) . . . . .	13
6.5	0.3.6 (2016-01-29) . . . . .	13
6.6	0.3.5 (2015-08-17) . . . . .	13
6.7	0.3.4 (2015-08-09) . . . . .	14
6.8	0.3.3 (2015-07-23) . . . . .	14
6.9	0.3.2 (2015-07-18) . . . . .	14
6.10	0.3.1 (2015-04-23) . . . . .	14
6.11	0.3.0 (2015-04-23) . . . . .	14
6.12	0.2.3 (2015-04-17) . . . . .	14
6.13	0.2.2 (2015-04-12) . . . . .	14
6.14	0.2.1 (2015-04-11) . . . . .	15
6.15	0.2.0 (2015-04-11) . . . . .	15

6.16	0.1.3 (2015-01-31)	15
6.17	0.1.2 (2014-07-20)	15
6.18	0.1.1 (2014-07-20)	15
6.19	0.1.0 (2014-07-20)	15
6.20	0.0.2 (2014-07-13)	16
6.21	0.0.1 (2014-07-10)	16
<b>7</b>	<b>Indices and tables</b>	<b>17</b>

Contents:



---

# **PyParadox**

---

PyParadox is a nix launcher for Paradox titles.

- Free software: GPLv3+
- Documentation: <https://pyparadox.readthedocs.org>
- Source code: <https://gitlab.com/carmenbbakker/pyparadox>
- PyPI: <https://pypi.python.org/pypi/pyparadox>

## **1.1 Requirements**

- Operating system:
  - Linux: Any modern distribution with up-to-date packages
  - OS X: Latest version
  - Windows: 7 and later
- Python: 3.3+
- Qt: Qt 5.3+ via PyQt5

## **1.2 Features**

- Games: PyParadox supports all cross-platform grand strategy titles of Paradox Interactive that use the Clausewitz engine. To date, these include:
  - Crusader Kings II
  - Europa Universalis IV
  - Stellaris
- Steam: PyParadox can be launched from Steam when starting the title from the Steam interface. All Steam integration should work just fine. A small configuration inside Steam is required to make sure that PyParadox is used instead of the stock launcher. See the documentatation on how to set this up.
- PyParadox separates the expansion DLCs from the regular DLCs. This should make it easier to start up your game with the exact expansions you want.

- PyParadox fixes an anomaly under Linux that causes the game to mess with the screen resolution when alt+tabbing. This anomaly was introduced when Paradox released their (new) stock launcher for Linux. By bypassing this launcher, PyParadox also bypasses the anomaly.

## 1.3 Installation

[Read the documentation.](#)



---

## Installation

---

PyParadox requires Python 3, pip and PyQt5. The ways of obtaining these three pieces of software differ per platform.

### 2.1 Arch Linux

PyParadox is [available in the AUR](#).

### 2.2 openSUSE and Fedora

PyParadox is [available in the Open Build Service](#).

### 2.3 Linux

You need your distribution's pip 3, PyQt 5 and QML packages installed.

For Ubuntu, that would look as follows:

```
sudo apt-get install python3-pip python3-pyqt5 python3-pyqt5.qtquick qml-module-qtquick-controls qml-
```

Now that you've satisfied the dependencies of PyParadox, you can download and install it. This should be as easy as:

```
sudo pip3 install pyparadox
```

If PyParadox installed correctly, the [Usage](#) section should be able to help you next.

### 2.4 OS X

OS X installation is a little tricky, because Apple provides its own version of Python that should be very much left alone. You do not want to mess around with this version of Python, ever. Never mind the fact that OS X's version of Python is too outdated to run PyParadox.

The best way to install PyParadox under OS X is to make a separate Python installation from Apple's version. An easy way to do this is to use Homebrew. If you have touched Homebrew before or have messed around with installations of Python before, proceed with extra caution.

To install Homebrew, open your terminal, copy and paste the following command and hit return:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

After you have installed Homebrew, you can use Homebrew to install Python 3 and PyQt5:

```
brew install python3
brew install pyqt5 --with-python3
```

You now need pip to be able to install PyParadox. Unfortunately, the Homebrew version of Python does not come with pip. It does, however, come with a tool to install pip:

```
easy_install-3.4 pip
```

Finally, install PyParadox using pip:

```
pip3.4 install pyparadox
```

If PyParadox installed correctly, the [Usage](#) section should be able to help you next.

## 2.5 Windows

This should be about as easy as the Linux instructions. Simply download and install Python and PyQt5. Python can be downloaded from [the official Python downloads page](#) and PyQt5 can be downloaded from the [Riverbank Computing downloads page](#).

After these two components are installed, open up a command line interface. In Windows, this is more commonly referred to as `cmd.exe`. There are various ways of accessing this program, but by far the most convenient ways to get there are:

- Right-clicking on your start menu button and clicking `Command Prompt`.
- Searching through your programs from `cmd.exe`.
  - Under Windows 8 this means typing `cmd.exe` on the Modern UI start screen, and watching the results pop up on the right side of the screen.
  - Under Windows 7, this means opening up your start menu and typing `cmd.exe` into the search box.

After the console is opened, type the following line into the console and hit enter:

```
pip install pyparadox
```

After this, you should be all set, and [Usage](#) should be able to help you out further.

---

## Usage

---

To use PyParadox from the command line, type one of the following commands:

- `pyparadox {ck2|eu4|stellaris}` (console)
- `pyparadox-qt {ck2|eu4|stellaris}` (QtWidgets GUI, recommended stable)
- `pyparadox-qml {ck2|eu4|stellaris}` (QML GUI, experimental)

Ideally, however, PyParadox is launched from Steam. To do this, right click on your game in Steam, click Properties, then click Set launch options... in the General tab. You want to enter the following value in the case of Crusader Kings II:

```
pyparadox-qt ck2 %command%
```

This will (hopefully) start up PyParadox when you start your game from Steam.

PyParadox comes with some default settings depending on your platform. In case PyParadox is not configured directly, change the paths to your game executable and mod path in the Configure dialog.

---

**Important:** A known issue is that PyParadox can be launched from the command line, but not from within Steam. This is because the Qt system libraries that PyParadox uses are incompatible with the ancient libstdc++ library in the Steam runtime.

Removing the libstdc++ library fixes this:

```
find ~/.steam/root/ -name "libstdc++.so*" -print -delete
```

The following command also removes other libraries that might cause issues in other games:

```
find ~/.steam/root/ \( -name "libgcc_s.so*" -o -name "libstdc++.so*" -o -name "libxcb.so*" \) -print
```

---

**Note:** Another known issue is that PyParadox might look distorted when launched from Steam. This does not affect the functionality of the program or the performance of the game. There is no known fix other than disabling the Steam runtime altogether. This can be done in Arch Linux by installing the `steam-native` package from the AUR.

An unfortunate side effect is that the game will then fail to launch.

---



---

## Contributing

---

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 4.1 Types of Contributions

#### 4.1.1 Report Bugs

Report bugs at <https://gitlab.com/carmenbbakker/pyparadox/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

#### 4.1.2 Fix Issues

Look through the GitLab issues for bugs. Any open issues are open for anybody to contribute to.

#### 4.1.3 Write Documentation

PyParadox could always use more documentation, primarily in the official docs and in the docstrings.

#### 4.1.4 Submit Feedback

The best way to send feedback is to file an issue at <https://gitlab.com/carmenbbakker/pyparadox/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *pyparadox* for local development.

1. Fork the *pyparadox* repo on GitLab.
2. Clone your fork locally:

```
git clone git@gitlab.com:your_name_here/pyparadox.git
cd pyparadox/
```

3. Install PyParadox and all its development dependencies into a virtualenv. The below commands assume you have *virtualenvwrapper* installed. PyParadox will be installed in-place, and the development dependencies will be installed into your virtualenv's *site-packages*. Also link your local installation of PyQt5 into your virtualenv:

```
mkvirtualenv pyparadox
pip install -r requirements/dev.txt
make link-pyqt
```

4. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, verify that your changes still pass all tests:

```
tox
```

6. Commit your changes and push your branch to GitLab:

```
git add your_files
git commit -m "Your detailed description of your changes."
git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitLab website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in *README.rst*.

---

**Credits**

---

## 5.1 Development Lead

- Carmen Bianca Bakker <[c.b.bakker@carmenbianca.eu](mailto:c.b.bakker@carmenbianca.eu)>

## 5.2 Contributors

- Robin C. Thomas <[rc.thomas90@gmail.com](mailto:rc.thomas90@gmail.com)> - Code snippets from ck2launcher





---

## History

---

### 6.1 0.3.10 (2016-11-17)

- Added The Reaper's Due and Rights of Man.

### 6.2 0.3.9 (2016-05-13)

- Added Mare Nostrum EU4 expansion.

### 6.3 0.3.8 (2016-05-12)

- Added Stellaris.

### 6.4 0.3.7 (2016-02-13)

- Added Conclave and The Cossacks to expansions.
- Added testing for gui module.

### 6.5 0.3.6 (2016-01-29)

- Improved logging.
- Improved main algorithm.
- Raise error on Python < 3.3 installation attempts.
- Improved documentation.
- Added COPYING file. Apparently this was lacking all along.

### 6.6 0.3.5 (2015-08-17)

- Added basic logging.

## 6.7 0.3.4 (2015-08-09)

- Moved Linux default game directory from `~/.local/share/Steam/...` to `~/.steam/root/...`. The former directory was deprecated by Ubuntu, but not by Arch. The latter directory works on both distributions.
- Fixed OS X compatibility. Now launches the game correctly.

## 6.8 0.3.3 (2015-07-23)

- Fixed bug that caused `pyparadox-qml` to fail to launch. This presumably only happens in more recent versions of Qt/PyQt5.

## 6.9 0.3.2 (2015-07-18)

- Added new expansions to the list of expansions.

## 6.10 0.3.1 (2015-04-23)

- Scrollbars in the QML front-end are now always visible.
- QML front-end now uses the absolute minimum imports that I could find. This has resulted in the use of Qt 5.3 rather than Qt 5.2. It is still possible to use Qt 5.1 with the QtWidgets version.

## 6.11 0.3.0 (2015-04-23)

- Implemented QML front-end that is almost identical to the QtWidgets front-end. This is a proof of concept to indicate that the backend is fairly decoupled. There *is* some code repetition, but almost no repetition in business logic. Then again, the business logic of this program is fairly small.

## 6.12 0.2.3 (2015-04-17)

- Added error message when program fails to launch.
- Greatly improved documentation.
- Fixed bug that caused a failed launch on Ubuntu 14.04. Incidentally, this also causes the menu bar to be used locally rather than integrated into the top panel bar. Fortunately, PyParadox doesn't *have* a menu bar. Yet.

## 6.13 0.2.2 (2015-04-12)

- Patched a bug that caused PyParadox to fail to launch from Steam.
- Removed Python 2.7 support from PyPI.

## 6.14 0.2.1 (2015-04-11)

- Minor patches to documentation because apparently I can't release anything without messing that stuff up.

## 6.15 0.2.0 (2015-04-11)

- Complete rewrite of the project, with reuse of some code.
- Now no longer supports Python 2.7 and Qt 4. This program will only work with Python 3.3+ and PyQt5.
- Configuration files relocated to a different location. Old configuration files no longer valid.
- Graphical user interface makes use of Qt .ui files, rather than hardcoded QtGui/QtWidgets code.
- A lot of functionality has been decoupled from the user interface. While the UI code performs *some* glue logic, it should be relatively trivial to cook up a user interface in a different framework. An experimental QML front-end is under consideration.
- Unit tests are a lot leaner, using fewer mocks and patches, instead relying much more on dependency injection.
- Documentation rewritten.

## 6.16 0.1.3 (2015-01-31)

- Added `-pyqt4` argument to force the usage of PyQt4.
- Fixed PyQt4 compatibility issues.

## 6.17 0.1.2 (2014-07-20)

- Fixed typo that caused README to display incorrectly.

## 6.18 0.1.1 (2014-07-20)

- Sweetened up the README with images of the program.

## 6.19 0.1.0 (2014-07-20)

- Added descriptive error message when game fails to run.
- Added experimental PyQt4 support.
- Detailed installation instructions per platform added.
- A logo of each game is now displayed.
- Mods and DLCs are now sorted.
- The application now has an icon.

## 6.20 0.0.2 (2014-07-13)

- Windows compatibility added.
- Window titles set correctly.
- Small UI tweaks (alt-shortcuts).
- Better game process management.
- Better unit testing, though incomplete.
- Just general code refactoring.
- Tested on Kubuntu 14.04 and Windows 8.1. Still not sure about OS X.

## 6.21 0.0.1 (2014-07-10)

- First release on PyPI.
- Basic functionality. No polish yet.
- Only tested on Kubuntu 14.04.

---

## Indices and tables

---

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