

---

# **pygeoip Documentation**

***Release 0.3.2***

**Jennifer Ennis, William Tisäter**

October 29, 2014



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Issues and Contribution</b>	<b>5</b>
<b>3</b>	<b>Documentation</b>	<b>7</b>
3.1	Getting Started . . . . .	7
3.2	Supported Databases . . . . .	8
3.3	API Reference . . . . .	9



This library is based on [Maxmind's GeoIP C API](#).

Tested with Python version 2.6, 2.7, 3.2 and 3.3.



---

# Installation

---

You can easily install pygeoip from PyPi.

```
pip install pygeoip
```





---

## Issues and Contribution

---

Bug reports are done by [creating an issue on Github](#). If you want to contribute you can always [create a pull request](#) for discussion and code submission.



## 3.1 Getting Started

Create your GeoIP instance with appropriate access flag. STANDARD reads data from disk when needed, MEMORY\_CACHE loads database into memory on instantiation and MMAP\_CACHE loads database into memory using mmap.

```
>>> import pygeoip
>>> gi = pygeoip.GeoIP('GeoIP.dat')
>>> gi.country_name_by_addr('64.233.161.99')
'United States'
```

### 3.1.1 Country Lookup

```
>>> gi = pygeoip.GeoIP('GeoIP.dat')
>>> gi.country_code_by_name('google.com')
'US'
>>> gi.country_code_by_addr('64.233.161.99')
'US'
>>> gi.country_name_by_addr('64.233.161.99')
'United States'

>>> gi = pygeoip.GeoIP('GeoIPv6.dat')
>>> gi.country_code_by_addr('2a00:1450:400f:802::1006')
'IE'
```

### 3.1.2 Region Lookup

```
>>> gi = pygeoip.GeoIP('GeoIPRegion.dat')
>>> gi.region_by_name('apple.com')
{'region_code': 'CA', 'country_code': 'US'}
```

### 3.1.3 City Lookup

```
>>> gi = pygeoip.GeoIP('GeoIPCity.dat')
>>> gi.record_by_addr('64.233.161.99')
{
    'city': u'Mountain View',
```

```
    'region_code': u'CA',
    'area_code': 650,
    'time_zone': 'America/Los_Angeles',
    'dma_code': 807,
    'metro_code': 'San Francisco, CA',
    'country_code3': 'USA',
    'latitude': 37.419199999999999,
    'postal_code': u'94043',
    'longitude': -122.0574,
    'country_code': 'US',
    'country_name': 'United States',
    'continent': 'NA'
}
>>> gi.time_zone_by_addr('64.233.161.99')
'America/Los_Angeles'
```

### 3.1.4 Organization Lookup

```
>>> gi = pygeoip.GeoIP('GeoIPOrg.dat')
>>> gi.org_by_name('dell.com')
'Dell Computer Corporation'
```

### 3.1.5 ISP Lookup

```
>>> gi = pygeoip.GeoIP('GeoIPISP.dat')
>>> gi.isp_by_name('cnn.com')
'Turner Broadcasting System'
```

### 3.1.6 ASN Lookup

```
>>> gi = pygeoip.GeoIP('GeoIPASNum.dat')
>>> gi.asn_by_name('cnn.com')
'AS5662 Turner Broadcasting'
```

## 3.2 Supported Databases

Type	IPv4	IPv6	Details
Country	Yes	Yes	<a href="#">MaxMind Country product page</a>
City	Yes	Yes	<a href="#">MaxMind City product page</a>
Organization	Yes		<a href="#">MaxMind Organization product page</a>
ISP	Yes		<a href="#">MaxMind ISP product page</a>
Region	Yes		<a href="#">MaxMind Region product page</a>
ASN	Yes	Yes	<a href="#">MaxMind ASN product page</a>
Netspeed	Yes		<a href="#">MaxMind Netspeed product page</a>

## 3.3 API Reference

### 3.3.1 GeoIP

`class pygeoip.GeoIP (filename, flags=0, cache=True)`

`__init__ (filename, flags=0, cache=True)`  
Create and return an GeoIP instance.

#### Parameters

- **filename** – File path to a GeoIP database
- **flags** – Flags that affect how the database is processed. Currently supported flags are STANDARD (default), MEMORY\_CACHE (preload the whole file into memory) and MMAP\_CACHE (access the file via mmap)
- **cache** – Used in tests to skip instance caching

`country_code_by_addr (addr)`  
Returns 2-letter country code (e.g. US) from IP address.

**Parameters** **addr** – IP address (e.g. 203.0.113.30)

`country_code_by_name (hostname)`  
Returns 2-letter country code (e.g. US) from hostname.

**Parameters** **hostname** – Hostname (e.g. example.com)

`country_name_by_addr (addr)`  
Returns full country name for specified IP address.

**Parameters** **addr** – IP address (e.g. 203.0.113.30)

`country_name_by_name (hostname)`  
Returns full country name for specified hostname.

**Parameters** **hostname** – Hostname (e.g. example.com)

`id_by_addr (addr)`  
Returns the database ID for specified address. The ID might be useful as array index. 0 is unknown.

**Parameters** **addr** – IPv4 or IPv6 address (eg. 203.0.113.30)

`last_netmask ()`  
Returns the netmask depth of the last lookup.

`netspeed_by_addr (addr)`  
Returns NetSpeed name from address.

**Parameters** **addr** – IP address (e.g. 203.0.113.30)

`netspeed_by_name (hostname)`  
Returns NetSpeed name from hostname. Can be Unknown, Dial-up, Cable, or Corporate.

**Parameters** **hostname** – Hostname (e.g. example.com)

`org_by_addr (addr)`  
Returns Organization, ISP, or ASNum name for given IP address.

**Parameters** **addr** – IP address (e.g. 203.0.113.30)

**org\_by\_name** (*hostname*)

Returns Organization, ISP, or ASNum name for given hostname.

**Parameters** **hostname** – Hostname (e.g. example.com)

**record\_by\_addr** (*addr*)

Returns dictionary with city data containing *country\_code*, *country\_name*, *region*, *city*, *postal\_code*, *latitude*, *longitude*, *dma\_code*, *metro\_code*, *area\_code*, *region\_code* and *time\_zone*.

**Parameters** **addr** – IP address (e.g. 203.0.113.30)

**record\_by\_name** (*hostname*)

Returns dictionary with city data containing *country\_code*, *country\_name*, *region*, *city*, *postal\_code*, *latitude*, *longitude*, *dma\_code*, *metro\_code*, *area\_code*, *region\_code* and *time\_zone*.

**Parameters** **hostname** – Hostname (e.g. example.com)

**region\_by\_addr** (*addr*)

Returns dictionary containing *country\_code* and *region\_code*.

**Parameters** **addr** – IP address (e.g. 203.0.113.30)

**region\_by\_name** (*hostname*)

Returns dictionary containing *country\_code* and *region\_code*.

**Parameters** **hostname** – Hostname (e.g. example.com)

**time\_zone\_by\_addr** (*addr*)

Returns time zone in tzdata format (e.g. America/New\_York or Europe/Paris)

**Parameters** **addr** – IP address (e.g. 203.0.113.30)

**time\_zone\_by\_name** (*hostname*)

Returns time zone in tzdata format (e.g. America/New\_York or Europe/Paris)

**Parameters** **hostname** – Hostname (e.g. example.com)

### 3.3.2 GeoIPError

**exception** `pygeoip.GeoIPError`

Thin wrapper of *Exception*, will be thrown in case of an unrecoverable error.

## Symbols

`__init__()` (pygeoip.GeoIP method), 9

## C

`country_code_by_addr()` (pygeoip.GeoIP method), 9

`country_code_by_name()` (pygeoip.GeoIP method), 9

`country_name_by_addr()` (pygeoip.GeoIP method), 9

`country_name_by_name()` (pygeoip.GeoIP method), 9

## G

GeoIP (class in pygeoip), 9

GeoIPError, 10

## I

`id_by_addr()` (pygeoip.GeoIP method), 9

## L

`last_netmask()` (pygeoip.GeoIP method), 9

## N

`netspeed_by_addr()` (pygeoip.GeoIP method), 9

`netspeed_by_name()` (pygeoip.GeoIP method), 9

## O

`org_by_addr()` (pygeoip.GeoIP method), 9

`org_by_name()` (pygeoip.GeoIP method), 9

## R

`record_by_addr()` (pygeoip.GeoIP method), 10

`record_by_name()` (pygeoip.GeoIP method), 10

`region_by_addr()` (pygeoip.GeoIP method), 10

`region_by_name()` (pygeoip.GeoIP method), 10

## T

`time_zone_by_addr()` (pygeoip.GeoIP method), 10

`time_zone_by_name()` (pygeoip.GeoIP method), 10