

---

**upref**

***Release 1.0.13***

**Mar 19, 2020**



---

## Contents:

---

<b>1</b>	<b>upref</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Basic Usage . . . . .	1
1.3	Code example . . . . .	2
1.4	Features . . . . .	2
1.5	License . . . . .	2
<b>2</b>	<b>MIT License</b>	<b>3</b>
<b>3</b>	<b>Examples</b>	<b>5</b>
3.1	Easy first Examples . . . . .	5
<b>4</b>	<b>Library API</b>	<b>7</b>
4.1	Class Hierarchy . . . . .	7
4.2	File Hierarchy . . . . .	7
4.3	Full API . . . . .	7
<b>5</b>	<b>Class Hierarchy</b>	<b>23</b>
<b>6</b>	<b>File Hierarchy</b>	<b>25</b>
<b>7</b>	<b>Full API</b>	<b>27</b>
7.1	Namespaces . . . . .	27
7.2	Classes and Structs . . . . .	27
7.3	Functions . . . . .	27
7.4	Variables . . . . .	27
<b>8</b>	<b>Indices and tables</b>	<b>29</b>
	<b>Index</b>	<b>31</b>



# CHAPTER 1

---

## upref

---

Upref is a small python module to grab and save some user data like login, password, url (mainly string data). After some projects, I want a simple method to find the right login and url from the user and save it in the personal data of the user.

## 1.1 Installation

Installing upref with pip:

```
$ pip install upref
```

## 1.2 Basic Usage

Here is the first example. The code fetch the URL from the user parameter

```
import upref

user_data = upref.get_pref(
    {"url": {"label": "The application url"}},
    "example01")

print("URL is {}".format(user_data['url']))
```

This code create a file in the user folder. And this file will be read again during the next run.

```
url:
  label: The application url
  value: http://www.test.org/
```

## 1.3 Code example

```
import upref
data = upref.get_pref(
    {
        'login': {'label': "Your login"},
        'passwd': {'label': "Your secret password",
                  'type': "Password"},
    },
    name="example02")
print("Your login is %s" % data['login'])
print("Your password (not so secret) is %s" % data['passwd'])
```

At the first execution, there is a window to grab the data. The second execution, there will be no windows, the data are only read from the file.

All data are saved in a yaml file at `%APPDATA%/.upref/example02.conf` or `~/.local/share/.upref/example02.conf`.

```
login:
  label: Your login
  value: My login
passwd:
  label: Your secret password
  value: Password
```

## 1.4 Features

- Read and save small amount of data (url, login, )
- Small gui to ask data from the user
- Text option is available

## 1.5 License

The upref is licensed under the terms of the MIT license and is available for free.

MIT © Florent

## CHAPTER 2

---

### MIT License

---

Copyright (c) 2019 Florent Tournois

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.





### 3.1 Easy first Examples

Read the interesting URL from the user. At the second run, nothing is asked to the user.

```
1 import upref
2
3 user_data = upref.get_pref(
4     {"url": {"label": "The application url"}},
5     "example01")
6
7 print("URL is {}".format(user_data['url']))
```

Here is the created file. The is located in the user folder.

Listing 1: example01.conf

```
1 url:
2   label: The application url
3   value: http://www.test.org/
```



### 4.1 Class Hierarchy

### 4.2 File Hierarchy

### 4.3 Full API

#### 4.3.1 Namespaces

##### Namespace upref

Copyright (c) 2018 Florent TOURNOIS.

##### Contents

- *Detailed Description*
- *Namespaces*
- *Variables*

##### Detailed Description

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT

NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Small tool to grab data from the user and to save it

## Namespaces

- *Namespace upref::core*
- *Namespace upref::gui*
- *Namespace upref::tty*
- *Namespace upref::version*

## Variables

- *Variable upref::\_\_all\_\_*
- *Variable upref::\_\_author\_\_*
- *Variable upref::\_\_copyright\_\_*
- *Variable upref::\_\_credits\_\_*
- *Variable upref::\_\_email\_\_*
- *Variable upref::\_\_license\_\_*
- *Variable upref::\_\_maintainer\_\_*
- *Variable upref::\_\_status\_\_*
- *Variable upref::\_\_version\_\_*

## Namespace upref::core

### Contents

- *Functions*
- *Variables*

## Functions

- *Function upref::core::\_\_get\_this\_filename*
- *Function upref::core::\_\_get\_this\_folder*
- *Function upref::core::\_\_main*
- *Function upref::core::\_\_set\_logging\_system*
- *Function upref::core::all\_values\_are\_set*
- *Function upref::core::conv\_description\_to\_raw*

- Function `upref::core::conv_raw_to_description`
- Function `upref::core::current_upref`
- Function `upref::core::default_conf`
- Function `upref::core::dict_merge`
- Function `upref::core::get_pref`
- Function `upref::core::is_frozen`
- Function `upref::core::load_conf`
- Function `upref::core::load_data`
- Function `upref::core::remove_pref`
- Function `upref::core::save_conf`
- Function `upref::core::save_data`
- Function `upref::core::set_pref`
- Function `upref::core::upref_filename`

## Variables

- Variable `upref::core::__DEFAULT_CONF_FILENAME__`
- Variable `upref::core::__EXT_FILENAME__`
- Variable `upref::core::__UPREF_FOLDER__`

## Namespace `upref::gui`

### Contents

- *Classes*
- *Functions*
- *Variables*

## Classes

- Class `PrefDialog`

## Functions

- Function `upref::gui::__get_this_filename`
- Function `upref::gui::__get_this_folder`
- Function `upref::gui::__main`
- Function `upref::gui::__set_logging_system`

- *Function upref::gui::get\_data*
- *Function upref::gui::get\_widget*
- *Function upref::gui::get\_widget\_style*
- *Function upref::gui::is\_frozen*
- *Function upref::gui::message*

## Variables

- *Variable upref::gui::\_\_all\_\_*

## Namespace upref::tty

### Contents

- *Functions*
- *Variables*

## Functions

- *Function upref::tty::\_\_get\_this\_filename*
- *Function upref::tty::\_\_get\_this\_folder*
- *Function upref::tty::\_\_main*
- *Function upref::tty::\_\_set\_logging\_system*
- *Function upref::tty::get\_data*
- *Function upref::tty::is\_frozen*
- *Function upref::tty::message*

## Variables

- *Variable upref::tty::\_\_all\_\_*

## Namespace upref::version

### Contents

- *Variables*

## Variables

- Variable `upref::version::__release_date__`
- Variable `upref::version::__version_info__`

## 4.3.2 Classes and Structs

### Class PrefDialog

- Defined in file `upref_gui.py`

### Inheritance Relationships

#### Base Type

- `public Dialog`

### Class Documentation

**class PrefDialog**  
Main dialog in wxPython.

#### Public Functions

`__init__ (self self, parent parent, data data)`  
`init_ui (self self)`  
`on_ok (self self, event event)`  
`on_close (self self, event event)`

#### Public Members

`data_description`  
`panel`  
`data_widget`

## 4.3.3 Functions

### Function `upref::core::__get_this_filename`

- Defined in file `upref_core.py`

## Function Documentation

`upref.core.__get_this_filename()`

Find the filename of this file (depend on the frozen or not) This function return the filename of this script.

The function is complex for the frozen system

**Return** the filename of THIS script.

## Function `upref::core::__get_this_folder`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.__get_this_folder()`

Find the filename of this file (depend on the frozen or not) This function return the filename of this script.

The function is complex for the frozen system

**Return** the folder of THIS script.

## Function `upref::core::__main`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.__main()`

Main script call only if this script is runned directly.

## Function `upref::core::__set_logging_system`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.__set_logging_system()`

Set up the logging system.

## Function `upref::core::all_values_are_set`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.all_values_are_set(data_description data_description)`

Conversion from the direct pref dict to a dict of descripton.



### Function `upref::core::conv_description_to_raw`

- Defined in `file_upref_core.py`

#### Function Documentation

`upref.core.conv_description_to_raw(data_description data_description)`  
Conversion from the direct pref dict to a dict of description.

### Function `upref::core::conv_raw_to_description`

- Defined in `file_upref_core.py`

#### Function Documentation

`upref.core.conv_raw_to_description(data data)`  
Conversion from the direct pref dict to a dict of description.

### Function `upref::core::current_upref`

- Defined in `file_upref_core.py`

#### Function Documentation

`upref.core.current_upref(name name)`  
Read the current upref of the user.

**Return** the dict with the preference

**Parameters**

- `name`: the name of the preference file (we add the extension `.upref`)

### Function `upref::core::default_conf`

- Defined in `file_upref_core.py`

#### Function Documentation

`upref.core.default_conf()`  
Read a conf file and return the dict.

### Function `upref::core::dict_merge`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.dict_merge(dct dct, merge_dct merge_dct, add_keys add_keys = True)`

Recursive dict merge.

Inspired by `:meth:dict.update()`, instead of updating only top-level keys, `dict_merge` recurses down into dicts nested to an arbitrary depth, updating keys. The `merge_dct` is merged into `dct`.

This version will return a copy of the dictionary and leave the original arguments untouched.

The optional argument `add_keys`, determines whether keys which are present in `merge_dct` but not `dct` should be included in the new dict.

Code from <https://gist.github.com/angstwad/bf22d1822c38a92ec0a9>

Args: `dct` (dict) onto which the merge is executed `merge_dct` (dict): `dct` merged into `dct` `add_keys` (bool): whether to add new keys

Returns: dict: updated dict

## Function `upref::core::get_pref`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.get_pref(data_description data_description, name name, interface interface = "gui", force_renew force_renew = False, mandatory mandatory = True)`

Get the value of preference or setting.

## Function `upref::core::is_frozen`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.is_frozen()`

Test the frozen situation of the executable.

## Function `upref::core::load_conf`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.load_conf(filename filename)`

Read a conf file and return the dict.

## Function `upref::core::load_data`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.load_data` (*name name, default\_data default\_data = None*)

Read the current upref of the user.

**Return** the dict with the preference

### Parameters

- `name`: the name of the preference file (we add the extension `.upref`)

## Function `upref::core::remove_pref`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.remove_pref` (*name name*)

Remove the preference file.

## Function `upref::core::save_conf`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.save_conf` (*conf conf, filename filename*)

Read a conf file and return the dict.

## Function `upref::core::save_data`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.save_data` (*data data, name name*)

Read the current upref of the user.

**Return** the dict with the preference

### Parameters

- `name`: the name of the preference file (we add the extension `.upref`)

## Function `upref::core::set_pref`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.set_pref (data data, name name, data_description data_description = None)`  
Get the value of preference or setting.

## Function `upref::core::upref_filename`

- Defined in `file_upref_core.py`

## Function Documentation

`upref.core.upref_filename (name name)`  
Read the current upref of the user.

**Return** the complet filename of the upref file

### Parameters

- `name`: the name of the preference file (we add the extension `.upref`)

## Function `upref::gui::__get_this_filename`

- Defined in `file_upref_gui.py`

## Function Documentation

`upref.gui.__get_this_filename ()`  
Find the filename of this file (depend on the frozen or not) This function return the filename of this script.  
The function is complex for the frozen system

**Return** the filename of THIS script.

## Function `upref::gui::__get_this_folder`

- Defined in `file_upref_gui.py`

## Function Documentation

`upref.gui.__get_this_folder ()`  
Find the filename of this file (depend on the frozen or not) This function return the filename of this script.  
The function is complex for the frozen system

**Return** the folder of THIS script.

## Function `upref::gui::__main`

- Defined in `file_upref_gui.py`

## Function Documentation

`upref.gui.__main()`

Main script call only if this script is runned directly.

## Function `upref::gui::__set_logging_system`

- Defined in file `upref_gui.py`

## Function Documentation

`upref.gui.__set_logging_system()`

Set up the logging system.

## Function `upref::gui::get_data`

- Defined in file `upref_gui.py`

## Function Documentation

`upref.gui.get_data (data_description data_description)`

## Function `upref::gui::get_widget`

- Defined in file `upref_gui.py`

## Function Documentation

`upref.gui.get_widget (parent parent, data data)`

Build a widget to collect the data.

## Function `upref::gui::get_widget_style`

- Defined in file `upref_gui.py`

## Function Documentation

`upref.gui.get_widget_style (style style)`

Build a widget to collect the data.

## Function `upref::gui::is_frozen`

- Defined in file `upref_gui.py`

## Function Documentation

`upref.gui.is_frozen()`

Test the frozen situation of the executable.

## Function `upref::gui::message`

- Defined in `file_upref_gui.py`

## Function Documentation

`upref.gui.message(msg_txt msg_txt, title title)`

Display a message.

## Function `upref::tty::__get_this_filename`

- Defined in `file_upref_tty.py`

## Function Documentation

`upref.tty.__get_this_filename()`

Find the filename of this file (depend on the frozen or not) This function return the filename of this script.

The function is complex for the frozen system

**Return** the filename of THIS script.

## Function `upref::tty::__get_this_folder`

- Defined in `file_upref_tty.py`

## Function Documentation

`upref.tty.__get_this_folder()`

Find the filename of this file (depend on the frozen or not) This function return the filename of this script.

The function is complex for the frozen system

**Return** the folder of THIS script.

## Function `upref::tty::__main`

- Defined in `file_upref_tty.py`

## Function Documentation

`upref.tty.__main()`

Main script call only if this script is runned directly.

### Function upref::tty::\_\_set\_logging\_system

- Defined in file\_upref\_tty.py

#### Function Documentation

upref.tty.\_\_set\_logging\_system()  
Set up the logging system.

### Function upref::tty::get\_data

- Defined in file\_upref\_tty.py

#### Function Documentation

upref.tty.get\_data(*data\_description data\_description*)  
Get the data from the user.

### Function upref::tty::is\_frozen

- Defined in file\_upref\_tty.py

#### Function Documentation

upref.tty.is\_frozen()  
Test the frozen situation of the executable.

### Function upref::tty::message

- Defined in file\_upref\_tty.py

#### Function Documentation

upref.tty.message(*msg\_txt msg\_txt, title title*)  
Display a message.

## 4.3.4 Variables

### Variable upref::\_\_all\_\_

- Defined in file\_upref\_\_init\_\_.py

#### Variable Documentation

```
list upref.__all__ = [ 'get_pref', 'set_pref', 'remove_pref', 'upref_filename', '']
```

#### Variable upref::\_\_author\_\_

- Defined in file\_upref\_\_init\_\_.py

#### Variable Documentation

```
string upref.__author__ = "Florent Tournois"
```

#### Variable upref::\_\_copyright\_\_

- Defined in file\_upref\_\_init\_\_.py

#### Variable Documentation

```
string upref.__copyright__ = "Copyright 2018, Florent Tournois"
```

#### Variable upref::\_\_credits\_\_

- Defined in file\_upref\_\_init\_\_.py

#### Variable Documentation

```
list upref.__credits__ = ["Arnaud Boidard, Bernard Migaud"]
```

#### Variable upref::\_\_email\_\_

- Defined in file\_upref\_\_init\_\_.py

#### Variable Documentation

```
string upref.__email__ = "florent.tournois@gmail.fr"
```

#### Variable upref::\_\_license\_\_

- Defined in file\_upref\_\_init\_\_.py

#### Variable Documentation

```
string upref.__license__ = "MIT"
```

#### Variable upref::\_\_maintainer\_\_

- Defined in file\_upref\_\_init\_\_.py



### Variable Documentation

```
string upref.__maintainer__ = "Florent Tournois"
```

#### Variable upref::\_\_status\_\_

- Defined in file\_upref\_\_init\_\_.py

### Variable Documentation

```
string upref.__status__ = "Production"
```

#### Variable upref::\_\_version\_\_

- Defined in file\_upref\_\_init\_\_.py

### Variable Documentation

```
string upref.__version__ = ''.join(str(c) for c in __version_info__)
```

#### Variable upref::core::\_\_DEFAULT\_CONF\_FILENAME\_\_

- Defined in file\_upref\_core.py

### Variable Documentation

```
string upref.core.__DEFAULT_CONF_FILENAME__ = "default.conf"
```

The filename of the default configuration.

#### Variable upref::core::\_\_EXT\_FILENAME\_\_

- Defined in file\_upref\_core.py

### Variable Documentation

```
string upref.core.__EXT_FILENAME__ = ".conf"
```

The filename of the default configuration.

#### Variable upref::core::\_\_UPREF\_FOLDER\_\_

- Defined in file\_upref\_core.py

### Variable Documentation

`string upref.core.__UPREF_FOLDER__ = ".upref"`  
The filename of the default configuration.

### Variable `upref::gui::__all__`

- Defined in file `_upref_gui.py`

### Variable Documentation

`list upref.gui.__all__ = ['get_data']`

### Variable `upref::tty::__all__`

- Defined in file `_upref_tty.py`

### Variable Documentation

`list upref.tty.__all__ = ['get_data']`

### Variable `upref::version::__release_date__`

- Defined in file `_upref_version.py`

### Variable Documentation

`string upref.version.__release_date__ = '2019-09-26'`

### Variable `upref::version::__version_info__`

- Defined in file `_upref_version.py`

### Variable Documentation

`tuple upref.version.__version_info__ = (1, 0, 13)`

## CHAPTER 5

---

### Class Hierarchy

---



## CHAPTER 6

---

### File Hierarchy

---



### 7.1 Namespaces

### 7.2 Classes and Structs

### 7.3 Functions

### 7.4 Variables





## CHAPTER 8

---

### Indices and tables

---

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



**D**

data\_description (*upref.gui.PrefDialog* attribute), 11

data\_widget (*upref.gui.PrefDialog* attribute), 11

**P**

panel (*upref.gui.PrefDialog* attribute), 11

**U**

upref.core.\_\_get\_this\_filename() (*built-in function*), 12

upref.core.\_\_get\_this\_folder() (*built-in function*), 12

upref.core.\_\_main() (*built-in function*), 12

upref.core.\_\_set\_logging\_system() (*built-in function*), 12

upref.core.all\_values\_are\_set() (*built-in function*), 12

upref.core.conv\_description\_to\_raw() (*built-in function*), 13

upref.core.conv\_raw\_to\_description() (*built-in function*), 13

upref.core.current\_upref() (*built-in function*), 13

upref.core.default\_conf() (*built-in function*), 13

upref.core.dict\_merge() (*built-in function*), 14

upref.core.get\_pref() (*built-in function*), 14

upref.core.is\_frozen() (*built-in function*), 14

upref.core.load\_conf() (*built-in function*), 14

upref.core.load\_data() (*built-in function*), 15

upref.core.remove\_pref() (*built-in function*), 15

upref.core.save\_conf() (*built-in function*), 15

upref.core.save\_data() (*built-in function*), 15

upref.core.set\_pref() (*built-in function*), 16

upref.core.upref\_filename() (*built-in function*), 16

upref.gui.\_\_get\_this\_filename() (*built-in function*), 16

upref.gui.\_\_get\_this\_folder() (*built-in function*), 16

upref.gui.\_\_main() (*built-in function*), 17

upref.gui.\_\_set\_logging\_system() (*built-in function*), 17

upref.gui.get\_data() (*built-in function*), 17

upref.gui.get\_widget() (*built-in function*), 17

upref.gui.get\_widget\_style() (*built-in function*), 17

upref.gui.is\_frozen() (*built-in function*), 18

upref.gui.message() (*built-in function*), 18

upref.gui.PrefDialog (*built-in class*), 11

upref.gui.PrefDialog.\_\_init\_\_() (*built-in function*), 11

upref.gui.PrefDialog.init\_ui() (*built-in function*), 11

upref.gui.PrefDialog.on\_close() (*built-in function*), 11

upref.gui.PrefDialog.on\_ok() (*built-in function*), 11

upref.tty.\_\_get\_this\_filename() (*built-in function*), 18

upref.tty.\_\_get\_this\_folder() (*built-in function*), 18

upref.tty.\_\_main() (*built-in function*), 18

upref.tty.\_\_set\_logging\_system() (*built-in function*), 19

upref.tty.get\_data() (*built-in function*), 19

upref.tty.is\_frozen() (*built-in function*), 19

upref.tty.message() (*built-in function*), 19