



**IP2LOCATION™**

**ACTIVEX COMPONENT**

User Manual

By Hexasoft Development Sdn. Bhd.

Copyright © 2001-2017 IP2Location.com.  
All Rights Reserved.

Visit our Web site at  
<http://www.ip2location.com>

~ Bringing Geography to the Internet ~

# TABLE OF CONTENTS

- 1. INTRODUCTION..... 2
- 2. SYSTEM REQUIREMENTS ..... 2
  - 2.1 Quick Start Guide ..... 3
- 3. DOWNLOAD AND INSTALLATION ..... 3
  - 3.1 Download ZIP File ..... 3
  - 3.2 BIN Database Download ..... 4
  - 3.3 Component Installation ..... 4
- 4. TESTING ..... 5
  - 4.1 Folders and Files Structure..... 5
  - 4.2 Sample Codes in PHP ..... 5
  - 4.3 Sample Codes in Visual Basic 6 ..... 5
  - 4.4 Sample Codes in C#..... 5
  - 4.5 Sample Codes in VB.NET..... 5
  - 4.6 Sample Codes in Cold Fusion ..... 5
  - 4.7 Sample Codes in Visual C++ ..... 6
  - 4.8 Sample Codes in Delphi ..... 7
  - 4.9 Sample Codes in Perl ..... 8
- 5. IMPLEMENTATION ..... 10
  - 5.1 IP2Location™ ActiveX Method Reference ..... 10
  - 5.2 IP2Location™ ActiveX Property Reference ..... 10
- 6. LICENSE PURCHASE ..... 11
  - 6.1 License Information ..... 11
  - 6.2 Purchase Instruction ..... 11
- 7. UPDATE COMPONENT ..... 12
  - 7.1 Update Component Internal Database..... 12
  - 7.2 Upgrade Component ..... 12
  - 7.3 Uninstall Component ..... 12
- 8. APPENDIX I: ISO3166 COUNTRY CODE ..... 13

# 1. INTRODUCTION

**IP2Location™** is the solution to identify *country, region, city, latitude & longitude of city, ZIP code, time zone, connection speed, ISP, domain name, IDD country code, area code, weather station code and name, mobile carrier, elevation and usage type* of your site visitors in real-time by using IP address. This document provides important information to help you quickly get up to speed with our products.

Developers use this ActiveX/COM component for software and web projects related to, but not limited to:

- Display native language and currency
- Redirect based on country
- Digital rights management
- Prevent password sharing and abuse of service
- Reduce credit card fraud
- Web log stats and analysis
- Auto-selection of country on forms
- Filter access from countries you do not do business with
- Geo-targeting for increased sales and click-through

IP2Location™ ActiveX/COM component is written and optimized for Windows® operating system performance.

# 2. SYSTEM REQUIREMENTS

IP2Location™ ActiveX/COM Component has been developed to run exclusively under Windows based x86/x64 machines with support for both standalone desktop and web server application. This component has built-in IP-COUNTRY database but can read a local IP2Location™ BIN database for lookup purpose. No Internet connection required to execute this component.

## Minimum Configuration Requirements

Scenario	CPU Required	RAM Required
Client	Pentium 90 MHz*	32 MB**
Server	Pentium 133 MHz*	128 MB**

\* Or the minimum CPU required running the operating system, whichever is higher.

\*\* Or the minimum RAM required running the operating system, whichever is higher.

### Supported Platforms

- Microsoft® Windows XP (Home or Professional)
- Microsoft® Windows 2000 (Professional, Server, or Advanced Server) with the latest Windows service pack and critical updates available from the Microsoft Security Web site.
- Microsoft® Windows Server 2003 family
- Microsoft® Windows Vista
- Microsoft® Windows 7
- Microsoft® Windows 8
- Microsoft® Windows 10
- Microsoft® Windows Server 2008 family
- Microsoft® Windows Server 2012 family
- Microsoft® Windows Server 2016 family

## 2.1 Quick Start Guide

In this document, you will go through these basic steps to use IP2Location™ ActiveX/COM Component

1. Download and installing IP2Location ActiveX/COM Component
2. Test your ActiveX/COM Component installation
3. Start using IP2Location™ ActiveX/COM Component in your projects
4. Purchase license to remove random 5 second delay in demo version
5. Update IP2Location™ Component database

**Note: For unregistered or demo, the component applies random 5-second in query. Registered version has unlimited real-time query without delay.**

## 3. DOWNLOAD AND INSTALLATION

### 3.1 Download ZIP File

Please use the following URL to download the latest IP2Location™ ActiveX/COM Component together with sample database.

- <https://www.ip2location.com/software/active-x>

## 3.2 BIN Database Download

You may download free sample databases, DB1 to DB24, at <https://www.ip2location.com/developers> for your testing. Please take note that the sample databases only contain IP address from **0.0.0.0 to 99.255.255.255 (IPv4)** or **2A04:0:0:0:0:0:0:0 to 2A04:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF (IPv6)** only.

If you need full IP address ranges but lower accuracy for testing, you may consider to download the LITE database at <https://lite.ip2location.com>.

## 3.3 Component Installation

Installing the IP2Location ActiveX/COM component system is very easy.

General steps in installing IP2Location™ ActiveX/COM component.

1. Unzip the demo package into a temporary directory.
2. Just copy the **ip2location.dll** file and sample database "IP2LOCATION-LITE-DB1. BIN" to a system directory in Web server. Directory *Windows\system32\* or *Winnt\system32\* is recommended.
3. Register the DLL component using one of the following options :

### Option 1

From the command prompt, navigate to the directory where you installed ip2location.dll and then run the command:

```
C:\Windows\system32> regsvr32 ip2location.dll
```

**OR**

### Option 2

Click on the Windows® "Start" button, then "Run" and typing in:

```
regsvr32 <fullpath>ip2location.dll
```

Make sure and replace "<fullpath>" with the actual full path that the ip2location.dll resides in.

4. In Explorer, right click ip2location.dll, select "Properties" and go to "Security" tab. Add "Read & Execute" permission to the Internet user account ie. "IUSR\_<machine>".
5. In Explorer, right click IP2LOCATION-LITE-DB1.BIN, select "Properties" and go to "Security" tab. Add "Read" permission to the Internet user account. i.e. "IUSR <machine>"

**\*Note: Please use x64ip2location.dll in 64bit machine.**

## 4. TESTING

### 4.1 Folders and Files Structure

Several sample files were enclosed in component zip file. The sample filenames by default are :

/asp/ip2location.asp	(ASP demo script)
/php/ip2location.php	(PHP demo script)
/vb6/ip2location.frm	(Visual Basic 6 demo application)
/asp.net/vb.net/ip2location.aspx.vb	(VB.NET demo script)
/asp.net/c#/ip2location.aspx.cs	(C# demo script)

Copy one of these samples to a Web directory configured to run Web project in specific language. You need to copy the IP2Location Sample BIN file IP2LOCATION-LITE-DB1.IPV6.BIN to the directory as well. Then run these samples using Web Browser.

If the IP2Location™ has been installed correctly, the sample script will execute and demonstrate results from the IP2Location products. For demo version, there is a 5-minutes delay per query. Registered version enjoys unlimited real-time query without delay.

*\*Note: If you are getting this error "Attempted to load a 64-bit assembly on a 32-bit platform. Use ReflectionOnlyLoad() instead of trying to load for reflection purposes." Please disable "Enable 32-Bit Applications" in your IIS Application Pools.*

### 4.2 Sample Codes in PHP

Please refer to the PHP sample codes in the setup package for more information.

### 4.3 Sample Codes in Visual Basic 6

Please refer to the VB6 sample codes in the setup package for more information.

### 4.4 Sample Codes in C#

Please refer to the C# sample codes in the setup package for more information.

### 4.5 Sample Codes in VB.NET

Please refer to the VB.NET sample codes in the setup package for more information.

### 4.6 Sample Codes in Cold Fusion

Query Country Information using Internal Database

```
<!-- connect to COM object -->
```

```

<CFOBJECT ACTION="Create" NAME="ipObj" CLASS="IP2Location.Country">
<!-- initialize IP2LocationT Component -->
<CFSET result_code = ipObj.Initialize("demo")>
<!-- it will return "US" -->
<CFSET result_code = ipObj.LookUpShortName("205.188.145.215")>
<!-- it will return "United States" -->
<CFSET result_code = ipObj.LookUpFullName("205.188.145.215")>
<!-- it will return "Japan" -->
<CFSET result_code = ipObj.CountryCodeToName("JP")>
<!-- it will return the version number -->
<CFSET result_code = ipObj.Version()>

```

## Query Other Information using External Database

```

<!-- connect to COM object -->
<CFOBJECT ACTION="Create" NAME="ipObj" CLASS="IP2Location.Country">
<!-- initialize IP2LocationT Component -->
<CFSET result_code = ipObj.Initialize("demo")>
<!-- query external IP2Location database -->
<CFSET result_code = ipObj.LookUp("IP2LOCATION-LITE-
DB1.BIN", "15.188.145.215")>
<CFSET result_code = ipObj.CountryShort>
<CFSET result_code = ipObj.CountryLong>
<CFSET result_code = ipObj.Region>
<CFSET result_code = ipObj.City>
<CFSET result_code = ipObj.Latitude>
<CFSET result_code = ipObj.Longitude>
<CFSET result_code = ipObj.ZIPCode>
<CFSET result_code = ipObj.ISP>
<CFSET result_code = ipObj.Domain>
<CFSET result_code = ipObj.TimeZone>
<CFSET result_code = ipObj.NetSpeed>
<CFSET result_code = ipObj.IDDCode>
<CFSET result_code = ipObj.AreaCode>
<CFSET result_code = ipObj.WeatherStationCode>
<CFSET result_code = ipObj.WeatherStationName>
<CFSET result_code = ipObj.MCC>
<CFSET result_code = ipObj.MNC>
<CFSET result_code = ipObj.MobileBrand>
<CFSET result_code = ipObj.Elevation>
<CFSET result_code = ipObj.UsageType>

```

## 4.7 Sample Codes in Visual C++

### Query Country Information using Internal Database

```

#import "c:\winnt\system32\ip2location.dll" no_namespace
HRESULT hr = OleInitialize(NULL);
ICountry ipObj("IP2Location.Country");
if ((hr == S_OK) || (hr == S_FALSE))
{
    // initialize IP2LocationT Component
    if (ipObj->Initialize("demo") != "OK")
    {

```

```

        // initialization failed
    }
    // it will return "US"
    AfxMessageBox(ipObj->LookUpShortName("205.188.145.215"));
    // it will return "United States"
    AfxMessageBox(ipObj->LookUpFullName("205.188.145.215"));
    // it will return "Japan"
    AfxMessageBox(ipObj->CountryCodeToName("JP"));
    // it will return the version number
    AfxMessageBox(ipObj->Version());
}
OleUninitialize();

```

## Query Other Information using External Database

```

#import "c:\winnt\system32\ip2location.dll" no_namespace
HRESULT hr = OleInitialize(NULL);
ICountry ipObj("IP2Location.Country");
if ((hr == S_OK) || (hr == S_FALSE))
{
    // initialize IP2Location Component
    if (ipObj->Initialize("demo") != "OK")
    {
        // initialization failed
    }
    AfxMessageBox(ipObj->LookUp("IP2LOCATION-LITE-DB1.BIN", "15.188.145.215"));
    AfxMessageBox(ipObj->CountryShort);
    AfxMessageBox(ipObj->CountryLong);
    AfxMessageBox(ipObj->Region);
    AfxMessageBox(ipObj->City);
    AfxMessageBox(ipObj->Latitude);
    AfxMessageBox(ipObj->Longitude);
    AfxMessageBox(ipObj->ZIPCode);
    AfxMessageBox(ipObj->ISP);
    AfxMessageBox(ipObj->Domain);
    AfxMessageBox(ipObj->TimeZone);
    AfxMessageBox(ipObj->NetSpeed);
    AfxMessageBox(ipObj->IDDCCode);
    AfxMessageBox(ipObj->AreaCode);
    AfxMessageBox(ipObj->WeatherStationCode);
    AfxMessageBox(ipObj->WeatherStationName);
    AfxMessageBox(ipObj->MCC);
    AfxMessageBox(ipObj->MNC);
    AfxMessageBox(ipObj->MobileBrand);
    AfxMessageBox(ipObj->Elevation);
    AfxMessageBox(ipObj->UsageType);
}
OleUninitialize();

```

## 4.8 Sample Codes in Delphi

### Query Country Information using Internal Database

```

uses ComObj;
var

```



```

ipObj: Variant;
nResult: Variant;
begin
    // create IP2LocationT object
    ipObj:=CreateOleObject('IP2Location.Country');
    // initialize IP2LocationT Component
    nResult := ipObj.Initialize('demo');
    if nResult = 'OK' then begin;
        // initialization passed
    else
        // initialization failed
    end;
    // it will return "US"
    nResult := ipObj.LookUpShortName('205.188.145.215');
    // it will return "United States"
    nResult := ipObj.LookUpFullName('205.188.145.215');
    // it will return "Japan"
    nResult := ipObj.CountryCodeToName('JP');
    // it will return the version number
    nResult := ipObj.Version();
end;

```

### Query Other Information using External Database

```

uses ComObj;
var
ipObj: Variant;
nResult: Variant;
begin
    // create IP2LocationT object
    ipObj:=CreateOleObject('IP2Location.Country');
    // initialize IP2LocationT Component
    nResult := ipObj.Initialize('demo');
    nResult := ipObj.LookUp('IP2LOCATION-LITE-DB1.BIN', '15.188.145.215');
    nResult := ipObj.CountryShort;
    nResult := ipObj.CountryLong;
    nResult := ipObj.Region;
    nResult := ipObj.City;
    nResult := ipObj.Latitude;
    nResult := ipObj.Longitude;
    nResult := ipObj.ZIPCode;
    nResult := ipObj.ISP;
    nResult := ipObj.Domain;
    nResult := ipObj.TimeZone;
    nResult := ipObj.NetSpeed;
    nResult := ipObj.IDDCode;
    nResult := ipObj.AreaCode;
    nResult := ipObj.WeatherStationCode;
    nResult := ipObj.WeatherStationName;
    nResult := ipObj.MCC;
    nResult := ipObj.MNC;
    nResult := ipObj.MobileBrand;
    nResult := ipObj.Elevation;
    nResult := ipObj.UsageType;
end;

```

## 4.9 Sample Codes in Perl

## Query Country Information using Internal Database

```

use OLE;
$code = Win32::OLECreateObject('IP2Location.Country',$ipObj) or
die "Couldn't create new ipObj Object!";
# initialize IP2Location Component
if ($ipObj->Initialize("demo") <> "OK")
{
    # initialization failed
}
# it will return "US"
$result = ipObj->LookupShortName("205.188.145.215");
# it will return "United States"
$result = ipObj->LookupFullName("205.188.145.215");
# it will return "Japan"
$result = ipObj->CountryCodeToName("JP");
# it will return the version number
$result = ipObj->Version();
# free object
$ipObj->Quit();

```

## Query Other Information using External Database

```

use OLE;
$code = Win32::OLECreateObject('IP2Location.Country',$ipObj) or
die "Couldn't create new ipObj Object!";
# initialize IP2Location Component
if ($ipObj->Initialize("demo") <> "OK")
{
    # initialization failed
}
$result = ipObj->Lookup('IP2LOCATION-LITE-DB1.BIN','15.188.145.215');
$result = ipObj->CountryShort();
$result = ipObj->CountryLong();
$result = ipObj->Region();
$result = ipObj->City();
$result = ipObj->Latitude();
$result = ipObj->Longitude();
$result = ipObj->ZIPCode();
$result = ipObj->ISP();
$result = ipObj->Domain();
$result = ipObj->TimeZone();
$result = ipObj->IDDCode();
$result = ipObj->AreaCode();
$result = ipObj->WeatherStationCode();
$result = ipObj->WeatherStationName();
$result = ipObj->MCC();
$result = ipObj->MNC();
$result = ipObj->MobileBrand();
$result = ipObj->Elevation();
$result = ipObj->UsageType();
# free object
$ipObj->Quit();

```

## 5. IMPLEMENTATION

### 5.1 IP2Location™ ActiveX Method Reference

Method	Description
Initialize ( RegCode As String )	Initialize component and apply registration code to remove 5-seconds query delay. Please use "demo" for evaluation purpose.
CountryCodeToName (Code As String)	Return full country name by country code. Refer Appendix I.
LookUpFullName ( IPAddress As String)	Return full country name of a specified IP address using Internal IP database.
LookUpShortName ( IPAddress As String)	Return country code (two letters) of a specified IP address using Internal IP database. Refer Appendix I.
Version ()	Return component version.
IsRegistered()	Return true if component have been registered correctly and n delay in query.
LookUp ( BINPath As String, IPAddress As String)	Query external IP2Location BIN database as in BINPath and stores all values in Object Properties.

### 5.2 IP2Location™ ActiveX Property Reference

Property	Type	Description
IPAddress	String	IP Address.
CountryShort	String	Two-character country code based on ISO 3166.
CountryLong	String	Country name based on ISO 3166.
Region	String	Region or state name.
City	String	City name.
Latitude	String	City latitude.
Longitude	String	City longitude.
ZIPCode	String	United States ZIP code.
ISP	String	Internet Service Provider (ISP) name.
Domain	String	Internet domain name associated to IP address range.
Timezone	String	Time zone of city
NetSpeed	String	Connection speed

IDDCode	String	The IDD prefix to call the city from another country
AreaCode	String	A varying length number assigned to geographic areas for call between cities
WeatherStationCode	String	Special code to identify the nearest weather observation station
WeatherStationName	String	Name of the nearest weather observation station
MCC	String	Mobile country code
MNC	String	Mobile network code
MobileBrand	String	Mobile brand
Elevation	String	Elevation in meters
UsageType	String	Usage type of ISP or organization

\*Note : The result will return a not supported field message if the field of the result is not supported by the current database.

## 6. LICENSE PURCHASE

### 6.1 License Information

A license is required for each machine the product is installed on, including development or staging machines. Please refer to the **License Agreement** section for the complete license agreement.

Trial/Demo license is only meant to be used for the evaluation purpose.

### 6.2 Purchase Instruction

Please visit <https://www.ip2location.com/software/active-x> and select the package that you wish to purchase. Fill in the online form and complete the payment transaction. Once your order has been approved, you will receive your license file (License.Key) immediately through email. Initialize the ActiveX component with the registration code in your script will remove the 5-seconds query delay.

For example, if the registration code is [demo@yahoo.com-JGSVXgzjdDNS](#)

#### ASP

```
Set ipObj = Server.CreateObject("IP2Location.Country")
result = ipObj.Initialize("demo@yahoo.com-JGSVXgzjdDNS")
```

```
...
Set ipObj = Nothing
```

## **PHP**

```
$ipObj = new COM("IP2Location.Country");
$result = $ipObj->Initialize(demo@yahoo.com-JGSVXgzjdDNS);
...
$ipObj = null;
```

# 7. UPDATE COMPONENT

## 7.1 Update Component Internal Database

The IP2Location™ ActiveX/COM Component depends on an internal database for lookup purpose. All users with valid license will be allowed to download the monthly updates from the IP2Location™ download area during subscription period. To update, users need to replace the BIN database with the latest one from the download area. The component will use the latest database once the local copy has been updated. Please refer to the welcome email for more information regarding download updates and account information.

## 7.2 Upgrade Component

To upgrade a new version of IP-Country database, you can overwrite the IP2Location™ ActiveX DLL Component after uninstalling the old component.

**Note:** You need to stop the IIS web server before overwrite your old component.

First, stop the Web server

```
C:\>net stop iisadmin /y
```

The, un-register the component and update it with the latest component.

```
C:\>regsvr32 c:\windows\system32\ip2location.dll /u
C:\>copy c:\new\ip2location.dll c:\windows\system32\ip2location.dll
```

NOTE: In this case, "c:\new\ip2location.dll" is the latest version of component. Meanwhile "c:\path\ip2location.dll" is the destination. We assume the installation path is c:\windows\system32.

Finally, register component and restart the web server.

```
C:\>regsvr32 c:\windows\system32\ip2location.dll
C:\>net start w3svc
```

## 7.3 Uninstall Component

To uninstall IP2Location™ ActiveX/COM DLL Component, you need to stop and restart the IIS services to remove the old version of the DLL from the memory, as follows:

```
C:\>net stop iisadmin /
```

Then

```
C:\>regsvr32 c:\windows\system32\ip2location.dll /u
C:\>delete c:\windows\system32\ip2location.dll
```

```
C:\>net start w3svc
```

## 8. APPENDIX I: ISO3166 COUNTRY CODE

This table lists all valid ISO3166 two characters country codes that returns from IP2Location™ ActiveX/COM component and explain the full country names for these country codes.

Country Code	Country Name
AD	ANDORRA
AE	UNITED ARAB EMIRATES
AF	AFGHANISTAN
AG	ANTIGUA AND BARBUDA
AI	ANGUILLA
AL	ALBANIA
AM	ARMENIA
AO	ANGOLA
AQ	ANTARCTICA
AR	ARGENTINA
AS	AMERICAN SAMOA
AT	AUSTRIA
AU	AUSTRALIA
AW	ARUBA
AX	ÅLAND ISLANDS
AZ	AZERBAIJAN
BA	BOSNIA AND HERZEGOWINA
BB	BARBADOS
BD	BANGLADESH
BE	BELGIUM
BF	BURKINA FASO
BG	BULGARIA
BH	BAHRAIN
BI	BURUNDI
BJ	BENIN
BL	SAINT BARTHÉLEMY
BM	BERMUDA

Country Code	Country Name
BN	BRUNEI DARUSSALAM
BO	BOLIVIA
BQ	BONAIRE, SINT EUSTATIUS AND SABA
BR	BRAZIL
BS	BAHAMAS
BT	BHUTAN
BV	BOUVET ISLAND
BW	BOTSWANA
BY	BELARUS
BZ	BELIZE
CA	CANADA
CC	COCOS (KEELING) ISLANDS
CD	CONGO, THE DEMOCRATIC REPUBLIC OF THE
CF	CENTRAL AFRICAN REPUBLIC
CG	CONGO
CH	SWITZERLAND
CI	CÔTE D'IVOIRE
CK	COOK ISLANDS
CL	CHILE
CM	CAMEROON
CN	CHINA
CO	COLOMBIA
CR	COSTA RICA
CU	CUBA
CV	CAPE VERDE
CW	CURAÇAO
CX	CHRISTMAS ISLAND
CY	CYPRUS
CZ	CZECH REPUBLIC
DE	GERMANY
DJ	DJIBOUTI
DK	DENMARK
DM	DOMINICA
DO	DOMINICAN REPUBLIC
DZ	ALGERIA
EC	ECUADOR
EE	ESTONIA
EG	EGYPT
EH	WESTERN SAHARA
ER	ERITREA
ES	SPAIN
ET	ETHIOPIA
FI	FINLAND
FJ	FIJI
FK	FALKLAND ISLANDS (MALVINAS)
FM	MICRONESIA, FEDERATED STATES OF

Country Code	Country Name
FO	FAROE ISLANDS
FR	FRANCE
GA	GABON
GB	GREAT BRITAIN
GD	GRENADA
GE	GEORGIA
GF	FRENCH GUIANA
GG	GUERNSEY
GH	GHANA
GI	GIBRALTAR
GL	GREENLAND
GM	GAMBIA
GN	GUINEA
GP	GADELOUPE
GQ	EQUATORIAL GUINEA
GR	GREECE
GS	SOUTH GEORGIA & SOUTH SANDWICH ISLANDS
GT	GUATEMALA
GU	GUAM
GW	GUINEA-BISSAU
GY	GUYANA
HK	HONG KONG
HM	HEARD ISLAND AND MCDONALD ISLANDS
HN	HONDURAS
HR	CROATIA
HT	HAITI
HU	HUNGARY
ID	INDONESIA
IE	IRELAND
IL	ISRAEL
IM	ISLE OF MAN
IN	INDIA
IO	BRITISH INDIAN OCEAN TERRITORY
IQ	IRAQ
IR	IRAN, ISLAMIC REPUBLIC OF
IS	ICELAND
IT	ITALY
JE	JERSEY
JM	JAMAICA
JO	JORDAN
JP	JAPAN
KE	KENYA
KG	KYRGYZSTAN
KH	CAMBODIA
KI	KIRIBATI
KM	COMOROS



Country Code	Country Name
KN	SAINT KITTS AND NEVIS
KP	KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF
KR	KOREA, REPUBLIC OF
KW	KUWAIT
KY	CAYMAN ISLANDS
KZ	KAZAKSTAN
LA	LAO PEOPLE'S DEMOCRATIC REPUBLIC
LB	LEBANON
LC	SAINT LUCIA
LI	LIECHTENSTEIN
LK	SRI LANKA
LR	LIBERIA
LS	LESOTHO
LT	LITHUANIA
LU	LUXEMBOURG
LV	LATVIA
LY	LIBYA
MA	MOROCCO
MC	MONACO
MD	MOLDOVA, REPUBLIC OF
ME	MONTENEGRO
MF	SAINT MARTIN (FRENCH PART)
MG	MADAGASCAR
MH	MARSHALL ISLANDS
MK	MACEDONIA, THE FORMER YUGOSLAV
ML	MALI
MM	MYANMAR
MN	MONGOLIA
MO	MACAU
MP	NORTHERN MARIANA ISLANDS
MQ	MARTINIQUE
MR	MAURITANIA
MS	MONTserrat
MT	MALTA
MU	MAURITIUS
MV	MALDIVES
MW	MALAWI
MX	MEXICO
MY	MALAYSIA
MZ	MOZAMBIQUE
NA	NAMIBIA
NC	NEW CALEDONIA
NE	NIGER
NF	NORFOLK ISLAND
NG	NIGERIA
NI	NICARAGUA

Country Code	Country Name
NL	NETHERLANDS
NO	NORWAY
NP	NEPAL
NR	NAURU
NU	NIUE
NZ	NEW ZEALAND
OM	OMAN
PA	PANAMA
PE	PERU
PF	FRENCH POLYNESIA
PG	PAPUA NEW GUINEA
PH	PHILIPPINES
PK	PAKISTAN
PL	POLAND
PM	SAINT PIERRE AND MIQUELON
PN	PITCAIRN
PR	PUERTO RICO
PS	PALESTINIAN, STATE OF
PT	PORTUGAL
PW	PALAU
PY	PARAGUAY
QA	QATAR
RE	RÉUNION
RO	ROMANIA
RS	SERBIA
RU	RUSSIAN FEDERATION
RW	RWANDA
SA	SAUDI ARABIA
SB	SOLOMON ISLANDS
SC	SEYCHELLES
SD	SUDAN
SE	SWEDEN
SG	SINGAPORE
SH	SAINT HELENA, ASCENSION AND TRISTAN DA CUNHA
SI	SLOVENIA
SJ	SVALBARD AND JAN MAYEN
SK	SLOVAKIA
SL	SIERRA LEONE
SM	SAN MARINO
SN	SENEGAL
SO	SOMALIA
SR	SURINAME
SS	SOUTH SUDAN
ST	SAO TOME AND PRINCIPE
SV	EL SALVADOR
SX	SINT MAARTEN (DUTCH PART)

Country Code	Country Name
SY	SYRIAN ARAB REPUBLIC
SZ	SWAZILAND
TC	TURKS AND CAICOS ISLANDS
TD	CHAD
TF	FRENCH SOUTHERN TERRITORIES
TG	TOGO
TH	THAILAND
TJ	TAJIKISTAN
TK	TOKELAU
TL	TIMOR-LESTE
TM	TURKMENISTAN
TN	TUNISIA
TO	TONGA
TR	TURKEY
TT	TRINIDAD AND TOBAGO
TV	TUVALU
TW	TAIWAN, PROVINCE OF CHINA
TZ	TANZANIA, UNITED REPUBLIC OF
UA	UKRAINE
UG	UGANDA
UK	UNITED KINGDOM
UM	UNITED STATES MINOR OUTLYING ISLANDS
US	UNITED STATES
UY	URUGUAY
UZ	UZBEKISTAN
VA	HOLY SEE (VATICAN CITY STATE)
VC	SAINT VINCENT AND THE GRENADINES
VE	VENEZUELA
VG	VIRGIN ISLANDS, BRITISH
VI	VIRGIN ISLANDS, U.S.
VN	VIET NAM
VU	VANUATU
WF	WALLIS AND FUTUNA
WS	SAMOA
YE	YEMEN
YT	MAYOTTE
ZA	SOUTH AFRICA
ZM	ZAMBIA
ZW	ZIMBABWE

Copyright © 2017 Hexasoft Development Sdn. Bhd., All Rights Reserved.

**Hexasoft Development Sdn. Bhd.**

1-2-15 Mayang Mall Kompleks,  
Jalan Mayang Pasir 1,  
11950 Bayan Baru,  
Pulau Pinang,  
Malaysia.

Email: [sales@ip2location.com](mailto:sales@ip2location.com)