

# **AreaCodeWorld™ Area Code Web Service**

<b>1. Overview</b> .....	3
1.1 Overview of the AreaCodeWorld Web Service .....	4
1.2 Front End of AreaCodeWorld Area Code Web Service .....	4
1.2.1. Integration with In-house System.....	5
1.2.2. Direct Access to Hosted Web Service.....	6
1.3 Back End of AreaCodeWorld Area Code Web Service.....	7
1.4 Process Flow Overview .....	7
<b>2. Implementation</b> .....	8
2.1 Implementation Details .....	9
2.2 Basic Parameters - Input .....	12
2.3 Basic Parameters - Output .....	12
2.4 List of possible value for MESSAGE field .....	13
2.5 Structure of the Request and Response .....	14
2.5.1. Request by SOAP .....	14
2.5.2. Response by SOAP .....	14
2.5.3. Request by HTTP-GET .....	15
2.5.4. Response by HTTP-GET .....	15
2.5.5. Request by HTTP-POST.....	16
2.5.6. Response by HTTP-POST .....	16
<b>3. Design Information</b> .....	18
3.1 Placement of AreaCodeWorld Area Code Web Service .....	18
<b>Appendix I : Data File Spesification</b> .....	19
<b>Appendix II : Sample Code</b> .....	22

## **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 AreaCodeWorld Area Code Web Service.

For more information, please visit <http://www.fraudlabs.com> or contact your AreaCodeWorld representative:

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

## 1.1 Overview of the AreaCodeWorld Web Service

AreaCodeWorld delivers scalable web services solutions that help you to obtain geographical and other information about North America, including the United States, Canada and most of the Caribbean. We are able to identify the NPA (area code), NXX (exchange), country, state, county, latitude, longitude, LATA, timezone, day light saving, population, FIPS, MSA, PMSA, CBSA as well as other important information by using our affordable database and technology.

AreaCodeWorld Area 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.

AreaCodeWorld Area Code Web Service is very easy to set up and use as it uses platform independent XML format.

### **Key Features Include:**

- Pinpoints the precise records of North America, including United States, Canada and most of the Caribbean, by NPA (area code), NXX (exchange) lookups
- Provides a complete XML-based Web Services API
- Contains sample code examples for ease integration

## 1.2 Front End of AreaCodeWorld Area 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 AreaCodeWorld Area 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/AreaCodeWorldsamplecodes.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/AreaCodeWorldWebServiceClientVB.zip>
  - ii. Microsoft ASP.Net - C#:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientCSharp.zip>
  - iii. Microsoft ASP 3.0:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientAsp.zip>
  - iv. Java/ Apache:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientJava.zip>
  - v. PHP:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientPHP.zip>
  - vi. Python:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientPython.zip>
  - vii. Perl:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientPerl.zip>
  - viii. ColdFusion MX:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientColdFusion.zip>
  - ix. Access 2000:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientAccess.zip>
  - x. Excel 2000:  
<http://www.fraudlabs.com/samplecode/AreaCodeWorldWebServiceClientExcel.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 AreaCodeWorld™ , 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 AreaCodeWorld™ , 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/AreaCodeWorldWebService.aspx>
- II. Click on "AreaCodeWorld"
- 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 AreaCodeWorld™ , 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 AreaCodeWorld™, 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 AreaCodeWorld Area 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 AreaCodeWorld Area Code Web Service.

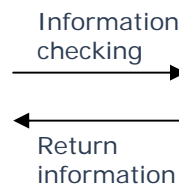
The process works as follows:

1. User enters NPA and NXX 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



**Step 1**  
User enters NPA and  
NXX



**Step 2**  
AreaCodeWorld Web Service  
(Geographical information retrieval)

## **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 AreaCodeWorld Area Code Web Service.

The implementation section covers the following topics:

- AreaCodeWorld Function

For those with solid SOAP programming knowledge, integration basics are included below. POST all requests to:

<http://ws.fraudlabs.com/AreaCodeWorldWebService.asmx>

For more information about AreaCodeWorld Area Code Web Service implementation, please visit <http://www.fraudlabs.com> or contact your AreaCodeWorld representative:

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

## 2.1 Implementation Details

This section was created for those who wish to develop applications that make use of AreaCodeWorld Area Code Web Service.

### SOAP – AreaCodeWorld Function Request

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

<?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>
    <AreaCodeWorld xmlns="http://ws.fraudlabs.com/">
      <NPA>string</NPA>
      <NXX>string</NXX>
      <LICENSE>string</LICENSE>
    </AreaCodeWorld>
  </soap:Body>
</soap:Envelope>
```

### SOAP – AreaCodeWorld 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>
    <AreaCodeWorldResponse xmlns="http://ws.fraudlabs.com/">
<AreaCodeWorldResult>
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <NPA>string</NPA>
  <STATE>string</STATE>
  <CITY>string</CITY>
  <NXX>string</NXX>
  <COUNTRY>string</COUNTRY>
  <COUNTY>string</COUNTY>
  <LATITUDE>float</LATITUDE>
  <LONGITUDE>float</LONGITUDE>
  <TIMEZONE>string</TIMEZONE>
  <LATA>string</LATA>
  <OBSERVES_DST>string</OBSERVES_DST>
```

```

<COUNTY_POPULATION>string</COUNTY_POPULATION>
<FIPS_COUNTY_CODE>string</FIPS_COUNTY_CODE>
<MSA_COUNTY_CODE>string</MSA_COUNTY_CODE>
<PMSA_COUNTY_CODE>string</PMSA_COUNTY_CODE>
<CBSA_COUNTY_CODE>string</CBSA_COUNTY_CODE>
<ZIPCODE_POSTALCODE>string</ZIPCODE_POSTALCODE>
<ZIPCODE_COUNT>string</ZIPCODE_COUNT>
<ZIPCODE_FREQUENCY>string</ZIPCODE_FREQUENCY>
<NEW_NPA>string</NEW_NPA>
<NXX_USE_TYPE>string</NXX_USE_TYPE>
<NXX_INTRO_VERSION>string</NXX_INTRO_VERSION>
<OVERLAY>string</OVERLAY>
<OCN>string</OCN>
<COMPANY>string</COMPANY>
<RATE_CENTER>string</RATE_CENTER>
<SWITCH_CLLI>string</SWITCH_CLLI>
<RC_VERTICAL>string</RC_VERTICAL>
<RC_HORIZONTAL>string</RC_HORIZONTAL>
<MESSAGE>string</MESSAGE>
  </AreaCodeWorldResult>
</AreaCodeWorldResponse>
</soap:Body>
</soap:Envelope>

```

## HTTP GET- AreaCodeWorld Function Request

```

GET
/ws.fraudlabs.com_non_ssl/AreaCodeWorldWebService.asmx/AreaCodeWorld?NP
A=string&NXX=string&LICENSE=string HTTP/1.1
Host: ws.fraudlabs.com

```

## HTTP GET- AreaCodeWorld Function Response

```

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

<?xml version="1.0" encoding="utf-8"?>
<AREACODEWORLD xmlns="http://ws.fraudlabs.com/">
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <NPA>string</NPA>
  <STATE>string</STATE>
  <CITY>string</CITY>
  <NXX>string</NXX>
  <COUNTRY>string</COUNTRY>
  <COUNTY>string</COUNTY>
  <LATITUDE>float</LATITUDE>
  <LONGITUDE>float</LONGITUDE>
  <TIMEZONE>string</TIMEZONE>
  <LATA>string</LATA>
  <OBSERVES_DST>string</OBSERVES_DST>
  <COUNTY_POPULATION>string</COUNTY_POPULATION>

```

```
<FIPS_COUNTY_CODE>string</FIPS_COUNTY_CODE>
<MSA_COUNTY_CODE>string</MSA_COUNTY_CODE>
<PMSA_COUNTY_CODE>string</PMSA_COUNTY_CODE>
<CBSA_COUNTY_CODE>string</CBSA_COUNTY_CODE>
<ZIPCODE_POSTALCODE>string</ZIPCODE_POSTALCODE>
<ZIPCODE_COUNT>string</ZIPCODE_COUNT>
<ZIPCODE_FREQUENCY>string</ZIPCODE_FREQUENCY>
<NEW_NPA>string</NEW_NPA>
<NXX_USE_TYPE>string</NXX_USE_TYPE>
<NXX_INTRO_VERSION>string</NXX_INTRO_VERSION>
<OVERLAY>string</OVERLAY>
<OCN>string</OCN>
<COMPANY>string</COMPANY>
<RATE_CENTER>string</RATE_CENTER>
<SWITCH_CLLI>string</SWITCH_CLLI>
<RC_VERTICAL>string</RC_VERTICAL>
<RC_HORIZONTAL>string</RC_HORIZONTAL>
<MESSAGE>string</MESSAGE>
</AREACODEWORLD>
```

## HTTP POST- AreaCodeWorld Function Request

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

NPA=string&NXX=string&LICENSE=string
```

## HTTP POST- AreaCodeWorld Function Response

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

<?xml version="1.0" encoding="utf-8"?>
<AREACODEWORLD xmlns="http://ws.fraudlabs.com/">
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <NPA>string</NPA>
  <STATE>string</STATE>
  <CITY>string</CITY>
  <NXX>string</NXX>
  <COUNTRY>string</COUNTRY>
  <COUNTY>string</COUNTY>
  <LATITUDE>float</LATITUDE>
  <LONGITUDE>float</LONGITUDE>
  <TIMEZONE>string</TIMEZONE>
  <LATA>string</LATA>
  <OBSERVES_DST>string</OBSERVES_DST>
  <COUNTY_POPULATION>string</COUNTY_POPULATION>
  <FIPS_COUNTY_CODE>string</FIPS_COUNTY_CODE>
```

```

<MSA_COUNTY_CODE>string</MSA_COUNTY_CODE>
<PMSA_COUNTY_CODE>string</PMSA_COUNTY_CODE>
<CBSA_COUNTY_CODE>string</CBSA_COUNTY_CODE>
<ZIPCODE_POSTALCODE>string</ZIPCODE_POSTALCODE>
<ZIPCODE_COUNT>string</ZIPCODE_COUNT>
<ZIPCODE_FREQUENCY>string</ZIPCODE_FREQUENCY>
<NEW_NPA>string</NEW_NPA>
<NXX_USE_TYPE>string</NXX_USE_TYPE>
<NXX_INTRO_VERSION>string</NXX_INTRO_VERSION>
<OVERLAY>string</OVERLAY>
<OCN>string</OCN>
<COMPANY>string</COMPANY>
<RATE_CENTER>string</RATE_CENTER>
<SWITCH_CLLI>string</SWITCH_CLLI>
<RC_VERTICAL>string</RC_VERTICAL>
<RC_HORIZONTAL>string</RC_HORIZONTAL>
<MESSAGE>string</MESSAGE>
</AREACODEWORLD>
    
```

A WSDL is available at:

<http://ws.fraudlabs.com/AreaCodeWorldWebService.asmx?wsdl>

A description of the AreaCodeWorldWebService operation is available at:

<http://ws.fraudlabs.com/AreaCodeWorldWebService.asmx?op=AreaCodeWorld>

## 2.2 Basic Parameters - Input

Field	Format	Description
NPA	Required	Three-digit North American Numbering Plan Area Code.
NXX	Required	Three digit local telephone number exchange.
LICENSE	Required	License key for free license and premium users.

## 2.3 Basic Parameters - Output

Field	Format	Description
NPA	string	Three-digit North American Numbering Plan Area Code.
NXX	string	Three digit local telephone number exchange.
CITY	String	Name of city.
STATE	String	Two letter USPS state abbreviation.
COUNTRY	String	Two character ISO 3166-1 country code.
COUNTY	String	Name of county.

LATA	String	Local Access Transport Area.
TIMEZONE	String	Offset from GMT.
OBSERVES_DST	String	Daylight Saving Time.
COUNTY_POPULATION	String	Estimation of county population.
FIPS_COUNTY_CODE	String	Federal Information Processing System (FIPS) county code.
MSA_COUNTY_CODE	double	Metropolitan Statistical Area.
PMSA_COUNTY_CODE	double	Primary Metropolitan Statistical Area.
LATITUDE	double	Latitude of city.
LONGITUDE	double	Longitude of city.
CBSA_COUNTY_CODE	double	Core Based Statistical Area.
ZIPCODE_POSTALCODE	String	ZIP code or postal code.
ZIPCODE_COUNT	String	Approximate ZIP code occurrences.
ZIPCODE_FREQUENCY	String	Percent frequency of ZIP code occurrence.
NEW_NPA	String	New area code.
NXX_USE_TYPE	String	Primary use designation.
NXX_INTRO_VERSION	String	Version of this product that NXX first appeared.
OVERLAY	String	Overlay.
OCN	String	Telephone Operating Company Number.
COMPANY	String	Name of the telephone company.
RATE_CENTER	String	Rate Center name.
SWITCH_CLLI	String	COMMON LANGUAGE® identifier.
RC_VERTICAL	String	V&H Coordinate system vertical coordinates.
RC_HORIZONTAL	String	V&H Coordinate system horizontal coordinates.
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

AreaCodeWorld Web Service Error Message
<ul style="list-style-type: none"> <li>Invalid license key</li> </ul>
<ul style="list-style-type: none"> <li>Area Code not found in database</li> </ul>
<ul style="list-style-type: none"> <li>NPA and NXX and license key cannot be blank</li> </ul>
<ul style="list-style-type: none"> <li>No credit available</li> </ul>
<ul style="list-style-type: none"> <li>License key was expired</li> </ul>

## 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>
    <AreaCodeWorld xmlns="http://ws.fraudlabs.com/">
      <NPA>201</NPA>
      <NXX>200</NXX>
      <LICENSE>xx-xxxx-xxxx</LICENSE>
    </AreaCodeWorld >
  </soap:Body>
</soap:Envelope>
```

### 2.5.2. Response by SOAP

After posting a valid request, AreaCodeWorld will return several parameters. An example of a response returned AreaCodeWorld 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>
    <AreaCodeWorldResponse xmlns="http://ws.fraudlabs.com/">
      <AreaCodeWorldResult>
        <CREDITSAVAILABLE>4512</CREDITSAVAILABLE>
        <NPA>201</NPA>
        <STATE>JE</STATE>
        <CITY>NJ</CITY>
        <NXX>200</NXX>
        <COUNTRY>US</COUNTRY>
        <COUNTY>HUDSON</COUNTY>
        <LATITUDE>40.71</LATITUDE>
        <LONGITUDE>-74.07</LONGITUDE>
        <TIMEZONE>-</TIMEZONE>
        <LATA>224</LATA>
        <OBSERVES_DST>1</OBSERVES_DST>
        <COUNTY_POPULATION>609000</COUNTY_POPULATION>
        <FIPS_COUNTY_CODE>34017</FIPS_COUNTY_CODE>
        <MSA_COUNTY_CODE>5602</MSA_COUNTY_CODE>
        <PMSA_COUNTY_CODE>3640</PMSA_COUNTY_CODE>
        <CBSA_COUNTY_CODE>35620</CBSA_COUNTY_CODE>
        <ZIPCODE_POSTALCODE>07305 / 07302 / 07304 / 7306</ZIPCODE_POSTALCODE>
        <ZIPCODE_COUNT>245 / 194 / 168 / 132</ZIPCODE_COUNT>
        <ZIPCODE_FREQUENCY>33 / 26 / 23 / 18</ZIPCODE_FREQUENCY>
        <NEW_NPA/>
        <NXX_USE_TYPE>L</NXX_USE_TYPE>
```

```
<NXX_INTRO_VERSION>1998-01-25</NXX_INTRO_VERSION>
<OVERLAY>0</OVERLAY>
<OCN>9206</OCN>
<COMPANY>VERIZON NEW JERSEY, INC. - NJ</COMPANY>
<RATE_CENTER>JERSEYCITY</RATE_CENTER>
<SWITCH_CLLI>JRCYNJBRDS5</SWITCH_CLLI>
<RC_VERTICAL>5009</RC_VERTICAL>
<RC_HORIZONTAL>1414</RC_HORIZONTAL>
<MESSAGE/>
  </AreaCodeWorldResult>
</AreaCodeWorldResponse>
</soap:Body>
</soap:Envelope>
```

### 2.5.3. Request by HTTP-GET

A valid request posted by user:

```
GET
/ws.fraudlabs.com_non_ssl/AreaCodeWorld_WebService.asmx/AreaCodeWorld?N
PA=201&NXX=200&LICENSE=XX-XXXX-XXXX HTTP/1.1
Host: ws.fraudlabs.com
```

### 2.5.4. Response by HTTP-GET

An example of a response returned AreaCodeWorld 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"?>
<AreaCodeWorld xmlns="http://ws.fraudlabs.com/">
<CREDITSAVAILABLE>4512</CREDITSAVAILABLE>
  <NPA>201</NPA>
  <STATE>JE</STATE>
  <CITY>NJ</CITY>
  <NXX>200</NXX>
  <COUNTRY>US</COUNTRY>
  <COUNTY>HUDSON</COUNTY>
  <LATITUDE>40.71</LATITUDE>
  <LONGITUDE>-74.07</LONGITUDE>
  <TIMEZONE>-</TIMEZONE>
  <LATA>224</LATA>
  <OBSERVES_DST>1</OBSERVES_DST>
  <COUNTY_POPULATION>609000</COUNTY_POPULATION>
  <FIPS_COUNTY_CODE>34017</FIPS_COUNTY_CODE>
  <MSA_COUNTY_CODE>5602</MSA_COUNTY_CODE>
  <PMSA_COUNTY_CODE>3640</PMSA_COUNTY_CODE>
  <CBSA_COUNTY_CODE>35620</CBSA_COUNTY_CODE>
  <ZIPCODE_POSTALCODE>07305 / 07302 / 07304 / 7306</ZIPCODE_POSTALCODE>
  <ZIPCODE_COUNT>245 / 194 / 168 / 132</ZIPCODE_COUNT>
  <ZIPCODE_FREQUENCY>33 / 26 / 23 / 18</ZIPCODE_FREQUENCY>
  <NEW_NPA/>
```

```

<NXX_USE_TYPE>L</NXX_USE_TYPE>
<NXX_INTRO_VERSION>1998-01-25</NXX_INTRO_VERSION>
<OVERLAY>0</OVERLAY>
<OCN>9206</OCN>
<COMPANY>VERIZON NEW JERSEY, INC. - NJ</COMPANY>
<RATE_CENTER>JERSEYCITY</RATE_CENTER>
<SWITCH_CLLI>JRCYNJBRDS5</SWITCH_CLLI>
<RC_VERTICAL>5009</RC_VERTICAL>
<RC_HORIZONTAL>1414</RC_HORIZONTAL>
<MESSAGE/>
</AreaCodeWorld>

```

### 2.5.5. Request by HTTP-POST

A valid request posted by user:

```

POST
/ws.fraudlabs.com_non_ssl/AreaCodeWorldWebService.asmx/AreaCodeWorld
HTTP/1.1
Host: ws.fraudlabs.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

NPA=201&NXX=200&LICENSE=XX-XXXX-XXXX

```

### 2.5.6. Response by HTTP-POST

An example of a response returned by AreaCodeWorld 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"?>
<AreaCodeWorld xmlns="http://ws.fraudlabs.com/">
<CREDITSAVAILABLE>4512</CREDITSAVAILABLE>
  <NPA>201</NPA>
  <STATE>JE</STATE>
  <CITY>NJ</CITY>
  <NXX>200</NXX>
  <COUNTRY>US</COUNTRY>
  <COUNTY>HUDSON</COUNTY>
  <LATITUDE>40.71</LATITUDE>
  <LONGITUDE>-74.07</LONGITUDE>
  <TIMEZONE>-</TIMEZONE>
  <LATA>224</LATA>
  <OBSERVES_DST>1</OBSERVES_DST>
  <COUNTY_POPULATION>609000</COUNTY_POPULATION>
  <FIPS_COUNTY_CODE>34017</FIPS_COUNTY_CODE>
  <MSA_COUNTY_CODE>5602</MSA_COUNTY_CODE>
  <PMSA_COUNTY_CODE>3640</PMSA_COUNTY_CODE>
  <CBSA_COUNTY_CODE>35620</CBSA_COUNTY_CODE>
  <ZIPCODE_POSTALCODE>07305 / 07302 / 07304 / 7306</ZIPCODE_POSTALCODE>

```

```
<ZIPCODE_COUNT>245 / 194 / 168 / 132</ZIPCODE_COUNT>
<ZIPCODE_FREQUENCY>33 / 26 / 23 / 18</ZIPCODE_FREQUENCY>
<NEW_NPA/>
<NXX_USE_TYPE>L</NXX_USE_TYPE>
<NXX_INTRO_VERSION>1998-01-25</NXX_INTRO_VERSION>
<OVERLAY>0</OVERLAY>
<OCN>9206</OCN>
<COMPANY>VERIZON NEW JERSEY, INC. - NJ</COMPANY>
<RATE_CENTER>JERSEYCITY</RATE_CENTER>
<SWITCH_CLLI>JRCYNJBRDS5</SWITCH_CLLI>
<RC_VERTICAL>5009</RC_VERTICAL>
<RC_HORIZONTAL>1414</RC_HORIZONTAL>
<MESSAGE/>
</AreaCodeWorld>
```

### **3. Design Information**

This section provides suggestion regarding the placing of AreaCodeWorld Area 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 AreaCodeWorld Area Code Web Service**

As AreaCodeWorld Area Code Web Service provides you very informative details by using NPA and NXX, 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 North America, including United States, Canada and the Caribbean.

**Level of security:** High

**Site Type(s):**

- e-Commerce Solutions
- Internet Retailers
- Software Developers

**Benefits:**

- With our database, applications can be built easily to support business logics
- Obtain additional information on North America addressing system and any other aspect of postal operations
- Enhance customer's information in North America
- Improve data entry speed and accuracy

## Appendix I : Data File Specification

### 1. NPA

Three-digit North American Numbering Plan Area Code for the exchange. Widely known as Area Code.

### 2. NXX

Three digit local telephone number exchange (Central Office Code Assignment) within the NPA. Generally 201 thru 999 (N=2-9, X=0-9).

### 3. COUNTRY

Two character International Standard ISO 3166-1 Country Code for the NPA/NXX Wire Center location.

CODE	COUNTRY
US	United States
CA	Canada
BS	Bahamas
BB	Barbados
AI	Anguilla
AG	Antigua and Barbuda
VG	Virgin Islands, British
KY	Cayman Islands
BM	Bermuda
GD	Grenada
TC	Turks and Caicos Islands
MS	Montserrat
LC	Saint Lucia
DM	Dominica
VC	Saint Vincent and the Grenadines
DO	Dominican Republic
TT	Trinidad and Tobago
KN	Saint Kitts and Nevis
JM	Jamaica

### 4. STATE

Two letter USPS state abbreviation (US, Canada and US territories) for the NPA/NXX Wire Center location.

### 5. CITY

Name or general location of the NPA/NXX Wire Center.

### 6. COUNTY

County name of the NPA/NXX Wire Center.

### 7. LATA

Three digits specify the Local Access Transport Area (LATA) of the NPA/NXX Wire Center is located in. LATA is a predefined boundary for local toll calls.

### 8. OBSERVES\_DST

Day-light-saving flag. 1 = Observes DST (Daylight Saving Time) during the normal DST observance period, 0 = Does not observe DST at all.

## **9. COUNTY\_POPULATION**

Estimation of county population based on Census 2000.

## **10. FIPS\_COUNTY\_CODE**

Five digit United States Census Bureau assigned Federal Information Processing System (FIPS) County Code that identifies the county location of the NPA/NXX. (United States only)

## **11. MSA\_COUNTY\_CODE**

Four digit Metropolitan Statistical Area Code (MSA) the NPA/NXX is located in if any. These codes are defined by the Office of Management and Budget and maintained by the United States Census Bureau. (United States only)

## **12. PMSA\_COUNTY\_CODE**

Four digit Primary Metropolitan Statistical Area Code (MSA) the NPA/NXX is located in if any. These codes are defined by the Office of Management and Budget and maintained by the United States Census Bureau. (United States only)

## **13. LATITUDE**

Latitude in decimal degrees of the center of the NPA/NXX Wire Center.

## **14. LONGITUDE**

Longitude in decimal degrees of the center of the NPA/NXX Wire Center.

## **15. CBSA\_COUNTY\_CODE**

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. (United States only)

## **16. ZIPCODE\_POSTALCODE**

Primary United States ZIP Code or Canadian postal code associated with the NPA/NXX.

## **17. ZIPCODE\_COUNT**

Approximate number of actual ZIP Code occurrences within NPA/NXX. If actual count unknown, 0 is found in the field.

## **18. ZIPCODE\_FREQUENCY**

Percent frequency of occurrence of this ZIP Code in the central office coverage area (United States only). If -1 is found in field, the most frequent but % unknown. If -2 is found, second most frequent but % unknown.

## **19. NEW\_NPA**

New Area Code. During the permissive dialing period, the old Area Code may still be displayed in the "NPA" field.

## **20. NXX\_USE\_TYPE**

Primary use designation. L=Landline and W=Wireless.

## **21. NXX\_INTRO\_VERSION**

Version of this product that NXX first appeared. This is not necessarily the NXX service date, but it should be within 12 months. Since 1994-02-04 was the first version that included the intro date, then all older NXXs will have an intro date of 1994-02-04 regardless of actual service date.

## **22. OVERLAY**

Single character field. If present, determines how the NEW\_NPA field is used. If OVERLAY=1, the NPA is being overlaid and the exchange should use the NEW\_NPA.

## **23. OCN**

Four character telephone Operating Company Number (OCN) assigned by the National Exchange Carrier Association (NECA).

## **24. COMPANY**

Name of the telephone company the exchange is assigned to.

## **25. RATE\_CENTER**

Ten character Rate Center name for Central Office. Blank if Rate Center name unknown.

## **26. SWITCH\_CLLI**

Eleven character COMMON LANGUAGE® identifier of the telephone switch used by the exchange.

## **27. RC\_VERTICAL**

NECA Tariff No. 4 Rate Center V&H Coordinate system vertical coordinate.

## **28. RC\_HORIZONTAL**

NECA Tariff No. 4 Rate Center V&H Coordinate system horizontal coordinate.

## **29. TIMEZONE**

Timezone. Offset from GMT (5=East, 6=Central, 7=Mountain, 8=West, 9=Pacific)

## Appendix II : Sample Code

AreaCodeWorld 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/AreaCodeWorldsamplecodes.aspx>

### i. ASP.NET – VB.NET (SOAP)

```
Private Sub AreaCodeWorldWebService()  
    Dim x_AreaCodeWorld As New  
AreaCodeWorldWebService.AreaCodeWorldWebService  
    Dim oAreaCodeWorld As New AREACODEWORLD  
  
    Try  
oAreaCodeWorld = x_AreaCodeWorld.AreaCodeWorld(Me.txtNPA.Text,  
Me.txtNXX.Text, Me.txtLicense.Text)  
  
Me.txtResult.Text = "NPA:" + oAreaCodeWorld.NPA & vbNewLine  
Me.txtResult.Text += "NXX:" + oAreaCodeWorld.NXX & vbNewLine  
Me.txtResult.Text += "CITY:" + oAreaCodeWorld.CITY & vbNewLine  
Me.txtResult.Text += "COUNTRY:" + oAreaCodeWorld.COUNTRY & vbNewLine  
Me.txtResult.Text += "STATE:" + oAreaCodeWorld.STATE & vbNewLine  
Me.txtResult.Text += "LATITUDE:" + oAreaCodeWorld.LATITUDE.ToString &  
vbNewLine  
Me.txtResult.Text += "LONGITUDE:" + oAreaCodeWorld.LONGITUDE.ToString &  
vbNewLine  
Me.txtResult.Text += "TIME_ZONE:" + oAreaCodeWorld.TIMEZONE.ToString &  
vbNewLine  
Me.txtResult.Text += "COUNTY:" + oAreaCodeWorld.COUNTY & vbNewLine  
Me.txtResult.Text += "LATA:" + oAreaCodeWorld.LATA & vbNewLine  
Me.txtResult.Text += "OBSERVES_DST:" + oAreaCodeWorld.OBSERVES_DST &  
vbNewLine  
Me.txtResult.Text += "COUNTY_POPULATION:" +  
oAreaCodeWorld.COUNTY_POPULATION & vbNewLine  
Me.txtResult.Text += "FIPS_COUNTY_CODE:" +  
oAreaCodeWorld.FIPS_COUNTY_CODE & vbNewLine  
Me.txtResult.Text += "MSA_COUNTY_CODE:" + AreaCodeWorld.MSA_COUNTY_CODE  
& vbNewLine  
Me.txtResult.Text += "PMSA_COUNTY_CODE:" +  
oAreaCodeWorld.PMSA_COUNTY_CODE & vbNewLine  
Me.txtResult.Text += "CBSA_COUNTY_CODE:" +  
oAreaCodeWorld.CBSA_COUNTY_CODE & vbNewLine  
Me.txtResult.Text += "ZIPCODE_POSTALCODE:" +  
oAreaCodeWorld.ZIPCODE_POSTALCODE & vbNewLine  
Me.txtResult.Text += "ZIPCODE_COUNT:" + oAreaCodeWorld.ZIPCODE_COUNT &  
vbNewLine  
Me.txtResult.Text += "ZIPCODE_FREQUENCY:" +  
oAreaCodeWorld.ZIPCODE_FREQUENCY & vbNewLine  
Me.txtResult.Text += "NEW_NPA:" + oAreaCodeWorld.NEW_NPA & vbNewLine  
Me.txtResult.Text += "NXX_USE_TYPE:" + oAreaCodeWorld.NXX_USE_TYPE &  
vbNewLine  
Me.txtResult.Text += "NXX_INTRO_VERSION:" +
```

```

oAreaCodeWorld.NXX_INTRO_VERSION & vbNewLine
Me.txtResult.Text += "OVERLAY:" + oAreaCodeWorld.OVERLAY & vbNewLine
Me.txtResult.Text += "OCN:" + oAreaCodeWorld.OCN & vbNewLine
Me.txtResult.Text += "COMPANY:" + oAreaCodeWorld.COMPANY & vbNewLine
Me.txtResult.Text += "RATE_CENTER:" + oAreaCodeWorld.RATE_CENTER &
vbNewLine
Me.txtResult.Text += "SWITCH_CLLI:" + oAreaCodeWorld.SWITCH_CLLI &
vbNewLine
Me.txtResult.Text += "RC_VERTICAL:" + oAreaCodeWorld.RC_VERTICAL &
vbNewLine
Me.txtResult.Text += "RC_HORIZONTAL:" + oAreaCodeWorld.RC_HORIZONTAL &
vbNewLine
Me.txtResult.Text += "CREDITS AVAILABLE:" +
oAreaCodeWorld.CREDITSAVAILABLE & vbNewLine
Me.txtResult.Text += "MESSAGE:" + oAreaCodeWorld.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_AreaCodeWorld As New
AreaCodeWorldWebService.AreaCodeWorldWebService
    Dim oAreaCodeWorld As New AREACODEWORLD

Try
    ' create the request URL
Dim x_builder As String
    ' add the host element of the URL
x_builder =
"http://ws.fraudlabs.com/AreaCodeWorldWebService.asmx/AreaCodeWorld?" &
_
    "NPA=" + Me.txtNPA.Text & _
    "&NXX=" + Me.txtNXX.Text & _
    "&LICENSE=" + Me.txtLicense.Text
    ' create the web client and obtain the response data
    ' as a byte array
Dim x_web_client As WebClient = New WebClient
Dim x_response() As Byte =
x_web_client.DownloadData(x_builder.ToString())
    ' process the string result to obtain a validation result
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 = "<NPA>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</NPA>")
Me.txtResult.Text = "NPA: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If

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

strParam = "<COUNTRY>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</COUNTRY>")
Me.txtResult.Text += vbNewLine & "COUNTRY: "
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 = "<COUNTY>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</COUNTY>")
Me.txtResult.Text += vbNewLine & "COUNTY: "
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 = "<TIMEZONE>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</TIMEZONE>")  
Me.txtResult.Text += vbNewLine & "TIME ZONE: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
  
strParam = "<LATA>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</LATA>")  
Me.txtResult.Text += vbNewLine & "LATA: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
  
strParam = "<OBSERVES_DST>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</OBSERVES_DST>")  
Me.txtResult.Text += vbNewLine & "OBSERVES_DST: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
  
strParam = "<COUNTY_POPULATION>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</COUNTY_POPULATION>")  
Me.txtResult.Text += vbNewLine & "COUNTY_POPULATION: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
  
strParam = "<FIPS_COUNTY_CODE>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</FIPS_COUNTY_CODE>")  
Me.txtResult.Text += vbNewLine & "FIPS_COUNTY_CODE: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
```

```
indexOpen)
End If

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

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

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

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

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

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

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

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

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

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

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

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

strParam = "<RATE_CENTER>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</RATE_CENTER>")
```

```
Me.txtResult.Text += vbNewLine & "RATE_CENTER: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
  
strParam = "<SWITCH_CLLI>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</SWITCH_CLLI>")  
Me.txtResult.Text += vbNewLine & "SWITCH_CLLI: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
  
strParam = "<RC_VERTICAL>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</RC_VERTICAL>")  
Me.txtResult.Text += vbNewLine & "RC_VERTICAL: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
  
strParam = "<RC_HORIZONTAL>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</RC_HORIZONTAL>")  
Me.txtResult.Text += vbNewLine & "RC_HORIZONTAL: "  
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 AreaCodeWorldWebService()
{
AreaCodeWorldWebService.AreaCodeWorldWebService x_AreaCodeWorld = new
AreaCodeWorldWebService.AreaCodeWorldWebService();

AREACODEWORLD oAreaCodeWorld = new AREACODEWORLD();

try
{
oAreaCodeWorld = xAreaCodeWorld.AreaCodeWorld(this.txtNPA.Text,
this.txtNXX.Text, this.txtLicense.Text);

this.txtResult.Text = "NPA:" + oAreaCodeWorld.NPA + "\n";
this.txtResult.Text += "NXX:" + oAreaCodeWorld.NXX + "\n";
this.txtResult.Text += "CITY:" + oAreaCodeWorld.CITY + "\n";
this.txtResult.Text += "COUNTRY:" + oAreaCodeWorld.COUNTRY + "\n";
this.txtResult.Text += "STATE:" + oAreaCodeWorld.STATE + "\n";
this.txtResult.Text += "LATITUDE:" + oAreaCodeWorld.LATITUDE + "\n";
this.txtResult.Text += "LONGITUDE:" + oAreaCodeWorld.LONGITUDE + "\n";
this.txtResult.Text += "TIME_ZONE:" + oAreaCodeWorld.TIMEZONE + "\n";
this.txtResult.Text += "COUNTY:" + oAreaCodeWorld.COUNTY + "\n";
this.txtResult.Text += "LATA:" + oAreaCodeWorld.LATA + "\n";
this.txtResult.Text += "OBSERVES_DST:" + oAreaCodeWorld.OBSERVES_DST +
"\n";
this.txtResult.Text += "COUNTY_POPULATION:" +
oAreaCodeWorld.COUNTY_POPULATION + "\n";
this.txtResult.Text += "FIPS_COUNTY_CODE:" +
oAreaCodeWorld.FIPS_COUNTY_CODE + "\n";
this.txtResult.Text += "MSA_COUNTY_CODE:" +
oAreaCodeWorld.MSA_COUNTY_CODE + "\n";
this.txtResult.Text += "PMSA_COUNTY_CODE:" +
oAreaCodeWorld.PMSA_COUNTY_CODE + "\n";
this.txtResult.Text += "CBSA_COUNTY_CODE:" +
oAreaCodeWorld.CBSA_COUNTY_CODE + "\n";
this.txtResult.Text += "ZIPCODE_POSTALCODE:" +
oAreaCodeWorld.ZIPCODE_POSTALCODE + "\n";
this.txtResult.Text += "ZIPCODE_COUNT:" + oAreaCodeWorld.ZIPCODE_COUNT
+ "\n";
this.txtResult.Text += "ZIPCODE_FREQUENCY:" +
oAreaCodeWorld.ZIPCODE_FREQUENCY + "\n";
this.txtResult.Text += "NEW_NPA:" + oAreaCodeWorld.NEW_NPA + "\n";
this.txtResult.Text += "NXX_USE_TYPE:" + oAreaCodeWorld.NXX_USE_TYPE +
"\n";
this.txtResult.Text += "NXX_INTRO_VERSION:" +
oAreaCodeWorld.NXX_INTRO_VERSION + "\n";
this.txtResult.Text += "OVERLAY:" + oAreaCodeWorld.OVERLAY + "\n";
this.txtResult.Text += "OCN:" + oAreaCodeWorld.OCN + "\n";
this.txtResult.Text += "COMPANY:" + oAreaCodeWorld.COMPANY + "\n";
this.txtResult.Text += "RATE_CENTER:" + oAreaCodeWorld.RATE_CENTER +
"\n";
this.txtResult.Text += "SWITCH_CLLI:" + oAreaCodeWorld.SWITCH_CLLI +
"\n";
this.txtResult.Text += "RC_VERTICAL:" + oAreaCodeWorld.RC_VERTICAL +
"\n";
this.txtResult.Text += "RC_HORIZONTAL:" + oAreaCodeWorld.RC_HORIZONTAL
+ "\n";
this.txtResult.Text += "CREDITS_AVAILABLE:" +
```

```
oAreaCodeWorld.CREDITSAVAILABLE + "\n";
this.txtResult.Text += "MESSAGE:" + oAreaCodeWorld.MESSAGE + "\n"; }
catch (Exception ex)
{
    Response.Write(ex.Message);
}
}
```

#### iv. ASP.NET – C#.NET (HTTP)

```
private void Lookup()
{
    AreaCodeWorldWebService.AreaCodeWorldWebService x_AreaCodeWorld = new
    AreaCodeWorldWebService.AreaCodeWorldWebService();

    AREACODEWORLD oAreaCodeWorld = new AREACODEWORLD();

    try
    {
        //create the request URL
        string x_builder;
        //add the host element of the URL
        x_builder =
        "http://ws.fraudlabs.com/AreaCodeWorldWebService.asmx/AreaCodeWorld?";
        x_builder += "NPA=" + this.txtNPA.Text;
        x_builder += "&NXX=" + this.txtNXX.Text;
        x_builder += "&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 = "<NPA>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</NPA>");
    this.txtResult.Text = "NPA: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
        indexOpen);
    }
}
```

```
        strParam = "<NXX>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</NXX>");
        this.txtResult.Text += "\rNXX: ";
        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 = "<COUNTRY>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</COUNTRY>");
        this.txtResult.Text += "\rCOUNTRY: ";
        if (indexClose != -1)
        {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
        }
        strParam = "<COUNTY>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</COUNTY>");
        this.txtResult.Text += "\rCOUNTY: ";
        if (indexClose != -1)
        {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
        }
        strParam = "<LATA>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</LATA>");
        this.txtResult.Text += "\rLATA: ";
        if (indexClose != -1)
        {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
        }
        strParam = "<OBSERVES_DST>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</OBSERVES_DST>");
```

```
        this.txtResult.Text += "\rOBSERVES_DST: ";
        if (indexClose != -1)
        {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
        }
        strParam = "<COUNTY_POPULATION>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</COUNTY_POPULATION>");
        this.txtResult.Text += "\rCOUNTY_POPULATION: ";
        if (indexClose != -1)
        {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
        }
        strParam = "<FIPS_COUNTY_CODE>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</FIPS_COUNTY_CODE>");
        this.txtResult.Text += "\rFIPS_COUNTY_CODE: ";
        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 = "<MSA_COUNTY_CODE>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</MSA_COUNTY_CODE>");
        this.txtResult.Text += "\rMSA_COUNTY_CODE: ";
        if (indexClose != -1)
        {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
        }
        strParam = "<TIMEZONE>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</TIMEZONE>");
        this.txtResult.Text += "\rTime Zone: ";
        if (indexClose != -1)
        {
```

```
        this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
    }
    strParam = "<PMSA_COUNTY_CODE>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</PMSA_COUNTY_CODE>");
    this.txtResult.Text += "\rPMSA_COUNTY_CODE: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
    }
    strParam = "<CBSA_COUNTY_CODE>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</CBSA_COUNTY_CODE>");
    this.txtResult.Text += "\rCBSA_COUNTY_CODE: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
    }
    strParam = "<ZIPCODE_POSTALCODE>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</ZIPCODE_POSTALCODE>");
    this.txtResult.Text += "\rZIPCODE_POSTALCODE: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
    }
    strParam = "<ZIPCODE_COUNT>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</ZIPCODE_COUNT>");
    this.txtResult.Text += "\rZIPCODE_COUNT: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
    }
    strParam = "<ZIPCODE_FREQUENCY>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</ZIPCODE_FREQUENCY>");
    this.txtResult.Text += "\rZIPCODE_FREQUENCY: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
    }
    strParam = "<NEW_NPA>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</NEW_NPA>");
    this.txtResult.Text += "\rNEW_NPA: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
    }
}
```

```
strParam = "<NXX_USE_TYPE>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</NXX_USE_TYPE>");
this.txtResult.Text += "\rNXX_USE_TYPE: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
}
strParam = "<NXX_INTRO_VERSION>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</NXX_INTRO_VERSION>");
this.txtResult.Text += "\rNXX_INTRO_VERSION: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
}
strParam = "<OVERLAY>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</OVERLAY>");
this.txtResult.Text += "\rOVERLAY: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
}
strParam = "<OCN>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</OCN>");
this.txtResult.Text += "\rOCN: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
}
strParam = "<COMPANY>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</COMPANY>");
this.txtResult.Text += "\rCOMPANY: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
}
strParam = "<RATE_CENTER>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</RATE_CENTER>");
this.txtResult.Text += "\rRATE_CENTER: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
}
strParam = "<SWITCH_CLLI>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</SWITCH_CLLI>");
```

```

        this.txtResult.Text += "\rSWITCH_CLLI: ";
        if (indexClose != -1)
        {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
        }
        strParam = "<RC_VERTICAL>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</RC_VERTICAL>");
        this.txtResult.Text += "\rRC_VERTICAL: ";
        if (indexClose != -1)
        {
            this.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen);
        }
        strParam = "<RC_HORIZONTAL>";
        indexOpen = p_result.IndexOf(strParam) + strParam.Length;
        indexClose = p_result.IndexOf("</RC_HORIZONTAL>");
        this.txtResult.Text += "\rRC_HORIZONTAL: ";
        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(strNPA, strNXX, strLICENSE)
dim SoapBody
if((strNPA <> "") AND (strNXX <> "") AND (strLICENSE <> "") AND
(strLICENSE <> "<Your License Key>")) then
SoapBody = xmlSoap(strNPA, strNXX, 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("NPA")) = "IXMLDOMEElement") then
%>
<tr>
<td align="left">NPA: </td>
<td><%=oNode.selectSingleNode("NPA").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("NXX")) = "IXMLDOMEElement") then
%>
<tr>
<td align="left">NXX: </td>
<td><%=oNode.selectSingleNode("NXX").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CITY")) = "IXMLDOMEElement") then
%>
<tr>
<td align="left">City: </td>
<td><%=oNode.selectSingleNode("CITY").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("STATE")) = "IXMLDOMEElement")
then %>
<tr>
<td align="left">State: </td>
<td><%=oNode.selectSingleNode("STATE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("LATA")) = "IXMLDOMEElement") then
%>
<tr>
<td align="left">LATA:
</td><td><%=oNode.selectSingleNode("LATA").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("COUNTY")) = "IXMLDOMEElement")
then %>
<tr>
<td align="left">County: </td>
<td><%=oNode.selectSingleNode("COUNTY").nodeTypedValue%></td>
</tr>
<% end if %>
```

```
<% if(TypeName(oNode.selectSingleNode("LATITUDE")) = "IXMLDOMEElement")
then %>
<tr>
<td align="left">Latitude: </td>
<td><%=oNode.selectSingleNode("LATITUDE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("LONGITUDE")) = "IXMLDOMEElement")
then %>
<tr>
<td align="left">Longitude: </td>
<td><%=oNode.selectSingleNode("LONGITUDE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("TIMEZONE")) = "IXMLDOMEElement")
then %>
<tr>
<td align="left">Time Zone: </td>
<td><%=oNode.selectSingleNode("TIMEZONE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("OBSERVES_DST")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">OBSERVES_DST: </td>
<td><%=oNode.selectSingleNode("OBSERVES_DST").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("COUNTY_POPULATION")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">COUNTRY_POPULATION: </td>
<td><%=oNode.selectSingleNode("COUNTY_POPULATION").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("FIPS_COUNTY_CODE")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">FIPS_COUNTY_CODE: </td>
<td><%=oNode.selectSingleNode("FIPS_COUNTY_CODE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("MSA_COUNTY_CODE")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">MSA_COUNTY_CODE: </td>
<td><%=oNode.selectSingleNode("MSA_COUNTY_CODE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("PMSA_COUNTY_CODE")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">PMSA_COUNTY_CODE: </td>
<td><%=oNode.selectSingleNode("PMSA_COUNTY_CODE").nodeTypedValue%></td>
</tr>
<% end if %>
```

```
<% if(TypeName(oNode.selectSingleNode("CBSA_COUNTY_CODE")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">CBSA_COUNTY_CODE: </td>
<td><%=oNode.selectSingleNode("CBSA_COUNTY_CODE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("ZIPCODE_POSTALCODE")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">ZIPCODE_POSTALCODE: </td>
<td><%=oNode.selectSingleNode("ZIPCODE_POSTALCODE").nodeTypedValue%></t
d>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("ZIPCODE_COUNT")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">ZIPCODE_COUNT: </td>
<td><%=oNode.selectSingleNode("ZIPCODE_COUNT").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("ZIPCODE_FREQUENCY")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">ZIPCODE_FREQUENCY: </td>
<td><%=oNode.selectSingleNode("ZIPCODE_FREQUENCY").nodeTypedValue%></td
>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("NEW_NPA")) = "IXMLDOMEElement")
then %>
<tr>
<td align="left">NEW_NPA: </td>
<td><%=oNode.selectSingleNode("NEW_NPA").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("NXX_USE_TYPE")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">NXX_USE_TYPE: </td>
<td><%=oNode.selectSingleNode("NXX_USE_TYPE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("NXX_INTRO_VERSION")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">NXX_INTRO_VERSION: </td>
<td><%=oNode.selectSingleNode("NXX_INTRO_VERSION").nodeTypedValue%></td
>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("OVERLAY")) = "IXMLDOMEElement")
then %>
<tr>
<td align="left">OVERLAY: </td>
<td><%=oNode.selectSingleNode("OVERLAY").nodeTypedValue%></td>
```

```
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("OCN")) = "IXMLDOMEElement") then
%>
<tr>
<td align="left">OCN: </td>
<td><%=oNode.selectSingleNode("OCN").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("COMPANY")) = "IXMLDOMEElement")
then %>
<tr>
<td align="left">COMPANY: </td>
<td><%=oNode.selectSingleNode("COMPANY").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("RATE_CENTER")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">RATE_CENTER: </td>
<td><%=oNode.selectSingleNode("RATE_CENTER").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("SWITCH_CLLI")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">SWITCH_CLLI: </td>
<td><%=oNode.selectSingleNode("SWITCH_CLLI").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("RC_VERTICAL")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">RC_VERTICAL: </td>
<td><%=oNode.selectSingleNode("RC_VERTICAL").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("RC_HORIZONTAL")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">RC_HORIZONTAL: </td>
<td><%=oNode.selectSingleNode("RC_HORIZONTAL").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CREDITSAVAILABLE")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">Credits Available: </td>
<td><%=oNode.selectSingleNode("CREDITSAVAILABLE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("MESSAGE")) = "IXMLDOMEElement")
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(strNPA, strNXX, strLICENSE)
' Instantiate objects to hold the XML DOM and the HTTP/XML
communication:
' 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 & "">" & _
"<NPA>" & strNPA & "</NPA>" & _
"<NXX>" & strNXX & "</NXX>" & _
"<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          ' Response from server was success
xmlSoap = xmlhttp.responseText
' note xmlSoap is the function name hence the return value is a string
Variant
Else                                  ' Response from server failed
xmlSoap = ""
' Tell administrator what went wrong - maybe not users though
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
if (!isset($_POST['submit'])) {} // if page is not submitted to itself
echo the form
else
{
$NPA = $_POST["NPA"];
$NXX = $_POST["NXX"];
$License = $_POST["License"];

if ($License == "<Enter License Key>" || $License == "")
{
echo "License key are required field." ;
}

else
{
require_once('./lib/nusoap.php');

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

// Create a new SOAP object
$soapclient = new
soapclient_nusoap('http://ws.fraudlabs.com/AreaCodeWorldWebService.asmx

```

```

?wsdl','wsdl');
if (!$soapclient->fault) {
    $result = $soapclient->call('AreaCodeWorld', array($msg));
    if(!isset($result['Error'])) {
        // No Error. Retrieve results.
        echo "NPA = " . $result["NPA"] . "<br>";
        echo "NXX = " . $result["NXX"] . "<br>";
        echo "COUNTRY = " . $result["COUNTRY"] . "<br>";
        echo "STATE = " . $result["STATE"] . "<br>";
        echo "CITY = " . $result["CITY"] . "<br>";
        echo "COUNTY = " . $result["COUNTY"] . "<br>";
        echo "LATITUDE = " . $result["LATITUDE"] . "<br>";
        echo "LONGITUDE = " . $result["LONGITUDE"] . "<br>";
        echo "TIME_ZONE = " . $result["TIMEZONE"] . "<br>";
        echo "LATA = " . $result["LATA"] . "<br>";
        echo "OBSERVES_DST = " . $result["OBSERVES_DST"] . "<br>";
        echo "COUNTY_POPULATION = " . $result["COUNTY_POPULATION"] .
"<br>";
        echo "FIPS_COUNTY_CODE = " . $result["FIPS_COUNTY_CODE"] .
"<br>";
        echo "MSA_COUNTY_CODE = " . $result["MSA_COUNTY_CODE"] . "<br>";
        echo "PMSA_COUNTY_CODE = " . $result["PMSA_COUNTY_CODE"] .
"<br>";
        echo "CBSA_COUNTY_CODE = " . $result["CBSA_COUNTY_CODE"] .
"<br>";
        echo "ZIPCODE_POSTALCODE = " . $result["ZIPCODE_POSTALCODE"] .
"<br>";
        echo "ZIPCODE_COUNT = " . $result["ZIPCODE_COUNT"] . "<br>";
        echo "ZIPCODE_FREQUENCY = " . $result["ZIPCODE_FREQUENCY"] .
"<br>";
        echo "NEW_NPA = " . $result["NEW_NPA"] . "<br>";
        echo "NXX_USE_TYPE = " . $result["NXX_USE_TYPE"] . "<br>";
        echo "NXX_INTRO_VERSION = " . $result["NXX_INTRO_VERSION"] .
"<br>";
        echo "OVERLAY = " . $result["OVERLAY"] . "<br>";
        echo "OCN = " . $result["OCN"] . "<br>";
        echo "COMPANY = " . $result["COMPANY"] . "<br>";
        echo "RATE_CENTER = " . $result["RATE_CENTER"] . "<br>";
        echo "SWITCH_CLLI = " . $result["SWITCH_CLLI"] . "<br>";
        echo "RC_VERTICAL = " . $result["RC_VERTICAL"] . "<br>";
        echo "RC_HORIZONTAL = " . $result["RC_HORIZONTAL"] . "<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 ();

            String result = proxy.AreaCodeWorldUS("201", "200", "<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 ();
        }
    }
}
```