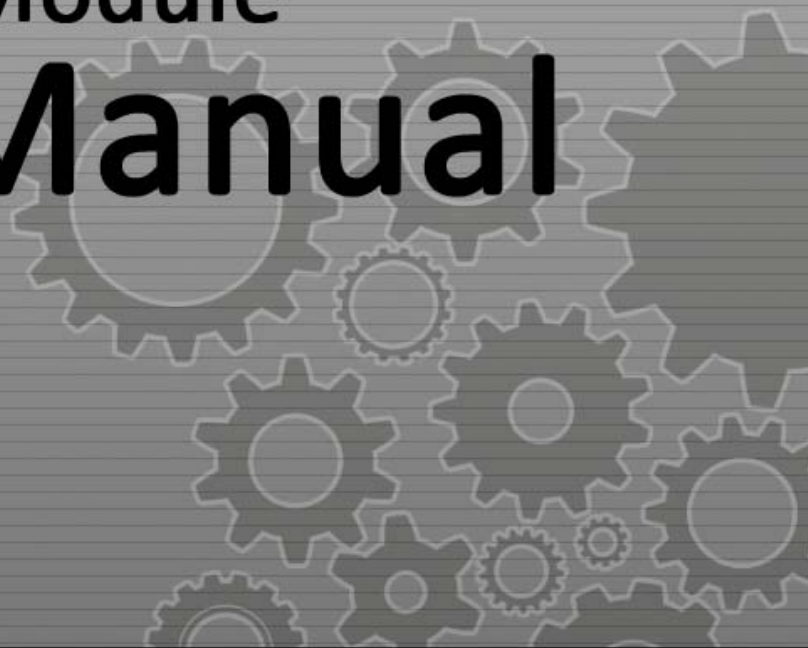


BrowserObject[™]

PHP Module

User Manual



Overview	1
Introduction of the BrowserObject™	2
Quick Start Guide.....	3
Step 1: Download and Installation.....	3
Step 2: Create Secured Folder.....	3
Step 3: Import IP2Location™ BIN Data File	4
Step 4: Import Definition File.....	4
Step 5: Import Java Applet Class File	4
Step 6: Testing	4
Installation Guide	5
Minimum System Requirements	5
Functions in BrowserObject™ PHP	6
Quick Functions List	64
Integration with the IP2Location™ component.....	73
Sample Codes for integration with IP2Location™	74
Legal Disclaimer	75

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™.

If you are new to BrowserObject™, you may proceed to Quick Start Guide for further assistance on importing license key and get started.

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

Email: sales@browserobject.com

Introduction of the BrowserObject™

BrowserObject™ is a module detecting Web Browser capabilities such as supported technologies and version that is running on the clients.

As we are heading towards electronic world today, the Internet boom occurred and business people of all stripes tried to figure out how to monetize the Web. Therefore, great job and smart software packages are crucial for preventing and minimizing site failures that due to technical issues.

BrowserObject™ is the solution for detecting the unique Web Browser settings and system capabilities of each Web site visitor. It enables developers to get rid of browser incompatibility problems by routing their visitor to pages optimized for their particular browser and therefore to ensure that their site works properly and is accessible to the largest possible audience always.

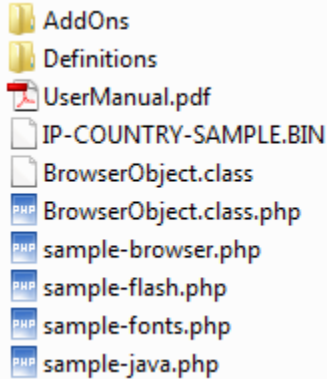
On top of that, BrowserObject™ eases the complexity and time consuming of troubleshooting and supporting jobs and thus providing richer web experiences to users.

Quick Start Guide

This quick start guide will help you get started with BrowserObject™ and will walk you through the steps for using it. The steps are as follows:

Step 1: Download and Installation

Download “browserobjectphp2.zip” from our server and extract all files to your local drive. You should get these files:



Step 2: Create Secured Folder

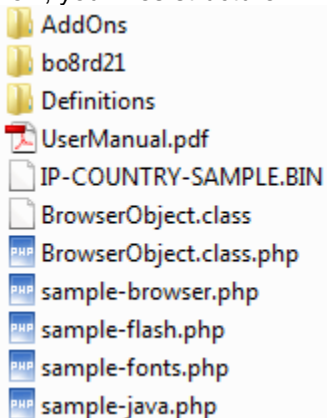
To run BrowserObject™ PHP module, you will need to put your “license.key” into your web directory. Please **DO NOT** change the content of the “license.key” or it will not working.

Note: Your license key is attached in the email we sent you.

Due to some security reasons, you are allowed to place your license key in any web directory that you want. We suggest you to put it into a folder with randomed name such as “bo8rd21”. (This file name will be used for the coming examples)

Note: Please rename your folder to other name instead of “bo8rd21”.

Now, your files structure will be like this:



If you wish to integrate IP2Location™ geolocation technology with BrowserObject™, proceed to Step 3, or other wise you can skip to Step 4.

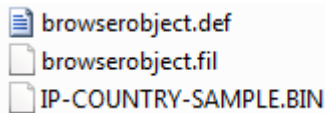
Step 3: Import IP2Location™ BIN Data File

If you are an existing user of IP2Location™, please copy your IP2Location BIN data file to the “bo8rd21” folder.

If you never use IP2Location™ before, you can use the IP2Location sample BIN Data File, “IP-COUNTRY-SAMPLE.BIN” which comes with this package.

Step 4: Import Definition File

BrowserObject™ definition files should store in to secured folder. We move “browserobject.def” and “browserobject.fil” from “Definitions” folder to “bo8rd21” folder. Now, we should have 3 files, “browserobjct.def”, “browserobject.fil” and IP2Location™ BIN data file in our “bo8rd21” folder. Delete the “Definitions” folder after you moved “browserobject.def” and “browserobject.fil”.



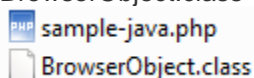
Step 5: Import Java Applet Class File

This step is needed for detecting Java based properties.

BrowserObject™ is able to detect details about the user's Java Virtual Machine configuration and operating system. BrowserObject™ uses an applet as a part of its testing to get this information, and the information is provided through the JavaVendor, JavaVersion, VirtualMachineName, VirtualMachineVersion, MSJVMBuild, OSName, OSVersion, and OSArchitecture properties.

The applet is named “BrowserObject.class” and please makes sure it’s same directory with your detection page.

For example, if you have a page called “sample-java.php, you have to copy the “BrowserObject.class” file to this directory.



Step 6: Testing

You can now review our sample source codes for more information and test it by upload them to your Web server.

Installation Guide

Minimum System Requirements

This section describes the minimum configuration requirements for running BrowserObject™. If the minimum requirements are not met, BrowserObject™ might not be working properly.

Below are the minimum requirements for running BrowserObject™:

Minimum Platform Requirements

- ✓ Microsoft® Windows Family,
- ✓ Unix/Linux based OS,

With IIS or Apache Web server

Minimum Configuration Requirements

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

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

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

Compatible PHP version

- ✓ PHP 4.4+ and
- ✓ PHP 5.2+

Addition Plugins

- ✓ ionCube Loader 3.0+ or
- ✓ Zend Optimizer 3.3.0+

Note: Both ionCube Loader and Zend Optimizer are free. You can get them for free at the following URL:

ionCube Loader: <http://www.ioncube.com/loaders.php>

Zend Optimizer: <http://www.zend.com/en/products/guard/optimizer/>

Working with the BrowserObject™ PHP Module

This section provides information on how to use BrowserObject™ module in your application to determine the exact browser capabilities and settings from your online visitors.

BrowserObject™ is a server side module that runs on your Web server, and is used in web application to detect the Web Browser capability information and settings of your visitors. The default BrowserObject™ is returned with default values. When there is a hit by your visitor, BrowserObject™ will immediately scan through the Web Browser and get the related information by examining the browser's user agent string. If a match is found with the information stored in BrowserObject™ definition file, then BrowserObject™ will return the results.

Functions in BrowserObject™ PHP

Creating BrowserObject™ Object

You are required to include "BrowserObject.class.php" top of your page before you want to use BrowserObject™ detection in your page. You can select either **ONE** of the following codes to include "BrowserObject.class.php" to your php page.

```
<php  
require "<Path to BrowserObject.class.php>";  
?>
```

```
<php  
require_once "<Path to BrowserObject.class.php>";  
?>
```

```
<php  
include "<Path to BrowserObject.class.php>";  
?>
```

```
<php  
include_once "<Path to BrowserObject.class.php>";  
?>
```

Usage example:

```
<php  
require "BrowserObject.class.php";  
?>
```

In this case, we assume that "BrowserObject.class.php" is same directory with our detection page.

Once you include the “BrowserObject.class.php”, you can now using BrowserObject™ by create an object. The syntax is shown as below:

```
&new BrowserObject([string $directory]);
```

Description:

This statement creates a BrowserObject™ object. If `$directory` isn't specified, the object will assume that “browserobject.def”, “browserobject.fil” and “license.key” are same directory with your php page. However, this is not recommended due security issue, your visitors might able to download your definition and license files. So, we created the random name folder.

Usage example:

```
<?php
$data = &new BrowserObject("bo8rd21");
?>
```

We create a BrowserObject™ object and assign it to a variable called `$data`. We specified the directory for “browserobject.def”, “browserobject.fil” and “license.key” is under our secured folder called “bo8rd21”.

Message()

```
string Message(void)
```

Description:

This function will return a string which is the error message and helps in error handling. It is recommended you read the message before displaying the results, so that you are able to handle the errors in an appropriate way. If there is more than once, each message is separated by “,”.

Usage example:

```
<?php
$message = explode(",", $data->Message()); //Split each message by ","
for($i=0; $i<count($message); $i++){ //Display message
    echo $message[$i]. "<br>";
}
?>
```

BrowserObjectVersion()

```
string BrowserObjectVersion(void);
```

Description:

This function returns the version of BrowserObject™ PHP module.

Usage example:

```
<?php  
echo $data->BrowserObjectVersion();  
?>
```

Output:

```
2.0.0
```

BrowserObjectDefinition()

```
string BrowserObjectDefinition(void);
```

Description:

This function returns the version of BrowserObject™ definition is currently in use.

Usage example:

```
<?php  
echo $data->BrowserObjectDefinition();  
?>
```

Output:

```
2007122601
```

AcrobatReaderInstalled()

```
string AcrobatReaderInstalled(void);
```

Description:

This function returns “True” or “False” to indicate if Adobe Acrobat Reader plug-in is installed. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->AcrobatReaderInstalled();  
?>
```

AcrobatReaderVersion()

Description:

This function returns the version of Adobe Acrobat Reader if it's installed, e.g. “8.1.1”. It returns “False” if Adobe Acrobat Reader is not installed. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->AcrobatReaderVersion();  
?>
```

ActiveXControls()

```
string ActiveXControls(void);
```

Description:

This function returns “True” or “False” to indicate if the browser supports ActiveX Controls.

Usage example:

```
<?php  
echo $data->ActiveXControls();  
?>
```

ActiveXEnabled()

```
string ActiveXEnabled(void);
```

Description:

This function returns “True” or “False” to indicate if the browser has ActiveX Controls enabled or disabled.

Usage example:

```
<?php  
echo $data->ActiveXEnabled();  
?>
```

AJAXSupported()

```
string AJAXSupported(void);
```

Description:

This function returns “True” or “False” to determine if the browser supports Asynchronization of JavaScript and XML (AJAX).

Usage example:

```
<?php  
echo $data->AJAXSupported();  
?>
```

Alpha()

```
string Alpha(void);
```

Description:

Alpha returns “True” or “False” to indicate if the browser is an alpha version and still under development.

Usage example:

```
<?php  
echo $data->Alpha();  
?>
```

AOL()

```
string AOL(void);
```

Description:

This function returns “True” or “False” to indicate if the browser is an America Online branded browser.

Usage example:

```
<?php  
echo $data->AOL();  
?>
```

AOLVersion()

```
string AOLVersion(void);
```

Description:

This function returns the version number of AOL if the browser is an America Online branded browser.

Usage example:

```
<?php  
echo $data->AOLVersion();  
?>
```

AuthenticodeUpdate()

```
string AuthenticodeUpdate(void);
```

Description:

This function returns the version of Authenticode (digitally signing code) which the browser supports.

Usage example:

```
<?php  
echo $data->AuthenticodeUpdate();  
?>
```

BackgroundSounds()

```
string BackgroundSounds(void);
```

Description:

This function returns “True” or “False” to determine if the browser can play background sounds.

Usage example:

```
<?php  
echo $data->BackgroundSounds();  
?>
```

Beta()

```
string Beta(void);
```

Description:

This function returns “True” or “False” to determine if the browser is a beta version.

Usage example:

```
<?php  
echo $data->Beta();  
?>
```

Browser()

```
string Browser(void);
```

Description:

This function returns the browser’s name, e.g. “IE”, “Netscape”, “Mozilla”, “Opera” and so on.

Usage example:

```
<?php  
echo $data->Browser();  
?>
```

BrowserBuild()

```
string BrowserBuild(void);
```

Description:

This function returns a string containing the build number of the browser, e.g. "7.0.5730.11".

Usage example:

```
<?php  
echo $data->BrowserBuild();  
?>
```

CDF()

```
string CDF(void);
```

Description:

This function returns "True" or "False" to determine if the browser supports the Channel Definition Format for webcasting.

Usage example:

```
<?php  
echo $data->CDF();  
?>
```

CharHeightAvailable()

```
string CharHeightAvailable(void);
```

Description:

This function returns the number of text lines that can fit on the device without causing scrolling. This applies only to wireless devices, based on Western character set.

Usage example:

```
<?php  
echo $data->CharHeightAvailable();  
?>
```

CharWidthAvailable()

```
string CharWidthAvailable(void);
```

Description:

This function returns the text characters that can fit across one line on the device without causing a line break. This function applies only to wireless devices, based on Western character set.

Usage example:

```
<?php  
echo $data->CharWidthAvailable();  
?>
```

CitrixInstalled()

```
string CitrixInstalled(void);
```

Description:

This function returns “True” or “False” to determine if Citrix Viewer plug-in is installed. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->CitrixInstalled();  
?>
```


City()

```
string City(void);
```

Description:

This function returns the city name for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->City();  
?>
```

Output:

```
New York
```

ClientDate()

```
string ClientDate(void);
```

Description:

This function returns the date of client browser, e.g. "2008-01-25"

Usage example:

```
<?php  
echo $data->ClientDate();  
?>
```

Output:

```
2008-01-25
```

ClientTime()

```
string ClientTime(void);
```

Description:

This function returns the time of client browser, e.g. "10:09:35"

Usage example:

```
<?php  
echo $data->ClientTime();  
?>
```

Output:

```
10:09:35
```

CLRVersion()

```
string CLRVersion(void);
```

Description:

This function returns the version of the .NET Common Language Runtime (CLR) if installed, e.g. "2.0.50727". If you have more than one version of .NET framework installed, it will return the latest version only.

Usage example:

```
<?php  
echo $data->CLRVersion();  
?>
```

ColorDepth()

```
string ColorDepth(void);
```

Description:

This function returns bits of the screen's resolution, e.g. "8", "16" and so on.

Usage example:

```
<?php  
echo $data->ColorDepth();  
?>
```

CompressGZIP()

```
string CompressGZIP(void);
```

Description:

This function returns "True" or "False" to determine if browser accepts compressed content in GZip format.

Usage example:

```
<?php  
echo $data->CompressGZIP();  
?>
```

ConnectionType()

```
string ConnectionType(void);
```

Description:

This function returns a string to determine the type of network connection of your visitor, e.g. "LAN", "MODEM", "OFFLINE" and so on. This function is only available with IE 5 and higher version on Windows.

Usage example:

```
<?php  
echo $data->ConnectionType();  
?>
```

CookiesEnabled()

```
string CookiesEnabled(void);
```

Description:

This function returns “True” or “False” to determine if cookies is enabled. It will return “False” if your visitor disabled the cookies.

Usage example:

```
<?php  
echo $data->CookiesEnabled();  
?>
```

CookiesSupported()

```
string CookiesSupported(void);
```

Description:

This function returns “True” or “False” to determine if browser supports cookies. The result is not affected if visitor enabled or disabled their browser cookies.

Usage example:

```
<?php  
echo $data->CookiesSupported();  
?>
```

CountryLong()

```
string CountryLong(void);
```

Description:

This function returns the country name for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->CountryLong();  
?>
```

Output:

```
United States
```

CountryShort()

```
string CountryShort(void);
```

Description:

This function returns the ISO-3166 two character country name for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->CountryShort();  
?>
```

Output:

```
US
```

Crawler()

```
string Crawler(void);
```

Description:

This function returns “True” or “False” to indicate if the browser is a crawler, spider or any other agents of search engine.

Usage example:

```
<?php  
echo $data->Crawler();  
?>
```

CrystalReportInstalled()

```
string CrystalReportInstalled(void);
```

Description:

This function returns “True” or “False” to indicate if Seagate Crystal Report plug-in has been installed. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->CrystalReportInstalled();  
?>
```

CSSSupported()

```
string CSSSupported(void);
```

Description:

This function returns “True” or “False” to indicate if the browser supports Cascading Style Sheets (CSS).

Usage example:

```
<?php  
echo $data->CSSSupported();  
?>
```

CSSVersion()

```
string CSSVersion(void);
```

Description:

This function returns the version of the supported style sheets.

Usage example:

```
<?php  
echo $data->CSSVersion();  
?>
```

DeviceXDPI()

```
string DeviceXDPI(void);
```

Description:

This function returns actual number of horizontal dots per inch of the display screen, e.g. "120"

Usage example:

```
<?php  
echo $data->DeviceXDPI();  
?>
```

DeviceYDPI()

```
string DeviceYDPI(void);
```

Description:

This function returns the actual number of vertical dots per inch of the display screen, e.g. "120"

Usage example:

```
<?php  
echo $data->DeviceYDPI();  
?>
```

DHTML()

```
string DHTML(void);
```

Description:

This function returns “True” or “False” to determine if the browser supports DHTML script.

Usage example:

```
<?php  
echo $data->DHTML();  
?>
```

DirectorInstalled()

```
string DirectorInstalled(void);
```

Description:

This function returns “True” or “False” to indicate if Adobe Director plug-in is installed. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->DirectorInstalled();  
?>
```

DirectorVersion()

```
string DirectorVersion(void);
```

Description:

This function returns the version of Adobe Director plug-in. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected. It returns “False” if Adobe Director is not installed.

Usage example:

```
<?php  
echo $data->DirectorVersion();  
?>
```


Domain()

```
string Domain(void);
```

Description:

This function returns the internet domain name for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->Domain();  
?>
```

EmailCrawler()

```
string EmailCrawler(void);
```

Description:

This function returns "True" or "False" to determine if the visiting agent is a crawler that collects email addresses.

Usage example:

```
<?php  
echo $data->EmailCrawler();  
?>
```

FileUpload()

```
string FileUpload(void);
```

Description:

This function returns "True" or "False" to determine if the browser enables file upload to a server.

Usage example:

```
<?php  
echo $data->FileUpload();  
?>
```

Firewall()

```
string Firewall(void);
```

Description:

This function returns “True” or “False” to indicate if there is a firewall behind blocking traffic on certain ports.

Usage example:

```
<?php  
echo $data->Firewall();  
?>
```

FlashInstalled()

```
string FlashInstalled(void);
```

Description:

This function returns “True” or “False” to indicate if Adobe Flash plug-in is installed. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->FlashInstalled();  
?>
```

FlashVersion()

```
string FlashVersion(void);
```

Description:

This function returns the version of Adobe Flash plug-in, e.g. "9,0,45,0" for IE on Window, and "9.0 r45" for Mozilla FireFox. It returns false if Flash is not installed.

Usage example:

```
<?php  
echo $data->FlashVersion();  
?>
```

FontColor()

```
string FontColor(void);
```

Description:

This function returns "True" or "False" to indicate if the browser supports the display of different colored text.

Usage example:

```
<?php  
echo $data->FontColor();  
?>
```

FontsInstalled()

```
string FontsInstalled([string $fonts]);
```

Description:

This function returns fonts installed on client machine. If *\$fonts* isn't specified, it will return all fonts installed in the client's machine, each font name is separated by ",". You can specify to detect the fonts you wanted only by separated the font name by ",".

Usage example:

```
<?php
$fonts = $data->FontsInstalled(); //Show all fonts
$font = split(",", $fonts); //Split fonts by ","

for($i=0; $i<count($font); $i++){ //Display each font name
    echo $font[$i]."<br>";
}
?>
```

FontSize()

```
string FontSize(void);
```

Description:

This function returns "True" or "False" that indicates if the browser supports the ability of displaying varying font sizes.

Usage example:

```
<?php
echo $data->FontSize();
?>
```

FontSmoothing()

```
string FontSmoothing(void);
```

Description:

This function returns “True” or “False” that indicates if the Smooth Edges of Screen Fonts option in Control Panel has been selected. This function is only available for IE 5 and later on Windows 32 and 64-bit platforms.

Usage example:

```
<?php  
echo $data->FontSmoothing();  
?>
```

Frames()

```
string Frames(void);
```

Description:

This function returns “True” or “False” that determine if the browser supports frames.

Usage example:

```
<?php  
echo $data->Frames();  
?>
```

FullVersion()

```
string FullVersion(void);
```

Description:

This function returns a string that representing the version of the entire browser, e.g. “4.04”.

Usage example:

```
<?php  
echo $data->FullVersion();  
?>
```

Gecko()

```
string Gecko(void);
```

Description:

This function returns “True” or “False” to indicate that if the browser is using the Gecko engine, e.g Mozilla FireFox and other variants.

Usage example:

```
<?php  
echo $data->Gecko();  
?>
```

GeckoActiveX()

```
string GeckoActiveX(void);
```

Description:

This function returns “True” or “False” to indicate that if the browser supports ActiveX in Gecko engine, e.g Mozilla FireFox and other variants.

Usage example:

```
<?php  
echo $data->GeckoActiveX();  
?>
```

GeckoBuildDate()

```
string GeckoBuildDate(void);
```

Description:

This function returns the build date of the Gecko engine, e.g. “20060612”.

Usage example:

```
<?php  
echo $data->GeckoBuildDate();  
?>
```

HDML()

```
string HDHL(void);
```

Description:

This function returns “True” or “False” to determine if the wireless devices support WAP’s predecessor.

Usage example:

```
<?php  
echo $data->HDML();  
?>
```

Height()

```
string Height(void);
```

Description:

This function returns the height of the the visitor’s screen size resolution.

Usage example:

```
<?php  
echo $data->Height();  
?>
```

HeightAvailable()

```
string HeightAvailable(void);
```

Description:

This function returns the actual amount of height available in the browser window itself.

Usage example:

```
<?php  
echo $data->HeightAvailable();  
?>
```

HighSecurity()

```
string HighSecurity(void);
```

Description:

HighSecurity returns "True" or "False" to determine if the security settings in Internet Explorer is set to High. Please note that BrowserObject™ may not function well and will return you the default value if security is set to High.

Usage example:

```
<?php  
echo $data->HighSecurity();  
?>
```

IFrames()

```
string IFrames(void);
```

Description:

This function returns "True" or "False" to determine if the browser supports the HTML IFrame tag.

Usage example:

```
<?php  
echo $data->IFrames();  
?>
```


ImageEnabled()

```
string ImageEnabled(void);
```

Description:

This function returns “True” or “False” to determine if the browser is configured to display images. This function is not accessible in all of the Web Browser.

Usage example:

```
<?php  
echo $data->ImageEnabled();  
?>
```

InternalIP()

```
string InternalIP(void);
```

Description:

This function returns the non-routable IP address in your visitor's network.

Usage example:

```
<?php  
echo $data->InternalIP();  
?>
```

InternetConnectionWizardVersion()

```
string InternetConnectionWizardVersion(void);
```

Description:

This function returns the version of Internet Connection Wizard, e.g.” 5.0.2918.1900”. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->InternetConnectionWizardVersion();  
?>
```

IPAddress()

```
string IPAddress(void);
```

Description:

This function returns the IP address of your visitor.

Usage example:

```
<?php  
echo $data->IPAddress();  
?>
```

IPIXViewerInstalled()

```
string IPIXViewerInstalled(void);
```

Description:

This function returns "True" or "False" to indicate if the plug-in of IPIX Image Viewer is installed. Not accessible by all of the browsers, it will return "Not tested" if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->IPIXViewerInstalled();  
?>
```

IsBanned()

```
string IsBanned(void);
```

Description:

This function returns "True" or "False" to determine if the browser is banned user agent, typically for documented abuse or a reputation.

Usage example:

```
<?php  
echo $data->IsBanned();  
?>
```

IsMobileDevice()

```
string IsMobileDevice(void);
```

Description:

This function returns “True” or “False” to determine if your visitor is using recognized mobile device.

Usage example:

```
<?php  
echo $data->IsMobileDevice();  
?>
```

ISP()

```
string ISP(void);
```

Description:

This function returns the Internet Service Provider or Company's name for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->ISP();  
?>
```

IsProxy()

```
string IsProxy(void);
```

Description:

This function returns “True” or “False” to determine if your visitor is behind a proxy server. Not all connections through a proxy server can be detected as such.

Usage example:

```
<?php  
echo $data->IsProxy();  
?>
```

IsSyndicationReader()

```
string IsSyndicationReader(void);
```

Description:

This function returns “True” or “False” to determine if the browser is actually an RSS, Atom, or other XML-based feed reader or aggregator.

Usage example:

```
<?php  
echo $data->IsSyndicationReader();  
?>
```

JavaAppletsSupported()

```
string JavaAppletsSupported(void);
```

Description:

This function returns “True” or “False” to determine if Java™ applets are supported by the browser.

Usage example:

```
<?php  
echo $data->JavaAppletsSupported();  
?>
```

JavaEnabled()

```
string JavaEnabled(void);
```

Description:

This function returns “True” or “False” to determine if the browser has Java allowed and enabled.

Usage example:

```
<?php  
echo $data->JavaEnabled();  
?>
```

JavaScriptSupported()

```
string JavaScriptSupported(void);
```

Description:

This function returns “True” or “False” to determine if the browser supports JavaScript.

Usage example:

```
<?php  
echo $data->JavaScriptSupported();  
?>
```

JavaScriptBuild()

```
string JavaScriptBuild(void);
```

Description:

This function returns the build number of JavaScript engine in your visitor’s browser, e.g. “5.7.5730”.

Usage example:

```
<?php  
echo $data->JavaScriptBuild();  
?>
```

JavaScriptEnabled()

```
string JavaScriptEnabled(void);
```

Description:

This function returns “True” or “False” to indicate if the browser has allow JavaScript and enabled it.

Usage example:

```
<?php  
echo $data->JavaScriptEnabled();  
?>
```

JavaScriptVersion()

```
string JavaScriptVersion(void);
```

Description:

This function returns the version of supported JavaScript, e.g. "1.5".

Usage example:

```
<?php
echo $data->JavaScriptVersion();
?>
```

JavaVendor()

```
string JavaVendor(void);
```

Description:

This function returns the Java Virtual Machine version, e.g. "Sun Microsystems Inc.". To test this property, the browser must support JavaApplets and enabled it.

Usage example:

```
<?php
echo $data->JavaVendor();
?>
```

JavaVersion()

```
string JavaVersion(void);
```

Description:

This function returns the version of Java Virtual Machine version, e.g. "1.6.0_03". To test this property, the browser must support JavaApplets and enabled it.

Usage example:

```
<?php
echo $data->JavaVersion();
?>
```

Latitude()

```
string Latitude(void);
```

Description:

This function returns the city's latitude for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->Latitude();  
?>
```

LogicalXDPI()

```
string LogicalXDPI(void);
```

Description:

This function returns the normal number of horizontal dots per inch (DPI) of the display screen, e.g. "120"

Usage example:

```
<?php  
echo $data->LogicalXDPI();  
?>
```

LogicalYDPI()

```
string LogicalYDPI(void);
```

Description:

LogicalXDPI returns the normal number of vertical dots per inch (DPI) of the display screen, e.g. "120"

Usage example:

```
<?php  
echo $data->LogicalYDPI();  
?>
```

Longitude()

```
string Longitude(void);
```

Description:

This function returns the city's longitude for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->Longitude();  
?>
```

MajorVersion()

```
string MajorVersion(void);
```

Description:

This function returns the major version of your visitor's browser, e.g. "7"

Usage example:

```
<?php  
echo $data->MajorVersion();  
?>
```

MapGuideInstalled()

```
string MapGuideInstalled(void);
```

Description:

This function returns "True" or "False" to determine if Autodesk MapGuide plug-in is installed.

Usage example:

```
<?php  
echo $data->MapGuideInstalled();  
?>
```


MinorVersion()

```
string MinorVersion(void);
```

Description:

This function returns the minor version of your visitor's browser, e.g. "0".

Usage example:

```
<?php  
echo $data->MinorVersion();  
?>
```

MinorVersionLetter()

```
string MinorVersionLetter(void);
```

Description:

This function returns the minor version letter of your visitor's browser, e.g. "C".

Usage example:

```
<?php  
echo $data->MinorVersionLetter();  
?>
```

MouseOver()

```
string MouseOver(void);
```

Description:

This function returns "True" or "False" to indicate if the browser supports the mouse over effects.

Usage example:

```
<?php  
echo $data->MouseOver();  
?>
```

MSJVMBuild()

```
string MSJVMBuild(void);
```

Description:

This function returns version of Microsoft Java Virtual Machine Build number, e.g.” 5.0.5000.0”. This function is only available for IE5.0 and later version.

Usage example:

```
<?php  
echo $data->MSJVMBuild();  
?>
```

MSNNetwork()

```
string MSNNetwork(void);
```

Description:

This function returns “True” or “False” to indicate if your visitor is from MSN network.

Usage example:

```
<?php  
echo $data->MSNNetwork();  
?>
```

NETCLR()

```
string NETCLR(void);
```

Description:

This function returns “True” or “False” indication if .NET Common Language Runtime (CLR) is installed.

Usage example:

```
<?php  
echo $data->NETCLR();  
?>
```

NetMeetingBuild()

```
string NetMeetingBuild(void);
```

Description:

This function returns the build number of Microsoft Netmeeting Control, e.g. "4.4.0.3400".

Usage example:

```
<?php
echo $data->NetMeetingBuild();
?>
```

NetscapeGold()

```
string NetscapeGold(void);
```

Description:

This function returns "True" or "False" to determine if the browser is a Gold version of Netscape.

Usage example:

```
<?php
echo $data->NetscapeGold();
?>
```

NetSpeed()

```
string NetSpeed(void);
```

Description:

This function returns the internet connection type for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php
echo $data->NetSpeed();
?>
```

OpenedPorts()

```
string OpenedPorts(int $ports);
```

Description:

This function returns the targeted port is remain opened or closed. You MUST specify the port number to check. You can maximum define up to 8 difference ports for checking. Each port need to separated by ",".

Usage example:

```
<?php  
//Check for port 80, 81, 23, 25 and 3389  
echo $data->OpenedPorts("80, 81, 23, 25, 3389");  
?>
```

Output:

```
80-Open,81-Open,23-Close,25-Open,3389-Close
```

OSArchitecture()

```
string OSArchitecture(void);
```

Description:

This function returns a string indicating the architecture of the operating system. To run this property test, Java and Java Applets have to be enabled.

Usage example:

```
<?php  
echo $data->OSArchitecture();  
?>
```

OSInformation()

```
string OSInformation(void);
```

Description:

This function returns a string indicating the information of operating system, e.g."Microsoft".

Usage example:

```
<?php  
echo $data->OSInformation();  
?>
```

OSName()

```
string OSName(void);
```

Description:

This function returns a string indicating the name of operating system, e.g." Windows XP". To run this property test, Java and Java Applets have to be enabled.

Usage example:

```
<?php  
echo $data->OSName();  
?>
```

OSVersion()

```
string OSVersion(void);
```

Description:

This function returns the version of operating system, e.g." 5.1". To run this property test, Java and Java Applets have to be enabled.

Usage example:

```
<?php  
echo $data->OSVersion();  
?>
```

OutlookVersion()

```
string OutlookVersion(void);
```

Description:

This function returns version of Microsoft Outlook, e.g." 6.0.2900.2180". This function is only available for IE5.0 and later version.

Usage example:

```
<?php  
echo $data->OutlookVersion();  
?>
```

Platform()

```
string Platform(void);
```

Description:

This function returns the platform of your visitor's machine, e.g. "Win XP".

Usage example:

```
<?php  
echo $data->Platform();  
?>
```

PNGSupported()

```
string PNGSupported(void);
```

Description:

This function returns "True" or "False" to determine if the browser supports Portable Network Graphic (PNG) image file format. To run this test, the browser must be able to display images.

Usage example:

```
<?php  
echo $data->PNGSupported();  
?>
```

PopupBlocked()

```
string PopupBlocked(void);
```

Description:

This function returns "True" or "False" to determine if the browser has its Pop-up blocker turned on, either a pop-up is being blocked by the browser itself or through a third party software.

Usage example:

```
<?php  
echo $data->PopupBlocked();  
?>
```

ProxyString()

```
string ProxyString(void);
```

Description:

This function returns proxy information that sent by the proxy server if your visitor is behind a proxy server.

Usage example:

```
<?php  
echo $data->ProxyString();  
?>
```

QuickTimeInstalled()

```
string QuickTimeInstalled(void);
```

Description:

This function return “True” or “False” to indicate if the Apple Quick Time Player plug-in is installed. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->QuickTimeInstalled();  
?>
```

QuickTimeVersion()

```
string QuickTimeVersion(void);
```

Description:

This function returns the version of Apple Quick Time Player plug-in. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->QuickTimeVersion();  
?>
```


RealPlayerInstalled()

```
string RealPlayerInstalled(void);
```

Description:

This function return “True” or “False” to indicate if the Real Player plug-in is installed. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->RealPlayerInstalled();  
?>
```

RealPlayerVersion()

```
string RealPlayerVersion(void);
```

Description:

This function returns the build number of Real Player plug-in, e.g. “6.0.12.1578”. Not accessible by all of the browsers, it will return “Not tested” if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->RealPlayerVersion();  
?>
```

Region()

```
string Region(void);
```

Description:

This function returns the region or state name for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->Region();  
?>
```

Output:

```
Florida
```

ReverseDNS()

```
string ReverseDNS(void);
```

Description:

This function returns the host name or host associated with your visitor's IP Address.

Usage example:

```
<?php  
echo $data->ReverseDNS();  
?>
```

ScreenHeight()

```
string ScreenHeight(void);
```

Description:

This function returns the height of your visitor's screen size resolution, e.g. "600".

Usage example:

```
<?php  
echo $data->ScreenHeight();  
?>
```

ScreenWidth()

```
string ScreenWidth(void);
```

Description:

This function returns the width of your visitor's screen size resolution, e.g. "800".

Usage example:

```
<?php  
echo $data->ScreenWidth();  
?>
```

ServerDate()

```
string ServerDate(void);
```

Description:

This function returns the date of server, e.g. "2008-01-25"

Usage example:

```
<?php  
echo $data->ServerDate();  
?>
```

Output:

```
2008-01-25
```

ServerTime()

```
string ServerTime(void);
```

Description:

This function returns the time of server, e.g. "10:09:35"

Usage example:

```
<?php  
echo $data->ServerTime();  
?>
```

Output:

```
10:09:35
```

SilverlightInstalled()

```
string SilverlightInstalled(void);
```

Description:

This function returns "True" or "False" to indicate if Microsoft Silverlight plug-in is installed. Not accessible by all of the browsers, it will return "Not tested" if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->SilverlightInstalled();  
?>
```

SilverlightVersion()

```
string SilverlightVersion(void);
```

Description:

This function returns the version of Microsoft Silverlight plug-in, e.g. "1.0". Not accessible by all of the browsers, it will return "Not tested" if the plug-in cannot be detected. It returns "False" if Microsoft Silverlight plug-in is not installed.

Usage example:

```
<?php  
echo $data->SilverlightVersion();  
?>
```

SoundCardInstalled()

```
string SoundCardInstalled(void);
```

Description:

This function returns "True" or "False" to indicate if there is any audio output. Not accessible by all of the browsers, it will return "Not tested" if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->SoundCarInstalled();  
?>
```

SSLSupported()

```
string SSLSupported(void);
```

Description:

This function returns "True" or "False" to indicate if Secure Sockets Layer (SSL) is supported by the browser.

Usage example:

```
<?php  
echo $data->SSLSupported();  
?>
```

SSLActive()

```
string SSLActive(void);
```

Description:

This function 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.

Usage example:

```
<?php  
echo $data->SSLActive();  
?>
```

SSLCipherSuite()

```
string SSLCipherSuite(void);
```

Description:

This function returns the SSL cipher suite used for SSL session. This test is only available if your visitor is connected through an active SSL connection.

Usage example:

```
<?php  
echo $data->SSLCipherSuite();  
?>
```

SSLEnabled()

```
string SSLEnabled(void);
```

Description:

This function returns “True” or “False” to indicate if your visitor’s browser enables SSL. To run this test, the browser must be able to display images.

Usage example:

```
<?php  
echo $data->SSLEnabled();  
?>
```

SSLKeySize()

```
string SSLKeySize(void);
```

Description:

This function returns the SSL key size which is supported by your visitor's browser, e.g. "128" bit. This test is not available on all web servers and platforms.

Usage example:

```
<?php  
echo $data->SSLKeySize();  
?>
```

SVGViewerInstalled()

```
string SVGViewerInstalled(void);
```

Description:

This function returns "True" or "False" to indicate if SVG Viewer plug-in is installed. Not accessible by all of the browsers, it will return "Not tested" if the plug-in cannot be detected.

Usage example:

```
<?php  
echo $data->SVGViewerInstalled();  
?>
```

SystemLanguage()

```
string SystemLanguage(void);
```

Description:

This function returns the language edition of your visitor's operating system in language code format, e.g "en-us".

Usage example:

```
<?php  
echo $data->SystemLanguage();  
?>
```

TableBackgroundColor()

```
string TableBackgroundColor(void);
```

Description:

This function returns “True” or “False” to indicate if the browser supports the ability to control the table display color.

Usage example:

```
<?php  
echo $data->TableBackgroundColor();  
?>
```

TableBackgroundImage()

```
string TableBackgroundImage(void);
```

Description:

This function returns “True” or “False” to indicate if the browser supports and displays table background image.

Usage example:

```
<?php  
echo $data->TableBackgroundImage();  
?>
```

Tables()

```
string Tables(void);
```

Description:

This function returns “True” or “False” to determine if the browser supports and display HTML table.

Usage example:

```
<?php  
echo $data->Tables();  
?>
```


TextSize()

```
string TextSize(void);
```

Description:

This function returns string to indicate the size of text that shown in your visitor's browser, e.g. "Medium", "Larger" and so on. This function is only available with IE5 and later.

Usage example:

```
<?php  
echo $data->TextSize();  
?>
```

TimeZone()

```
string TimeZone(void);
```

Description:

This function returns the UTC time zone for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->TimeZone();  
?>
```

TimeZoneDifference()

```
string TimeZoneDifference(void);
```

Description:

This function returns the time zone difference between the server's time zone and the time zone where the visitor's browser is running in, e.g. "-8".

Usage example:

```
<?php  
echo $data->TimeZoneDifference();  
?>
```

TotalFontsInstalled()

```
string TotalFontsInstalled(void);
```

Description:

This function returns total fonts installed on the client machine.

Usage example:

```
<?php  
echo $data->TotalFontsInstalled();  
?>
```

URLReferrer()

```
string URLReferrer(void);
```

Description:

This function returns the URL of the referring page if any.

Usage example:

```
<?php  
echo $data->URLReferrer();  
?>
```

UserLanguage()

```
string UserLanguage(void);
```

Description:

This function returns the language code corresponding to the regional settings from your visitor, e.g. "EN-US".

Usage example:

```
<?php  
echo $data->UserLanguage();  
?>
```

VBScriptSupported()

```
string VBScriptSupported(void);
```

Description:

This function returns “True” or “False” to determine if the browser supports VBScript.

Usage example:

```
<?php  
echo $data->VBScriptSupported();  
?>
```

VBScriptBuild()

```
string VBScriptBuild(void);
```

Description:

This function returns the build number of VBScript engine in your visitor’s browser, e.g. “5.7.5730”.

Usage example:

```
<?php  
echo $data->VBScriptBuild();  
?>
```

ViewpointInstalled()

```
string ViewpointInstalled(void);
```

Description:

This function returns “True” or “False” to indicate if the Viewpoint plug-in is installed.

Usage example:

```
<?php  
echo $data->ViewpointInstalled();  
?>
```

VirtualMachineName()

```
string VirtualMachineName(void);
```

Description:

This function returns the name of Java Virtual Machine, e.g. "Java HotSpot(TM) Client VM". To run this property test, Java and Java Applets have to be enabled.

Usage example:

```
<?php  
echo $data->VirtualMachineName();  
?>
```

VirtualMachineVersion()

```
string VirtualMachineVersion(void);
```

Description:

This function returns version of Java Virtual Machine, e.g. "1.6.0_03-b05". To run this property test, Java and Java Applets have to be enabled.

Usage example:

```
<?php  
echo $data->VirtualMachineVersion();  
?>
```

WAPDeviceModel()

```
string WAPDeviceModel(void);
```

Description:

This function returns the model of known device.

Usage example:

```
<?php  
echo $data->WAPDeviceModel();  
?>
```

WAPGateway()

```
string WAPGateway(void);
```

Description:

This function returns the information of the UP.Link gateway.

Usage example:

```
<?php  
echo $data->WAPGateway();  
?>
```

WAPMaximumDeckSize()

```
string WAPMaximumDeckSize(void);
```

Description:

This function returns the supported maximum number of bytes in a deck.

Usage example:

```
<?php  
echo $data->WAPMaximumDeckSize();  
?>
```

WAPSubscriberID()

```
string WAPSubscriberID(void);
```

Description:

This function returns the unique subscriber ID of WAP visitor.

Usage example:

```
<?php  
echo $data->WAPSubscriberID();  
?>
```

WidthAvailable()

```
string WidthAvailable(void);
```

Description:

This function returns the actual amount of width available in the browser window itself.

Usage example:

```
<?php
echo $data->WidthAvailable();
?>
```

Win16()

```
string Win16(void);
```

Description:

This function returns “True” or “False” to determine if the browser is running on a 16-bit operating system.

Usage example:

```
<?php
echo $data->Win16();
?>
```

Win32()

```
string Win32(void);
```

Description:

This function returns “True” or “False” to determine if the browser is running on a 32-bit operating system.

Usage example:

```
<?php
echo $data->Win32();
?>
```

Win64()

```
string Win64(void);
```

Description:

This function returns “True” of “False” to determine if the browser is running on a 64-bit operating system.

Usage example:

```
<?php  
echo $data->Win64();  
?>
```

WMPInstalled ()

```
string WMPInstalled(void);
```

Description:

This function returns “True” of “False” to determine if Windows Media Player plug-in is installed.

Usage example:

```
<?php  
echo $data->WMPInstalled();  
?>
```

WMPVersion()

```
string WMPVersion(void);
```

Description:

This function returns the version of Windows Media Player plug-in, e.g. “11.0.5721.5230”.

Usage example:

```
<?php  
echo $data->WMPVersion();  
?>
```

WindowsMessengerInstalled()

```
string WindowsMessengerInstalled(void);
```

Description:

This function returns “True” or “False” to determine if the Windows Messenger is installed.

Usage example:

```
<?php  
echo $data->WindowsMessengerInstalled();  
?>
```

WindowsMessengerStatus()

```
string WindowsMessengerStatus(void);
```

Description:

This function returns the status of Windows Messenger, e.g. “Busy”.

Usage example:

```
<?php  
echo $data->WindowsMessengerStatus();  
?>
```

XMLSupported()

```
string XMLSupported(void);
```

Description:

This function returns “True” or “False” to determine if the browser supports XML (extended markup language).

Usage example:

```
<?php  
echo $data->XMLSupported();  
?>
```


XMLVersion()

```
string XMLVersion(void);
```

Description:

This function returns the version of supported XML, e.g. "6".

Usage example:

```
<?php  
echo $data->XMLVersion();  
?>
```

ZipCode()

```
string ZipCode(void);
```

Description:

This function returns the US ZIP code for your visitor. This function is only available with the purchase of IP2Location's data product.

Usage example:

```
<?php  
echo $data->ZipCode();  
?>
```

Quick Functions List

Function Name	Description
Message()	This function will return a string which is the error message and helps in error handling.
BrowserObjectVersion()	This function returns the version of BrowserObject™ PHP module.
BrowserObjectDefinition()	This function returns the version of BrowserObject™ definition is currently in use.
AcrobatReaderInstalled()	This function returns “True” or “False” to indicate if Adobe Acrobat Reader plug-in is installed.
AcrobatReaderVersion()	This function returns the version of Adobe Acrobat Reader if it's installed, e.g. “8.1.1”.
ActiveXControls()	This function returns “True” or “False” to indicate if the browser supports ActiveX Controls.
ActiveXEnabled()	This function returns “True” or “False” to indicate if the browser has ActiveX Controls enabled or disabled.
AJAXSupported()	This function returns “True” or “False” to determine if the browser supports Asynchronization of JavaScript and XML (AJAX).
Alpha()	Alpha returns “True” or “False” to indicate if the browser is an alpha version and still under development.
AOL()	This function returns “True” or “False” to indicate if the browser is an America Online branded browser.
AOLVersion()	This function returns the version number of AOL if the browser is an America Online branded browser.
AuthenticcodeUpdate()	This function returns the version of Authenticcode (digitally signing code) which the browser supports.
BackgroundSounds()	This function returns “True” or “False” to determine if the browser can play background sounds.
Beta()	This function returns “True” or “False” to determine if the browser is a beta version.
Browser()	This function returns the browser's name, e.g. “IE”, “Netscape”, “Mozilla”, “Opera” and so on.
BrowserBuild()	This function returns a string containing the build number of the browser, e.g. “7.0.5730.11”.
CDF()	This function returns “True” or “False” to determine if the browser supports the Channel Definition Format for webcasting.

CharHeightAvailable()	This function returns the number of text lines that can fit on the device without causing scrolling.
CharWidthAvailable()	This function returns the text characters that can fit across one line on the device without causing a line break.
CitrixInstalled()	This function returns "True" or "False" to determine if Citrix Viewer plug-in is installed.
City()	This function returns the city name for your visitor.
ClientDate()	This function returns the date of client browser, e.g. "2008-01-25"
ClientTime()	This function returns the time of client browser, e.g. "10:09:35"
CLRVersion()	This function returns the version of the .NET Common Language Runtime (CLR) if installed, e.g. "2.0.50727".
ColorDepth()	This function returns bits of the screen's resolution, e.g. "8", "16" and so on.
CompressGZIP()	This function returns "True" or "False" to determine if browser accepts compressed content in GZip format.
ConnectionType()	This function returns a string to determine the type of network connection of your visitor, e.g "LAN", "MODEM", "OFFLINE" and so on.
CookiesEnabled()	This function returns "True" or "False" to determine if cookies is enabled. It will return "False" if your visitor disabled the cookies.
CookiesSupported()	This function returns "True" or "False" to determine if browser supports cookies.
CountryLong()	This function returns the country name for your visitor.
CountryShort()	This function returns the ISO-3166 two character country name for your visitor.
Crawler()	This function returns "True" or "False" to indicate if the browser is a crawler, spider or any other agents of search engine.
CrystalReportInstalled()	This function returns "True" or "False" to indicate if Seagate Crystal Report plug-in has been installed.
CSSSupported()	This function returns "True" or "False" to indicate if the browser supports Cascading Style Sheets (CSS).
CSSVersion()	This function returns the version of the supported style sheets.
DeviceXDPI()	This function returns actual number of horizontal dots per inch of the display screen, e.g. "120"

DeviceYDPI()	This function returns the actual number of vertical dots per inch of the display screen, e.g. "120"
DHTML()	This function returns "True" or "False" to determine if the browser supports DHTML script.
DirectorInstalled()	This function returns "True" or "False" to indicate if Adobe Director plug-in is installed.
DirectorVersion()	This function returns the version of Adobe Director plug-in.
Domain()	This function returns the internet domain name for your visitor.
EmailCrawler()	This function returns "True" or "False" to determine if the visiting agent is a crawler that collects email addresses.
FileUpload()	This function returns "True" or "False" to determine if the browser enables file upload to a server.
Firewall()	This function returns "True" or "False" to indicate if there is a firewall behind blocking traffic on certain ports.
FlashInstalled()	This function returns "True" or "False" to indicate if Adobe Flash plug-in is installed.
FlashVersion()	This function returns the version of Adobe Flash plug-in.
FontColor()	This function returns "True" or "False" to indicate if the browser supports the display of different colored text.
FontsInstalled()	This function returns fonts installed on client machine.
FontSize()	This function returns "True" or "False" that indicates if the browser supports the ability of displaying varying font sizes.
FontSmoothing()	This function returns "True" or "False" that indicates if the Smooth Edges of Screen Fonts option in Control Panel has been selected.
Frames()	This function returns "True" or "False" that determine if the browser supports frames.
FullVersion()	This function returns a string that representing the version of the entire browser, e.g. "4.04".
Gecko()	This function returns "True" or "False" to indicate that if the browser is using the Gecko engine.
GeckoActiveX()	This function returns "True" or "False" to indicate that if the browser supports ActiveX in Gecko engine, e.g Mozilla FireFox and other variants.
GeckoBuildDate()	This function returns the build date of the Gecko engine, e.g. "20060612".

HDML()	This function returns "True" or "False" to determine if the wireless devices support WAP's predecessor.
Height()	This function returns the height of the the visitor's screen size resolution.
HeightAvailable()	This function returns the actual amount of height available in the browser window itself.
HighSecurity()	HighSecurity returns "True" or "False" to determine if the security settings in Internet Explorer is set to High.
IFrames()	This function returns "True" or "False" to determine if the browser supports the HTML IFrame tag.
ImageEnabled()	This function returns "True" or "False" to determine if the browser is configured to display images. This function is not accessible in all of the Web Browser.
InternalIP()	This function returns the non-routable IP address in your visitor's network.
InternetConnectionWizardVersion()	This function returns the version of Internet Connection Wizard.
IPAddress()	This function returns the IP address of your visitor.
IPIXViewerInstalled()	This function returns "True" or "False" to indicate if the plug-in of IPIX Image Viewer is installed.
IsBanned()	This function returns "True" of "False" to determine if the browser is banned user agent.
IsMobileDevice()	This function returns "True" of "False" to determine if your visitor is using recognized mobile device.
ISP()	This function returns the Internet Service Provider or Company's name for your visitor.
IsProxy()	This function returns "True" of "False" to determine if your visitor is behind a proxy server.
IsSyndicationReader()	This function returns "True" of "False" to determine if the browser is actually an RSS, Atom, or other XML-based feed reader or aggregator.
JavaAppletsSupported()	This function returns "True" of "False" to determine if Java™ applets are supported by the browser.
JavaEnabled()	This function returns "True" of "False" to determine if the browser has Java allowed and enabled.
JavaScriptSupported()	This function returns "True" of "False" to determine if the browser supports JavaScript.

JavaScriptBuild()	This function returns the build number of JavaScript engine in your visitor's browser, e.g. "5.7.5730".
JavaScriptEnabled()	This function returns "True" or "False" to indicate if the browser has allow JavaScript and enabled it.
JavaScriptVersion()	This function returns the version of supported JavaScript, e.g. "1.5".
JavaVendor()	This function returns the Java Virtual Machine version, e.g. "Sun Microsystems Inc.".
JavaVersion()	This function returns the version of Java Virtual Machine version, e.g. "1.6.0_03".
Latitude()	This function returns the city's latitude for your visitor.
LogicalXDPI()	This function returns the normal number of horizontal dots per inch (DPI) of the display screen, e.g. "120"
LogicalYDPI()	LogicalXDPI returns the normal number of vertical dots per inch (DPI) of the display screen, e.g. "120"
Longitude()	This function returns the city's longitude for your visitor.
MajorVersion()	This function returns the major version of your visitor's browser, e.g. "7"
MapGuideInstalled()	This function returns "True" or "False" to determine if Autodesk MapGuide plug-in is installed.
MinorVersion()	This function returns the minor version of your visitor's browser, e.g. "0".
MinorVersionLetter()	This function returns the minor version letter of your visitor's browser, e.g. "C".
MouseOver()	This function returns "True" or "False" to indicate if the browser supports the mouse over effects.
MSJVMBuild()	This function returns version of Microsoft Java Virtual Machine Build number, e.g. "5.0.5000.0".
MSNNetwork()	This function returns "True" or "False" to indicate if your visitor is from MSN network.
NETCLR()	This function returns "True" or "False" indication if .NET Common Language Runtime (CLR) is installed.
NetMeetingBuild()	This function returns the build number of Microsoft Netmeeting Control, e.g. "4.4.0.3400".
NetscapeGold()	This function returns "True" or "False" to determine if the browser is a Gold version of Netscape.

NetSpeed()	This function returns the internet connection type for your visitor.
OpenedPorts()	This function returns the targeted port is remain opened or closed.
OSArchitecture()	This function returns a string indicating the architecture of the operating system.
OSInformation()	This function returns a string indicating the information of operating system, e.g."Microsoft".
OSName()	This function returns a string indicating the name of operating system, e.g." Windows XP".
OSVersion()	This function returns the version of operating system, e.g." 5.1".
OutlookVersion()	This function returns version of Microsoft Outlook, e.g." 6.0.2900.2180".
Platform()	This function returns the platform of your visitor's machine, e.g." Win XP".
PNGSupported()	This fuction returns "True" or "False" to determine if the browser supports Portable Network Graphic (PNG) image file format.
PopupBlocked()	This function returns "True" or "False" to determine if the browser has its Pop-up blocker turned on, either a pop-up is being blocked by the browser itself or through a third party software.
ProxyString()	This function returns proxy information that sent by the proxy server if your visitor is behind a proxy server.
QuickTimeInstalled()	This function return "True" or "False" to indicate if the Apple Quick Time Player plug-in is installed.
QuickTimeVersion()	This function returns the version of Apple Quick Time Player plug-in.
RealPlayerInstalled()	This function return "True" or "False" to indicate if the Real Player plug-in is installed.
RealPlayerVersion()	This function returns the build number of Real Player plug-in, e.g. "6.0.12.1578".
Region()	This function returns the region or state name for your visitor.
ReverseDNS()	function returns the host name or host associated with your visitor's IP Address.

ScreenHeight()	This function returns the height of your visitor's screen size resolution, e.g. "600".
ScreenWidth()	This function returns the width of your visitor's screen size resolution, e.g. "800".
ServerDate()	This function returns the date of server, e.g. "2008-01-25"
ServerTime()	This function returns the time of server, e.g. "10:09:35"
SilverlightInstalled()	This function returns "True" or "False" to indicate if Microsoft Silverlight plug-in is installed.
SilverlightVersion()	This function returns the version of Microsoft Silverlight plug-in, e.g. "1.0".
SoundCardInstalled()	This function returns "True" or "False" to indicate if there is any audio output.
SSLSupported()	This function returns "True" or "False" to indicate if Secure Sockets Layer (SSL) is supported by the browser.
SSLActive()	This function 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()	This function returns the SSL cipher suite used for SSL session.
SSLEnabled()	This function returns "True" or "False" to indicate if your visitor's browser enables SSL. To run this test, the browser must be able to display images.
SSLKeySize()	This function returns the SSL key size which is supported by your visitor's browser, e.g. "128" bit.
SVGViewerInstalled()	This function returns "True" or "False" to indicate if SVG Viewer plug-in is installed.
SystemLanguage()	This function returns the language edition of your visitor's operating system in language code format, e.g "en-us".
TableBackgroundColor()	This function returns "True" or "False" to indicate if the browser supports the ability to control the table display color.
TableBackgroundImage()	This function returns "True" or "False" to determine if the browser supports and display HTML table.
Tables()	This function returns "True" or "False" to determine if the browser supports and display HTML table.
TextSize()	This function returns string to indicate the size of text that shown in your visitor's browser.

TimeZone()	This function returns the UTC time zone for your visitor.
TimeZoneDifference()	This function returns the time zone difference between the server's time zone and the time zone where the visitor's browser is running in, e.g. "-8".
TotalFontsInstalled()	This function returns total fonts installed on the client machine.
URLReferrer()	This function returns the URL of the referring page if any.
UserLanguage()	This function returns the language code corresponding to the regional settings from your visitor, e.g. "EN-US".
VBScriptSupported()	This function returns "True" or "False" to determine if the browser supports VBScript.
VBScriptBuild()	This function returns the build number of VBScript engine in your visitor's browser, e.g. "5.7.5730".
ViewpointInstalled()	This function returns "True" or "False" to indicate if the Viewpoint plug-in is installed.
VirtualMachineName()	This function returns the name of Java Virtual Machine, e.g. "Java HotSpot(TM) Client VM".
VirtualMachineVersion()	This function returns version of Java Virtual Machine, e.g. "1.6.0_03-b05".
WAPDeviceModel()	This function returns the model of known device.
WAPGateway()	This function returns the information of the UP.Link gateway.
WAPMaximumDeckSize()	This function returns the supported maximum number of bytes in a deck.
WAPSubscriberID()	This function returns the unique subscriber ID of WAP visitor.
WidthAvailable()	This function returns the actual amount of width available in the browser window itself.
Win16()	This function returns "True" or "False" to determine if the browser is running on a 16-bit operating system.
Win32()	This function returns "True" or "False" to determine if the browser is running on a 32-bit operating system.
Win64()	This function returns "True" or "False" to determine if the browser is running on a 64-bit operating system.
WMPInstalled ()	This function returns "True" or "False" to determine if Windows Media Player plug-in is installed.

WMPVersion()	This function returns the version of Windows Media Player plug-in, e.g. "11.0.5721.5230".
WindowsMessengerInstalled()	This function returns "True" of "False" to determine if the Windows Messenger is installed.
WindowsMessengerStatus()	This function returns the status of Windows Messenger, e.g. "Busy".
XMLSupported()	This function returns "True" of "False" to determine if the browser supports XML (extended markup language).
XMLVersion()	This function returns the version of supported XML, e.g. "6".
ZipCode()	This function returns the US ZIP code for your visitor. This function is only available with the purchase of IP2Location's data product.

Integration with the IP2Location™ component

If you have subscribed BrowserObject™ Component Basic Edition or onwards, and if you are existing user of IP2Location™, now you can easily integrate the two components and you are able to get your user's geolocation information at the same time.

To place IP2Location™ in BrowserObject™, import BIN data file in your web directory, you can choose to place in sub directory if you want, but you will have to define the path in your function.

To define the path of IP2Location BIN data file, please use the following code:

```
<?php
echo $data->SetIP2LocationPath( ("bo8rd21/IP-COUNTRY-REGION-CITY-LATITUDE-
LONGITUDE-ZIPCODE-TIMEZONE-ISP-DOMAIN-NETSPPEED.BIN" ) ;
?>
```

This code defined the IP2Location BIN file is located in the folder called "bo8rd21" and the BIN file is **"IP-COUNTRY-REGION-CITY-LATITUDE-LONGITUDE-ZIPCODE-TIMEZONE-ISP-DOMAIN-NETSPPEED.BIN"**.

For more information, please visit www.ip2location.com or contact your IP2Location representative:

Email: sales@ip2location.com

Sample Codes for integration with IP2Location™

Usage example:

```
<?php
echo $data->CountryShort();
?>
```

Output:

US

Usage example:

```
<?php
echo $data->CountryLong();
?>
```

Output:

United States

Legal Disclaimer

Welcome to **BrowserObject.com** Web site operated by **Hexasoft Development Sdn. Bhd. ("HDSB")**. Our services subject to the following Terms of Service ("TOS"), which we may modify this TOS at any time, and such amendments shall be effective immediately upon posting of the amended TOS. In addition, when using particular **BrowserObject.com** services, you and **HDSB** shall be subject to any posted guidelines or rules applicable to such services which may be posted from time to time. All such guidelines or rules are hereby incorporated by reference into the TOS.

Accordingly, you agree to review this TOS periodically, and your continued use of our services shall be deemed your acceptance of the amended TOS.

Description of Service

HDSB currently provides customers with access to a number of online resources, including software components and browser information ("Product"). Unless explicitly stated otherwise, any new features that augments or enhances the existing Service, including the release of new **BrowserObject.com** products, shall be subject to the TOS. You understand and agree that neither **HDSB** nor its affiliates are responsible or liable for any claim, loss, or damage directly or indirectly resulting from your use of the Product or the information or the resources contained on or accessible through this site. **HDSB** and its affiliates will attempt to keep the information and the resources on or accessible through this site timely and accurate, but make no guarantees, and disclaim any implied warranty or representation, about its accuracy, relevance, timeliness, completeness, or appropriateness for a particular purpose.

Links

The Service may provide, or third parties may provide, links to other Internet sites or resources. Be aware that **HDSB** does not control, makes no guarantees about, and disclaims any express or implied representations or warranties about the accuracy, relevance, timeliness, completeness, or appropriateness for a particular purpose of the information or the resources contained on these or any other Internet sites. Further, the inclusion of these links is not intended to reflect the importance of these other sites or endorse any of the views expressed in, or products or services offered by, these other sites. Your access and use these and any other Internet sites, including the information, services, products, materials, and any other resources contained on them, solely at your own risk.

Disclaimer of Warranties

You expressly understand and agree that:

Your use of The Service and The Product are at your sole risk. The Service/The Product is provided on an "As Is" and "As Available" basis. **HDSB** expressly disclaims all warranties of any kind, whether express or implied, including, but not limited to the implied warranties of merchantability, fitness for a particular purpose and non-infringement. **HDSB** makes no warranty that: (a) The Service/Product will meet your requirements, (b) The Service/Product will be uninterrupted, timely, secure, or error-free, (c) The results that may be obtained from the use of The Service/Product will be accurate or reliable, (d) The quality of any products, services, information, or other material obtained by you through The Service/Product will meet your expectations, and (e) Any errors in the software will be corrected.

No advice or information, whether oral or written, obtained by you from **HDSB** or through or from The Service shall create any warranty not expressly stated in the Terms Of Service.

THE SERVICES AND ALL INFORMATION, PRODUCTS AND OTHER CONTENT (INCLUDING THIRD PARTY INFORMATION, PRODUCTS AND CONTENT) INCLUDED IN OR ACCESSIBLE FROM THIS WEB SITE ARE PROVIDED "AS IS" AND ARE SUBJECT TO CHANGE AT ANY TIME WITHOUT NOTICE TO YOU. TO THE FULLEST EXTENT PERMITTED BY LAW, **BrowserObject.com** AND **HDSB** DISCLAIM ALL REPRESENTATIONS AND WARRANTIES (EXPRESS, IMPLIED AND STATUTORY, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND ON-INFRINGEMENT OF PROPRIETARY RIGHTS) AS TO THE SERVICES AND ALL INFORMATION, PRODUCTS AND OTHER CONTENT (INCLUDING THIRD PARTY INFORMATION, PRODUCTS AND CONTENT) INCLUDED IN OR ACCESSIBLE FROM THIS WEB SITE.

Limitation of Liability

To the maximum extent permitted by applicable law, in no event shall **HDSB** or its affiliates be liable for any direct, indirect, special, punitive, incidental, exemplary or consequential damages, or any damages whatsoever, resulting from any loss of use, loss of data, loss of profits, business interruption, litigation or any other pecuniary loss, whether based on breach of contract, tort (including negligence), product liability or otherwise, arising out of or in any way connected with the use or performance of other sites, with the delay or inability to use other sites, or with the provision of or failure to make available any information, services, products, materials, or other resources contained on other sites, even if advised of the possibility of such damages.

Indemnity

You agree to indemnify, defend, and hold harmless **HDSB** and its affiliates from any liability, loss, claim, and expense (including reasonable attorneys' fees) related to your violation of this Service.

Changes to Sites

HDSB may change or modify the information, services, products, materials, and any other resources contained on or accessible through their sites, or discontinue their sites altogether, at any time without notice.

No License Granted

Nothing on this Web site or your use of The Service shall be construed as conferring any license or other rights under the intellectual property or other proprietary rights of **HDSB** or any third party, whether by implication or otherwise.

Copyright

All software and technology used on this site is the property of **HDSB** or its software suppliers and protected by Malaysian and international copyright laws. The content and software on this site may be used as a promotion and selling purpose with the written consent from **HDSB**. Any other use, including the reproduction, modification, distribution, transmission, republication, display, or performance, of the content on this site is strictly prohibited. Reverse engineering and data-mining of our services are serious offenses and legal action will be taken against any violation