

BrowserObject™ Web Service

~~ Browser Detection Made Very Easy ~~

1. Overview	3
1.1 Overview of the BrowserObject Web Service	4
1.2 Front End of BrowserObject 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 BrowserObject Web Service.....	7
1.3.1. How to get client's User Agent string	7
1.3.2. Record Matching.....	8
1.4 Process Flow Overview	9
2. Implementation	10
2.1 Implementation Details	11
2.2 Basic Parameters - Input	16
2.3 Basic Parameters - Output	16
2.4 List of possible value for MESSAGE field	19
2.5 Structure of the Request and Response.....	20
2.5.1. Request by SOAP	20
2.5.2. Response by SOAP	20
2.5.3. Request by HTTP-GET	21
2.5.4. Response by HTTP-GET	22
2.5.5. Request by HTTP-POST.....	23
2.5.6. Response by HTTP-POST	23
3. Design Information	25
3.1 Placement of BrowserObject Web Service	25
Appendix II : Sample Code	26

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 BrowserObject Web Services.

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

Email : sales@browserobject.com

1.1 Overview of the BrowserObject Web Service

BrowserObject delivers scalable web services solutions that help you to determine your online visitors' browser's information. We are able to identify user's Web Browser's name, version, platform, and to check if the browser supports SSL Protocol, frame, JavaScript, PNG, Compress GZip and so on using proprietary User Agent lookup database and technology.

BrowserObject Web Service is hosted, programmable XML Web Service that allows exchange of data between systems. 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 BrowserObject.com, the leading browser detection provider.

BrowserObject Web Service is very easy to set up and use. You may understand your visitors better, as well as preventing and minimizing Web site failures that due to technical issues by detecting your clients' Web Browser's capabilities within a minute of time. From the information that the web service return to you, you may also get rid of browser incompatibility problems by routing your visitor to pages optimized for their particular browser.

Key Features Include:

- Detects Major Search Engines
- Detects Web Browsers Type and Version
- Detects Wireless Phones, PDAs and Other Devices
- Detects New Browser and Automatic Updates
- Provides a complete XML-based Web Services API
- Contains sample code examples for ease integration

1.2 Front End of BrowserObject 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 BrowserObject 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/browserobjectsamples.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/BrowserObjectWebServiceClientVB.zip>
 - ii. Microsoft ASP.Net - C#:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientCSharp.zip>
 - iii. Microsoft ASP 3.0:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientAsp.zip>
 - iv. Java/ Apache:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientJava.zip>
 - v. PHP:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientPHP.zip>
 - vi. Python:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientPython.zip>
 - vii. Perl:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientPerl.zip>
 - viii. ColdFusion MX:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientColdFusion.zip>
 - ix. Access 2000:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientAccess.zip>
 - x. Excel 2000:
<http://www.fraudlabs.com/samplecode/BrowserObjectWebServiceClientExcel.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 BrowserObject™ Browser Detection, 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 BrowserObject™ Browser Detection, 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/browserobjectwebservice.asmx>
- II. Click on "BrowserObject"
- 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 BrowserObject™ Browser Detection, 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 BrowserObject™ Browser Detection, 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 BrowserObject 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 BrowserObject Web Service.

The process works as follows:

1. User submits User Agent that to be analyzed
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 perform the User Agent check
4. Returns and display the Web browser’s information

1.3.1. How to get client’s User Agent string

ASP:

```
...  
Dim strUserAgent = Request.ServerVariables("HTTP_USER_AGENT")  
....
```

PHP:

```
...  
$strUserAgent = $_SERVER['HTTP_USER_AGENT'];  
...
```

JavaScript:

```
...  
var strUserAgent = navigator.userAgent;  
...
```

VB.NET:

```
...  
Dim strUserAgent As String =  
Request.ServerVariables("HTTP_USER_AGENT")  
...
```

C#:

```
...  
string strUserAgent = Request.ServerVariables["HTTP_USER_AGENT"];  
...
```

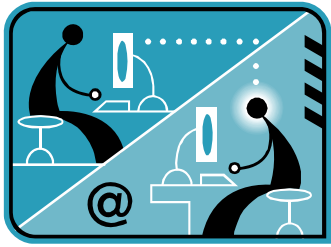
Perl:

```
...  
require LWP::UserAgent;  
my $strUserAgent = LWP::UserAgent->new; $ua->timeout(10); $ua->  
>env_proxy;  
...
```

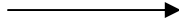
1.3.2. Record Matching

First, we analyze the User Agent String. Then we use our own algorithm to match case and match with the browser's information.

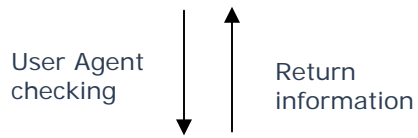
1.4 Process Flow Overview



Step 1
Visitors visit websites



Step 2
Visitor's User Agent String captured by merchant



Step 3
BrowserObject Web Service
(Browser's 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 BrowserObject Web Service.

The implementation section covers the following topics:

- BrowserObject Function

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

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

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

Email : sales@BrowserObject.com

2.1 Implementation Details

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

SOAP – BrowserObject Function Request

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

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

SOAP – BrowserObject 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>
    <BrowserObjectResponse xmlns="http://ws.fraudlabs.com/">
      <BrowserObjectResult>
        <ActiveXControls>string</ActiveXControls>
        <Alpha>string</Alpha>
        <AOL>string</AOL>
        <AOLVersion>string</AOLVersion>
        <AuthenticcodeUpdate>string</AuthenticcodeUpdate>
        <BackgroundSounds>string</BackgroundSounds>
        <Beta>string</Beta>
        <Browser>string</Browser>
        <CDF>string</CDF>
        <CharHeightAvail>string</CharHeightAvail>
        <CharWidthAvail>string</CharWidthAvail>
        <CLRVersion>string</CLRVersion>
        <CompressGZIP>string</CompressGZIP>
```

```
<CookiesSupported>string</CookiesSupported>
<Crawler>string</Crawler>
<CSSVersion>string</CSSVersion>
<EmailCrawler>string</EmailCrawler>
<FileUpload>string</FileUpload>
<FontColor>string</FontColor>
<FontSize>string</FontSize>
<Frames>string</Frames>
<Gecko>string</Gecko>
<NetscapeGold>string</NetscapeGold>
<HDML>string</HDML>
<IFrames>string</IFrames>
<IsBanned>string</IsBanned>
<IsMobileDevice>string</IsMobileDevice>
<IsSyndicationReader>string</IsSyndicationReader>
<JavaAppletsSupported>string</JavaAppletsSupported>
<JavaScriptSupported>string</JavaScriptSupported>
<MajorVersion>string</MajorVersion>
<MinorVersion>string</MinorVersion>
<MinorVersionLetter>string</MinorVersionLetter>
<MouseOver>string</MouseOver>
<MSN>string</MSN>
<NETCLR>string</NETCLR>
<OSInformation>string</OSInformation>
<PNGSupported>string</PNGSupported>
<Platform>string</Platform>
<IsProxy>string</IsProxy>
<SSLSupported>string</SSLSupported>
<SSLActive>string</SSLActive>
<SSLCipherSuite>string</SSLCipherSuite>
<TableBackgroundColor>string</TableBackgroundColor>
<TableBackgroundImage>string</TableBackgroundImage>
<Tables>string</Tables>
<UserLanguage>string</UserLanguage>
<VBScriptSupported>string</VBScriptSupported>
<WAPDeviceModel>string</WAPDeviceModel>
<WAPGateway>string</WAPGateway>
<WAPMaximumDeckSize>string</WAPMaximumDeckSize>
<WAPSubscriberID>string</WAPSubscriberID>
<WAP>string</WAP>
<Win16>string</Win16>
<Win32>string</Win32>
<Win64>string</Win64>
<XMLSupported>string</XMLSupported>
<UserAgent>string</UserAgent>
<Version>string</Version>
<CREDITSAVAILABLE>string</CREDITSAVAILABLE>
<MESSAGE>string</MESSAGE>
</BrowserObjectResult>
</BrowserObjectResponse>
</soap:Body>
</soap:Envelope>
```

HTTP GET- BrowserObject Function Request

```
GET
/ws.fraudlabs.com_non_ssl/browserobjectwebservice.asmx/BrowserObject?US
ERAGENTSTRING=string&LICENSE=string HTTP/1.1
Host: ws.fraudlabs.com
```

HTTP GET- BrowserObject Function Response

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

<?xml version="1.0" encoding="utf-8"?>
<BROWSEROBJECT xmlns="http://ws.fraudlabs.com/">
  <ActiveXControls>string</ActiveXControls>
  <Alpha>string</Alpha>
  <AOL>string</AOL>
  <AOLVersion>string</AOLVersion>
  <AuthenticcodeUpdate>string</AuthenticcodeUpdate>
  <BackgroundSounds>string</BackgroundSounds>
  <Beta>string</Beta>
  <Browser>string</Browser>
  <CDF>string</CDF>
  <CharHeightAvail>string</CharHeightAvail>
  <CharWidthAvail>string</CharWidthAvail>
  <CLRVersion>string</CLRVersion>
  <CompressGZIP>string</CompressGZIP>
  <CookiesSupported>string</CookiesSupported>
  <Crawler>string</Crawler>
  <CSSVersion>string</CSSVersion>
  <EmailCrawler>string</EmailCrawler>
  <FileUpload>string</FileUpload>
  <FontColor>string</FontColor>
  <FontSize>string</FontSize>
  <Frames>string</Frames>
  <Gecko>string</Gecko>
  <NetscapeGold>string</NetscapeGold>
  <HDML>string</HDML>
  <IFrames>string</IFrames>
  <IsBanned>string</IsBanned>
  <IsMobileDevice>string</IsMobileDevice>
  <IsSyndicationReader>string</IsSyndicationReader>
  <JavaAppletsSupported>string</JavaAppletsSupported>
  <JavaScriptSupported>string</JavaScriptSupported>
  <MajorVersion>string</MajorVersion>
  <MinorVersion>string</MinorVersion>
  <MinorVersionLetter>string</MinorVersionLetter>
  <MouseOver>string</MouseOver>
  <MSN>string</MSN>
  <NETCLR>string</NETCLR>
  <OSInformation>string</OSInformation>
  <PNGSupported>string</PNGSupported>
  <Platform>string</Platform>
  <IsProxy>string</IsProxy>
  <SSLSupported>string</SSLSupported>
  <SSLActive>string</SSLActive>
```

```

<SSLCipherSuite>string</SSLCipherSuite>
<TableBackgroundColor>string</TableBackgroundColor>
<TableBackgroundImage>string</TableBackgroundImage>
<Tables>string</Tables>
<UserLanguage>string</UserLanguage>
<VBScriptSupported>string</VBScriptSupported>
<WAPDeviceModel>string</WAPDeviceModel>
<WAPGateway>string</WAPGateway>
<WAPMaximumDeckSize>string</WAPMaximumDeckSize>
<WAPSubscriberID>string</WAPSubscriberID>
<WAP>string</WAP>
<Win16>string</Win16>
<Win32>string</Win32>
<Win64>string</Win64>
<XMLSupported>string</XMLSupported>
<UserAgent>string</UserAgent>
<Version>string</Version>
<CREDITSAVAILABLE>string</CREDITSAVAILABLE>
<MESSAGE>string</MESSAGE>
</BROWSEROBJECT>

```

HTTP POST- BrowserObject Function Request

```

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

USERAGENTSTRING=string&LICENSE=string

```

HTTP POST- BrowserObject Function Response

```

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

<?xml version="1.0" encoding="utf-8"?>
<BROWSEROBJECT xmlns="http://ws.fraudlabs.com/">
  <ActiveXControls>string</ActiveXControls>
  <Alpha>string</Alpha>
  <AOL>string</AOL>
  <AOLVersion>string</AOLVersion>
  <AuthenticcodeUpdate>string</AuthenticcodeUpdate>
  <BackgroundSounds>string</BackgroundSounds>
  <Beta>string</Beta>
  <Browser>string</Browser>
  <CDF>string</CDF>
  <CharHeightAvail>string</CharHeightAvail>
  <CharWidthAvail>string</CharWidthAvail>
  <CLRVersion>string</CLRVersion>
  <CompressGZIP>string</CompressGZIP>

```

```
<CookiesSupported>string</CookiesSupported>
<Crawler>string</Crawler>
<CSSVersion>string</CSSVersion>
<EmailCrawler>string</EmailCrawler>
<FileUpload>string</FileUpload>
<FontColor>string</FontColor>
<FontSize>string</FontSize>
<Frames>string</Frames>
<Gecko>string</Gecko>
<NetscapeGold>string</NetscapeGold>
<HDML>string</HDML>
<IFrames>string</IFrames>
<IsBanned>string</IsBanned>
<IsMobileDevice>string</IsMobileDevice>
<IsSyndicationReader>string</IsSyndicationReader>
<JavaAppletsSupported>string</JavaAppletsSupported>
<JavaScriptSupported>string</JavaScriptSupported>
<MajorVersion>string</MajorVersion>
<MinorVersion>string</MinorVersion>
<MinorVersionLetter>string</MinorVersionLetter>
<MouseOver>string</MouseOver>
<MSN>string</MSN>
<NETCLR>string</NETCLR>
<OSInformation>string</OSInformation>
<PNGSupported>string</PNGSupported>
<Platform>string</Platform>
<IsProxy>string</IsProxy>
<SSLSupported>string</SSLSupported>
<SSLActive>string</SSLActive>
<SSLCipherSuite>string</SSLCipherSuite>
<TableBackgroundColor>string</TableBackgroundColor>
<TableBackgroundImage>string</TableBackgroundImage>
<Tables>string</Tables>
<UserLanguage>string</UserLanguage>
<VBScriptSupported>string</VBScriptSupported>
<WAPDeviceModel>string</WAPDeviceModel>
<WAPGateway>string</WAPGateway>
<WAPMaximumDeckSize>string</WAPMaximumDeckSize>
<WAPSubscriberID>string</WAPSubscriberID>
<WAP>string</WAP>
<Win16>string</Win16>
<Win32>string</Win32>
<Win64>string</Win64>
<XMLSupported>string</XMLSupported>
<UserAgent>string</UserAgent>
<Version>string</Version>
<CREDITSAVAILABLE>string</CREDITSAVAILABLE>
<MESSAGE>string</MESSAGE>
</BROWSEROBJECT>
```

A WSDL is available at:

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

A description of the FraudLabs operation is available at:

<http://ws.fraudlabs.com/browserobjectwebservice.asmx?op=BrowserObject>

2.2 Basic Parameters - Input

Field	Format	Description
USERAGENTSTRING	Required	User Agent of Internet visitor
LICENSE	Required	License key for free license and premium users.

2.3 Basic Parameters - Output

Field	Format	Description
BROWSERNAME	String	This property returns the browser's name, e.g. "IE", "Netscape", "Mozilla", "Opera" and so on.
FULLVERSION	String	This property returns a string that representing the version of the entire browser, e.g. "4.04"
GECKO	String	This property returns "True" or "False" to indicate that if the browser is using the Gecko engine
MAJORVERSION	String	This property returns the major version of your visitor's browser, e.g. "7"
MINORVERSION	String	This property returns the minor version of your visitor's browser, e.g. "0".
MINORVERSIONLETTER	String	This property returns the minor version letter of your visitor's browser, e.g. "C".
CRAWLER	String	This property returns "True" or "False" to indicate if the browser is a crawler, spider or any other agents of search engine.
EMAILCRAWLER	String	This property returns "True" or "False" to determine if the visiting agent is a crawler that collects email addresses
ALPHA	String	Alpha returns "True" or "False" to indicate if the browser is an alpha version and still under development
BETA	String	This property returns "True" or "False" to determine if the browser is a beta version
BACKGROUNDOUNDS	String	This property returns "True" or "False" to determine if the browser

		can play background sounds.
AUTHENTICCODEUPDATE	String	This property returns the version of Authenticode (digitally signing code) which the browser supports.
NETSCAPEGOLD	String	This property returns "True" or "False" to determine if the browser is a Gold version of Netscape
ISBANNED	String	This property returns "True" or "False" to determine if the browser is banned user agent.
ISSYNDICATIONREADER	String	This property returns "True" or "False" to determine if the browser is actually an RSS, Atom, or other XML-based feed reader or aggregator
MOUSEOVER	String	This property returns "True" or "False" to indicate if the browser supports the mouse over effects
SSLSUPPORTED	String	This property returns "True" or "False" to indicate if Secure Sockets Layer (SSL) is supported by the browser
SSLACTIVE	String	This property returns "True" or "False" to indicate if your visitor is connected through an active SSL connection. This test is not available on all web servers and platforms
SSLCIPHERSUITE	String	This property returns the SSL cipher suite used for SSL session
CSSSUPPORTED	String	This property returns "True" or "False" to indicate if the browser supports Cascading Style Sheets (CSS)
PNGSUPPORTED	String	This function returns "True" or "False" to determine if the browser supports Portable Network Graphic (PNG) image file format
FONTCOLORSUPPORTED	String	This property returns "True" or "False" to indicate if the browser supports the display of different colored text
FONTSIZESUPPORTED	String	This property returns "True" or "False" that indicates if the browser supports the ability of displaying varying font sizes
TABLESSUPPORTED	String	This property returns "True" or "False" to determine if the browser supports and display HTML table
TABLEBACKGROUNDCOLORSUPPORTED	String	This property returns "True" or "False" to indicate if the browser supports the ability to control the table display color
TABLEBACKGROUNDIMAGESUPPORTED	String	This property returns "True" or "False" to determine if the browser

		supports and display HTML table
FRAMES	String	This property returns "True" or "False" that determine if the browser supports frames
IFRAMES	String	This property returns "True" or "False" to determine if the browser supports the HTML IFrame tag
AOLVERSION	String	This property returns the version number of AOL if the browser is an America Online branded browser
AOL	String	This property returns "True" or "False" to indicate if the browser is an America Online branded browser.
ISPROXY	String	This property returns "True" of "False" to determine if your visitor is behind a proxy server
COMPRESSGZIP	String	This property returns "True" or "False" to determine if browser accepts compressed content in GZip format
MSNNETWORK	String	This property returns "True" or "False" to indicate if your visitor is from MSN network
ISMOBILEDEVICE	String	This property returns "True" of "False" to determine if your visitor is using recognized mobile device.
WAP	String	This property returns "True" of "False" to determine if the device supports WML and WAP (Wireless Application Protocol).
HDML	String	This property returns "True" or "False" to determine if the wireless devices support WAP's predecessor.
WAPDEVICEMODEL	String	This property returns the model of known device
WAPGATEWAY	String	This property returns the information of the UP.Link gateway.
WAPMAXIMUMDECKSIZE	String	This property returns the supported maximum number of bytes in a deck.
WAPSUBSCRIBERID	String	This property returns the unique subscriber ID of WAP visitor.
PLATFORM	String	This property returns the platform of your visitor's machine, e.g." Win XP".
WIN16	String	This property returns "True" of "False" to determine if the browser is running on a 16-bit operating system.
WIN32	String	This property returns "True" of "False" to determine if the browser

		is running on a 32-bit operating system.
WIN64	String	This property returns "True" or "False" to determine if the browser is running on a 64-bit operating system.
NETCLR	String	This property returns "True" or "False" indication if .NET Common Language Runtime (CLR) is installed.
CLRVERSION	String	This property returns the version of the .NET Common Language Runtime (CLR) if installed, e.g. "2.0.50727".
ACTIVEXCONTROLS	String	This property returns "True" or "False" to indicate if the browser supports ActiveX Controls
JAVASCRIPTSUPPORTED	String	This property returns "True" or "False" to determine if the browser supports JavaScript
VBSCRIPTSUPPORTED	String	This property returns "True" or "False" to determine if the browser supports VBScript
XMLSUPPORTED	String	This property returns "True" or "False" to determine if the browser supports XML (extended markup language)
FILEUPLOAD	String	This property returns "True" or "False" to determine if the browser enables file upload to a server.
USERLANGUAGE	String	This property returns the language code corresponding to the regional settings from your visitor, e.g. "ENGLISH"
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

BrowserObject Web Service Error Message
• Invalid license key
• User Agent 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>
    <BrowserObject xmlns="http://ws.fraudlabs.com/">
      <USERAGENTSTRING>Mozilla/4.0 (compatible; MSIE 7.0; Windows NT
5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; MS-RTC LM 8; .NET CLR
3.0.04506.648; .NET CLR 3.5.21022)</USERAGENTSTRING>
      <LICENSE>xx-xxxx-xxxx</LICENSE>
    </BrowserObject>
  </soap:Body>
</soap:Envelope>
```

2.5.2. Response by SOAP

After posting a valid request, BrowserObject will return several parameters. An example of a response returned by BrowserObject 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>
    <BrowserObjectResponse xmlns="http://ws.fraudlabs.com/">
      <BrowserObjectResult>
        <ActiveXControls>True</ActiveXControls>
        <Alpha>False</Alpha>
        <AOL>False </AOL>
        <AOLVersion>0</AOLVersion>
        <AuthenticcodeUpdate>Unknown</AuthenticcodeUpdate>
        <BackgroundSounds>True</BackgroundSounds>
        <Beta>False</Beta>
        <Browser>IE</Browser>
        <CDF>True</CDF>
        <CharHeightAvail>0</CharHeightAvail>
        <CharWidthAvail>0</CharWidthAvail>
        <CLRVersion>0</CLRVersion>
        <CompressGZIP>True</CompressGZIP>
        <CookiesSupported>True</CookiesSupported>
        <Crawler>False</Crawler>
        <CSSVersion>2</CSSVersion>
        <EmailCrawler>False</EmailCrawler>
        <FileUpload>True</FileUpload>
        <FontColor>True</FontColor>
        <FontSize>True</FontSize>
        <Frames>True</Frames>
```

```

<Gecko>False</Gecko>
<NetscapeGold>False</NetscapeGold>
<HDML>False</HDML>
<IFrames>True</IFrames>
<IsBanned>False</IsBanned>
<IsMobileDevice>False</IsMobileDevice>
<IsSyndicationReader>False</IsSyndicationReader>
<JavaAppletsSupported>True</JavaAppletsSupported>
<JavaScriptSupported>True</JavaScriptSupported>
<MajorVersion>7</MajorVersion>
<MinorVersion>0</MinorVersion>
<MinorVersionLetter>0</MinorVersionLetter>
<MouseOver>True</MouseOver>
<MSN>False</MSN>
<NETCLR>False</NETCLR>
<OSInformation>Unknown</OSInformation>
<PNGSupported>True</PNGSupported>
<Platform>WinXP</Platform>
<IsProxy>False</IsProxy>
<SSLSupported>True</SSLSupported>
<SSLActive>False</SSLActive>
<SSLCipherSuite>Unknown</SSLCipherSuite>
<TableBackgroundColor>True</TableBackgroundColor>
<TableBackgroundImage>True</TableBackgroundImage>
<Tables>True</Tables>
<UserLanguage>ENGLISH</UserLanguage>
<VBScriptSupported>True</VBScriptSupported>
<WAPDeviceModel>Unknown</WAPDeviceModel>
<WAPGateway>Unknown</WAPGateway>
<WAPMaximumDeckSize>0</WAPMaximumDeckSize>
<WAPSubscriberID>Unknown</WAPSubscriberID>
<WAP>False</WAP>
<Win16>False</Win16>
<Win32>True</Win32>
<Win64>False</Win64>
<XMLSupported>True</XMLSupported>
<UserAgent>Mozilla/4.0 (compatible; MSIE 7.0; Windows NT
5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; MS-RTC LM 8; .NET CLR
3.0.04506.648; .NET CLR 3.5.21022)</UserAgent>
<Version>7.0</Version>
<CREDITSAVAILABLE>210</CREDITSAVAILABLE>
<MESSAGE />
</BrowserObjectResult>
</BrowserObjectResponse>
</soap:Body>
</soap:Envelope>

```

2.5.3. Request by HTTP-GET

A valid request posted by user:

```

GET
/ws.fraudlabs.com_non_ssl/BrowserObjectwebservice.asmx/BrowserObject?US
ERAGENTSTRING= Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET
CLR 1.1.4322; .NET CLR 2.0.50727; MS-RTC LM 8; .NET CLR
3.0.04506.648; .NET CLR 3.5.21022)&LICENSE=XX-XXXX-XXXX HTTP/1.1

```

```
Host: ws.fraudlabs.com
```

2.5.4. Response by HTTP-GET

An example of a response returned by BrowserObject 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"?>
<BROWSEROBJECT xmlns="http://ws.fraudlabs.com/">
  <ActiveXControls>True</ActiveXControls>
  <Alpha>False</Alpha>
  <AOL>False </AOL>
  <AOLVersion>0</AOLVersion>
  <AuthenticodeUpdate>Unknown</AuthenticodeUpdate>
  <BackgroundSounds>True</BackgroundSounds>
  <Beta>False</Beta>
  <Browser>IE</Browser>
  <CDF>True</CDF>
  <CharHeightAvail>0</CharHeightAvail>
  <CharWidthAvail>0</CharWidthAvail>
  <CLRVersion>0</CLRVersion>
  <CompressGZIP>True</CompressGZIP>
  <CookiesSupported>True</CookiesSupported>
  <Crawler>False</Crawler>
  <CSSVersion>2</CSSVersion>
  <EmailCrawler>False</EmailCrawler>
  <FileUpload>True</FileUpload>
  <FontColor>True</FontColor>
  <FontSize>True</FontSize>
  <Frames>True</Frames>
  <Gecko>False</Gecko>
  <NetscapeGold>False</NetscapeGold>
  <HDML>False</HDML>
  <IFrames>True</IFrames>
  <IsBanned>False</IsBanned>
  <IsMobileDevice>False</IsMobileDevice>
  <IsSyndicationReader>False</IsSyndicationReader>
  <JavaAppletsSupported>True</JavaAppletsSupported>
  <JavaScriptSupported>True</JavaScriptSupported>
  <MajorVersion>7</MajorVersion>
  <MinorVersion>0</MinorVersion>
  <MinorVersionLetter>0</MinorVersionLetter>
  <MouseOver>True</MouseOver>
  <MSN>False</MSN>
  <NETCLR>False</NETCLR>
  <OSInformation>Unknown</OSInformation>
  <PNGSupported>True</PNGSupported>
  <Platform>WinXP</Platform>
  <IsProxy>False</IsProxy>
  <SSLSupported>True</SSLSupported>
  <SSLActive>False</SSLActive>
  <SSLCipherSuite>Unknown</SSLCipherSuite>
  <TableBackgroundColor>True</TableBackgroundColor>
```

```

<TableBackgroundImage>True</TableBackgroundImage>
<Tables>True</Tables>
<UserLanguage>ENGLISH</UserLanguage>
<VBScriptSupported>True</VBScriptSupported>
<WAPDeviceModel>Unknown</WAPDeviceModel>
<WAPGateway>Unknown</WAPGateway>
<WAPMaximumDeckSize>0</WAPMaximumDeckSize>
<WAPSubscriberID>Unknown</WAPSubscriberID>
<WAP>False</WAP>
<Win16>False</Win16>
<Win32>True</Win32>
<Win64>False</Win64>
<XMLSupported>True</XMLSupported>
<UserAgent>Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET
CLR 1.1.4322; .NET CLR 2.0.50727; MS-RTC LM 8; .NET CLR
3.0.04506.648; .NET CLR 3.5.21022)</UserAgent>
<Version>7.0</Version>
<CREDITSAVAILABLE>210</CREDITSAVAILABLE>
<MESSAGE />
</BROWSEROBJECT>

```

2.5.5. Request by HTTP-POST

A valid request posted by user:

```

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

USERAGENTSTRING=Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET
CLR 1.1.4322; .NET CLR 2.0.50727; MS-RTC LM 8; .NET CLR
3.0.04506.648; .NET CLR 3.5.21022)&LICENSE=XX-XXXX-XXXX

```

2.5.6. Response by HTTP-POST

An example of a response returned by BrowserObject 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"?>
<BROWSEROBJECT xmlns="http://ws.fraudlabs.com/">
  <ActiveXControls>True</ActiveXControls>
  <Alpha>False</Alpha>
  <AOL>False </AOL>
  <AOLVersion>0</AOLVersion>
  <AuthenticcodeUpdate>Unknown</AuthenticcodeUpdate>
  <BackgroundSounds>True</BackgroundSounds>
  <Beta>False</Beta>
  <Browser>IE</Browser>

```

```
<CDF>True</CDF>
<CharHeightAvail>0</CharHeightAvail>
<CharWidthAvail>0</CharWidthAvail>
<CLRVersion>0</CLRVersion>
<CompressGZIP>True</CompressGZIP>
<CookiesSupported>True</CookiesSupported>
<Crawler>False</Crawler>
<CSSVersion>2</CSSVersion>
<EmailCrawler>False</EmailCrawler>
<FileUpload>True</FileUpload>
<FontColor>True</FontColor>
<FontSize>True</FontSize>
<Frames>True</Frames>
<Gecko>False</Gecko>
<NetscapeGold>False</NetscapeGold>
<HDML>False</HDML>
<IFrames>True</IFrames>
<IsBanned>False</IsBanned>
<IsMobileDevice>False</IsMobileDevice>
<IsSyndicationReader>False</IsSyndicationReader>
<JavaAppletsSupported>True</JavaAppletsSupported>
<JavaScriptSupported>True</JavaScriptSupported>
<MajorVersion>7</MajorVersion>
<MinorVersion>0</MinorVersion>
<MinorVersionLetter>0</MinorVersionLetter>
<MouseOver>True</MouseOver>
<MSN>False</MSN>
<NETCLR>False</NETCLR>
<OSInformation>Unknown</OSInformation>
<PNGSupported>True</PNGSupported>
<Platform>WinXP</Platform>
<IsProxy>False</IsProxy>
<SSLSupported>True</SSLSupported>
<SSLActive>False</SSLActive>
<SSLCipherSuite>Unknown</SSLCipherSuite>
<TableBackgroundColor>True</TableBackgroundColor>
<TableBackgroundImage>True</TableBackgroundImage>
<Tables>True</Tables>
<UserLanguage>ENGLISH</UserLanguage>
<VBScriptSupported>True</VBScriptSupported>
<WAPDeviceModel>Unknown</WAPDeviceModel>
<WAPGateway>Unknown</WAPGateway>
<WAPMaximumDeckSize>0</WAPMaximumDeckSize>
<WAPSubscriberID>Unknown</WAPSubscriberID>
<WAP>False</WAP>
<Win16>False</Win16>
<Win32>True</Win32>
<Win64>False</Win64>
<XMLSupported>True</XMLSupported>
<UserAgent>Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET
CLR 1.1.4322; .NET CLR 2.0.50727; MS-RTC LM 8; .NET CLR
3.0.04506.648; .NET CLR 3.5.21022)</UserAgent>
<Version>7.0</Version>
<CREDITSAVAILABLE>210</CREDITSAVAILABLE>
<MESSAGE />
</BROWSEROBJECT>
```

Design Information

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

2.6 Placement of BrowserObject Web Service

As BrowserObject web service provides you very informative details regarding identification of online visitors' browser information by using User Agent, this service can be placed on any forms on your website, offering a seamless integration. In fact, you may already be using this kind of service without realizing it.

Description: Visitors Web browsers information is needed to minimize site failure and to reduce the complexity of troubleshooting jobs.

Level of security: High

Site Type(s):

- Internet Retailers
- Software Developers

Benefits:

- Understand and get your visitor's browser setting
- Prevent site failure due to browser setting mismatch
- Redirect web pages based on your site's requirement
- Eliminating browser errors and to ease the troubleshooting jobs
- Web log statistics and analysis to understand your visitors better

Appendix II : Sample Code

BrowserObject 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/browserobjectsamplescodes.aspx>

i. ASP.NET – VB.NET (SOAP)

```
Private Sub BrowserObjectWebService()  
    Dim x_BrowserObject As New  
    BrowserObjectWebService.BrowserObjectWebService  
    Dim oBrowserObject As New BROWSEROBJECT  
    Try  
        oBrowserObject =  
x_BrowserObject.BrowserObject(Me.txtLookup.Text, Me.txtLicense.Text)  
        Me.txtResult.Text = "UserAgent: " &  
oBrowserObject.UserAgent & vbNewLine  
        Me.txtResult.Text += "ActiveXControls: " &  
oBrowserObject.ActiveXControls & vbNewLine  
        Me.txtResult.Text += "Alpha: " & oBrowserObject.Alpha &  
vbNewLine  
        Me.txtResult.Text += "AOL: " & oBrowserObject.AOL &  
vbNewLine  
        Me.txtResult.Text += "AOLVersion: " &  
oBrowserObject.AOLVersion & vbNewLine  
        Me.txtResult.Text += "AuthenticcodeUpdate: " &  
oBrowserObject.AuthenticcodeUpdate & vbNewLine  
        Me.txtResult.Text += "BackgroundSounds: " &  
oBrowserObject.BackgroundSounds & vbNewLine  
        Me.txtResult.Text += "BETA: " & oBrowserObject.Beta &  
vbNewLine  
        Me.txtResult.Text += "Browser: " & oBrowserObject.Browser &  
vbNewLine  
        Me.txtResult.Text += "CDF: " & oBrowserObject.CDF &  
vbNewLine  
        Me.txtResult.Text += "CharHeightAvail: " &  
oBrowserObject.CharHeightAvail & vbNewLine  
        Me.txtResult.Text += "CharWidthAvail: " &  
oBrowserObject.CharWidthAvail & vbNewLine  
        Me.txtResult.Text += "CLRVersion: " &  
oBrowserObject.CLRVersion & vbNewLine  
        Me.txtResult.Text += "CompressGZIP: " &  
oBrowserObject.CompressGZIP & vbNewLine  
        Me.txtResult.Text += "CookiesSupported: " &  
oBrowserObject.CookiesSupported & vbNewLine  
        Me.txtResult.Text += "Crawler: " & oBrowserObject.Crawler &  
vbNewLine  
        Me.txtResult.Text += "CSSVersion: " &  
oBrowserObject.CSSVersion & vbNewLine  
        Me.txtResult.Text += "EmailCrawler: " &  
oBrowserObject.EmailCrawler & vbNewLine  
        Me.txtResult.Text += "FileUpload: " &
```

```
oBrowserObject.FileUpload & vbNewLine
    Me.txtResult.Text += "FontColor: " &
oBrowserObject.FontColor & vbNewLine
    Me.txtResult.Text += "FontSize: " & oBrowserObject.FontSize &
vbNewLine
    Me.txtResult.Text += "Frames: " & oBrowserObject.Frames &
vbNewLine
    Me.txtResult.Text += "Gecko: " & oBrowserObject.Gecko &
vbNewLine
    Me.txtResult.Text += "NetscapeGold: " &
oBrowserObject.NetscapeGold & vbNewLine
    Me.txtResult.Text += "HDML: " & oBrowserObject.HDML &
vbNewLine
    Me.txtResult.Text += "IFrames: " & oBrowserObject.IFrames &
vbNewLine
    Me.txtResult.Text += "IsBanned: " & oBrowserObject.IsBanned &
vbNewLine
    Me.txtResult.Text += "IsMobileDevice: " &
oBrowserObject.IsMobileDevice & vbNewLine
    Me.txtResult.Text += "IsSyndicationReader: " &
oBrowserObject.IsSyndicationReader & vbNewLine
    Me.txtResult.Text += "JavaAppletsSupported: " &
oBrowserObject.JavaAppletsSupported & vbNewLine
    Me.txtResult.Text += "JavaScriptSupported: " &
oBrowserObject.JavaScriptSupported & vbNewLine
    Me.txtResult.Text += "MajorVersion: " &
oBrowserObject.MajorVersion & vbNewLine
    Me.txtResult.Text += "MinorVersion: " &
oBrowserObject.MinorVersion & vbNewLine
    Me.txtResult.Text += "MinorVersionLetter: " &
oBrowserObject.MinorVersionLetter & vbNewLine
    Me.txtResult.Text += "MouseOver: " &
oBrowserObject.MouseOver & vbNewLine
    Me.txtResult.Text += "MSN: " & oBrowserObject.MSN &
vbNewLine
    Me.txtResult.Text += "NETCLR: " & oBrowserObject.NETCLR &
vbNewLine
    Me.txtResult.Text += "OSInformation: " &
oBrowserObject.OSInformation & vbNewLine
    Me.txtResult.Text += "PNGSupported: " &
oBrowserObject.PNGSupported & vbNewLine
    Me.txtResult.Text += "Platform: " & oBrowserObject.Platform &
vbNewLine
    Me.txtResult.Text += "IsProxy: " & oBrowserObject.IsProxy &
vbNewLine
    Me.txtResult.Text += "SSLSupported: " &
oBrowserObject.SSLSupported & vbNewLine
    Me.txtResult.Text += "SSLActive: " &
oBrowserObject.SSLActive & vbNewLine
    Me.txtResult.Text += "SSLCipherSuite: " &
oBrowserObject.SSLCipherSuite & vbNewLine
    Me.txtResult.Text += "TableBackgroundColor: " &
oBrowserObject.TableBackgroundColor & vbNewLine
    Me.txtResult.Text += "TableBackgroundImage: " &
oBrowserObject.TableBackgroundImage & vbNewLine
    Me.txtResult.Text += "Tables: " & oBrowserObject.Tables &
vbNewLine
```

```

    Me.txtResult.Text += "UserLanguage: " &
oBrowserObject.UserLanguage & vbNewLine
    Me.txtResult.Text += "VBScriptSupported: " &
oBrowserObject.VBScriptSupported & vbNewLine
    Me.txtResult.Text += "WAPDeviceModel: " &
oBrowserObject.WAPDeviceModel & vbNewLine
    Me.txtResult.Text += "WAPGateway: " &
oBrowserObject.WAPGateway & vbNewLine
    Me.txtResult.Text += "WAPMaximumDeckSize: " &
oBrowserObject.WAPMaximumDeckSize & vbNewLine
    Me.txtResult.Text += "WAPSubscriberID: " &
oBrowserObject.WAPSubscriberID & vbNewLine
    Me.txtResult.Text += "WAP: " & oBrowserObject.WAP &
vbNewLine
    Me.txtResult.Text += "Win16: " & oBrowserObject.Win16 &
vbNewLine
    Me.txtResult.Text += "Win32: " & oBrowserObject.Win32 &
vbNewLine
    Me.txtResult.Text += "Win64: " & oBrowserObject.Win64 &
vbNewLine
    Me.txtResult.Text += "XMLSupported: " &
oBrowserObject.XMLSupported & vbNewLine
    Me.txtResult.Text += "Version: " & oBrowserObject.Version &
vbNewLine
    Me.txtResult.Text += "CREDITSAVAILABLE: " &
oBrowserObject.CREDITSAVAILABLE & vbNewLine
    Me.txtResult.Text += "MESSAGE: " & oBrowserObject.MESSAGE &
vbNewLine
    Catch ex As Exception
        Response.Write(ex.Message)
    End TryEnd Sub

```

ii. ASP.NET – VB.NET (HTML)

```

Private Sub Lookup()
    Dim x_BrowserObject As New
BrowserObjectWebService.BrowserObjectWebService
    Dim oBrowserObject As New BROWSEROBJECT
    Try
        ' create the request URL
        Dim x_builder As String
        ' add the host element of the URL
        x_builder =
"http://ws.fraudlabs.com/BrowserObjectWebService.asmx/BrowserObject?" &
_
        "USERAGENTSTRING=" & Me.txtLookup.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 = "<ActiveXControls>"
    indexOpen = p_result.IndexOf(strParam) + strParam.Length
    indexClose = p_result.IndexOf("</ActiveXControls>")
    Me.txtResult.Text = "ActiveXControls: "
    If indexClose <> -1 Then
        Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
    End If

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

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

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

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

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

```
Me.txtResult.Text += vbNewLine & "BackgroundSounds: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<Beta>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</Beta>")  
Me.txtResult.Text += vbNewLine & "Beta: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<Browser>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</Browser>")  
Me.txtResult.Text += vbNewLine & "Browser: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<CDF>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</CDF>")  
Me.txtResult.Text += vbNewLine & "CDF: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<CharHeightAvail>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</CharHeightAvail>")  
Me.txtResult.Text += vbNewLine & "CharHeightAvail: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<CharWidthAvail>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</CharWidthAvail>")  
Me.txtResult.Text += vbNewLine & "CharWidthAvail: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<CLRVersion>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</CLRVersion>")  
Me.txtResult.Text += vbNewLine & "CLRVersion: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,
```

```
indexClose - indexOpen)
    End If

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Me.txtResult.Text += vbNewLine & "IsBanned: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<IsMobileDevice>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</IsMobileDevice>")  
Me.txtResult.Text += vbNewLine & "IsMobileDevice: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<IsSyndicationReader>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</IsSyndicationReader>")  
Me.txtResult.Text += vbNewLine & "IsSyndicationReader: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<JavaAppletsSupported>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</JavaAppletsSupported>")  
Me.txtResult.Text += vbNewLine & "JavaAppletsSupported: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<JavaScriptSupported>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</JavaScriptSupported>")  
Me.txtResult.Text += vbNewLine & "JavaScriptSupported: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<MajorVersion>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</MajorVersion>")  
Me.txtResult.Text += vbNewLine & "MajorVersion: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<MinorVersion>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</MinorVersion>")  
Me.txtResult.Text += vbNewLine & "MinorVersion: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,
```

```
indexClose - indexOpen)
    End If

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Me.txtResult.Text += vbNewLine & "TableBackgroundColor: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<TableBackgroundImage>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</TableBackgroundImage>")  
Me.txtResult.Text += vbNewLine & "TableBackgroundImage: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<Tables>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</Tables>")  
Me.txtResult.Text += vbNewLine & "Tables: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<UserLanguage>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</UserLanguage>")  
Me.txtResult.Text += vbNewLine & "UserLanguage: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<VBScriptSupported>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</VBScriptSupported>")  
Me.txtResult.Text += vbNewLine & "VBScriptSupported: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<WAPDeviceModel>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</WAPDeviceModel>")  
Me.txtResult.Text += vbNewLine & "WAPDeviceModel: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,  
indexClose - indexOpen)  
End If  
  
strParam = "<WAPGateway>"  
indexOpen = p_result.IndexOf(strParam) + strParam.Length  
indexClose = p_result.IndexOf("</WAPGateway>")  
Me.txtResult.Text += vbNewLine & "WAPGateway: "  
If indexClose <> -1 Then  
    Me.txtResult.Text += p_result.Substring(indexOpen,
```

```
indexClose - indexOpen)
    End If

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

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

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

    strParam = "<Win16>"
    indexOpen = p_result.IndexOf(strParam) + strParam.Length
    indexClose = p_result.IndexOf("</Win16>")
    Me.txtResult.Text += vbNewLine & "Win16: "
    If indexClose <> -1 Then
        Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
    End If

    strParam = "<Win32>"
    indexOpen = p_result.IndexOf(strParam) + strParam.Length
    indexClose = p_result.IndexOf("</Win32>")
    Me.txtResult.Text += vbNewLine & "Win32: "
    If indexClose <> -1 Then
        Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
    End If

    strParam = "<Win64>"
    indexOpen = p_result.IndexOf(strParam) + strParam.Length
    indexClose = p_result.IndexOf("</Win64>")
    Me.txtResult.Text += vbNewLine & "Win64: "
    If indexClose <> -1 Then
        Me.txtResult.Text += p_result.Substring(indexOpen,
indexClose - indexOpen)
    End If
```

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

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

strParam = "<Version>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</Version>")
Me.txtResult.Text += vbNewLine & "Version: "
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 BrowserObjectWebService()
{
    BrowserObjectWebService.BrowserObjectWebService x_BrowserObject = new
BrowserObjectWebService.BrowserObjectWebService();
    BROWSEROBJECT oBrowserObject = new BROWSEROBJECT();
    try
        {
```

```
oBrowserObject =
x_BrowserObject.BrowserObject(this.txtLookup.Text,
this.txtLicense.Text);
    this.txtResult.Text = "UserAgent: " +
oBrowserObject.UserAgent + "\n";
    this.txtResult.Text += "ActiveXControls: " +
oBrowserObject.ActiveXControls + "\n";
    this.txtResult.Text += "Alpha: " +
oBrowserObject.Alpha + "\n";
    this.txtResult.Text += "AOL: " +
oBrowserObject.AOL + "\n";
    this.txtResult.Text += "AOLVersion: " +
oBrowserObject.AOLVersion + "\n";
    this.txtResult.Text += "AuthenticcodeUpdate: " +
oBrowserObject.AuthenticcodeUpdate + "\n";
    this.txtResult.Text += "BackgroundSounds: " +
oBrowserObject.BackgroundSounds + "\n";
    this.txtResult.Text += "BETA: " +
oBrowserObject.Beta + "\n";
    this.txtResult.Text += "Browser: " +
oBrowserObject.Browser + "\n";
    this.txtResult.Text += "CDF: " +
oBrowserObject.CDF + "\n";
    this.txtResult.Text += "CharHeightAvail: " +
oBrowserObject.CharHeightAvail + "\n";
    this.txtResult.Text += "CharWidthAvail: " +
oBrowserObject.CharWidthAvail + "\n";
    this.txtResult.Text += "CLRVersion: " +
oBrowserObject.CLRVersion + "\n";
    this.txtResult.Text += "CompressGZIP: " +
oBrowserObject.CompressGZIP + "\n";
    this.txtResult.Text += "CookiesSupported: " +
oBrowserObject.CookiesSupported + "\n";
    this.txtResult.Text += "Crawler: " +
oBrowserObject.Crawler + "\n";
    this.txtResult.Text += "CSSVersion: " +
oBrowserObject.CSSVersion + "\n";
    this.txtResult.Text += "EmailCrawler: " +
oBrowserObject.EmailCrawler + "\n";
    this.txtResult.Text += "FileUpload: " +
oBrowserObject.FileUpload + "\n";
    this.txtResult.Text += "FontColor: " +
oBrowserObject.FontColor + "\n";
    this.txtResult.Text += "FontSize: " +
oBrowserObject.FontSize + "\n";
    this.txtResult.Text += "Frames: " +
oBrowserObject.Frames + "\n";
    this.txtResult.Text += "Gecko: " +
oBrowserObject.Gecko + "\n";
    this.txtResult.Text += "NetscapeGold: " +
oBrowserObject.NetscapeGold + "\n";
    this.txtResult.Text += "HDML: " +
oBrowserObject.HDML + "\n";
    this.txtResult.Text += "IFrames: " +
oBrowserObject.IFrames + "\n";
    this.txtResult.Text += "IsBanned: " +
oBrowserObject.IsBanned + "\n";
```

```
        this.txtResult.Text += "IsMobileDevice: " +
oBrowserObject.IsMobileDevice + "\n";
        this.txtResult.Text += "IsSyndicationReader: "
+ oBrowserObject.IsSyndicationReader + "\n";
        this.txtResult.Text += "JavaAppletsSupported: "
+ oBrowserObject.JavaAppletsSupported + "\n";
        this.txtResult.Text += "JavaScriptSupported: "
+ oBrowserObject.JavaScriptSupported + "\n";
        this.txtResult.Text += "MajorVersion: " +
oBrowserObject.MajorVersion + "\n";
        this.txtResult.Text += "MinorVersion: " +
oBrowserObject.MinorVersion + "\n";
        this.txtResult.Text += "MinorVersionLetter: " +
oBrowserObject.MinorVersionLetter + "\n";
        this.txtResult.Text += "MouseOver: " +
oBrowserObject.MouseOver + "\n";
        this.txtResult.Text += "MSN: " +
oBrowserObject.MSN + "\n";
        this.txtResult.Text += "NETCLR: " +
oBrowserObject.NETCLR + "\n";
        this.txtResult.Text += "OSInformation: " +
oBrowserObject.OSInformation + "\n";
        this.txtResult.Text += "PNGSupported: " +
oBrowserObject.PNGSupported + "\n";
        this.txtResult.Text += "Platform: " +
oBrowserObject.Platform + "\n";
        this.txtResult.Text += "IsProxy: " +
oBrowserObject.IsProxy + "\n";
        this.txtResult.Text += "SSLSupported: " +
oBrowserObject.SSLSupported + "\n";
        this.txtResult.Text += "SSLActive: " +
oBrowserObject.SSLActive + "\n";
        this.txtResult.Text += "SSLCipherSuite: " +
oBrowserObject.SSLCipherSuite + "\n";
        this.txtResult.Text += "TableBackgroundColor: "
+ oBrowserObject.TableBackgroundColor + "\n";
        this.txtResult.Text += "TableBackgroundImage: "
+ oBrowserObject.TableBackgroundImage + "\n";
        this.txtResult.Text += "Tables: " +
oBrowserObject.Tables + "\n";
        this.txtResult.Text += "UserLanguage: " +
oBrowserObject.UserLanguage + "\n";
        this.txtResult.Text += "VBScriptSupported: " +
oBrowserObject.VBScriptSupported + "\n";
        this.txtResult.Text += "WAPDeviceModel: " +
oBrowserObject.WAPDeviceModel + "\n";
        this.txtResult.Text += "WAPGateway: " +
oBrowserObject.WAPGateway + "\n";
        this.txtResult.Text += "WAPMaximumDeckSize: " +
oBrowserObject.WAPMaximumDeckSize + "\n";
        this.txtResult.Text += "WAPSubscriberID: " +
oBrowserObject.WAPSubscriberID + "\n";
        this.txtResult.Text += "WAP: " +
oBrowserObject.WAP + "\n";
        this.txtResult.Text += "Win16: " +
oBrowserObject.Win16 + "\n";
        this.txtResult.Text += "Win32: " +
```

```

oBrowserObject.Win32 + "\n";
    this.txtResult.Text += "Win64: " +
oBrowserObject.Win64 + "\n";
    this.txtResult.Text += "XMLSupported: " +
oBrowserObject.XMLSupported + "\n";
    this.txtResult.Text += "Version: " +
oBrowserObject.Version + "\n";
    this.txtResult.Text += "CREDITSAVAILABLE: " +
oBrowserObject.CREDITSAVAILABLE + "\n";
    this.txtResult.Text += "MESSAGE: " +
oBrowserObject.MESSAGE + "\n";
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}

```

iv. ASP.NET – C#.NET (HTML)

```

private void Lookup()
{
    BrowserObjectWebService.BrowserObjectWebService x_BrowserObject = new
    BrowserObjectWebServiceClientCSharp.BrowserObjectWebService.BrowserObjec
    ctWebService();
    BROWSEROBJECT oBrowserObject = new BROWSEROBJECT();
    try
    {
        string x_builder;

        x_builder =
"http://ws.fraudlabs.com/BrowserObjectWebService.asmx/BrowserObject?";
        x_builder += "USERAGENTSTRING=" + this.txtLookup.Text +
"&LICENSE=" + this.txtLicense.Text;

        WebClient x_web_client = new WebClient();
        byte [] x_response =
x_web_client.DownloadData(x_builder.ToString());

        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 = "<ActiveXControls>";

```

```
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</ActiveXControls>");
this.txtResult.Text = "ActiveXControls: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Alpha>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Alpha>");
this.txtResult.Text += "\rAlpha: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<AOL>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</AOL>");
this.txtResult.Text += "\rAOL: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<AOLVersion>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</AOLVersion>");
this.txtResult.Text += "\rAOLVersion: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<AuthenticcodeUpdate>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</AuthenticcodeUpdate>");
this.txtResult.Text += "\rAuthenticcodeUpdate: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<BackgroundSounds>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</BackgroundSounds>");
this.txtResult.Text += "\rBackgroundSounds: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
```

```
}  
  
strParam = "<Beta>";  
indexOpen = p_result.IndexOf(strParam) + strParam.Length;  
indexClose = p_result.IndexOf("</Beta>");  
this.txtResult.Text += "\rBeta: ";  
if (indexClose != -1)  
{  
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
    indexOpen);  
}  
  
strParam = "<Browser>";  
indexOpen = p_result.IndexOf(strParam) + strParam.Length;  
indexClose = p_result.IndexOf("</Browser>");  
this.txtResult.Text += "\rBrowser: ";  
if (indexClose != -1)  
{  
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
    indexOpen);  
}  
  
strParam = "<CDF>";  
indexOpen = p_result.IndexOf(strParam) + strParam.Length;  
indexClose = p_result.IndexOf("</CDF>");  
this.txtResult.Text += "\rCDF: ";  
if (indexClose != -1)  
{  
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
    indexOpen);  
}  
  
strParam = "<CharHeightAvail>";  
indexOpen = p_result.IndexOf(strParam) + strParam.Length;  
indexClose = p_result.IndexOf("</CharHeightAvail>");  
this.txtResult.Text += "\rCharHeightAvail: ";  
if (indexClose != -1)  
{  
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
    indexOpen);  
}  
  
strParam = "<CharWidthAvail>";  
indexOpen = p_result.IndexOf(strParam) + strParam.Length;  
indexClose = p_result.IndexOf("</CharWidthAvail>");  
this.txtResult.Text += "\rCharWidthAvail: ";  
if (indexClose != -1)  
{  
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -  
    indexOpen);  
}  
  
strParam = "<CLRVersion>";  
indexOpen = p_result.IndexOf(strParam) + strParam.Length;  
indexClose = p_result.IndexOf("</CLRVersion>");  
this.txtResult.Text += "\rCLRVersion: ";  
if (indexClose != -1)
```

```
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<CompressGZIP>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</CompressGZIP>");
this.txtResult.Text += "\rCompressGZIP: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<CookiesSupported>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</CookiesSupported>");
this.txtResult.Text += "\rCookiesSupported: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Crawler>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Crawler>");
this.txtResult.Text += "\rCrawler: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<CSSVersion>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</CSSVersion>");
this.txtResult.Text += "\rCSSVersion: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<FileUpload>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</FileUpload>");
this.txtResult.Text += "\rFileUpload: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<FontColor>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
```

```
indexClose = p_result.IndexOf("</FontColor>");
this.txtResult.Text += "\rFontColor: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<FontSize>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</FontSize>");
this.txtResult.Text += "\rFontSize: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Frames>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Frames>");
this.txtResult.Text += "\rFrames: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Gecko>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Gecko>");
this.txtResult.Text += "\rGecko: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<NetscapeGold>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</NetscapeGold>");
this.txtResult.Text += "\rNetscapeGold: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<HDML>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</HDML>");
this.txtResult.Text += "\rHDML: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
```

```
strParam = "<IFrames>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</IFrames>");
this.txtResult.Text += "\rIFrames: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<IsBanned>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</IsBanned>");
this.txtResult.Text += "\rIsBanned: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<IsMobileDevice>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</IsMobileDevice>");
this.txtResult.Text += "\rIsMobileDevice: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<IsSyndicationReader>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</IsSyndicationReader>");
this.txtResult.Text += "\rIsSyndicationReader: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<JavaAppletsSupported>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</JavaAppletsSupported>");
this.txtResult.Text += "\rJavaAppletsSupported: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<JavaScriptSupported>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</JavaScriptSupported>");
this.txtResult.Text += "\rJavaScriptSupported: ";
if (indexClose != -1)
{
```

```
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<MajorVersion>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</MajorVersion>");
this.txtResult.Text += "\rMajorVersion: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<MinorVersion>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</MinorVersion>");
this.txtResult.Text += "\rMinorVersion: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<MinorVersionLetter>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</MinorVersionLetter>");
this.txtResult.Text += "\rMinorVersionLetter: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<EmailCrawler>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</EmailCrawler>");
this.txtResult.Text += "\rEmailCrawler: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<MouseOver>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</MouseOver>");
this.txtResult.Text += "\rMouseOver: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

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

```
this.txtResult.Text += "\rMSN: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<NETCLR>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</NETCLR>");
this.txtResult.Text += "\rNETCLR: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<OSInformation>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</OSInformation>");
this.txtResult.Text += "\rOSInformation: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<PNGSupported>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</PNGSupported>");
this.txtResult.Text += "\rPNGSupported: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Platform>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Platform>");
this.txtResult.Text += "\rPlatform: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<IsProxy>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</IsProxy>");
this.txtResult.Text += "\rIsProxy: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
}
```

```
strParam = "<SSLSupported>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</SSLSupported>");
this.txtResult.Text += "\rSSLSupported: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<SSLActive>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</SSLActive>");
this.txtResult.Text += "\rSSLActive: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<SSLCipherSuite>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</SSLCipherSuite>");
this.txtResult.Text += "\rSSLCipherSuite: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<TableBackgroundColor>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</TableBackgroundColor>");
this.txtResult.Text += "\rTableBackgroundColor: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<TableBackgroundImage>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</TableBackgroundImage>");
this.txtResult.Text += "\rTableBackgroundImage: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Tables>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Tables>");
this.txtResult.Text += "\rTables: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
```

```
indexOpen);
}

strParam = "<UserLanguage>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</UserLanguage>");
this.txtResult.Text += "\rUserLanguage: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<VBScriptSupported>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</VBScriptSupported>");
this.txtResult.Text += "\rVBScriptSupported: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<WAPDeviceModel>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</WAPDeviceModel>");
this.txtResult.Text += "\rWAPDeviceModel: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<WAPGateway>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</WAPGateway>"); this.txtResult.Text +=
"\rWAPGateway: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<WAPMaximumDeckSize>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</WAPMaximumDeckSize>");
this.txtResult.Text += "\rWAPMaximumDeckSize: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<WAPSubscriberID>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</WAPSubscriberID>");
this.txtResult.Text += "\rWAPSubscriberID: ";
```

```
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<WAP>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</WAP>");
this.txtResult.Text += "\rWAP: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Win16>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Win16>");
this.txtResult.Text += "\rWin16: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Win32>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Win32>");
this.txtResult.Text += "\rWin32: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Win64>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Win64>");
this.txtResult.Text += "\rWin64: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<XMLSupported>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</XMLSupported>");
this.txtResult.Text += "\rXMLSupported: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<UserAgent>";
```

```
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</UserAgent>");
this.txtResult.Text += "\rUserAgent: ";
if (indexClose != -1)
{
    this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}

strParam = "<Version>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</Version>");
this.txtResult.Text += "\rVersion: ";
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. PHP

```
<?php
    if (!isset($_POST['submit'])) {} // if page is not submitted to
itself echo the form
    else
    {
        $UAS = $_POST["UAS"];
        $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 = '<BrowserObject xmlns="http://ws.fraudlabs.com/">
<USERAGENTSTRING>'. $UAS. '</USERAGENTSTRING>
<LICENSE>'. $License. '</LICENSE>

</BrowserObject>';

// Create a new SOAP object
$soapclient = new
soapclient_nusoap('http://ws.fraudlabs.com/BrowserObjectWebService.asmx
?wsdl', 'wsdl');
if (!$soapclient->fault) {

    $result = $soapclient->call('BrowserObject', array($msg));
    if(!isset($result['Error'])) {
        // No Error. Retrieve results.

echo "ActiveXControls = " . $result["ActiveXControls"] . "<br>";
echo "Alpha = " . $result["Alpha"] . "<br>";
echo "AOL = " . $result["AOL"] . "<br>";
echo "AOLVersion = " . $result["AOLVersion"] . "<br>";
echo "AuthenticcodeUpdate = " . $result["AuthenticcodeUpdate"] .
"<br>";
echo "BackgroundSounds = " . $result["BackgroundSounds"] .
"<br>";
echo "Beta = " . $result["Beta"] . "<br>";
echo "Browser = " . $result["Browser"] . "<br>";
echo "CDF = " . $result["CDF"] . "<br>";
echo "CharHeightAvail = " . $result["CharHeightAvail"] . "<br>";
echo "CharWidthAvail = " . $result["CharWidthAvail"] . "<br>";
echo "CLRVersion = " . $result["CLRVersion"] . "<br>";
echo "CompressGZIP = " . $result["CompressGZIP"] . "<br>";
echo "CookiesSupported = " . $result["CookiesSupported"] .
"<br>";
echo "Crawler = " . $result["Crawler"] . "<br>";
echo "CSSVersion = " . $result["CSSVersion"] . "<br>";
echo "EmailCrawler = " . $result["EmailCrawler"] . "<br>";
echo "FileUpload = " . $result["FileUpload"] . "<br>";
echo "FontColor = " . $result["FontColor"] . "<br>";
echo "FontSize = " . $result["FontSize"] . "<br>";
echo "Frames = " . $result["Frames"] . "<br>";
echo "Gecko = " . $result["Gecko"] . "<br>";
echo "NetscapeGold = " . $result["NetscapeGold"] . "<br>";
echo "HDML = " . $result["HDML"] . "<br>";
echo "IFrames = " . $result["IFrames"] . "<br>";
echo "IsBanned = " . $result["IsBanned"] . "<br>";
echo "IsMobileDevice = " . $result["IsMobileDevice"] . "<br>";
echo "IsSyndicationReader = " . $result["IsSyndicationReader"] .
"<br>";
echo "JavaAppletsSupported = " .
$result["JavaAppletsSupported"] . "<br>";
```

```
    echo "JavaScriptSupported = " . $result["JavaScriptSupported"] .
"<br>";
    echo "MajorVersion = " . $result["MajorVersion"] . "<br>";
    echo "MinorVersion = " . $result["MinorVersion"] . "<br>";
    echo "MinorVersionLetter = " . $result["MinorVersionLetter"] .
"<br>";
    echo "MouseOver = " . $result["MouseOver"] . "<br>";
    echo "MSN = " . $result["MSN"] . "<br>";
    echo "NETCLR = " . $result["NETCLR"] . "<br>";
    echo "OSInformation = " . $result["OSInformation"] . "<br>";
    echo "PNGSupported = " . $result["PNGSupported"] . "<br>";
    echo "Platform = " . $result["Platform"] . "<br>";
    echo "IsProxy = " . $result["IsProxy"] . "<br>";
    echo "SSLSupported = " . $result["SSLSupported"] . "<br>";
    echo "SSLActive = " . $result["SSLActive"] . "<br>";
    echo "SSLCipherSuite = " . $result["SSLCipherSuite"] . "<br>";
    echo "TableBackgroundColor = " .
$result["TableBackgroundColor"] . "<br>";
    echo "TableBackgroundImage = " .
$result["TableBackgroundImage"] . "<br>";
    echo "Tables = " . $result["Tables"] . "<br>";
    echo "UserLanguage = " . $result["UserLanguage"] . "<br>";
    echo "VBScriptSupported = " . $result["VBScriptSupported"] .
"<br>";
    echo "WAPDeviceModel = " . $result["WAPDeviceModel"] . "<br>";
    echo "WAPGateway = " . $result["WAPGateway"] . "<br>";
    echo "WAPMaximumDeckSize = " . $result["WAPMaximumDeckSize"] .
"<br>";
    echo "WAPSubscriberID = " . $result["WAPSubscriberID"] . "<br>";
    echo "WAP = " . $result["WAP"] . "<br>";
    echo "Win16 = " . $result["Win16"] . "<br>";
    echo "Win32 = " . $result["Win32"] . "<br>";
    echo "Win64 = " . $result["Win64"] . "<br>";
    echo "XMLSupported = " . $result["XMLSupported"] . "<br>";
    echo "UserAgent = " . $result["UserAgent"] . "<br>";
    echo "Version = " . $result["Version"] . "<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
}
}
}
?>
```