

PostalCodeWorld™ Mexico Postal Code Web Service

1. Overview	3
1.1 Overview of the PostalCodeWorld Mexico Web Service.....	4
1.2 Front End of PostalCodeWorld Mexico Postal 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 PostalCodeWorld Mexico Postal Code Web Service.....	7
1.4 Process Flow Overview	7
2. Implementation	8
2.1 Implementation Details	9
2.2 Basic Parameters - Input	11
2.3 Basic Parameters - Output	11
2.4 List of possible value for MESSAGE field	12
2.5 Structure of the Request and Response	12
2.5.1. Request by SOAP	12
2.5.2. Response by SOAP	13
2.5.3. Request by HTTP-GET	13
2.5.4. Response by HTTP-GET	13
2.5.5. Request by HTTP-POST.....	14
2.5.6. Response by HTTP-POST	14
3. Design Information	15
3.1 Placement of PostalCodeWorld Mexico Postal Code Web Service	15
Appendix I : Data File Spesification	16
Appendix II : Sample Code	18

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 PostalCodeWorld Mexico Postal Code Web Service.

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

Email : sales@fraudlabs.com

1.1 Overview of the PostalCodeWorld Mexico Web Service

PostalCodeWorld Mexico delivers scalable web services solutions that help you to obtain geographical and other information about Mexico. We are able to identify the state name, city name, colony name, new area code, old area code, latitude, longitude, time zone as well as other important information by using our affordable PostalCodeWorld Mexico database and technology.

PostalCodeWorld Mexico Postal 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.

PostalCodeWorld Mexico Postal Code Web Service is very easy to set up and use as it uses platform independent XML format. You may get precise records by postalcodes geographical location within a minute of time.

Key Features Include:

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

1.2 Front End of PostalCodeWorld Mexico Postal 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 PostalCodeWorld Mexico Postal 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/postalcodeworldMexicosamplecodes.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/PostalCodeWorldMexico_WebServiceClientVB.zip
 - ii. Microsoft ASP.Net - C#:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_WebServiceClientCSharp.zip
 - iii. Microsoft ASP 3.0:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_WebServiceClientAsp.zip
 - iv. Java/ Apache:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_WebServiceClientJava.zip
 - v. PHP:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_WebServiceClientPHP.zip
 - vi. Python:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_WebServiceClientPython.zip
 - vii. Perl:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_WebServiceClientPerl.zip
 - viii. ColdFusion MX:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_WebServiceClientColdFusion.zip
 - ix. Access 2000:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_WebServiceClientAccess.zip
 - x. Excel 2000:
http://www.fraudlabs.com/samplecode/PostalCodeWorldMexico_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 PostalCodeWorld™ Mexico, 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 PostalCodeWorld™ Mexico, 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/PostalCodeWorldMexico_WebService.aspx
- II. Click on "PostalCodeWorld_Mexico"
- 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 PostalCodeWorld™ Mexico, 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 PostalCodeWorld™ Mexico, 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 PostalCodeWorld Mexico Postal 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 PostalCodeWorld Mexico Postal Code Web Service.

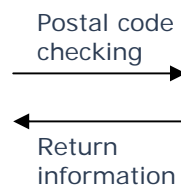
The process works as follows:

1. User enters Postal 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



Step 1
User enters Mexico
Postal code



Step 2
PostalCodeWorldMexico 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 PostalCodeWorld Mexico Postal Code Web Service.

The implementation section covers the following topics:

- PostalCodeWorld_Mexico Function

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

http://ws.fraudlabs.com/PostalCodeWorldMexico_WebService.asmx

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

Email : sales@fraudlabs.com

2.1 Implementation Details

This section was created for those who wish to develop applications that make use of PostalCodeWorld Mexico Postal Code Web Service.

SOAP – PostalCodeWorld Mexico Function Request

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

<?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>
    <PostalCodeWorld_Mexico xmlns="http://ws.fraudlabs.com/">
      <PostalCode>string</PostalCode>
      <LICENSE>string</LICENSE>
    </PostalCodeWorld_Mexico>
  </soap:Body>
</soap:Envelope>
```

SOAP – PostalCodeWorld Mexico 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>
    <PostalCodeWorld_MexicoResponse xmlns="http://ws.fraudlabs.com/">
      <PostalCodeWorld_MexicoResult>
        <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
        <POSTAL_CODE>string</POSTAL_CODE>
        <COLONY_NAME>string</COLONY_NAME>
        <CITY_NAME>string</CITY_NAME>
        <STATE_ABBR>string</STATE_ABBR>
        <STATE_NAME>string</STATE_NAME>
        <LATITUDE>float</LATITUDE>
        <LONGITUDE>float</LONGITUDE>
        <OLD_AREA_CODE>string</OLD_AREA_CODE>
        <NEW_AREA_CODE>string</NEW_AREA_CODE>
        <TIME_ZONE>float</TIME_ZONE>
        <DAY_LIGHT_SAVING>string</DAY_LIGHT_SAVING>
```

```
<MESSAGE>string</MESSAGE>
  </PostalCodeWorld_MexicoResult>
</PostalCodeWorld_MexicoResponse>
</soap:Body>
</soap:Envelope>
```

HTTP GET- PostalCodeWorld Mexico Function Request

```
GET
/ws.fraudlabs.com_non_ssl/postalcodeworldmexico_webservice.asmx/PostalC
odeWorld_Mexico?PostalCode=string&LICENSE=string HTTP/1.1
Host: ws.fraudlabs.com
```

HTTP GET- PostalCodeWorld Mexico Function Response

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

<?xml version="1.0" encoding="utf-8"?>
<POSTALCODEWORLD_MEXICO xmlns="http://ws.fraudlabs.com/">
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <POSTAL_CODE>string</POSTAL_CODE>
  <COLONY_NAME>string</COLONY_NAME>
  <CITY_NAME>string</CITY_NAME>
  <STATE_ABBR>string</STATE_ABBR>
  <STATE_NAME>string</STATE_NAME>
  <LATITUDE>float</LATITUDE>
  <LONGITUDE>float</LONGITUDE>
  <OLD_AREA_CODE>string</OLD_AREA_CODE>
  <NEW_AREA_CODE>string</NEW_AREA_CODE>
  <TIME_ZONE>float</TIME_ZONE>
  <DAY_LIGHT_SAVING>string</DAY_LIGHT_SAVING>
  <MESSAGE>string</MESSAGE>
</POSTALCODEWORLD_MEXICO>
```

HTTP POST- PostalCodeWorld Mexico Function Request

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

PostalCode=string&LICENSE=string
```

HTTP POST- PostalCodeWorld Mexico Function Response

```

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

<?xml version="1.0" encoding="utf-8"?>
<POSTALCODEWORLD_MEXICO xmlns="http://ws.fraudlabs.com/">
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <POSTAL_CODE>string</POSTAL_CODE>
  <COLONY_NAME>string</COLONY_NAME>
  <CITY_NAME>string</CITY_NAME>
  <STATE_ABBR>string</STATE_ABBR>
  <STATE_NAME>string</STATE_NAME>
  <LATITUDE>float</LATITUDE>
  <LONGITUDE>float</LONGITUDE>
  <OLD_AREA_CODE>string</OLD_AREA_CODE>
  <NEW_AREA_CODE>string</NEW_AREA_CODE>
  <TIME_ZONE>float</TIME_ZONE>
  <DAY_LIGHT_SAVING>string</DAY_LIGHT_SAVING>
  <MESSAGE>string</MESSAGE>
</POSTALCODEWORLD_MEXICO>
    
```

A WSDL is available at:

http://ws.fraudlabs.com/PostalCodeWorldMexico_WebService.asmx?wsdl

A description of the PostalCodeWorldMexico_WebService operation is available at:

http://ws.fraudlabs.com/PostalCodeWorldMexico_WebService.asmx?op=PostalCodeWorld_Mexico

2.2 Basic Parameters - Input

Field	Format	Description
PostalCode	Required	Mexico Postal Code
LICENSE	Required	License key for free license and premium users.

2.3 Basic Parameters - Output

Field	Format	Description
POSTAL_CODE	string	The Postal Code of the city.
COLONY_NAME	string	The colony name of the Postal Code.
CITY_NAME	string	The city name of the Postal Code.
STATE_ABBR	string	The telephone area code belonging to the particular postal code is located.

STATE_NAME	string	The state name of the Postal Code.
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 north or south of the equator.
LONGITUDE	float	The geographic coordinate of a point measured in degrees east or west of the Greenwich meridian.
OLD_AREA_CODE	string	Old one digit area code.
NEW_AREA_CODE	string	New three digit area code.
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

PostalCodeWorld Mexico Web Service Error Message
• Invalid license key
• Postal Code xx not found in database. Please try again.
• Postal 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>
    <PostalCodeWorld_Mexico xmlns="http://ws.fraudlabs.com/">
      <PostalCode>01000</ZIPCode>
      <LICENSE>xx-xxxx-xxxx</LICENSE>
    </PostalCodeWorld_Mexico >
  </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>
    <PostalCodeWorld_MexicoResponse xmlns="http://ws.fraudlabs.com/">
      <PostalCodeWorld_MexicoResult>
        <CREDITSAVAILABLE>463</CREDITSAVAILABLE>
        <Postal_CODE>01000</Postal_CODE>
        <COLONY_NAME>COL. SAN ANGEL / COL. VILLA OBREGON</COLONY_NAME>
        <CITYNAME>MEXICO</CITYNAME>
        <STATE_ABBR>DF</STATE_ABBR>
        <STATE_NAME>DISTRITO FEDERAL</STATE_NAME>
        <LATITUDE>19.34981</LATITUDE>
        <LONGITUDE>-99.1865</LONGITUDE>
        <OLD_AREA_CODE>5</OLD_AREA_CODE>
        <NEW_AREA_CODE>525</NEW_AREA_CODE>
        <TIME_ZONE>-6</TIME_ZONE>
        <DAY_LIGHT_SAVING>Y</DAY_LIGHT_SAVING>
        <MESSAGE/>
      </PostalCodeWorld_MexicoResult>
    </PostalCodeWorld_MexicoResponse>
  </soap:Body>
</soap:Envelope>
```

2.5.3. Request by HTTP-GET

A valid request posted by user:

```
GET
/ws.fraudlabs.com_non_ssl/PostalCodeWorldMexico_WebService.asmx/PostalC
odeWorld_Mexico?PostalCode=01000&LICENSE=XX-XXXX-XXXX HTTP/1.1
Host: ws.fraudlabs.com
```

2.5.4. Response by HTTP-GET

An example of a response returned PostalCodeWorld_Mexico 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"?>
<PostalCodeWorld_Mexico xmlns="http://ws.fraudlabs.com/">
  <CREDITSAVAILABLE>463</CREDITSAVAILABLE>
  <Postal_CODE>01000</Postal_CODE>
```

```
<COLONY_NAME>COL. SAN ANGEL / COL. VILLA OBREGON</COLONY_NAME>
<CITYNAME>MEXICO</CITYNAME>
<STATE_ABBR>DF</STATE_ABBR>
<STATE_NAME>DISTRITO FEDERAL</STATE_NAME>
<LATITUDE>19.34981</LATITUDE>
<LONGITUDE>-99.1865</LONGITUDE>
<OLD_AREA_CODE>5</OLD_AREA_CODE>
<NEW_AREA_CODE>525</NEW_AREA_CODE>
<TIME_ZONE>-6</TIME_ZONE>
<DAY_LIGHT_SAVING>Y</DAY_LIGHT_SAVING>
<MESSAGE/>
</PostalCodeWorld_Mexico>
```

2.5.5. Request by HTTP-POST

A valid request posted by user:

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

PostalCode=01000&LICENSE=XX-XXXX-XXXX
```

2.5.6. Response by HTTP-POST

An example of a response returned by PostalCodeWorld_Mexico 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"?>
<PostalCodeWorld_Mexico xmlns="http://ws.fraudlabs.com/">
  <CREDITSAVAILABLE>463</CREDITSAVAILABLE>
  <Postal_CODE>01000</Postal_CODE>
  <COLONY_NAME>COL. SAN ANGEL / COL. VILLA OBREGON</COLONY_NAME>
  <CITYNAME>MEXICO</CITYNAME>
  <STATE_ABBR>DF</STATE_ABBR>
  <STATE_NAME>DISTRITO FEDERAL</STATE_NAME>
  <LATITUDE>19.34981</LATITUDE>
  <LONGITUDE>-99.1865</LONGITUDE>
  <OLD_AREA_CODE>5</OLD_AREA_CODE>
  <NEW_AREA_CODE>525</NEW_AREA_CODE>
  <TIME_ZONE>-6</TIME_ZONE>
  <DAY_LIGHT_SAVING>Y</DAY_LIGHT_SAVING>
  <MESSAGE/>
</PostalCodeWorld_Mexico>
```

3. Design Information

This section provides suggestion regarding the placing of PostalCodeWorld Mexico Postal 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 PostalCodeWorld Mexico Postal Code Web Service

As PostalCodeWorld Mexico Postal Code Web Service provides you very informative details by using Postal 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 Mexico

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 58,000 precise records with 28,000 unique postal codes.
- With our PostalCodeWorld database, applications can be built easily to support business logics
- Obtain additional information on Mexico addressing system and any other aspect of postal operations
- Enhance customer's information in Mexico
- Improve data entry speed and accuracy

Appendix I : Data File Specification

1. POSTAL_CODE

Mexican postal codes are made up of 5 digits. It is used in mailing address to identify correct postal exchange. The following is the way to format a Mexican address:

1st LINE: Name

2nd LINE: Number and street

3rd LINE: Colony (if applicable for large cities)

4th LINE: Postal Code, City and State

5th LINE: MEXICO

2. COLONY_NAME

Some large cities are divided into colonia or colonies. Small towns do not have the colony name and the field is empty.

3. CITY_NAME

This field belongs to the city name of the Postal Code.

4. STATE_ABBR

This is an abbreviation name of the state.

5. STATE_NAME

This is the state name of the postal code. Following is complete table of 32 states in Mexico.

ABBR	STATE NAME
AGS	AGUASCALIENTES
BC	BAJA CALIFORNIA
BCS	BAJA CALIFORNIA SUR
CAM	CAMPECHE
COAH	COAHUILA
COL	COLIMA
CHIS	CHIAPAS
CHIH	CHIHUAHUA
DF	DISTRITO FEDERAL
DGO	DURANGO
GTO	GUANAJUATO
GRO	GUERRERO
HGO	HIDALGO
JAL	JALISCO
MEX	MEXICO
MICH	MICHOACAN
MOR	MORELOS
NAY	NAYARIT
NL	NUEVO LEON
OAX	OAXACA
PUE	PUEBLA
QRO	QUERETARO
QROO	QUINTANA ROO
SLP	SAN LUIS POTOSI

SIN	SINALOA
SON	SONORA
TAB	TABASCO
TAM	TAMAULIPAS
TLAX	TLAXCALA
VER	VERACRUZ
YUC	YUCATAN
ZAC	ZACATECAS

6. LATITUDE

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

7. LONGITUDE

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

8. OLD_AREA_CODE

This number represents the old one digit area code used in the postal code.

9. NEW_AREA_CODE

This number represents the new three digit area code used in the postal code.

10. TIME_ZONE

This number represents the hours past the Greenwich Meridian Time Zone (GMT/UTC).

11. DAY_LIGHT_SAVING

The flag represents whether the county observes day light saving during the summer. "Y" for YES; "N" for NO.

Appendix II : Sample Code

PostalCodeWorldMexico 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/postalcodeworldMexicosamplecodes.aspx>

i. ASP.NET – VB.NET (SOAP)

```
Private Sub PostalCodeWorldMexicoWebService()  
    Dim x_PostalCodeWorld As New  
    PostalCodeWorldMexico_WebService.PostalCodeWorldMexico_WebService  
    Dim oPostalCodeWorld As New POSTALCODEWORLD_MEXICO  
  
    Try  
        oPostalCodeWorld =  
        x_PostalCodeWorld.PostalCodeWorld_Mexico(Me.txtPostalCode.Text,  
        Me.txtLicense.Text)  
  
        Me.txtResult.Text = "Postal Code:" + oPostalCodeWorld.POSTAL_CODE &  
        vbNewLine  
        Me.txtResult.Text += "State Name:" + oPostalCodeWorld.STATE_NAME &  
        vbNewLine  
        Me.txtResult.Text += "City Name:" + oPostalCodeWorld.CITY_NAME &  
        vbNewLine  
        Me.txtResult.Text += "Colony Name:" + oPostalCodeWorld.COLONY_NAME &  
        vbNewLine  
        Me.txtResult.Text += "State ABBR:" + oPostalCodeWorld.STATE_ABBR &  
        vbNewLine  
        Me.txtResult.Text += "Latitude:" + oPostalCodeWorld.LATITUDE.ToString &  
        vbNewLine  
        Me.txtResult.Text += "Longitude:" + oPostalCodeWorld.LONGITUDE.ToString  
        & vbNewLine  
        Me.txtResult.Text += "Time Zone:" + oPostalCodeWorld.TIME_ZONE.ToString  
        & vbNewLine  
        Me.txtResult.Text += "Day Light Saving:" +  
        oPostalCodeWorld.DAY_LIGHT_SAVING & vbNewLine  
        Me.txtResult.Text += "Old Area Code:" + oPostalCodeWorld.OLD_AREA_CODE  
        & vbNewLine  
        Me.txtResult.Text += "New Area Code:" + oPostalCodeWorld.NEW_AREA_CODE  
        & vbNewLine  
        Me.txtResult.Text += "Credits Available:" +  
        oPostalCodeWorld.CREDITSAVAILABLE & vbNewLine  
        Me.txtResult.Text += "Message:" + oPostalCodeWorld.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_PostalCodeWorld As New
PostalCodeWorldMexico_WebService.PostalCodeWorldMexico_WebService
Dim oPostalCodeWorld As New POSTALCODEWORLD_MEXICO

Try
' create the request URL
Dim x_builder As String
' add the host element of the URL
x_builder =
"http://ws.fraudlabs.com/PostalCodeWorldMexico_WebService.asmx/PostalCo
deWorld_Mexico?" & _
"PostalCode=" + Me.txtPostalCode.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 = "<POSTAL_CODE>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</POSTAL_CODE>")
Me.txtResult.Text = "Postal Code: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If
strParam = "<STATE_NAME>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</STATE_NAME>")
Me.txtResult.Text += vbNewLine & "State Name: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If
strParam = "<CITY_NAME>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</CITY_NAME>")
Me.txtResult.Text += vbNewLine & "City Name: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If
strParam = "<COLONY_NAME>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</COLONY_NAME>")
```

```
Me.txtResult.Text += vbNewLine & "Colony Name: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
strParam = "<STATE_ABBR>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</STATE_ABBR>")  
Me.txtResult.Text += vbNewLine & "State ABBR: "  
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 = "<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 = "<OLD_AREA_CODE>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</OLD_AREA_CODE>")  
Me.txtResult.Text += vbNewLine & "Old Area Code: "  
If indexClose <> -1 Then  
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
indexOpen)  
End If  
strParam = "<NEW_AREA_CODE>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</NEW_AREA_CODE>")  
Me.txtResult.Text += vbNewLine & "New Area Code: "
```

```

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 PostalCodeWorldMexicoWebService()
{
PostalCodeWorldMexico_WebService.PostalCodeWorldMexico_WebService
xPostalCodeWorld = new
PostalCodeWorldMexico_WebService.PostalCodeWorldMexico_WebService();

POSTALCODEWORLD_MEXICO oPostalCodeWorld = new POSTALCODEWORLD_MEXICO();

try
{
oPostalCodeWorld =
xPostalCodeWorld.PostalCodeWorld_Mexico(this.txtPostalCode.Text,
this.txtLicense.Text);

this.txtResult.Text = "Postal Code:" + oPostalCodeWorld.POSTAL_CODE +
"\n";
this.txtResult.Text += "State Name:" + oPostalCodeWorld.STATE_NAME +
"\n";
this.txtResult.Text += "City Name:" + oPostalCodeWorld.CITY_NAME +
"\n";
this.txtResult.Text += "Colony Name:" + oPostalCodeWorld.COLONY_NAME +
"\n";
this.txtResult.Text += "State ABBR:" + oPostalCodeWorld.STATE_ABBR +
"\n";
this.txtResult.Text += "Latitude:" + oPostalCodeWorld.LATITUDE + "\n";
this.txtResult.Text += "Longitude:" + oPostalCodeWorld.LONGITUDE +
"\n";
this.txtResult.Text += "Time Zone:" + oPostalCodeWorld.TIME_ZONE +
"\n";
this.txtResult.Text += "Day Light Saving:" +
oPostalCodeWorld.DAY_LIGHT_SAVING + "\n";
}
}

```

```

this.txtResult.Text += "Old Area Code:" +
oPostalCodeWorld.OLD_AREA_CODE + "\n";
this.txtResult.Text += "New Area Code:" +
oPostalCodeWorld.NEW_AREA_CODE + "\n";
this.txtResult.Text += "Credits Available:" +
oPostalCodeWorld.CREDITSAVAILABLE + "\n";
this.txtResult.Text += "Message:" + oPostalCodeWorld.MESSAGE + "\n";
}
catch (Exception ex)
{
Response.Write(ex.Message);
}
}
}

```

iv. ASP.NET – C#.NET (HTTP)

```

private void Lookup()
{
PostalCodeWorldMexico_WebService.PostalCodeWorldMexico_WebService
x_PostalCodeWorld = new
PostalCodeWorldMexico_WebService.PostalCodeWorldMexico_WebService();

POSTALCODEWORLD_MEXICO oPostalCodeWorld = new POSTALCODEWORLD_MEXICO();

try
{
//create the request URL
string x_builder;
//add the host element of the URL
x_builder =
"http://ws.fraudlabs.com/PostalCodeWorldMexico_WebService.asmx/PostalCo
deWorld_Mexico?PostalCODE=" + this.txtPostalCode.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 = "<POSTAL_CODE>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</POSTAL_CODE>");
}
}

```

```
this.txtResult.Text = "Postal Code: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
strParam = "<STATE_NAME>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</STATE_NAME>");
this.txtResult.Text += "\rState Name: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
strParam = "<CITY_NAME>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</CITY_NAME>");
this.txtResult.Text += "\rCity Name: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
strParam = "<COLONY_NAME>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</COLONY_NAME>");
this.txtResult.Text += "\rColony Name: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
strParam = "<STATE_ABBR>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</STATE_ABBR>");
this.txtResult.Text += "\rCity ABBR: ";
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 = "<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 = "<OLD_AREA_CODE>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</OLD_AREA_CODE>");
this.txtResult.Text += "\rOld Area Code: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
strParam = "<NEW_AREA_CODE>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</NEW_AREA_CODE>");
this.txtResult.Text += "\rNew Area Code: ";
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(strPostalCode, strLICENSE)
dim SoapBody
if((strPostalCode <> "") AND (strLICENSE <> "") AND (strLICENSE <>
"<Your License Key>")) then
SoapBody = xmlSoap(strPostalCode, 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("POSTAL_CODE")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">Postal Code: </td>
<td><%=oNode.selectSingleNode("POSTAL_CODE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("CITY_NAME")) = "IXMLDOMElement")
then %>
<tr>
<td align="left">City Name: </td>
<td><%=oNode.selectSingleNode("CITY_NAME").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("STATE_NAME")) =
"IXMLDOMElement") then %>
<tr>
<td align="left">State Name: </td>
<td><%=oNode.selectSingleNode("STATE_NAME").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("COLONY_NAME")) =
"IXMLDOMElement") then %>
<tr>

```

```
<td align="left">Colony Name: </td>
<td><%=oNode.selectSingleNode("COLONY_NAME").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("STATE_ABBR")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">State ABBR: </td>

<td><%=oNode.selectSingleNode("STATE_ABBR").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("TIME_ZONE")) = "IXMLDOMEElement")
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")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">Day Light Saving: </td>

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

<% if(TypeName(oNode.selectSingleNode("OLD_AREA_CODE")) =
"IXMLDOMEElement") then %>
<tr>
<td align="left">Old Area Code: </td>

<td><%=oNode.selectSingleNode("OLD_AREA_CODE").nodeTypedValue%></td>
</tr>
<% end if %>
<% if(TypeName(oNode.selectSingleNode("NEW_AREA_CODE")) =
"IXMLDOMEElement") then %>
<tr>
```

```
<td align="left">New Area Code: </td>
<td><%=oNode.selectSingleNode("NEW_AREA_CODE").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(strPostalCode, 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 & "">" & _
"<PostalCode>" & strPostalCode & "</PostalCode>" & _
"<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 Varient
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
{
$PostalCode = $_POST["PostalCode"];
$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
$msg = '<PostalCodeWorld_Mexico xmlns="http://ws.fraudlabs.com/">
<PostalCode>'.$PostalCode.'</PostalCode>
<LICENSE>'.$License.'</LICENSE>
</PostalCodeWorld_Mexico>';

// Create a new SOAP object
$soapclient = new
soapclient_nusoap('http://ws.fraudlabs.com/PostalCodeWorldMexico_WebSer
vice.asmx?wsdl','wsdl');
if (!$soapclient->fault) {
$result = $soapclient->call('PostalCodeWorld_Mexico', array($msg));
if(!isset($result['Error'])) {
// No Error. Retrieve results.
echo "POSTALCODE = " . $result["POSTAL_CODE"] . "<br>";
echo "STATE NAME = " . $result["STATE_NAME"] . "<br>";
echo "CITY NAME= " . $result["CITY_NAME"] . "<br>";
echo "COLONY NAME = " . $result["COLONY_NAME"] . "<br>";
echo "STATE ABBR = " . $result["STATE_ABBR"] . "<br>";
echo "LATITUDE = " . $result["LATITUDE"] . "<br>";
echo "LONGITUDE = " . $result["LONGITUDE"] . "<br>";
echo "TIME_ZONE = " . $result["TIME_ZONE"] . "<br>";
echo "DAY LIGHT SAVING = " . $result["DAY_LIGHT_SAVING"] . "<br>";
echo "OLD AREA CODE = " . $result["OLD_AREA_CODE"] . "<br>";
echo "NEW AREA CODE = " . $result["NEW_AREA_CODE"] . "<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 PostalCode, String
LICENSE)
            String result = proxy.PostalCodeWorld_Mexico("01000", "<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 ();
        }
    }
}
```