ZIPCodeWorld™ United States ZIP Code Web Service

1. Overview

This documentation provides a basic understanding and information to help you get started with our products. Look over this documentation to gain a high-level understanding of the process flow that underlies the ZIPCodeWorld United States ZIP Code Web Service.

For more information, please visit http://www.fraudlabs.com or contact your ZIPCodeWorld United States representative:

Email : sales@fraudlabs.com

1.1 Overview of the ZIPCodeWorld United States Web Service

ZIPCodeWorld United States delivers scalable web services solutions that help you to obtain geographical and other information about United States. We are able to identify the state, city name, city alias name, area code, county FIPS, latitude, longitude, time zone as well as other important information by using our affordable United States ZIP Codes database and technology.

ZIPCodeWorld United States ZIP Code Web Service is hosted, programmable XML Web Service that allows exchange of data between systems. It is hosted on redundant servers and 24x7x365 monitoring. Customers can integrate our web service regardless of their web server and other business solutions. This XML web service is used under authorized license to ZipCodeWorld.com, the global leader in postal code service industry, as well as the leading provider of postal code information to small businesses worldwide.

ZIPCodeWorld United States ZIP Code Web Service is very easy to set up and use as it uses platform independent XML format. You may get precise records by zipcodes geographical location within a minute of time.

Key Features Include:

- Pinpoints the precise records of location, with county name, city, county FIPS, latitude and longitude, and many more using ZIP Code lookups
- Provides a complete XML-based Web Services API
- Contains sample code examples for ease integration

1.2 Front End of ZIPCodeWorld United States ZIP Code Web Service

The general idea is that front end is responsible for collection input from the user and conforms to some specification that the back end can use. Front end of ZIPCodeWorld United States ZIP Code Web Service is rather simple to understand. As we are using platform independent XML format to exchange data between systems, you may either integrate our web service to your in-house system, or direct access to our hosted web service.

1.2.1. Integration with In-house System

I. Our sample codes in different languages are available at: http://www.fraudlabs.com/zipcodeworldUSsamplecodes.aspx . Log on and download sample codes that you need (For sample codes in different languages please refer to Appendix II). Below are links to get sample codes:

i. Microsoft ASP.NET - VB.NET: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientVB.zip

ii. Microsoft ASP.Net - C#: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientCSharp.zip

iii. Microsoft ASP 3.0: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientAsp.zip

iv. Java/ Apache: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientJava.zip

v. PHP: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientPHP.zip

vi. Python: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientPython.zip

vii. Perl: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientPerl.zip

viii. ColdFusion MX: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientColdFusion.zip

ix. Access 2000: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientAccess.zip

x. Excel 2000: http://www.fraudlabs.com/samplecode/ZipCodeWorldUS_WebServiceClientExcel.zip

II. Please go through 'readme' file that we provide together with the sample codes for more set up information

III. In order to use our service, you need to get your own license key – *How?*

i. log on to http://www.fraudlabs.com

ii. sign up as our registered member

iii. check your email to complete user activation
iv. get license key :
  a) Free License Account:
    1) at the left menu bar, under category of ZIPCodeWorld™ United States, click “free license”
    2) view Terms of Use (*please note that you must agree with our Terms of Use before proceed)
    3) click on “Get Free License Now”
    4) the license key will be sent to your email
  b) Premium Subscription:
    1) at the left menu bar, under category of ZIPCodeWorld™ United States, click “subscribe now”
    2) fill in required field in the secure payment form
    3) click on “make payment”
    4) the license key will be sent to your email after your payment is confirmed by our online payment merchant

IV. Now fill in the required field

V. Click on “Submit” and result is provided

1.2.2. Direct Access to Hosted Web Service

I. To use our web service directly, please log on to: http://ws.fraudlabs.com/ZIPCodeWorldUS_WebService.asmx

II. Click on “ZIPCodeWorld_US”

III. In order to use our service, you need to get your own license key – *How?*

i. log on to http://www.fraudlabs.com
ii. sign up as our registered member
iii. check your email to complete user activation
iv. get license key :
  a) Free License Account:
    1) at the left menu bar, under category of ZIPCodeWorld™ United States, click “free license”
    2) view Terms of Use (*please note that you must agree with our Terms of Use before proceed)
    3) click on “Get Free License Now”
    4) the license key will be sent to your email
  b) Premium Subscription:

1) at the left menu bar, under category of ZIPCodeWorld™ United States, click “subscribe now”
2) fill in required field in the secure payment form
3) click on “make payment”
4) the license key will be sent to your email after your payment is confirmed by our online payment merchant

IV. Now you fill in the particular field

V. Click on “Invoke” and result is provided

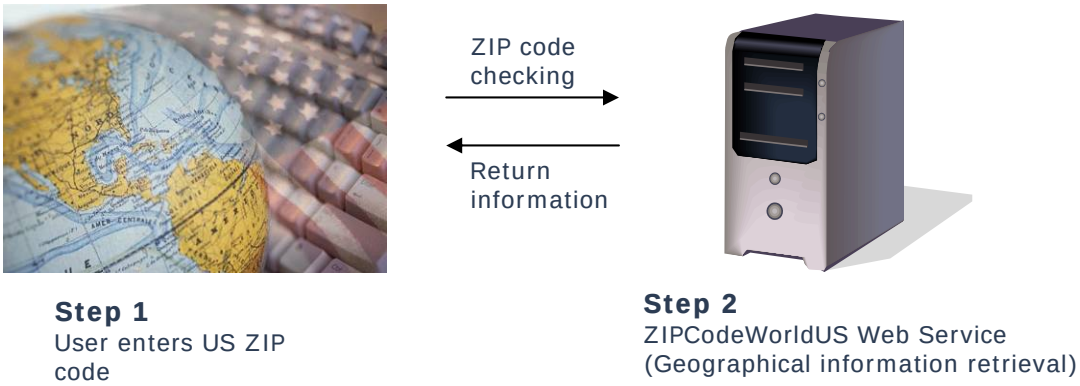
1.3 Back End of ZIPCodeWorld United States ZIP Code Web Service

Back end is the part that processes the input from the front end. The process is seamless to the end-user and works by interaction with a SOAP API through ZIPCodeWorld United States ZIP Code Web Service.

The process works as follows:

1. User enters ZIP code that to be searched
2. System verify the license key before proceed
3. If the license key is valid, it will proceed and check the credits availability. If the remaining credits still available, it will start to retrieve related information from database
4. Returns and display the precise information

1.4 Process Flow Overview



2. Implementation

This section provides basic information of the process of integrating the web service into your website. Look over this section to gain a high-level understanding of requesting ZIPCodeWorld United States ZIP Code Web Service.

The implementation section covers the following topics:

- ZIPCodeWorld_US Function

For those with solid SOAP programming knowledge, integration basics are included below. POST all requests to:
http://ws.fraudlabs.com/ZIPCodeWorldUS_WebService.asmx

For more information about ZIPCodeWorld United States ZIP Code Web Service implementation, please visit http://www.fraudlabs.com or contact your ZIPCodeWorld United States representative:

Email : sales@fraudlabs.com

2.1 Implementation Details

This section was created for those who wish to develop applications that make use of ZIPCodeWorld United States ZIP Code Web Service.

SOAP – ZIPCodeWorld United States Function Request

```
POST /ws.fraudlabs.com_non_ssl/ZipCodeWorldUS_WebService.asmx HTTP/1.1
Host: ws.fraudlabs.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://ws.fraudlabs.com/ZIPCodeWorld_US"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ZIPCodeWorld_US xmlns="http://ws.fraudlabs.com/">
      <ZIPCode>string</ZIPCode>
      <LICENSE>string</LICENSE>
    </ZIPCodeWorld_US>
  </soap:Body>
</soap:Envelope>
```

SOAP – ZIPCodeWorld United States Function Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ZIPCodeWorld_USResponse xmlns="http://ws.fraudlabs.com/">
      <ZIPCodeWorld_USResult>
        <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
        <ZIP_CODE>string</ZIP_CODE>
        <STATE>string</STATE>
        <CITY>string</CITY>
        <AREA_CODE>string</AREA_CODE>
        <CITY_ALIAS_ABBR>string</CITY_ALIAS_ABBR>
        <CITY_ALIAS_NAME>string</CITY_ALIAS_NAME>
        <STATE_FIPS>string</STATE_FIPS>
        <COUNTY_FIPS>string</COUNTY_FIPS>
        <LATITUDE>float</LATITUDE>
        <LONGITUDE>float</LONGITUDE>
        <CITY_TYPE>string</CITY_TYPE>
        <COUNTY_NAME>string</COUNTY_NAME>
```

```
        <TIME_ZONE>string</TIME_ZONE>
        <DAY_LIGHT_SAVING>string</DAY_LIGHT_SAVING>
        <ELEVATION>string</ELEVATION>
        <MSA2000>string</MSA2000>
        <PMSA>string</PMSA>
        <CBSA>string</CBSA>
        <CBSA_DIV>string</CBSA_DIV>
        <CBSA_TITLE>string</CBSA_TITLE>
        <PERSONS_PER_HOUSEHOLD>double</PERSONS_PER_HOUSEHOLD>
        <ZIPCODE_POPULATION>double</ZIPCODE_POPULATION>
        <COUNTIES_AREA>double</COUNTIES_AREA>
        <HOUSEHOLDS_PER_ZIPCODE>double</HOUSEHOLDS_PER_ZIPCODE>
        <WHITE_POPULATION>double</WHITE_POPULATION>
        <BLACK_POPULATION>double</BLACK_POPULATION>
        <HISPANIC_POPULATION>double</HISPANIC_POPULATION>
        <INCOME_PER_HOUSEHOLD>double</INCOME_PER_HOUSEHOLD>
        <AVERAGE_HOUSE_VALUE>double</AVERAGE_HOUSE_VALUE>
        <MESSAGE>string</MESSAGE>
      </ZIPCodeWorld_USResult>
    </ZIPCodeWorld_USResponse>
  </soap:Body>
</soap:Envelope>
```

HTTP GET- ZIPCodeWorld United States Function Request

```
GET
/ws.fraudlabs.com_non_ssl/ZIPCodeWorldUS_WebService.asmx/ZIPCodeWorld_U
S?ZIPCode=string&LICENSE=string HTTP/1.1
Host: ws.fraudlabs.com
```

HTTP GET- ZIPCodeWorld United States Function Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<ZIPCODEWORLD_US xmlns="http://ws.fraudlabs.com/">
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <ZIP_CODE>string</ZIP_CODE>
  <STATE>string</STATE>
  <CITY>string</CITY>
  <AREA_CODE>string</AREA_CODE>
  <CITY_ALIAS_ABBR>string</CITY_ALIAS_ABBR>
  <CITY_ALIAS_NAME>string</CITY_ALIAS_NAME>
  <STATE_FIPS>string</STATE_FIPS>
  <COUNTY_FIPS>string</COUNTY_FIPS>
  <LATITUDE>float</LATITUDE>
  <LONGITUDE>float</LONGITUDE>
  <CITY_TYPE>string</CITY_TYPE>
  <COUNTY_NAME>string</COUNTY_NAME>
  <TIME_ZONE>string</TIME_ZONE>
```

```
  <DAY_LIGHT_SAVING>string</DAY_LIGHT_SAVING>
  <ELEVATION>string</ELEVATION>
  <MSA2000>string</MSA2000>
  <PMSA>string</PMSA>
  <CBSA>string</CBSA>
  <CBSA_DIV>string</CBSA_DIV>
  <CBSA_TITLE>string</CBSA_TITLE>
  <PERSONS_PER_HOUSEHOLD>double</PERSONS_PER_HOUSEHOLD>
  <ZIPCODE_POPULATION>double</ZIPCODE_POPULATION>
  <COUNTIES_AREA>double</COUNTIES_AREA>
  <HOUSEHOLDS_PER_ZIPCODE>double</HOUSEHOLDS_PER_ZIPCODE>
  <WHITE_POPULATION>double</WHITE_POPULATION>
  <BLACK_POPULATION>double</BLACK_POPULATION>
  <HISPANIC_POPULATION>double</HISPANIC_POPULATION>
  <INCOME_PER_HOUSEHOLD>double</INCOME_PER_HOUSEHOLD>
  <AVERAGE_HOUSE_VALUE>double</AVERAGE_HOUSE_VALUE>
  <MESSAGE>string</MESSAGE>
</ZIPCODEWORLD_US>
```

HTTP POST- ZIPCodeWorld United States Function Request

```
POST
/ws.fraudlabs.com_non_ssl/ZIPCodeWorldUS_WebService.asmx/ZIPCodeWorld_U
S HTTP/1.1
Host: ws.fraudlabs.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

ZIPCode=string&LICENSE=string
```

HTTP POST- ZIPCodeWorld United States Function Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<ZIPCODEWORLD_US xmlns="http://ws.fraudlabs.com/">
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <ZIP_CODE>string</ZIP_CODE>
  <STATE>string</STATE>
  <CITY>string</CITY>
  <AREA_CODE>string</AREA_CODE>
  <CITY_ALIAS_ABBR>string</CITY_ALIAS_ABBR>
  <CITY_ALIAS_NAME>string</CITY_ALIAS_NAME>
  <STATE_FIPS>string</STATE_FIPS>
  <COUNTY_FIPS>string</COUNTY_FIPS>
  <LATITUDE>float</LATITUDE>
  <LONGITUDE>float</LONGITUDE>
  <CITY_TYPE>string</CITY_TYPE>
  <COUNTY_NAME>string</COUNTY_NAME>
  <TIME_ZONE>string</TIME_ZONE>
  <DAY_LIGHT_SAVING>string</DAY_LIGHT_SAVING>
```

```
  <ELEVATION>string</ELEVATION>
  <MSA2000>string</MSA2000>
  <PMSA>string</PMSA>
  <CBSA>string</CBSA>
  <CBSA_DIV>string</CBSA_DIV>
  <CBSA_TITLE>string</CBSA_TITLE>
  <PERSONS_PER_HOUSEHOLD>double</PERSONS_PER_HOUSEHOLD>
  <ZIPCODE_POPULATION>double</ZIPCODE_POPULATION>
  <COUNTIES_AREA>double</COUNTIES_AREA>
  <HOUSEHOLDS_PER_ZIPCODE>double</HOUSEHOLDS_PER_ZIPCODE>
  <WHITE_POPULATION>double</WHITE_POPULATION>
  <BLACK_POPULATION>double</BLACK_POPULATION>
  <HISPANIC_POPULATION>double</HISPANIC_POPULATION>
  <INCOME_PER_HOUSEHOLD>double</INCOME_PER_HOUSEHOLD>
  <AVERAGE_HOUSE_VALUE>double</AVERAGE_HOUSE_VALUE>
  <MESSAGE>string</MESSAGE>
</ZIPCODEWORLD_US>
```

A WSDL is available at:
http://ws.fraudlabs.com/ZIPCodeWorldUS_WebService.asmx?wsdl

A description of the ZIPCodeWorldUS_WebService operation is available at:
http://ws.fraudlabs.com/ZIPCodeWorldUS_WebService.asmx?op=ZIPCodeWorld_US

2.2 Basic Parameters - Input

Field	Format	Description
ZIPCode	Required	US ZIP Code
LICENSE	Required	License key for free license and premium users.

2.3 Basic Parameters - Output

Field	Format	Description
ZIP_CODE	string	The ZIP Code of the city designated by the USPS.
CITY	string	The name of the city designated by the USPS.
STATE	string	The two-letter abbreviation for the state to that particular ZIP code.
AREA_CODE	string	The telephone area code belonging to the particular postal code is located.
CITY_ALIAS_NAME	string	The alias name for the ZIP Code. This July be a city name, postal entity, community or a municipality name.
CITY_ALIAS_ABBR	string	This is a standard 13-character abbreviation for city alias name. This field is empty if

		names are less or equal to 13-character.
CITY_TYPE	string	The field indicates the type of locale identified in the city alias name such as Post Office, station or branch.
COUNTY_NAME	string	The name of the county or parish in which the 5-digit ZIP Code resides.
STATE_FIPS	string	The FIPS code assigned to state.
COUNTY_FIPS	string	The FIPS code assigned to given county or parish within a state.
TIME_ZONE	string	The time zone number represents by the hours past the Greenwich Time Zone.
DAY_LIGHT_SAVING	string	This flag indicates whether the city observe day light saving.
LATITUDE	float	The geographic coordinate of a point measured in degrees east or west of the Greenwich meridian.
LONGITUDE	float	The geographic coordinate of a point measured in degrees east or west of the Greenwich meridian.
ELEVATION	double	The counties average elevation.
MSA2000	string	Metropolitan Statistical Area based on Census 2000
PMSA	string	Primary Metropolitan Statistical Area
CBSA	string	Core Based Statistical Area, Metropolitan Statistical Area and Micropolitan Statistical Area released by OMB in Year 2003
CBSA_DIV	string	Metropolitan Division
CBSA_TITLE	string	Full title description of CBSA and MSA
PERSONS_PER_HOUSEHOLD	double	The average amount of persons per Household based on Census 2000.
ZIPCODE_POPULATION	double	The population of the ZIP Code based on Census 2000.
COUNTIES_AREA	double	The counties square mileage based on Census 2000.
HOUSEHOLDS_PER_ZIPCODE	double	Total estimated amount of households per ZIP Code based on Census 2000.
WHITE_POPULATION	double	The estimate of White population per ZIP code based on Census 2000.
BLACK_POPULATION	double	The estimate of Black population per ZIP code based on Census 2000.
HISPANIC_POPULATION	double	The estimate of Hispanic population per ZIP code based on Census 2000.
INCOME_PER_HOUSEHOLD	double	The average household income within county based on Census 2000.
AVERAGE_HOUSE_VALUE	double	The average house value within county based on Census 2000.
CREDITSAVAILABLE	Integer	Number of queries remaining in your account, can be used to alert you when you may need to add more queries to your account.
MESSAGE	String	Web Service Message Response

2.4 List of possible value for MESSAGE field

ZIPCodeWorld Web Service Error Message
• Invalid license key
• Zip Code xx not found in database.Please try again.
• ZIP code and License Key cannot be blank
• No credit available
• License key was expired

2.5 Structure of the Request and Response

2.5.1. Request by SOAP

A valid request posted from user should look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ZIPCodeWorld_US xmlns="http://ws.fraudlabs.com/">
      <ZIPCode>00501</ZIPCode>
      <LICENSE>xx-xxxx-xxxx</LICENSE>
    </ZIPCodeWorld_US >
  </soap:Body>
</soap:Envelope>
```

2.5.2. Response by SOAP

After posting a valid request, ZIPCodeWorld_US will return several parameters. An example of a response returned ZIPCodeWorld_US is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ZIPCodeWorld_USResponse xmlns="http://ws.fraudlabs.com/">
      <ZIPCodeWorld_USResult>
        <CREDITSAVAILABLE>4963</CREDITSAVAILABLE>
        <ZIP_CODE>00501</ZIP_CODE>
        <STATE>NY</STATE>
        <CITY>HOLTSVILLE</CITY>
        <AREA_CODE>631</AREA_CODE>
        <CITY_ALIAS_ABBR/>
        <CITY_ALIAS_NAME>Gholtsville / I R S SERVICE
CENTER</CITY_ALIAS_NAME>
        <STATE_FIPS>36</STATE_FIPS>
        <COUNTY_FIPS>103</COUNTY_FIPS>
```

```
        <LATITUDE>40.81518</LATITUDE>
        <LONGITUDE>-73.0455</LONGITUDE>
        <CITY_TYPE>P / O</CITY_TYPE>
        <COUNTY_NAME>SUFFOLK</COUNTY_NAME>
        <TIME_ZONE>5</TIME_ZONE>
        <DAY_LIGHT_SAVING>Y</DAY_LIGHT_SAVING>
        <ELEVATION>25</ELEVATION>
        <MSA2000>5602</MSA2000>
        <PMSA>5380</PMSA>
        <CBSA>35620</CBSA>
        <CBSA_DIV>35004</CBSA_DIV>
        <CBSA_TITLE>NEW YORK-NORTHERN NEW JERSEY-LONG ISLAND, NY-NJ-PA
METROPOLITAN STATISTICAL AREA</CBSA_TITLE>
        <PERSONS_PER_HOUSEHOLD>3.11</PERSONS_PER_HOUSEHOLD>
        <ZIPCODE_POPULATION>0</ZIPCODE_POPULATION>
        <COUNTIES_AREA>911</COUNTIES_AREA>
        <HOUSEHOLDS_PER_ZIPCODE>0</HOUSEHOLDS_PER_ZIPCODE>
        <WHITE_POPULATION>0</WHITE_POPULATION>
        <BLACK_POPULATION>0</BLACK_POPULATION>
        <HISPANIC_POPULATION>0</HISPANIC_POPULATION>
        <INCOME_PER_HOUSEHOLD>0</INCOME_PER_HOUSEHOLD>
        <AVERAGE_HOUSE_VALUE>0</AVERAGE_HOUSE_VALUE>
        <MESSAGE/>
      </ZIPCodeWorld_USResult>
    </ZIPCodeWorld_USResponse>
  </soap:Body>
</soap:Envelope>
```

2.5.3. Request by HTTP-GET

A valid request posted by user:

```
GET
/ws.fraudlabs.com_non_ssl/ZIPCodeWorldUS_WebService.asmx/ZIPCodeWorld_U
S?ZIPCode=00501&LICENSE=XX-XXXX-XXXX HTTP/1.1
Host: ws.fraudlabs.com
```

2.5.4. Response by HTTP-GET

An example of a response returned ZIPCodeWorld_US is shown below:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<ZIPCodeWorld_US xmlns="http://ws.fraudlabs.com/">
   <CREDITSAVAILABLE>4963</CREDITSAVAILABLE>
   <ZIP_CODE>00501</ZIP_CODE>
   <STATE>NY</STATE>
   <CITY>HOLTSVILLE</CITY>
   <AREA_CODE>631</AREA_CODE>
   <CITY_ALIAS_ABBR/>
```

```
   <CITY_ALIAS_NAME>Gholtsville / I R S SERVICE
CENTER</CITY_ALIAS_NAME>
   <STATE_FIPS>36</STATE_FIPS>
   <COUNTY_FIPS>103</COUNTY_FIPS>
   <LATITUDE>40.81518</LATITUDE>
   <LONGITUDE>-73.0455</LONGITUDE>
   <CITY_TYPE>P / O</CITY_TYPE>
   <COUNTY_NAME>SUFFOLK</COUNTY_NAME>
   <TIME_ZONE>5</TIME_ZONE>
   <DAY_LIGHT_SAVING>Y</DAY_LIGHT_SAVING>
   <ELEVATION>25</ELEVATION>
   <MSA2000>5602</MSA2000>
   <PMSA>5380</PMSA>
   <CBSA>35620</CBSA>
   <CBSA_DIV>35004</CBSA_DIV>
   <CBSA_TITLE>NEW YORK-NORTHERN NEW JERSEY-LONG ISLAND, NY-NJ-PA
METROPOLITAN STATISTICAL AREA</CBSA_TITLE>
   <PERSONS_PER_HOUSEHOLD>3.11</PERSONS_PER_HOUSEHOLD>
   <ZIPCODE_POPULATION>0</ZIPCODE_POPULATION>
   <COUNTIES_AREA>911</COUNTIES_AREA>
   <HOUSEHOLDS_PER_ZIPCODE>0</HOUSEHOLDS_PER_ZIPCODE>
   <WHITE_POPULATION>0</WHITE_POPULATION>
   <BLACK_POPULATION>0</BLACK_POPULATION>
   <HISPANIC_POPULATION>0</HISPANIC_POPULATION>
   <INCOME_PER_HOUSEHOLD>0</INCOME_PER_HOUSEHOLD>
   <AVERAGE_HOUSE_VALUE>0</AVERAGE_HOUSE_VALUE>
   <MESSAGE/>
</ZIPCodeWorld_US>
```

2.5.5. Request by HTTP-POST

A valid request posted by user:

```
POST
/ws.fraudlabs.com_non_ssl/ZIPCodeWorldUS_WebService.asmx/ZIPCodeWorld_U
S HTTP/1.1
Host: ws.fraudlabs.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

ZIPCode=00501&LICENSE=XX-XXXX-XXXX
```

2.5.6. Response by HTTP-POST

An example of a response returned by ZIPCodeWorld_US is shown below:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<ZIPCodeWorld_US xmlns="http://ws.fraudlabs.com/">
   <CREDITSAVAILABLE>4963</CREDITSAVAILABLE>
```

```
   <ZIP_CODE>00501</ZIP_CODE>
   <STATE>NY</STATE>
   <CITY>HOLTSVILLE</CITY>
   <AREA_CODE>631</AREA_CODE>
   <CITY_ALIAS_ABBR/>
   <CITY_ALIAS_NAME>Gholtsville / I R S SERVICE
CENTER</CITY_ALIAS_NAME>
   <STATE_FIPS>36</STATE_FIPS>
   <COUNTY_FIPS>103</COUNTY_FIPS>
   <LATITUDE>40.81518</LATITUDE>
   <LONGITUDE>-73.0455</LONGITUDE>
   <CITY_TYPE>P / O</CITY_TYPE>
   <COUNTY_NAME>SUFFOLK</COUNTY_NAME>
   <TIME_ZONE>5</TIME_ZONE>
   <DAY_LIGHT_SAVING>Y</DAY_LIGHT_SAVING>
   <ELEVATION>25</ELEVATION>
   <MSA2000>5602</MSA2000>
   <PMSA>5380</PMSA>
   <CBSA>35620</CBSA>
   <CBSA_DIV>35004</CBSA_DIV>
   <CBSA_TITLE>NEW YORK-NORTHERN NEW JERSEY-LONG ISLAND, NY-NJ-PA
METROPOLITAN STATISTICAL AREA</CBSA_TITLE>
   <PERSONS_PER_HOUSEHOLD>3.11</PERSONS_PER_HOUSEHOLD>
   <ZIPCODE_POPULATION>0</ZIPCODE_POPULATION>
   <COUNTIES_AREA>911</COUNTIES_AREA>
   <HOUSEHOLDS_PER_ZIPCODE>0</HOUSEHOLDS_PER_ZIPCODE>
   <WHITE_POPULATION>0</WHITE_POPULATION>
   <BLACK_POPULATION>0</BLACK_POPULATION>
   <HISPANIC_POPULATION>0</HISPANIC_POPULATION>
   <INCOME_PER_HOUSEHOLD>0</INCOME_PER_HOUSEHOLD>
   <AVERAGE_HOUSE_VALUE>0</AVERAGE_HOUSE_VALUE>
   <MESSAGE/>
</ZIPCodeWorld_US>
```

3. Design Information

This section provides suggestion regarding the placing of ZIPCodeWorld United States ZIP Code Web Service into Web pages. Look over this section to get a general idea for adding this service to your website.

3.1 Placement of ZIPCodeWorld United States ZIP Code Web Service

As ZIPCodeWorld United States ZIP Code Web Service provides you very informative details by using ZIP code, this service can be placed on any forms on your website, offering a seamless integration.

Description: Geographical information is needed for business purpose or even for online transaction in United States

Level of security: High

Site Type(s):
- e-Commerce Solutions
- Internet Retailers
- Software Developers

Benefits:
- Get information from affordable and complete database which contains over 76,000 precise records
- With our ZIPCodeWorld database, applications can be built easily to support business logics
- Obtain additional information on United States addressing system and any other aspect of postal operations
- Enhance customer's information in United States
- Improve data entry speed and accuracy

Appendix I : Data File Spesification

1. ZIP_CODE
A five-digit code (00000-999999) which identifies a specific geographic delivery area. ZIP also known as “Zone Improvement Plan”. ZIP Codes represent an area within a state or a single building or company with high mail volume. In very rare case, a ZIP Code shares different states boundaries.

2. CITY
This is the name of USPS preferred city. It is used at the last line of postal address. The name can represent a city, a postal entity, a community or a municipality name.

3. STATE
This field use 2-letter abbreviation to represent the name of the state or US Territory. For examples, AZ represents Arizona and CA represents California.

4. AREA_CODE
The telephone area code belonging to the particular ZIP code is located.

5. CITY_ALIAS_NAME
This may be the name of a city, a postal entity, a community or a city alternate name. There may be more than one alias names for a ZIP Code.

6. CITY_ALIAS_ABBR
If the CITY_ALIAS_NAME is more than 13-character long, this field represents a 13-character abbreviation for the CITY_ALIAS_NAME. This field is empty if otherwise.

7. CITY_TYPE
This field indicates the type of locale in field CITY_ALIAS_NAME. Refer to table for all city types.

CODE	DESCRIPTION
A	Airport Mail Facility (AMF)
B	Branch
C	Community Post Office (CPO)
D	Area Distribution Center (ASC)
E	Sectional Center Facility (SCF)
F	Delivery Distribution Center (DDC)
G	General Mailing Facility (GMF)
K	Bulk Mail Center (BMC)
M	Money Order Unit
N	Non Postal Community Name (Acceptable by USPS)
O	Non Postal Community Name (Not Acceptable by USPS)
P	Post Office
S	Station

U	Urbanization
V	Vehicle Maintenance Facility

8. COUNTY_NAME
The name of the county or parish in which the 5-digit ZIP Code resides.

9. STATE_FIPS
The FIPS code assigned to US state. No value for military state code such as AA, AE and AP.

10. COUNTY_FIPS
The FIPS code assigned to COUNTY_NAME within a state. In Alaska, it identifies a region within the state.

11. TIME_ZONE
This number represents the hours past the Greenwich Meridian Time Zone (GMT).

NUMBER	TIME ZONE
4	Atlantic
5	Eastern
6	Central
7	Mountain
8	Pacific
9	Alaska
10	Hawaii-Aleutian
11	Samoa
14	Guam
15	Palau

12. DAY_LIGHT_SAVING
The flag represents whether the county observes day light saving during the summer. “Y” for YES; “N” for NO.

13. LATITUDE
The geographical coordinate of a centroid measured in degree north or south of the equator.

14. LONGITUDE
The geographical coordinate of a centroid measured in degree east or west of the Greenwich meridian.

15. ELEVATION
The counties average elevation.

16. MSA2000
Metropolitan Statistical Area used in Census 2000 (outdated).

17. PMSA
Primary Metropolitan Statistical Area.

18. CBSA
The Core Based Statistical Area (CBSA) are five-digit numeric codes assigned to the 935 metropolitan and micropolitan statistical areas defined by the Office of Management and Budget (OMB) effective from Census 2000.

19. CBSA_DIV
A CBSA containing a single core with a population of 2.5 million or more may be subdivided to form smaller groupings of counties referred to as "metropolitan divisions"

20. CBSA_TITLE
Full title description of Metropolitan/Micropolitan Statistical Area.

21. PERSONS_PER_HOUSEHOLD
The average amount of persons per Household based on Census 2000.

22. ZIPCODE_POPULATION
The population of the ZIP Code based on Census 2000.

23. COUNTIES_AREA
The counties square mileage based on Census 2000.

24. HOUSEHOLDS_PER_ZIPCODE
Total estimated amount of households per ZIP Code based on Census 2000.

25. WHITE_POPULATION
The estimate of White population per ZIP code based on Census 2000.

26. BLACK_POPULATION
The estimate of Black population per ZIP code based on Census 2000.

27. HISPANIC_POPULATION
The estimate of Hispanic population per ZIP code based on Census 2000.

28. INCOME_PER_HOUSEHOLD
The average household income within a county based on Census 2000.

29. AVERAGE_HOUSE_VALUE
The average house value in the county based on Census 2000.

Appendix II : Sample Code

ZIPCodeWorldUS Web Service sample code is available in several different programming languages. Below are some examples, for more different programming languages please log on to: http://www.fraudlabs.com/zipcodeworldUSsamplecodes.aspx

i. ASP.NET – VB.NET (SOAP)

```
Private Sub ZIPCodeWorldUSWebService()
  Dim x_ZIPCodeWorld As New
ZIPCodeWorldUS_WebService.ZIPCodeWorldUS_WebService
  Dim oZIPCodeWorld As New ZIPCODEWORLD_US
Try
  oZIPCodeWorld = x_ZIPCodeWorld.ZIPCodeWorld_US(Me.txtZipCode.Text,
Me.txtLicense.Text)

Me.txtResult.Text = "ZIP Code:" + oZIPCodeWorld.ZIP_CODE & vbNewLine
Me.txtResult.Text += "State:" + oZIPCodeWorld.STATE & vbNewLine
Me.txtResult.Text += "City:" + oZIPCodeWorld.CITY & vbNewLine
Me.txtResult.Text += "Area Code:" + oZIPCodeWorld.AREA_CODE & vbNewLine
Me.txtResult.Text += "City Alias ABBR:" + oZIPCodeWorld.CITY_ALIAS_ABBR
& vbNewLine
Me.txtResult.Text += "City Alias Name:" + oZIPCodeWorld.CITY_ALIAS_NAME
& vbNewLine
Me.txtResult.Text += "State FIPS:" + oZIPCodeWorld.STATE_FIPS &
vbNewLine
Me.txtResult.Text += "County FIPS:" + oZIPCodeWorld.COUNTY_FIPS &
vbNewLine
Me.txtResult.Text += "Latitude:" + oZIPCodeWorld.LATITUDE.ToString &
vbNewLine
Me.txtResult.Text += "Longitude:" + oZIPCodeWorld.LONGITUDE.ToString &
vbNewLine
Me.txtResult.Text += "City Type:" + oZIPCodeWorld.CITY_TYPE & vbNewLine
Me.txtResult.Text += "County Name:" + oZIPCodeWorld.COUNTY_NAME &
vbNewLine
Me.txtResult.Text += "Time Zone:" + oZIPCodeWorld.TIME_ZONE & vbNewLine
Me.txtResult.Text += "Day Light Saving:" +
oZIPCodeWorld.DAY_LIGHT_SAVING & vbNewLine
Me.txtResult.Text += "Elevation:" + oZIPCodeWorld.ELEVATION & vbNewLine
Me.txtResult.Text += "MSA2000:" + oZIPCodeWorld.MSA2000 & vbNewLine
Me.txtResult.Text += "PMSA:" + oZIPCodeWorld.PMSA & vbNewLine
Me.txtResult.Text += "CBSA:" + oZIPCodeWorld.CBSA & vbNewLine
Me.txtResult.Text += "CBSA DIV:" + oZIPCodeWorld.CBSA_DIV & vbNewLine
Me.txtResult.Text += "CBSA Title:" + oZIPCodeWorld.CBSA_TITLE &
vbNewLine
Me.txtResult.Text += "Persons per Household:" +
oZIPCodeWorld.PERSONS_PER_HOUSEHOLD.ToString & vbNewLine
Me.txtResult.Text += "ZIPCode Population:" +
oZIPCodeWorld.ZIPCODE_POPULATION.ToString & vbNewLine
Me.txtResult.Text += "Counties Area:" +
oZIPCodeWorld.COUNTIES_AREA.ToString & vbNewLine
Me.txtResult.Text += "Household per ZIPCode:" +
oZIPCodeWorld.HOUSEHOLDS_PER_ZIPCODE.ToString & vbNewLine
```

```
Me.txtResult.Text += "White Population:" +
oZIPCodeWorld.WHITE_POPULATION.ToString & vbNewLine
Me.txtResult.Text += "Black Population:" +
oZIPCodeWorld.BLACK_POPULATION.ToString & vbNewLine
Me.txtResult.Text += "Hispanic Population:" +
oZIPCodeWorld.HISPANIC_POPULATION.ToString & vbNewLine
Me.txtResult.Text += "Income per Household:" +
oZIPCodeWorld.INCOME_PER_HOUSEHOLD.ToString & vbNewLine
Me.txtResult.Text += "Average House Value:" +
oZIPCodeWorld.AVERAGE_HOUSE_VALUE.ToString & vbNewLine
Me.txtResult.Text += "Credits Available:" +
oZIPCodeWorld.CREDITSAVAILABLE & vbNewLine
Me.txtResult.Text += "Message:" + oZIPCodeWorld.MESSAGE & vbNewLine
   Catch ex As Exception
            Response.Write(ex.Message)
   End Try
End Sub
```

ii. ASP.NET – VB.NET (HTTP)

```
Private Sub Lookup()
 Dim x_ZIPCodeWorld As New
ZIPCodeWorldUS_WebService.ZIPCodeWorldUS_WebService
 Dim oZIPCodeWorld As New ZIPCODEWORLD_US

 Try
   Dim x_builder As String
   x_builder =
"http://ws.fraudlabs.com/ZIPCodeWorldUS_WebService.asmx/ZIPCodeWorld_US
?ZIPCODE=" + Me.txtZipCode.Text & "&LICENSE=" + Me.txtLicense.Text

   Dim x_web_client As WebClient = New WebClient
   Dim x_response() As Byte =
x_web_client.DownloadData(x_builder.ToString())

processResultString(Encoding.Default.GetString(x_response))
   Catch ex As Exception
     Response.Write(ex.Message)
   End Try End Sub

Private Sub processResultString(ByVal p_result As String)
   Dim indexOpen As Integer
   Dim indexClose As Integer
   Dim strParam As String

   strParam = "<ZIP_CODE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</ZIP_CODE>")
        Me.txtResult.Text = "ZIP Code: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<STATE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
```

```
        indexClose = p_result.IndexOf("</STATE>")
        Me.txtResult.Text += vbNewLine & "State: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<CITY>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</CITY>")
        Me.txtResult.Text += vbNewLine & "City: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<AREA_CODE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</AREA_CODE>")
        Me.txtResult.Text += vbNewLine & "Area Code: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<CITY_ALIAS_ABBR>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</CITY_ALIAS_ABBR>")
        Me.txtResult.Text += vbNewLine & "City Alias ABBR: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<CITY_ALIAS_NAME>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</CITY_ALIAS_NAME>")
        Me.txtResult.Text += vbNewLine & "City Alias Name: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<STATE_FIPS>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</STATE_FIPS>")
        Me.txtResult.Text += vbNewLine & "State FIPS: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<COUNTY_FIPS>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</COUNTY_FIPS>")
        Me.txtResult.Text += vbNewLine & "County FIPS: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<COUNTY_NAME>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</COUNTY_NAME>")
```

```
        Me.txtResult.Text += vbNewLine & "County Name: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<LATITUDE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</LATITUDE>")
        Me.txtResult.Text += vbNewLine & "Latitude: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<LONGITUDE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</LONGITUDE>")
        Me.txtResult.Text += vbNewLine & "Longitude: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<CITY_TYPE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</CITY_TYPE>")
        Me.txtResult.Text += vbNewLine & "City Type: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<TIME_ZONE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</TIME_ZONE>")
        Me.txtResult.Text += vbNewLine & "Time Zone: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<DAY_LIGHT_SAVING>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</DAY_LIGHT_SAVING>")
        Me.txtResult.Text += vbNewLine & "Day Light Saving: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<ELEVATION>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</ELEVATION>")
        Me.txtResult.Text += vbNewLine & "Elevation: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<MSA2000>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</MSA2000>")
        Me.txtResult.Text += vbNewLine & "MSA2000: "
```

```
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<PMSA>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</PMSA>")
        Me.txtResult.Text += vbNewLine & "PMSA: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<CBSA>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</CBSA>")
        Me.txtResult.Text += vbNewLine & "CBSA: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<CBSA_DIV>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</CBSA_DIV>")
        Me.txtResult.Text += vbNewLine & "CBSA DIV: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<CBSA_TITLE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</CBSA_TITLE>")
        Me.txtResult.Text += vbNewLine & "CBSA Title: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<PERSONS_PER_HOUSEHOLD>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</PERSONS_PER_HOUSEHOLD>")
        Me.txtResult.Text += vbNewLine & "Persons per Household: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<ZIPCODE_POPULATION>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</ZIPCODE_POPULATION>")
        Me.txtResult.Text += vbNewLine & "ZIPCode Population: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<COUNTIES_AREA>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</COUNTIES_AREA>")
        Me.txtResult.Text += vbNewLine & "Counties Area: "
        If indexClose <> -1 Then
```

```
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<HOUSEHOLDS_PER_ZIPCODE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</HOUSEHOLDS_PER_ZIPCODE>")
        Me.txtResult.Text += vbNewLine & "Households per ZIPCode: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<WHITE_POPULATION>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</WHITE_POPULATION>")
        Me.txtResult.Text += vbNewLine & "White Population: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<BLACK_POPULATION>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</BLACK_POPULATION>")
        Me.txtResult.Text += vbNewLine & "Black Population: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<HISPANIC_POPULATION>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</HISPANIC_POPULATION>")
        Me.txtResult.Text += vbNewLine & "Hispanic Population: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<INCOME_PER_HOUSEHOLD>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</INCOME_PER_HOUSEHOLD>")
        Me.txtResult.Text += vbNewLine & "Income per Household: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<AVERAGE_HOUSE_VALUE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</AVERAGE_HOUSE_VALUE>")
        Me.txtResult.Text += vbNewLine & "Average House Value: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
        strParam = "<CREDITSAVAILABLE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</CREDITSAVAILABLE>")
        Me.txtResult.Text += vbNewLine & "Credits Available:"
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
```

```
indexClose - indexOpen)
        End If
        strParam = "<MESSAGE>"
        indexOpen = p_result.IndexOf(strParam) + strParam.Length
        indexClose = p_result.IndexOf("</MESSAGE>")
        Me.txtResult.Text += vbNewLine & "Message: "
        If indexClose <> -1 Then
            Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
        End If
End Sub
```

iii. ASP.NET – C#.NET (SOAP)

```
private void ZIPCodeWorldUSWebService()
{
ZIPCodeWorldUS_WebService.ZIPCodeWorldUS_WebService x_ZIPCodeWorld =
new ZIPCodeWorldUS_WebService.ZIPCodeWorldUS_WebService();

ZIPCODEWORLD_US oZIPCodeWorld = new ZIPCODEWORLD_US();

try
{
oZIPCodeWorld = x_ZIPCodeWorld.ZIPCodeWorld_US(this.txtZipCode.Text,
this.txtLicense.Text);

this.txtResult.Text = "ZIP Code:" + oZIPCodeWorld.ZIP_CODE + "\n";
this.txtResult.Text += "State:" + oZIPCodeWorld.STATE + "\n";
this.txtResult.Text += "City:" + oZIPCodeWorld.CITY + "\n";
this.txtResult.Text += "Area Code:" + oZIPCodeWorld.AREA_CODE + "\n";
this.txtResult.Text += "City Alias ABBR:" +
oZIPCodeWorld.CITY_ALIAS_ABBR + "\n";
this.txtResult.Text += "City Alias Name:" +
oZIPCodeWorld.CITY_ALIAS_NAME + "\n";
this.txtResult.Text += "State FIPS:" + oZIPCodeWorld.STATE_FIPS + "\n";
this.txtResult.Text += "County FIPS:" + oZIPCodeWorld.COUNTY_FIPS +
"\n";
this.txtResult.Text += "Latitude:" + oZIPCodeWorld.LATITUDE + "\n";
this.txtResult.Text += "Longitude:" + oZIPCodeWorld.LONGITUDE + "\n";
this.txtResult.Text += "City Type:" + oZIPCodeWorld.CITY_TYPE + "\n";
this.txtResult.Text += "County Name:" + oZIPCodeWorld.COUNTY_NAME +
"\n";
this.txtResult.Text += "Time Zone:" + oZIPCodeWorld.TIME_ZONE + "\n";
this.txtResult.Text += "Day Light Saving:" +
oZIPCodeWorld.DAY_LIGHT_SAVING + "\n";
this.txtResult.Text += "Elevation:" + oZIPCodeWorld.ELEVATION + "\n";
this.txtResult.Text += "MSA2000:" + oZIPCodeWorld.MSA2000 + "\n";
this.txtResult.Text += "PMSA:" + oZIPCodeWorld.PMSA + "\n";
this.txtResult.Text += "CBSA:" + oZIPCodeWorld.CBSA + "\n";
this.txtResult.Text += "CBSA DIV:" + oZIPCodeWorld.CBSA_DIV + "\n";
this.txtResult.Text += "CBSA Title:" + oZIPCodeWorld.CBSA_TITLE + "\n";
this.txtResult.Text += "Persons per Household:" +
oZIPCodeWorld.PERSONS_PER_HOUSEHOLD + "\n";
this.txtResult.Text += "ZIPCode Population:" +
oZIPCodeWorld.ZIPCODE_POPULATION + "\n";
```

```
this.txtResult.Text += "Counties Area:" + oZIPCodeWorld.COUNTIES_AREA +
"\n";
this.txtResult.Text += "Household per ZIPCode:" +
oZIPCodeWorld.HOUSEHOLDS_PER_ZIPCODE + "\n";
this.txtResult.Text += "White Population:" +
oZIPCodeWorld.WHITE_POPULATION + "\n";
this.txtResult.Text += "Black Population:" +
oZIPCodeWorld.BLACK_POPULATION + "\n";
this.txtResult.Text += "Hispanic Population:" +
oZIPCodeWorld.HISPANIC_POPULATION + "\n";
this.txtResult.Text += "Income per Household:" +
oZIPCodeWorld.INCOME_PER_HOUSEHOLD + "\n";
this.txtResult.Text += "Average House Value:" +
oZIPCodeWorld.AVERAGE_HOUSE_VALUE + "\n";
this.txtResult.Text += "Credits Available:" +
oZIPCodeWorld.CREDITSAVAILABLE + "\n";
this.txtResult.Text += "Message:" + oZIPCodeWorld.MESSAGE + "\n";
}
   catch (Exception ex)
   {
    Response.Write(ex.Message);
   }
}
```

iv. ASP.NET – C#.NET (HTTP)

```
private void Lookup()
{
ZIPCodeWorldUS_WebService.ZIPCodeWorldUS_WebService x_ZIPCodeWorld =
new ZIPCodeWorldUS_WebService.ZIPCodeWorldUS_WebService();

ZIPCODEWORLD_US oZIPCodeWorld = new ZIPCODEWORLD_US();

try
{
//create the request URL
string x_builder;
//add the host element of the URL
x_builder  =
"http://ws.fraudlabs.com/ZIPCodeWorldUS_WebService.asmx/ZIPCodeWorld_US
?ZIPCODE=" + this.txtZipCode.Text + "&LICENSE=" + this.txtLicense.Text;

// create the web client and obtain the response data as a byte array
WebClient x_web_client = new WebClient();
byte [] x_response = x_web_client.DownloadData(x_builder.ToString());
// process the string result to obtain a validation result

processResultString(Encoding.Default.GetString(x_response));
}
catch (Exception ex)
{
  Response.Write(ex.Message);
}
}
```

```
private void processResultString(String p_result)
{
 int indexOpen;
 int indexClose;
 string strParam;

strParam = "<ZIP_CODE>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</ZIP_CODE>");
this.txtResult.Text = "ZIP Code: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
)
      strParam = "<STATE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</STATE>");
      this.txtResult.Text += "\rState: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<CITY>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</CITY>");
      this.txtResult.Text += "\rCity: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<AREA_CODE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</AREA_CODE>");
      this.txtResult.Text += "\rArea Code: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<CITY_ALIAS_ABBR>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</CITY_ALIAS_ABBR>");
      this.txtResult.Text += "\rCity Alias ABBR: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<CITY_ALIAS_NAME>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</CITY_ALIAS_NAME>");
      this.txtResult.Text += "\rCity Alias Name: ";
      if (indexClose != -1)
      {
```

```
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<STATE_FIPS>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</STATE_FIPS>");
      this.txtResult.Text += "\rState FIPS: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<COUNTY_FIPS>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</COUNTY_FIPS>");
      this.txtResult.Text += "\rCounty FIPS: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<COUNTY_NAME>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</COUNTY_NAME>");
      this.txtResult.Text += "\rCounty Name: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<LATITUDE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</LATITUDE>");
      this.txtResult.Text += "\rLatitude: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<LONGITUDE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</LONGITUDE>");
      this.txtResult.Text += "\rLongitude: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<CITY_TYPE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</CITY_TYPE>");
      this.txtResult.Text += "\rCity Type: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
```

```
      strParam = "<TIME_ZONE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</TIME_ZONE>");
      this.txtResult.Text += "\rTime Zone: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<DAY_LIGHT_SAVING>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</DAY_LIGHT_SAVING>");
      this.txtResult.Text += "\rDay Light Saving: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<ELEVATION>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</ELEVATION>");
      this.txtResult.Text += "\rElevation: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<MSA2000>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</MSA2000>");
      this.txtResult.Text += "\rMSA2000: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<PMSA>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</PMSA>");
      this.txtResult.Text += "\rPMSA: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<CBSA>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</CBSA>");
      this.txtResult.Text += "\rCBSA: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<CBSA_DIV>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</CBSA_DIV>");
```

```
      this.txtResult.Text += "\rCBSA DIV: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<CBSA_TITLE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</CBSA_TITLE>");
      this.txtResult.Text += "\rCBSA Title: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<PERSONS_PER_HOUSEHOLD>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</PERSONS_PER_HOUSEHOLD>");
      this.txtResult.Text += "\rPersons per Household: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<ZIPCODE_POPULATION>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</ZIPCODE_POPULATION>");
      this.txtResult.Text += "\rZIPCode Population: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<COUNTIES_AREA>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</COUNTIES_AREA>");
      this.txtResult.Text += "\rCounties Area: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<HOUSEHOLDS_PER_ZIPCODE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</HOUSEHOLDS_PER_ZIPCODE>");
      this.txtResult.Text += "\rHouseholds per ZIPCode: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<WHITE_POPULATION>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</WHITE_POPULATION>");
      this.txtResult.Text += "\rWhite Population: ";
      if (indexClose != -1)
      {
```

```
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<BLACK_POPULATION>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</BLACK_POPULATION>");
      this.txtResult.Text += "\rBlack Population: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<HISPANIC_POPULATION>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</HISPANIC_POPULATION>");
      this.txtResult.Text += "\rHispanic Population: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<INCOME_PER_HOUSEHOLD>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</INCOME_PER_HOUSEHOLD>");
      this.txtResult.Text += "\rIncome per Household: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<AVERAGE_HOUSE_VALUE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</AVERAGE_HOUSE_VALUE>");
      this.txtResult.Text += "\rAverage House Value: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<CREDITSAVAILABLE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</CREDITSAVAILABLE>");
      this.txtResult.Text += "\rCredits Available:";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }
      strParam = "<MESSAGE>";
      indexOpen = p_result.IndexOf(strParam) + strParam.Length;
      indexClose = p_result.IndexOf("</MESSAGE>");
      this.txtResult.Text += "\rMessage: ";
      if (indexClose != -1)
      {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
      }}
```

v. ASP 3.0

```
Sub sendRequest(strZIPCode, strLICENSE)
dim SoapBody if((strZIPCode <> "") AND (strLICENSE <> "") AND
(strLICENSE <> "")) then
SoapBody = xmlSoap(strZIPCode, strLICENSE)
end if %>

<table border="0" cellspacing="1" cellpadding="3" rules="rows">
<%
if SoapBody = "" then
      %><tr><td><b>Empty Soap Body Response</b></td></tr><%
else
dim xml
on error resume next
set xml = Server.CreateObject("Microsoft.XMLDOM")
if(err.number = 0) then
      xml.async = False
      xml.loadxml(SoapBody)
      dim oNode : set oNode =
xml.selectSingleNode("soap:Envelope/soap:Body/" & SoapAction &
"Response/" & SoapAction & "Result")
      if (oNode.selectSingleNode("Error").nodeTypedValue = "") then
%>
<tr>
 <th colspan="2" align="left"><b>Result</b></th>
</tr>
<% if(TypeName(oNode.selectSingleNode("ZIP_CODE")) = "IXMLDOMElement")
then %>
<tr>
 <td align="left">ZIP Code: </td>

<td><%=oNode.selectSingleNode("ZIP_CODE").nodeTypedValue%></td>
</tr>
<% end if %>

<% if(TypeName(oNode.selectSingleNode("CITY")) = "IXMLDOMElement") then
%>
<tr>
<td align="left">City: </td>
<td><%=oNode.selectSingleNode("CITY").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("STATE")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">State: </td>
<td><%=oNode.selectSingleNode("STATE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("AREA_CODE")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">Area Code: </td>
<td><%=oNode.selectSingleNode("AREA_CODE").nodeTypedValue%></td>
```

```
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CITY_ALIAS_ABBR")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">City Alias ABBR: </td>

<td><%=oNode.selectSingleNode("CITY_ALIAS_ABBR").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CITY_ALIAS_NAME")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">City Alias Name: </td>

<td><%=oNode.selectSingleNode("CITY_ALIAS_NAME").nodeTypedValue%></td>
</tr>
<% end if %>

<% if(TypeName(oNode.selectSingleNode("STATE_FIPS")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">State FIPS: </td>
<td><%=oNode.selectSingleNode("STATE_FIPS").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("COUNTY_FIPS")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">County FIPS: </td>
<td><%=oNode.selectSingleNode("COUNTY_FIPS").nodeTypedValue%></td>
</tr>
<% if(TypeName(oNode.selectSingleNode("LATITUDE")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">Latitude: </td>
<td><%=oNode.selectSingleNode("LATITUDE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("LONGITUDE")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">Longitude: </td>
<td><%=oNode.selectSingleNode("LONGITUDE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("COUNTY_NAME")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">County Name: </td>
<td><%=oNode.selectSingleNode("COUNTY_NAME").nodeTypedValue%></td>
</tr>
<% end if %>

<% end if %>
<% if(TypeName(oNode.selectSingleNode("TIME_ZONE")) = "IXMLDOMElement")
then %>
```

```
<tr>
<td align="left">Time Zone: </td>
<td><%=oNode.selectSingleNode("TIME_ZONE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("DAY_LIGHT_SAVING")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Day Light Saving: </td>
<td><%=oNode.selectSingleNode("DAY_LIGHT_SAVING").nodeTypedValue%></td>
</tr>
<% end if %>
                                        <%
if(TypeName(oNode.selectSingleNode("ELEVATION")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">Elevation: </td>
<td><%=oNode.selectSingleNode("ELEVATION").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("MSA2000")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">MSA2000: </td>
<td><%=oNode.selectSingleNode("MSA2000").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("PMSA")) = "IXMLDOMElement") then
%>
<tr>
<td align="left">PMSA: </td>
<td><%=oNode.selectSingleNode("PMSA").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CBSA")) = "IXMLDOMElement") then
%>
<tr>
<td align="left">CBSA: </td>
<td><%=oNode.selectSingleNode("CBSA").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CBSA_DIV")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">CBSA DIV: </td>
<td><%=oNode.selectSingleNode("CBSA_DIV").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CBSA_TITLE")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">CBSA Title: </td>
<td><%=oNode.selectSingleNode("CBSA_TITLE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("PERSONS_PER_HOUSEHOLD")) =
"IXMLDOMElement") then %>
```

```
<tr>
<td align="left">Persons per Household: </td>
<td><%=oNode.selectSingleNode("PERSONS_PER_HOUSEHOLD").nodeTypedValue%>
</td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("ZIPCODE_POPULATION")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">ZIPCode Population: </td>
<td><%=oNode.selectSingleNode("ZIPCODE_POPULATION").nodeTypedValue%></t
d>
</tr>
<% end if %>

<% if(TypeName(oNode.selectSingleNode("COUNTIES_AREA")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Counties Area: </td>
<td><%=oNode.selectSingleNode("COUNTIES_AREA").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("HOUSEHOLDS_PER_ZIPCODE")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Households per ZIPCODE: </td>
<td><%=oNode.selectSingleNode("HOUSEHOLDS_PER_ZIPCODE").nodeTypedValue%
></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("WHITE_POPULATION")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">White Population: </td>
<td><%=oNode.selectSingleNode("WHITE_POPULATION").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("BLACK_POPULATION")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Black Population: </td>
<td><%=oNode.selectSingleNode("BLACK_POPULATION").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("HISPANIC_POPULATION")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Hispanic Population: </td>
<td><%=oNode.selectSingleNode("HISPANIC_POPULATION").nodeTypedValue%></
td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("INCOME_PER_HOUSEHOLD")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Income per Household: </td>
<td><%=oNode.selectSingleNode("INCOME_PER_HOUSEHOLD").nodeTypedValue%><
```

```
/td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("AVERATE_HOUSE_VALUE")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Average House Value: </td>
<td><%=oNode.selectSingleNode("AVERAGE_HOUSE_VALUE").nodeTypedValue%></
td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CREDITSAVAILABLE")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Credits Available: </td>
<td><%=oNode.selectSingleNode("CREDITSAVAILABLE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("MESSAGE")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">Message: </td>
<td><%=oNode.selectSingleNode("MESSAGE").nodeTypedValue%></td>
</tr>
<% end if %>
<%
else ' error element
%>
<tr>
<td align="left" colspan="2"><b>Error</b></td>
</tr>
<tr>
<td align="left">Description:</td>
<td><%=oNode.selectSingleNode("MESSAGE").nodeTypedValue%></td>
</tr>
<%
end if 'error element does not exist
set xml = nothing
else
%>
<tr>
<td align="left">This objects requires Microsofts XML Parser 3.0 SP2 or
greater.</td>
</tr>
<tr>
<td>Download here: <a
href="http://download.microsoft.com/download/xml/SP/3.20/W9X2KMeXP/EN-
US/msxml3sp2Setup.exe">http://download.microsoft.com/download/xml/SP/3.
20/W9X2KMeXP/EN-US/msxml3sp2Setup.exe</a></td>
</tr>
<%
Response.Write("Error: " & err.number)
end if 'DOM object is valid
end if 'soapbody is <> "" empty string
%>
</table> <%
end sub
```

```
function xmlSoap(strZIPCode, strLICENSE)
' Instantiate objects to hold the XML DOM and the HTTP/XML
ommunication:
' Instantiate object for HTTP/XML communication:
Dim xmlhttp, strSoap
Set xmlhttp = Server.CreateObject("Msxml2.ServerXMLHTTP")

' Build XML document:
strSoap =   "<?xml version=""1.0"" encoding=""utf-8""?>" & _
"<soap:Envelope xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
xmlns:xsd=""http://www.w3.org/2001/XMLSchema""
xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/"">" & _
"<soap:Body>" & _
"<" & SoapAction & " xmlns=""" & SoapNamespace & """>" & _
"<ZIPCode>" & strZIPCode & "</ZIPCode>" & _
"<LICENSE>" & strLICENSE & "</LICENSE>" & _
"</" & SoapAction & ">" & _
"</soap:Body>" & _
"</soap:Envelope>"
'Build custom HTTP header:
xmlhttp.Open "POST", "http://" & SoapServer & SoapPath, False     '
False = Do not respond immediately
xmlhttp.setRequestHeader "Man", "POST " & SoapPath & " HTTP/1.1"
xmlhttp.setRequestHeader "Host", SoapServer
xmlhttp.setRequestHeader "Content-Type", "text/xml; charset=utf-8"
xmlhttp.setRequestHeader "SOAPAction", SoapNamespace & SoapAction

'Send it using the header generated above:
xmlhttp.send(strSoap)
if xmlhttp.Status = 200 then
xmlSoap = xmlhttp.responseText
else
xmlSoap = ""
Response.Write("Server Error.<br>")
Response.Write("status = " & xmlhttp.status)
Response.Write("<br>" & xmlhttp.statusText & "<br>" &
xmlhttp.responseText)
Response.Write("<br><pre>" & Request.ServerVariables("ALL_HTTP") &
"</pre>")
end If
Set xmlhttp = nothing
end function
%>
```

vi. PHP

```
<?php

// Get your own license key from http://www.fraudlabs.com
$license = "<Enter License Key>";

// Replace these parameters to your values
$ZIPCode = "00501";
```

```
require_once('./lib/nusoap.php');

// Call the service with this message
$msg = '<ZIPCodeWorld_US xmlns="http://ws.fraudlabs.com/">
      <ZIPCode>'.$ZIPCode.'</ZIPCode>
      <LICENSE>'.$License.'</LICENSE>
      </ZIPCodeWorld_US>';

// Create a new SOAP object
$soapclient  = new
soapclient_nusoap('http://ws.fraudlabs.com/ZIPCodeWorldUS_WebService.as
mx?wsdl','wsdl');
if (!$soapclient->fault) {
$result = $soapclient->call('ZIPCodeWorld_US', array($msg));
if(!isset($result['Error'])) {
// No Error. Retrieve results.
      echo "ZIPCODE = " . $result["ZIP_CODE"] . "<br>";
      echo "STATE = " . $result["STATE"] . "<br>";
      echo "CITY = " . $result["CITY"] . "<br>";
      echo "AREA CODE = " . $result["AREA_CODE"] . "<br>";
      echo "CITY ALIAS ABBR = " . $result["CITY_ALIAS_ABBR"] . "<br>";
      echo "CITY ALIAS NAME = " . $result["CITY_ALIAS_NAME"] . "<br>";
      echo "STATE FIPS = " . $result["STATE_FIPS"] . "<br>";
      echo "COUNTY FIPS = " . $result["COUNTY_FIPS"] . "<br>";
      echo "LATITUDE = " . $result["LATITUDE"] . "<br>";
      echo "LONGITUDE = " . $result["LONGITUDE"] . "<br>";
      echo "CITY TYPE = " . $result["CITY_TYPE"] . "<br>";
      echo "COUNTY NAME = " . $result["COUNTY_NAME"] . "<br>";
      echo "TIME ZONE = " . $result["TIME_ZONE"] . "<br>";
      echo "DAY LIGHT SAVING = " . $result["DAY_LIGHT_SAVING"] ."<br>";
      echo "ELEVATION = " . $result["ELEVATION"] . "<br>";
      echo "MSA2000 = " . $result["MSA2000"] . "<br>";
      echo "PMSA = " . $result["PMSA"] . "<br>";
      echo "CBSA = " . $result["CBSA"] . "<br>";
      echo "CBSA DIV = " . $result["CBSA_DIV"] . "<br>";
      echo "CBSA TITLE = " . $result["CBSA_TITLE"] . "<br>";
      echo "PERSONS PER HOUSEHOLD = " .
$result["PERSONS_PER_HOUSEHOLD"] . "<br>";
      echo "ZIPCode Population = " . $result["ZIPCODE_POPULATION"] .
"<br>";
      echo "COUNTIES AREA = " . $result["COUNTIES_AREA"] . "<br>";
      echo "HOUSEHOLDS PER ZIPCODE = " .
$result["HOUSEHOLDS_PER_ZIPCODE"] . "<br>";
      echo "WHITE POPULATION = " . $result["WHITE_POPULATION"] .
"<br>";
      echo "BLACK POPULATION = " . $result["BLACK_POPULATION"] .
"<br>";
      echo "HISPANIC POPULATION = " . $result["HISPANIC_POPULATION"] .
"<br>";
      echo "INCOME PER HOUSEHOLD = " .
$result["INCOME_PER_HOUSEHOLD"] . "<br>";
      echo "AVERAGE HOUSE VALUE = " . $result["AVERAGE_HOUSE_VALUE"] .
"<br>";
      echo "CREDITSAVAILABLE = " . $result["CREDITSAVAILABLE"] .
"<br>";
      echo "MESSAGE = " . $result["MESSAGE"] . "<br>";
```

```
      } else {
  // Error
      echo "Error Description = " . $result["Error"]["Desc"] . "<br>";
      echo "Error Number = " . $result["Error"]["Number"] . "<br>";
      }
} else {
      echo "Error = {$soapclient->faultcode}<br>";
      echo "String = {$soapclient->faultstring}";
} // end if $soapclient->call is success

}
?>
```

vii. JAVA / APACHE

```
import java.io.*;

//Web Service Client Class
public class Client
{
  //Entry Point to this Application
  public static void main(String[] args)
  {
   try
   {
   //Create a proxy
   ApacheSoapProxy proxy = new ApacheSoapProxy ();

   //Invoke FraudLabsCheck() over SOAP and get the new OID
   //Require Parameter FraudLabsCheck(String ZIPCode, String LICENSE)
   String result = proxy.ZIPCodeWorld_US("01464", "<Enter License
Key>");

   //Print out the value
   System.out.println (result);
   }
   catch (java.net.MalformedURLException exception)
   {
      exception.printStackTrace ();
   }
   catch (org.apache.soap.SOAPException exception)
   {
      exception.printStackTrace ();
   }
}
}
```