

---

# PlugSy Documentation

**Sithis**

**May 30, 2018**



---

## Contents:

---

<b>1</b>	<b>About</b>	<b>1</b>
<b>2</b>	<b>Links</b>	<b>3</b>
<b>3</b>	<b>Project Goal</b>	<b>5</b>
<b>4</b>	<b>Features</b>	<b>7</b>
<b>5</b>	<b>Roadmap</b>	<b>9</b>
<b>6</b>	<b>Similar Projects</b>	<b>11</b>
<b>7</b>	<b>Example Usage</b>	<b>13</b>
7.1	PlugSy . . . . .	13
7.2	SDK . . . . .	14
<b>8</b>	<b>Installation</b>	<b>15</b>
<b>9</b>	<b>Dependencies</b>	<b>17</b>
<b>10</b>	<b>Indices and tables</b>	<b>19</b>



# CHAPTER 1

---

## About

---

PlugSy is a cross-platform threaded plugin framework and SDK for Python. With PlugSy, each plugin runs in a separate thread and has a handle to the global PlugSy plugin management object, allowing plugins to communicate and interact with each other. Whilst there are numerous existing Python Plugin frameworks, it's this built-in support for threading that makes PlugSy slightly different.



## CHAPTER 2

---

### Links

---

- home: <https://github.com/sithis993/PlugSy>
- docs: <http://plugsy.readthedocs.io>
- PyPI: <https://pypi.org/project/PlugSy/>





## CHAPTER 3

---

### Project Goal

---

The goal of this project is to aid with the rapid development of Python applications of varying size and complexity. PlugSy tries to make it easy for developers to create small modular components (plugins) that interact and combine to form the basis of a complete software piece. By keeping extensibility and modularity in mind, PlugSy allows you to move from a basic single-plugin application, to a large application with a multitude of complex plugins, at your own pace.



## CHAPTER 4

---

### Features

---

- Separation of plugins into core and addon packages
- Cross-platform support
- Pure-python, no dependencies required
- Built-in logging support for the PlugSy manager and each individual plugin
- SDK GUI for quickly creating and deleting plugins



## CHAPTER 5

---

### Roadmap

---

- **PyInstaller integration and executable production**
  - Combining PlugSy and any developed plugins into a PyInstaller executable via SDK and SDK GUI
  - Option to add a Windows service wrapper
- Further support for additional Python versions



## CHAPTER 6

---

### Similar Projects

---

- PluginBase: <http://pluginbase.pocoo.org/>
- Yapsy: <http://yapsy.sourceforge.net/>





---

## Example Usage

---

### 7.1 PlugSy

#### Initiating PlugSy

```
from plugsy import Plugsy  
  
plugsy = Plugsy()
```

#### Activating all existing plugins

```
plugsy.activate_plugins()
```

#### Activating specific plugins

```
plugsy.activate_plugins(["MainPlugin", "FirstPlugin", "FifthPlugin"])
```

#### Deactivating all plugins

```
plugsy.deactivate_plugins()
```

#### Deactivating specific plugins

```
plugsy.deactivate_plugins(["MainPlugin", "FirstPlugin", "FifthPlugin"])
```

#### Interacting with plugin objects

```
main_plugin = plugsy.get_plugin("MainPlugin")  
main_plugin.do_something()  
main_plugin.do_something_else()  
  
first_plugin = plugsy.get_plugin("FirstPlugin")  
first_plugin.take_an_action("some_action")  
some_data = first_plugin.get_some_data()
```

## 7.2 SDK

### Initiating the SDK

```
from plugsy.sdk.Sdk import Sdk

# Pass in path to directory containing plugins (or path to an empty dir)
sdk = Sdk(plugins_home_path=".\\plugins")
```

### Creating a new plugin

```
# Pass in the plugin type (core or addon) and plugin name
sdk.create_plugin(plugin_type="core", name="MyNewPlugin")
```

### Deleting a plugin

```
sdk.delete_plugin(name="MyNewPlugin")
```

## CHAPTER 8

---

### Installation

---

Installation of PlugSy is simple using Pip:

```
pip install plugsy
```



## CHAPTER 9

---

### Dependencies

---

- \*wxPython (4.0.1): <https://pypi.org/project/wxPython/#description>

\*wxPython is only required if you're running Windows and want to use the SDK GUI



## CHAPTER 10

---

### Indices and tables

---

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