## Contents

1 Getting started 3  
   1.1 Installation ................................................................. 3  
   1.2 Usage .............................................................. 3  

2 API 5  
   2.1 Package ................................................................. 5  
       2.1.1 Submodules ......................................................... 5  
   2.2 Indices and tables ..................................................... 8  

Python Module Index 9  

Index 11
A python library for managing a hosts file.
CHAPTER 1

Getting started

1.1 Installation

Using pip package manager

$ pip install python-hosts

From source

$ git clone https://github.com/jonhadfield/python-hosts
$ cd python-hosts
$ python setup.py install

1.2 Usage

Create an instance of a hosts file:

```python
from python_hosts import Hosts, HostsEntry
my_hosts = Hosts()
```

Add an entry:

```python
new_entry = HostsEntry(entry_type='ipv4', address='1.2.3.4', names=['example.com', 'example'])
my_hosts.add([new_entry])
```

Remove an entry/entries matching an address:

```python
my_hosts.remove_all_matching(address='1.2.3.4')
```
Remove an entry/entries matching an address:

```python
my_hosts.remove_all_matching(name='example.com')
```

Write entries:

```python
my_hosts.write()
```
CHAPTER 2

API

2.1 Package

This package contains all of the modules utilised by the python-hosts library.

hosts: Contains the Hosts and HostsEntry classes that represent instances of a hosts file and it’s individual lines/entries
util: Contains helper functions to check the available operations on a hosts file and the validity of a hosts file entry
exception: Contains the custom exceptions that are raised in the event of an error in processing a hosts file and its entries

2.1.1 Submodules

python_hosts.hosts module

This module contains classes: HostsEntry: A representation of a hosts file entry, i.e. a line containing an IP address and name(s), a comment, or a blank line/line separator.

Hosts: A representation of a hosts file, e.g. /etc/hosts and c:\windows\system32\drivers\etc\hosts for a linux or MS windows based machine respectively. Each entry being represented as an instance of the HostsEntry class.

class python_hosts.hosts.Hosts(path=None)
    Bases: object
    A hosts file.

    add(entries=None, force=False, allow_address_duplication=False, merge_names=False)
    Add instances of HostsEntry to the instance of Hosts. :param entries: A list of instances of HostsEntry
    :param force: Remove matching before adding :param allow_address_duplication: Allow using multiple entries for same address :param merge_names: Merge names where address already exists :return: The counts of successes and failures

count()
    Get a count of the number of host entries :return: The number of host entries
static **determine_hosts_path** *(platform=None)*
Return the hosts file path based on the supplied or detected platform. :param platform: a string used to identify the platform :return: detected filesystem path of the hosts file

**entries**

**exists** *(address=None, names=None, comment=None)*
Determine if the supplied address and/or names, or comment, exists in a HostsEntry within Hosts :param address: An ipv4 or ipv6 address to search for :param names: A list of names to search for :param comment: A comment to search for :return: True if a supplied address, name, or comment is found. Otherwise, False.

**static get_hosts_by_url** *(url=None)*
Request the content of a URL and return the response :param url: The URL of the hosts file to download :return: The content of the passed URL

**hosts_path**

**import_file** *(import_file_path=None)*
Read a list of host entries from a file, convert them into instances of HostsEntry and then append to the list of entries in Hosts :param import_file_path: The path to the file containing the host entries :return: Counts reflecting the attempted additions

**import_url** *(url=None, force=None)*
Read a list of host entries from a URL, convert them into instances of HostsEntry and then append to the list of entries in Hosts :param url: The URL of where to download a hosts file :return: Counts reflecting the attempted additions

**populate_entries** *
Called by the initialiser of Hosts. This reads the entries from the local hosts file, converts them into instances of HostsEntry and adds them to the Hosts list of entries. :return: None

**remove_all_matching** *(address=None, name=None)*
Remove all HostsEntry instances from the Hosts object where the supplied ip address or name matches :param address: An ipv4 or ipv6 address :param name: A host name :return: None

**write** *(path=None)*
Write all of the HostsEntry instances back to the hosts file :param path: override the write path :return: Dictionary containing counts

### class python_hosts.hosts.HostsEntry *(entry_type=None, address=None, comment=None, names=None)*
Bases: object
An entry in a hosts file.

**address**

**comment**

**entry_type**

**static get_entry_type** *(hosts_entry=None)*
Return the type of entry for the line of hosts file passed :param hosts_entry: A line from the hosts file :return: ‘comment’ | ‘blank’ | ‘ipv4’ | ‘ipv6’

**is_real_entry** *

**names**

**static str_to_hostentry** *(entry)*
Transform a line from a hosts file into an instance of HostsEntry :param entry: A line from the hosts file :return: An instance of HostsEntry
**python_hosts.utils module**

This module contains utility functions used by the Hosts and HostsEntry methods.

**dedupe_list**

Utility function to remove duplicates from a list.

**is_ipv4**

Check if the string provided is a valid ipv4 address.

**is_ipv6**

Check if the string provided is a valid ipv6 address.

**is_readable**

Test if the supplied filesystem path can be read.

**valid_hostnames**

Check if the supplied list of strings are valid hostnames.

**python_hosts.exception module**

**HostsEntryException**

Base exception class. All HostsEntry-specific exceptions should subclass this class.

**HostsException**

Base exception class. All Hosts-specific exceptions should subclass this class.

**InvalidComment**

Raised when a HostsEntry is defined as type ‘comment’ but with an invalid comment.

**InvalidIPv4Address**

Raised when a HostsEntry is defined as type ‘ipv4’ but with an invalid address.

**InvalidIPv6Address**

Raised when a HostsEntry is defined as type ‘ipv6’ but with an invalid address.

**UnableToWriteHosts**

Raised when a Hosts file cannot be written.
2.2 Indices and tables

- genindex
- modindex
- search
p

python_hosts, 5
python_hosts.exception, 7
python_hosts.hosts, 5
python_hosts.utils, 7
<table>
<thead>
<tr>
<th>Index</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>add() (python_hosts.hosts.Hosts method), 5</td>
</tr>
<tr>
<td>C</td>
<td>comment (python_hosts.hosts.HostsEntry attribute), 6</td>
</tr>
<tr>
<td>D</td>
<td>dedupe_list() (in module python_hosts.utils), 7</td>
</tr>
<tr>
<td>E</td>
<td>entries (python_hosts.hosts.Hosts attribute), 6</td>
</tr>
<tr>
<td>G</td>
<td>get_entry_type() (python_hosts.hosts.HostsEntry static method), 6</td>
</tr>
<tr>
<td>H</td>
<td>Hosts (class in python_hosts.hosts), 5</td>
</tr>
<tr>
<td>I</td>
<td>import_file() (python_hosts.hosts.Hosts method), 6</td>
</tr>
<tr>
<td>N</td>
<td>names (python_hosts.hosts.HostsEntry attribute), 6</td>
</tr>
<tr>
<td>P</td>
<td>populate_entries() (python_hosts.hosts.Hosts method), 6</td>
</tr>
<tr>
<td>R</td>
<td>remove_all_matching() (python_hosts.hosts.Hosts method), 6</td>
</tr>
<tr>
<td>S</td>
<td>str_to_hostentry() (python_hosts.hosts.HostsEntry static method), 6</td>
</tr>
<tr>
<td>U</td>
<td>UnableToWriteHosts, 7</td>
</tr>
<tr>
<td>V</td>
<td>valid_hostnames() (in module python_hosts.utils), 7</td>
</tr>
<tr>
<td>W</td>
<td>write() (python_hosts.hosts.Hosts method), 6</td>
</tr>
</tbody>
</table>