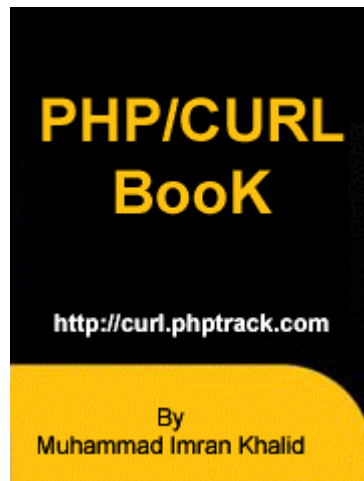


PHP/CURL Book with Examples Version 1.8



Author : Muhammad Imran Khalid
WebSite: <http://curl.phptrack.com>
Email: info@curl.phptrack.com

This Book is a collective work consisting of the following major Open Source companies: cURL, PHP, OpenSSL, libhttpheaders, ethereal each licensed under a separate Open Source License. <http://curl.phptrack.com> is not affiliated with nor endorsed by any of the above providers. See Copyrights for details http://curl.phptrack.com/copyrights/LICENSE_AGREEMENT.txt.

Version 1.1, June, 2005, Version 1.5, Jan, 2006, Version 1.7, Mar, 2006,
Version 1.8, Mar, 2006. This document is subject to change without notice.

Legal Disclaimer

PHP/CURL Book Copyright 2005-2006 <http://curl.phptrack.com> and Imran - All Rights Reserved. THIS COPYRIGHT INFORMATION MUST REMAIN INTACT AND MAY NOT BE MODIFIED IN ANY WAY.

By purchasing this Book you agreed to accept the terms of this Agreement. This Agreement is a legal contract, which specifies the terms of the license and warranty limitation between you and <http://curl.phptrack.com>. You should carefully read the following terms and conditions before using this Book. Unless you have a different license agreement obtained from <http://curl.phptrack.com>, installation or use of this Book indicates your acceptance of the license and warranty limitation terms contained in this Agreement. If you do not agree to the terms of this Agreement, promptly delete and destroy all copies of the Book.

This Book and its Examples are protected by copyright law. Unauthorized reproduction or distribution of this Book and its Examples, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

License to Use: You are allowed to use only one licensed copy of Book and its Examples on one web site. You are not allowed to distribute the Book and its Examples and/or documentation by any means without prior written permission from <http://curl.phptrack.com>. All rights to the Book and its Examples and documentation not expressly granted under this Agreement are reserved to <http://curl.phptrack.com>.

Disclaimer of Warranty:

THIS Book IS PROVIDED "AS IS", AND, ANY EXPRESS OR IMPLIED WARRANTIES INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS Book, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THE USER MUST ASSUME THE ENTIRE RISK OF USING THIS PROGRAM. IN NO CASE WILL Webradev.com OR LIABILITY EXCEED THE AMOUNT OF THE LICENSE FEE ACTUALLY PAID BY LICENSEE TO 'cURL PHP Book'. IF ANY PROVISION OF THIS AGREEMENT IS FOUND TO BE LEGALLY UNENFORCEABLE, THEN THAT PROVISION WILL BE DELETED FROM THIS AGREEMENT, AND THE REMAINDER OF THE AGREEMENT WILL CONTINUE IN FULL FORCE AND EFFECT.

Table of Contents

1.0 Welcome to PHP/CURL Book	05
1.1 About this Book.....	06
1.2 About the Author.....	07
2.0 Getting Started with CURL PHP Book	08
2.1 Before You Start.....	08
2.2 Technical Requirements.....	08
2.3 Installation.....	08
3.0 Curl Library	10
3.1 What is cURL ?.....	10
3.2 What is libcurl ?.....	11
3.3 What is PHP/CURL ?.....	12
3.4 Who write PHP/CURL functions?.....	12
3.5 The HTTP Protocol	12
3.6 To cURL or to libcurl?.....	12
4.0 PHP/CURL Functions with Examples	13
4.1. Using libcurl with PHP.....	13
4.2. Simple usage.....	13
4.3. Forms	15
1.1 GET	15
1.2 POST	16
1.3 Hidden Fields	17
1.4 Figure Out What A POST Looks Like	18
4.4. Download Image (Binnary File)	18
4.5 Authentication	19
4.6. Referer	19
4.7. User Agent	20
4.8. Redirects	21
4.9. Cookies	22
4.10. HTTPS (SSL)	25
4.11. Debug	26
4.12. XML, Credit Card Processing Solutions.....	28
4.13. Post Custom Header to the Remote (Target) Server	30
4.14. Upload a file to remote server from HTML form	32
4.15. FTP Upload a file	32
4.16. FTp Delete a File	33

5.0 LibCURL Options	35
6.0 Header Tracking Tools	42
7.0 RealTime Example	44
7.1. Ebay CURL PHP Login Script	44
7.2. Ebay Summary Page After Login with PHP.....	48
7.3 Calculate Fedex Shipping Charges with XML CURL PHP support.	51
7.4 Yahoo Email login and download Address Book	55
7.5 PayPal Login and download Transaction History.	55
7.6 Hotmail Login and download Contacts List.	55
7.7 Gmail address book download	55
8.0 Regular Expressions	56
8.1. What are Regular Expressions?	56
8.2. What exactly is possible with Regular Expressions?.....	56
8.3 Regular Expressions in PHP.....	56
8.4 Regular Expressions Examples	59
8.4.1 preg_match function	59
8.4.2 preg_match_all function	60
8.4.3 preg_replace function	62
8.4.4 preg_replace function	62
8.4.5 'ereg','eregi','ereg_replace'.....	63
8.4.6 Parsing Form for hidden fields	64
8.4.7 Parsing Images links from HTML page.....	65
Appendix A	67
Appendix B	70
References	71

1.0 Welcome to PHP/CURL Book

PHP/CURL Examples Book with Tutorial Help is a powerful Book that allows PHP Programmers, Web Development companies to achieve better Grip on PHP CURL Functions Library usage (<http://www.php.net/manual/en/ref.curl.php>) in Professional field to write your own Web boots, Spiders, Search Engines, Compare prices tools, Get online products prices from sites, Download images, copy latest news from news sites. Automatically Place banners on sites, download hotmail, Yahoo mails and EBay products, Calculate online Shipping charges, Process Credit cards and much more.

Book Examples solution designed to fit PHP Web Developer, Programmers, Web Development companies in their daily operation. Our Book solutions allow you to write Web boots spiders and big Applications for B2B companies like Yahoo mail, Hotmail, EBay sites. You can generate different types of Web Applications by using this book without hassle.

Features:

1. Examples with live Demo.
2. Step by Step guide to each code.
3. Real life PHP CURL Open Source codes.
4. Header Tracking Solutions.
5. Credit Card Processing Solutions.
6. XML Processing Solutions.
7. Ebay Login Script.
8. Write online web Price Compare sites.
9. Calculate realtime FedEx/UPS Shipping Charges.
10. Yahoo Email login and download Address Book.
11. Hotmail Email login and download Contacts List.
12. PayPal Login and download Transaction History.

1.1 About this Book

We try our best to explain PHP/CURL functions usage by Real Life Examples and Professional Real Time Applications developed in PHP. This book helps a lot for PHP beginner students, PHP professional experts and advance level programmer to write Commercial Applications by using PHP with cURL support.

The main Library we uses for our PHP/CURL Book for writing webbots and spiders is cURL and libcurl. Curl is a useful tool for webbot developers as it is indispensable for transferring files with URL syntax. With cURL you can easily get files using any of the following protocols: FTP, FTPS, HTTP, HTTPS, GOPHER, TELNET, DICT, FILE and LDAP. Curl also supports HTTPS certificates, HTTP POST, HTTP PUT, FTP uploading, kerberos, HTTP form based upload, proxies, cookies, HTTP (basic) user/password authentication. We helped numerous companies/ Developers by either developing cURL based applications in CURL PHP, or by helping their developers use cURL in our PHP CURL Forum.

The increasing amount of applications moving to the web has made "HTTP Scripting" more frequently requested and wanted. To be able to automatically extract information from the web, to fake users, to post or upload data to web servers are all important tasks today.

This tutorial is intended for PHP programmers and web developers interested in using their web server to transfer files or communicate with other servers. You will need some general knowledge of PHP language.

You can easily write webbots, search engines, spiders and screen scrapers after reading this book. Unlike web browsers from Microsoft and Netscape, a webbot, spider (or agent) can automatically surf the Net and filter data for relevance, and present the data within a unique business context.

Spiders and agents can also automate a nearly limitless number of processes, including: downloading information into PDAs, completing web-based forms, and alerting people when specific information is posted on the Internet. Spiders and agents can also connect the web to non-traditional web interfaces like pagers, fax machines and voice synthesizers. Spiders and web agents are only limited by the imagination of their developers.

Examples of what spiders/web-bots can do:

- * Search for the Internet for inappropriate use of intellectual property,
- * Receive real-time court records,
- * Monitor the status of online auctions and automatically make "rules-based" bids,
- * Compare your online prices to the online prices of your competitors,
- * Poll information from online appliances, and
- * Create an online digital "clipping service" on yourself, or your competition.

1.2 About the Author

Muhammad Imran Khalid

V/P.O. Dhonikey, Wazirabad, Gujranwala, Pakistan

Phone: +92.55.6602955, Cell: +92.345.6900444

Web: <http://imran.phptrack.com>, Email: imran@phptrack.com

Muhammad Imran Khalid Internet development experience started in 2000. Now a days, He serves his services around the World with Commercial Experience in PHP Web Development and maintenance of Company websites, providing regular updates to existing web sites, CSS, XHTML Search Engines Optimization Techniques, build websites to W3C standards and XML Google search sitemaps, custom state-of-the-art E-commerce sites, Online Shopping Stores, Webbot, Spiders and Web agents technologies--as well as dynamic data-driven web applications in LAMP (Linux + Apache + MySQL + PHP), PHP/CURL and XML based web services. He is also remotely managing MYSQL Database servers, IIS Web Servers.

It was here that he lead and developed Internet projects for his clients including Ebay, IMail, Hotmail and Techdata plugins. He is also PHP team Organizer of PHP meetings in Lahore Pakistan and his writing credits include: PHP/CURL BOOK Won P@SHA Awards 2005 in Research and Development, PHP GPL Scripts and Web Techniques Articles. In recent years, He also Help online on his site Forum <http://curl.phptrack.com> Web Usability and Internet Crawlers/ Web Spiders to students and Professional Developers.

Most recently, He has created an auto advertisement lister project. This tool lists your advertisement automatically to other sites with no human typing. He has solid expertise in writing Price comparison tools, Search engines, downloading web images.

2.0 Getting Started with PHP/CURL

2.1 Before You Start

This guide assumes that you have already installed a **Web Server (IIS, APACHE)** and running **PHP with cURL support** on it and you are familiar with how to use them. If you are not sure whether your system meets the requirements or how to use required tools, talk to your manager or system administrator.

We also explain in details how to install **Web Server (IIS, APACHE)** and running **PHP with cURL support** software in [Appendix A](#), [Appendix B](#).

2.2 Technical Requirements

Before you begin using the system, ensure that you have the appropriate software installed and configured on your system. All you will need is -

In order to use the CURL functions you need to install the CURL package (<http://curl.haxx.se>). PHP requires that you use CURL 7.0.2-beta or higher. PHP will not work with any version of CURL below version 7.0.2-beta. In PHP 4.2.3, you will need CURL version 7.9.0 or higher. From PHP 4.3.0, you will need a CURL version that's 7.9.8 or higher. PHP 5.0.0 will most likely require a CURL version greater than 7.10.5

2.3 Installation

To use PHP's CURL support you must also compile PHP `--with-curl[=DIR]` where DIR is the location of the directory containing the lib and include directories. In the "include" directory there should be a folder named "curl" which should contain the easy.h and curl.h files. There should be a file named libcurl.a located in the "lib" directory. Beginning with PHP 4.3.0 you can configure PHP to use CURL for URL streams `--with-curlwrappers`.

Note to Win32 Users: In order to enable this module on a Windows environment, you must copy libeay32.dll and ssleay32.dll from the DLL folder of the PHP/Win32 binary package to the SYSTEM folder of your Windows machine. (Ex: C:\WINNT\SYSTEM32 or C:\WINDOWS\SYSTEM) . Please also open you “php.ini” file . (Ex: C:\WINNT\SYSTEM32 or C:\WINDOWS\SYSTEM) and Change the line “;extension=php_curl.dll” To this “extension=php_curl.dll” (remove only the start semicolon)

For more details on Installation please view [Appendix A](#), [Appendix B](#).

3.0 cURL Library

The cURL libraries provide a nice interface for file transfers to and from a webserver. They have support for a wide variety of protocols (like HTTPS) giving them an edge over built-in PHP functions like `fsockopen()`. The libraries are thread-safe, IPv6 compatible, and will work with any technology that is built on top of HTTP. Whether you are building simple script to fetch a web page, or a secure payment gateway, leveraging the functions built into cURL can save a lot of time.

cURL and libcurl are libraries that allow a webserver to transfer files with a remote computer using a variety of Internet protocols. The libraries are highly configurable, allowing practically any type of client-server request to be performed. By using these tools, a webserver can act as a client, creating and responding to requests using any technology built on HTTP, like XML-RPC, SOAP, or PHPTRACK.

Curl Library is not written to do everything for you. It makes the requests, it gets the data, it sends data and it retrieves the information. You probably need to glue everything together using some kind of script language or repeated manual invokes.

3.1 What is cURL?

The name is a play on 'Client for URLs', originally with URL spelled in uppercase to make it obvious it deals with RLs. The fact it can also be pronounced 'see URL' also helped, it works as an abbreviation for "Client URL Request Library" or why not the recursive version: "Curl URL Request Library". The cURL project produces two products libcurl and curl.

libcurl

A free and easy-to-use client-side URL transfer library, supporting FTP, TPS, HTTP, HTTPS, GOPHER, TELNET, DICT, FILE and LDAP. libcurl supports TTSPS certificates, HTTP POST, HTTP PUT, FTP uploading, kerberos, HTTP orm based upload, proxies, cookies, user+password authentication, file ransfer resume, http proxy tunneling and more!

libcurl is highly portable, it builds and works identically on numerous latforms, including Solaris, NetBSD, FreeBSD, OpenBSD, Darwin, HPUX, RIX, AIX, Tru64, Linux, UnixWare, HURD, Windows, Amiga, OS/2, BeOs, Mac S X, Ultrix, QNX, OpenVMS, RISC OS, Novell NetWare, DOS and more...

libcurl is free, thread-safe, IPv6 compatible, feature rich, well upported and fast.

curl

A command line tool for getting or sending files using URL syntax.

Since curl uses libcurl, it supports a range of common Internet protocols, currently including HTTP, HTTPS, FTP, FTPS, GOPHER, LDAP, DICT, TELNET and FILE.

We pronounce curl and cURL with an initial k sound: [kurl].

There are numerous sub-projects and related projects that also use the word curl in the project names in various combinations, but you should take notice that this is directed at the command-line tool named curl (and libcurl the library), and may therefore not be valid for other curl-related projects.

3.2 What is libcurl?

libcurl is a reliable and portable library which provides you with an easy interface to a range of common Internet protocols. You can use libcurl for free in your application, be it open source, commercial or closed-source. libcurl is most probably the most portable, most powerful and most often used C-based multi-platform file transfer library on this planet - be it open source or commercial.

cURL stands for "Client URLs", and was developed by Daniel Stenberg in 1998 as a command line tool. libcurl is a portable library that provides an easy interface to the cURL functionality. It is thread safe, IPv6 compatible, and supports persistent connections. The libcurl PHP binding was added by Sterling Hughes.

Both cURL and libcurl can transfer files using a wide variety of protocols, including HTTP, HTTPS, FTP, FTPS, GOPHER, LDAP, DICT, TELNET and FILE. The libraries run on practically any *NIX operating system, as well as Windows, OS/2, BeOS, and many more.

The cURL libraries are truly open source, with an MIT/X derivative license. This license is very liberal, allowing the use of cURL for whatever you want, commercial or not. You can use libcurl for free, and even include and distribute it with your own application, whether commercial or closed-source.

cURL should not to be confused with the Curl Corporation, which is the commercial producer of the client side programming language, Curl.

3.3 What is PHP/CURL?

The module for PHP that makes it possible for PHP programs to access curl-functions from within PHP.

In the cURL project we call this module PHP/CURL to differentiate it from curl the command line tool and libcurl the library. The PHP team however does not refer to it like this (for unknown reasons). They call it plain CURL (often using all caps) which causes much confusion to users which in turn gives us a higher question load.

3.4 Who write PHP/CURL Functions?

PHP/CURL is a module that comes with the regular PHP package. It depends and uses libcurl, so you need to have libcurl installed properly first before PHP/CURL can be used. PHP/CURL is written by Sterling Hughes.

3.5 The HTTP Protocol

HTTP is the protocol used to fetch data from web servers. It is a very simple protocol that is built upon TCP/IP. The protocol also allows information to get sent to the server from the client using a few different methods, as will be shown here.

HTTP is plain ASCII text lines being sent by the client to a server to request a particular action, and then the server replies a few text lines before the actual requested content is sent to the client.

3.6 To cURL or to libcurl?

The decision as to whether to use cURL or libcurl depends on the situation. For instance, if I have a cron job running that e-mails me when a file changes on a remote server, or if my ISP doesn't have libcurl support in their PHP install, using cURL makes more sense. However, if I have libcurl support in PHP and I am building a PHP application requiring cURL functionality, libcurl is the right choice.

4.0 PHP/CURL Functions with Examples

If cURL support is enabled in PHP, the `phpinfo()` function will display it in its output as below. You are requested to check it before writing your first simple programme in PHP.

```
<?php
```

```
phpinfo();
```

```
?>
```

curl

CURL support	enabled
CURL Information	libcurl/7.11.2 OpenSSL/0.9.7c zlib/1.1.4

4.1 Using libcurl with PHP

While using cURL from within PHP is an option, using the libcurl PHP binding is much easier, especially for things like an HTTP POST operation.

The process of using libcurl from within PHP is a matter of following these basic steps:

Initialize the cURL session Set the cURL options (The order of the options is not important) Execute the options in the cURL session Close the curl session

4.2 Simple Usage

The simplest and most common request/operation made using HTTP is to get a URL. The URL could itself refer to a web page, an image or a file. The client issues a GET request to the server and receives the document it asked for.

The ***curl_init()*** will initialize a new session and return a CURL handle.

curl_exec(\$ch) This function should be called after you initialize a CURL session and all the options for the session are set. Its purpose is simply to execute the predefined CURL session (given by the ch).

curl_setopt(\$ch, option, value) Set an option for a CURL session identified by the ch parameter. option specifies which option to set, and value specifies the value for the option given.

curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, 1) Return Page contents. If we set 0 then no output will be returned.

curl_setopt(\$ch, CURLOPT_URL, \$url) Pass URL as parameter. This is your target server website address. This is the URL you want to get from internet.

curl_exec(\$ch) Grab URL and pass it to the variable for showing output.

curl_close(\$ch) close curl resource, and free up system resources.

1. Get the main page from PHPTrack web-server: Examples/001.1.php

```
<?php
// Example 001
// Simple Get Webpage
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/index.php"; // From URL to get webpage
contents.

$ch = curl_init(); // Initialize a CURL session.

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return Page contents.
curl_setopt($ch, CURLOPT_URL, $url); // Pass URL as parameter.
$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

4.3 Forms

Forms are the general way a web site can present a HTML page with fields for the user to enter data in, and then press some kind of 'OK' or 'submit' button to get that data sent to the server. The server then typically uses the posted data to decide how to act. Like using the entered words to search in a database, or to add the info in a bug track system, display the entered address on a map or using the info as a login-prompt verifying that the user is allowed to see what it is about to see.

Of course there has to be some kind of program in the server end to receive the data you send. You cannot just invent something out of the air.

1.1 GET

A GET-form uses the method GET, as specified in HTML like:

```
<form method="GET" action="post.php">
  <input type="text" name="first_name">
  <input type="text" name="age">
  <input type="submit" name="press" value="OK">
</form>
```

In your favorite browser, this form will appear with a text box to fill in and a press-button labeled "OK". If you fill in name=imran and age=25 and press the OK button, your browser will then create a new URL to get for you. The URL will get "post.php?name=imran&age=30&press=OK" appended to the path part of the previous URL.

If the original form was seen on the page "www.yahoo.com/when/birth.html", the second page you'll get will become
"www.yahoo.com/when/post.php?name=imran&age=30&press=OK".

Most search engines work this way.

Please use the PHP function "*rawurlencode*" to encode the values of the querystring. For example, if the data to contain a space, you need to replace that space with %20. The PHP function "*rawurlencode*" works best to perform this job.

Post the HTML Form with GET method to PHPTrack web-server page: Examples/002.1.php

```
<?php
// Example 002.1
// Pass Form Variables as method = GET
// Copyright http://curl.phptrack.com

$domain = "http://curl.phptrack.com/"; // URL to POST FORM.
$post_fields = 'get_page.php?fuseaction=forum&name=imran&age=30&press=OK';
$url = $domain . $post_fields;

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return Page contents.
curl_setopt($ch, CURLOPT_URL, $url); // Pass URL as parameter.
$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

1.2 POST

The GET method makes all input field names get displayed in the URL field of your browser. That's generally a good thing when you want to be able to bookmark that page with your given data, but it is an obvious disadvantage if you entered secret information in one of the fields or if there are a large amount of fields creating a very long and unreadable URL.

The HTTP protocol then offers the POST method. This way the client sends the data separated from the URL and thus you won't see any of it in the URL address field.

The form would look very similar to the previous one:

```
<form method="GET" action="post.php">
  <input type="text" name="first_name">
  <input type="text" name="age">
  <input type="submit" name="press" value="OK">
</form>
```

And to use curl to post this form with the same data filled in as before, we could do it like:

```
" name=imran&age=30&press=OK"
```

This kind of POST will use the Content-Type application/x-www-form-urlencoded and is the most widely used POST kind.

The data you send to the server **MUST** already be properly encoded, curl will not do that for you. For example, if you want the data to contain a space, you need to replace that space with %20 etc. Failing to comply with this will most likely cause your data to be received wrongly and messed up.

```
<?php
// Example 002.2
// Pass form Variables as method = POST
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/subscribe.php"; // URL to POST FORM.
(Action of Form)
// use PHP Fucntion url_encode() for post variable for application/x-www-
form-urlencoded
$post_fields = 'fuseaction=forum&name=imran%20khalid&age=30&press=OK'; //
form Fields.

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // Pass URL as parameter.
curl_setopt($ch, CURLOPT_POST, 1); // use this option to Post a form
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_fields); // Pass form Fields.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return Page contents.

$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

1.3 Hidden Fields

A very common way for HTML based application to pass state information between pages is to add hidden fields to the forms. Hidden fields are already filled in, they aren't displayed to the user and they get passed along just as all the other fields.

A similar example form with one visible field, one hidden field and one submit button could look like:

```
<form method="POST" action="process.php">
  <input type="text" name="first_name">
  <input type="hidden" name="status" value="married">
  <input type="submit" name="press" value="OK">
</form>
```

To post this with curl, you won't have to think about if the fields are hidden or not. To curl they're all the same:

first_name=khalid&press=OK&status=imran

1.4 Figure Out What A POST Looks Like

Please use Header Tracking Tools on page 25 to make it easy to view what's Request and Response Generated when you access a certain page.

When you're about to fill in a form and send to a server by using curl instead of a browser, you're of course very interested in sending a POST exactly the way your browser does.

An easy way to get to see this, is to save the HTML page with the form on your local disk, modify the 'method' to a GET, and press the submit button (you could also change the action URL if you want to).

You will then clearly see the data get appended to the URL, separated with a '?'-letter as GET forms are supposed to.

4.4 Download Image (Binary File)

You can also download an image from web or any Binary file like pdf, psd, doc, zip with curl. Google is using this technique to get all web images.

```
<?php
// Example 003
// Download Image (Binary File)
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/images/header.jpg"; // URL to Download
Image

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // Pass URL as parameter.

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return stream contents.
curl_setopt($ch, CURLOPT_BINARYTRANSFER, 1); // We'll be returning this
transfer, and the data is binary
$data = curl_exec($ch); // // Grab the jpg and save the contents in the
$data variable
curl_close($ch); // close curl resource, and free up system resources.

// Set the header to type image/jpeg, since that's what we're
// displaying
header("Content-type: image/jpeg");
echo $data; // Print stream contents.

?>
```

4.5 Authentication

Authentication is the ability to tell the server your username and password so that it can verify that you're allowed to do the request you're doing. The Basic authentication used in HTTP (which is the type curl uses by default) is *plain* *text* based, which means it sends username and password only slightly obfuscated, but still fully readable by anyone that sniffs on the network between you and the remote server.

Please use the below option for user password authentication in Browser Dialog Box Authentication:

```
<?php
// Example 004
// Login to site where Dialog Box Open for Authentication
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/login.php"; // URL to POST Login Data.
$post_fields = 'username:password'; // PopUpDialog Login Fields.
// Do not remove the ":" sign between username and password.
$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // Pass URL as parameter.
curl_setopt($ch, CURLOPT_USERPWD, $post_fields); // Dialog Box
Authentication.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return Page contents.

$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

4.6 Referer

A HTTP request may include a 'referer' field (yes it is misspelled), which can be used to tell from which URL the client got to this particular resource. Some programs/scripts check the referer field of requests to verify that this wasn't arriving from an external site or an unknown page. While this is a stupid way to check something so easily forged, many scripts still do it. Using curl, you can put anything you want in the referer-field and thus more easily be able to fool the server into serving your request.

Use libcurl to set the referer field with:

```
<?php
// Example 005
// Pass Refferal to the Target Site. This insure that request is from this
site.
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/login.php"; // URL
$reffer = "http://curl.phptrack.com/index.php"; // Refferal site

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // Pass URL as parameter.
curl_setopt($ch, CURLOPT_REFERER, $reffer); // Refferal site Setting.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return Page contents.

$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

4.7 User Agent

Very similar to the referer field, all HTTP requests may set the User-Agent field. It names what user agent (client) that is being used. Many applications use this information to decide how to display pages. Silly web programmers try to make different pages for users of different browsers to make them look the best possible for their particular browsers. They usually also do different kinds of javascript, vbscript etc.

At times, you will see that getting a page with curl will not return the same page that you see when getting the page with your browser. Then you know it is time to set the User Agent field to fool the server into thinking you're one of those browsers.

```
<?php
// Example 006
// Pass User Agent to the Target Site. This insure that request is from
// which Browser and Operating System.
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/login.php"; // URL to POST Login Data.
$agent = "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"; //Agent
Setting for Internet Explorer
$agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4)
Gecko/20030624 Netscape/7.1 (ax)"; //Agent Setting for Netscape

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // Pass URL as parameter.
curl_setopt($ch, CURLOPT_USERAGENT, $agent); // Agent Setting.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return Page contents.

$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

4.8 Redirects

When a resource is requested from a server, the reply from the server may include a hint about where the browser should go next to find this page, or a new page keeping newly generated output. The header that tells the browser to redirect is Location:.

Curl does not follow Location: headers by default, but will simply display such pages in the same manner it display all HTTP replies. It does however feature an option that will make it attempt to follow the Location: pointers.

To tell libcurl to follow a Location:

If you use curl to POST to a site that immediately redirects you to another page, you can safely use -L and -d/-F together. Curl will only use POST in the first request, and then revert to GET in the following operations.

```
<?php
// Example 007
// Redirect Page where Server transfer control after login verification etc.
// if this option is not provided then this will not go to welcome page.
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/login.php"; // URL

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // Pass URL as parameter.
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // Redirect to page where its
goes after login.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Return Page contents.

$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

4.9 Cookies

The way the web browsers do "client side state control" is by using cookies. Cookies are just names with associated contents. The cookies are sent to the client by the server. The server tells the client for what path and host name it wants the cookie sent back, and it also sends an expiration date and a few more properties.

When a client communicates with a server with a name and path as previously specified in a received cookie, the client sends back the cookies and their contents to the server, unless of course they are expired.

Many applications and servers use this method to connect a series of requests into a single logical session. To be able to use curl in such occasions, we must be able to record and send back cookies the way the web application expects them. The same way browsers deal with them.

This is very Important section of your PHP CURL Scripts for login verification or get pages in secure section of a website. There are three things required to implement it.

1- Your server Cookie File Path.

```
$cookie_file_path = "C:/Inetpub/wwwroot/spiders/cookie/cook"; //  
Please set your Cookie File path. This file must have CHMOD 777  
(Full Read / Write Option).
```

2- Two CURL Functions.

```
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path); // The name  
of the file containing the cookie data. The cookie file can be in  
Netscape format, or just plain HTTP-style headers dumped into a  
file.  
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path); // The name  
of a file to save all internal cookies to when the connection  
closes.
```

Please do as:-

1 Write a code in a file test.php and upload on your server.

```
<?php
```

```
print realpath('test.php');
```

```
?>
```

2 Access this file in your browser like yourdoamin.com/test.php and copy the output. This should be your server path. For Example
"C:/Inetpub/wwwroot/mydomain/folder/test.php"

3 Now create a txt file and save it as cookie.txt and upload on your server.

4 in ftp write click on your file and select CHMOD (777) click all options (read, write). Give full permission to file so that your script should be able to write in it. On windows OS, please right right click on your file on server and give full access to everyone.

5 Now make a string of your server path with of file cookie.txt by adding point 2 output in it.

For Example

```
"C:/Inetpub/wwwroot/mydomain/folder/cookie.txt"
```

```
<?php
// Example 008
// Use Cookies for Storing, Reading, setting and Passing
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/login.php"; // URL
$postfields = 'name=admin&password=guest&submit=save';
$reffer = "http://curl.phptrack.com/index.php";
$agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4)
Gecko/20030624 Netscape/7.1 (ax)";
$cookie_file_path = "C:/Inetpub/wwwroot/spiders/cookie/cook"; // Please
set your Cookie File path. This file must have CHMOD 777 (Full Read /
Write Option).

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // The URL to fetch. You can also
set this when initializing a session with curl_init().
curl_setopt($ch, CURLOPT_USERAGENT, $agent); // The contents of the "User-
Agent: " header to be used in a HTTP request.
curl_setopt($ch, CURLOPT_POST, 1); //TRUE to do a regular HTTP POST. This
POST is the normal application/x-www-form-urlencoded kind, most commonly
used by HTML forms.
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields); //The full data to post
in a HTTP "POST" operation.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // TRUE to return the
transfer as a string of the return value of curl_exec() instead of
outputting it out directly.
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // TRUE to follow any
"Location: " header that the server sends as part of the HTTP header (note
this is recursive, PHP will follow as many "Location: " headers that it is
sent, unless CURLOPT_MAXREDIRS is set).
curl_setopt($ch, CURLOPT_REFERER, $reffer); //The contents of the
"Referer: " header to be used in a HTTP request.

curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path); // The name of
the file containing the cookie data. The cookie file can be in Netscape
format, or just plain HTTP-style headers dumped into a file.
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path); // The name of a
file to save all internal cookies to when the connection closes.

$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```


4.10 HTTPS (SSL)

There are a few ways to do secure HTTP transfers. The by far most common protocol for doing this is what is generally known as HTTPS, HTTP over SSL. SSL encrypts all the data that is sent and received over the network and thus makes it harder for attackers to spy on sensitive information.

SSL (or TLS as the latest version of the standard is called) offers a truckload of advanced features to allow all those encryptions and key infrastructure mechanisms encrypted HTTP requires.

Curl supports encrypted fetches thanks to the freely available OpenSSL Libraries.

```
<?php
// Example 009
// HTTPS (SSL Pages)
// Copyright http://curl.phptrack.com

$url = "https://your_Secure_site.com/login.php"; // URL
$postfields = 'name=admin&password=guest&submit=save';
$referer = "https://your_Secure_site.com/index.php";
$agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4)
Gecko/20030624 Netscape/7.1 (ax)";
$cookie_file_path = "C:/Inetpub/wwwroot/spiders/cookie/cook"; // Please
set your Cookie File path. This file must have CHMOD 777 (Full Read /
Write Option).

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // The URL to fetch. You can also
set this when initializing a session with curl_init().
curl_setopt($ch, CURLOPT_USERAGENT, $agent); // The contents of the "User-
Agent: " header to be used in a HTTP request.
curl_setopt($ch, CURLOPT_POST, 1); //TRUE to do a regular HTTP POST. This
POST is the normal application/x-www-form-urlencoded kind, most commonly
used by HTML forms.
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields); //The full data to post
in a HTTP "POST" operation.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // TRUE to return the
transfer as a string of the return value of curl_exec() instead of
outputting it out directly.
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // TRUE to follow any
"Location: " header that the server sends as part of the HTTP header (note
this is recursive, PHP will follow as many "Location: " headers that it is
sent, unless CURLOPT_MAXREDIRS is set).
curl_setopt($ch, CURLOPT_REFERER, $referer); //The contents of the
"Referer: " header to be used in a HTTP request.
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path); // The name of
the file containing the cookie data. The cookie file can be in Netscape
format, or just plain HTTP-style headers dumped into a file.
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path); // The name of a
file to save all internal cookies to when the connection closes.
```

```
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false); //FALSE to stop CURL from
verifying the peer's certificate. Alternate certificates to verify against
can be specified with the CURLOPT_CAINFO option or a certificate directory
can be specified with the CURLOPT_CAPATH option. CURLOPT_SSL_VERIFYHOST
may also need to be TRUE or FALSE if CURLOPT_SSL_VERIFYPEER is disabled
(it defaults to 2). TRUE by default as of CURL 7.10. Default bundle
installed as of CURL 7.10.
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0); // 1 to check the existence
of a common name in the SSL peer certificate. 2 to check the existence of
a common name and also verify that it matches the hostname provided.

$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

4.11 Debug

Many times when you run curl on a site, you'll notice that the site doesn't seem to respond the same way to your curl requests as it does to your browser's.

Then you need to start making your curl requests more similar to your browser's requests:

- * Make sure you check for and use cookies when needed
- * Set user-agent to one like a recent popular browser does
- * Set referer like it is set by the browser
- * If you use POST, make sure you send all the fields and in the same order as the browser does it.

```
<?php
// Example 010
// Bug Tracking with Viewing Whats Header passed and Returns
// Copyright http://curl.phptrack.com

$url = "http://curl.phptrack.com/login.php"; // URL
$postfields = 'name=admin&password=guest&submit=save';
$referer = "http://curl.phptrack.com/index.php";
$agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4)
Gecko/20030624 Netscape/7.1 (ax)";
$cookie_file_path = "C:/Inetpub/wwwroot/spiders/cookie/cook"; // Please
set your Cookie File path. This file must have CHMOD 777 (Full Read /
Write Option).

$ch = curl_init(); // Initialize a CURL session.
curl_setopt($ch, CURLOPT_URL, $url); // The URL to fetch. You can also
set this when initializing a session with curl_init().
curl_setopt($ch, CURLOPT_USERAGENT, $agent); // The contents of the "User-
Agent: " header to be used in a HTTP request.
curl_setopt($ch, CURLOPT_POST, 1); //TRUE to do a regular HTTP POST. This
POST is the normal application/x-www-form-urlencoded kind, most commonly
used by HTML forms.
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields); //The full data to post
in a HTTP "POST" operation.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // TRUE to return the
transfer as a string of the return value of curl_exec() instead of
outputting it out directly.
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // TRUE to follow any
"Location: " header that the server sends as part of the HTTP header (note
this is recursive, PHP will follow as many "Location: " headers that it is
sent, unless CURLOPT_MAXREDIRS is set).
curl_setopt($ch, CURLOPT_REFERER, $referer); //The contents of the
"Referer: " header to be used in a HTTP request.
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path); // The name of
the file containing the cookie data. The cookie file can be in Netscape
format, or just plain HTTP-style headers dumped into a file.
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path); // The name of a
file to save all internal cookies to when the connection closes.

curl_setopt($ch, CURLOPT_HEADER, 1); // Bug Tracking TRUE to include the
header in the output.
curl_setopt($ch, CURLOPT_VERBOSE, 1); // Bug Tracking TRUE to output
verbose information. Writes output to STDERR, or the file specified using
CURLOPT_STDERR.

$result = curl_exec($ch); // grab URL and pass it to the variable.
curl_close($ch); // close curl resource, and free up system resources.

echo $result; // Print page contents.

?>
```

4.12 XML, Credit Card Processing Solutions.

Electronic commerce (e-commerce) - using remote, electronic connections for transacting business - is an integral part of our world today. But conducting credit card and check authorizations over the Internet, in a secure environment, is a very complex process. CURL PHP provides a simple payment solution for the e-commerce industry by offering XML CURL PHP Connect. Using XML CURL PHP Connect is the easiest method for connecting an online store to the powerful Secure Payment Gateway.

A merchant uses the API software modules to build the payment solution that fits the merchant's unique needs. The API is a tool for the merchant that needs a custom commerce solution. It is an application programming interface (API) used by merchants to build complex Web sites or other custom systems that process payments. The API enables full-featured, highly secure and reliable e-commerce Web sites and custom retail implementations.

First Create your XML String then Pass it as CURL POST Method Option. You also need to Pass the file path of the certificate.pem file.

```
<?php
// Example 012
//We will process the Credit Card Transaction.
//PREAUTH of Credit Card.
// Copyright http://curl.phptrack.com
////////////////////////////////////
//
function curl_process($data)
{
    // set up transaction variables
    $debugging = 0;
    $key       = $data["keyfile"];
    $xml       = $data["xml"];
    $url       = $data["host"] . ':' . $data['port'];

    $ch = curl_init ();
    curl_setopt ($ch, CURLOPT_URL,$url);
    curl_setopt($ch, CURLOPT_HEADER, 1);
    curl_setopt ($ch, CURLOPT_POST, 1);
    curl_setopt ($ch, CURLOPT_POSTFIELDS, $xml);
    curl_setopt ($ch, CURLOPT_SSLCERT, $key);
    curl_setopt ($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt ($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt ($ch, CURLOPT_RETURNTRANSFER, 1);

    if ($debugging)
        curl_setopt ($ch, CURLOPT_VERBOSE, 1);

    # use curl to send the xml SSL string
    $result = curl_exec ($ch);
    curl_close($ch);
    return $result;
}
```

```
}
////////////////////////////////////
////////////////////////////////////

// This XML String vary from Credit Card Processing company to company.
// Please your Read your Credit Card Processing company Manual.
$xml ="
    <order>
        <orderoptions>
            <result>GOOD</result>
            <ordertype>PREAUTH</ordertype>
        </orderoptions>
        <merchantinfo>
            <configfile>888999</configfile>
        </merchantinfo>
        <creditcard>
            <cardnumber>4111111111111111</cardnumber>
            <cardexpmonth>07</cardexpmonth>
            <cardexpyear>2006</cardexpyear>
            <cvmvalue>548</cvmvalue>
            <cvmindicator>provided</cvmindicator>

        </creditcard>
        <payment>
            <chargetotal>1</chargetotal>
        </payment>
        <transactiondetails>
            <oid>CODE1001</oid>
            <ponumber>web</ponumber>
        </transactiondetails>
        <billing>
            <name>Muhammad Imran</name>
            <address1>4673 Blue Street</address1>
            <city>Los Angeles</city>
            <state>CA</state>
            <zip>90016</zip>
            <country>US</country>
            <phone>7135566443</phone>
            <email>test@mail.com</email>
            <addrnum>4673</addrnum>

        </billing>
        <notes>
            <comments>Shopping on web.</comments>
        </notes>
    </order>";

$myorder["host"]      = "secure.your_merchant_server.net";
$myorder["port"]      = "1027";
$myorder["keyfile"]   =
"c:/inetpub/wwwroot/yoursite_path/certificate.pem";
$myorder["xml"]       = $xml;

$result = curl_process($myorder); # use curl methods
print $result;

?>
```

4.13 Post Custom Header to the Remote (Target) Server.

This might happen when we access certain web pages on web with CURL PHP scripts, they do not functions. This is due to some checks on remote server to not feed to visitors having some special Header in their request.

For Example you can not access the site <http://www.neteller.com/ab/> with above examples. This site need Custom Headers so that Remote server consider our request same as of Web Browser to send us the page output.

If you use the above Debug Example then you will get the what headers are posted by your Script. To get what Headers posted by your Web Browser you need to download and install LiveHTTPHeader tool. Please read in this book section 6.0 for more details on LiveHTTPHeader tool.

A common Header Always look like as:-

`http://www.neteller.com/ab/`

```
GET /ab/ HTTP/1.1
Host: www.neteller.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4) Gecko/20030624
Netscape/7.1 (ax)
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-
mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: CFTOKEN=1ea40fc1d; CP=null*; WEBTRENDS_ID=209FC; CFID=680689
```

To make Such Header we will use the `CURLOPT_HTTPHEADER` function in our script.

```
<?php
// Example custom_header.php
// Get Secure Page from www.neteller.com
// With Custom Header.
// Copyright http://curl.phptrack.com

$cookie_file_path =
"C:/Inetpub/wwwroot/07feb2005/phptrack/curl/cookie.txt";
$url = 'http://www.neteller.com/ab/';
$reffer = 'http://www.neteller.com';

$header_array[0] = "GET /ab/ HTTP/1.1";
$header_array[1]= "User-Agent: Mozilla/5.0 (Windows; U; Windows NT
5.0; en-US; rv:1.4) Gecko/20030624 Netscape/7.1 (ax)";
$header_array[2]= "Host: www.neteller.com";
$header_array[3]= "Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;
q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1";
$header_array[4]= "Accept-Language: en-us,en;q=0.5";
$header_array[5]= "Accept-Encoding: gzip,deflate";
$header_array[6]= "Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7";
$header_array[7]= "Keep-Alive: 300";
$header_array[8] = "Connection: Close";

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,$url);
curl_setopt($ch, CURLOPT_HTTPHEADER, $header_array);
curl_setopt($ch, CURLOPT_REFERER, $reffer);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
$result = curl_exec ($ch);
curl_close ($ch);
print $result;

?>
```

4.14 Upload a file to remote server from HTML form

You have to use the same option as for FORM POST METHOD but with some changes. Use can use this example where Form has enctype="multipart/form-data". Use the \$_FILE PHP variable to read the uploaded file and pass it to postfield with @ sign as:

```
curl_setopt($ch, CURLOPT_POSTFIELDS,  
array("$field_name"=>"@".$_FILES['file']['tmp_name']));
```

```
<?php
// Upload a file to remote server.
// Upload file field on form having multipart/form-data.
// written by imranlink@hotmail.com

    set_time_limit(0);
    $url = 'http://phptrack.com/upload.php'; // change to your form action url.
    $field_name = 'file'; // please edit it according to your form file field name.

    if (isset($_FILES['file']))
    {
        $ch = curl_init($url);
        curl_setopt($ch, CURLOPT_POSTFIELDS, array("$field_name"=>"@".$_FILES['file']['tmp_name']));
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $result = curl_exec($ch);
        curl_close($ch);
        echo $result;
    }
    else
    {
        print "<form enctype=\"multipart/form-data\" \"
            . \"action=\"".$_PHP_SELF.\" method=\"post\" >\n";
        print '<p>
            <input type="file" name="file">
            </p>
            <p>
            <input type="submit" name="Submit" value="Submit">
            </p>';

        print "</form>";
    }
?>
```


4.15 FTP Upload a file

You have the option in curl to use all ftp commands. For Example to upload a file, please use the sample code:

```
<?php
// http://curl.phptrack.com
// PHP/CURL FTP upload to a remote site
// Copyright imran@phptrack.com

$ftp_url = "ftp://id:password@phptrack.com/public_html/cu/dt.jpg";
$file_to_upload_path = realpath("product.jpg");
$file_size = filesize($file_to_upload_path);
$fp = fopen($file_to_upload_path, 'rb');

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $ftp_url);
curl_setopt($ch, CURLOPT_UPLOAD, 1);
curl_setopt($ch, CURLOPT_INFILE, $fp);
curl_setopt($ch, CURLOPT_INFILESIZE, $file_size);
$result = curl_exec ($ch);
curl_close ($ch);
print $result;

?>
```

4.16 FTP Delete a file

You have the option in curl to use all ftp commands. For Example to delete a file, please use the sample code:

```
<?php
// http://curl.phptrack.com
// PHP/CURL FTP delete a remote file
// Copyright imran@phptrack.com

$ftp_url = "ftp://id:password@phptrack.com/public_html/curl/examples/";

// this will delete file "dest.jpg" from ftp path "/public_html/curl/examples"
// please change your path/file name correctly.

$file_path = '/public_html/curl/examples';
$file_name = 'dest.jpg';

$POSTFIELDS[0]= 'CWD '.$file_path;
$POSTFIELDS[1]= 'DELE '.$file_name;

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $ftp_url);
curl_setopt($ch, CURLOPT_POSTQUOTE,$POSTFIELDS);
$result = curl_exec ($ch);
curl_close ($ch);
print $result;

?>
```

5.0 LibCURL Options

Option	Set value to	Notes
CURLOPT_AUTOREFERER	TRUE to automatically set the <i>Referer:</i> field in requests where it follows a <i>Location:</i> redirect.	Available since PHP 5.1.0.
CURLOPT_BINARYTRANSFER	TRUE to return the raw output when CURLOPT_RETURNTRANSFER is used.	
CURLOPT_COOKIESESSION	TRUE to mark this as a new cookie "session". It will force libcurl to ignore all cookies it is about to load that are "session cookies" from the previous session. By default, libcurl always stores and loads all cookies, independent if they are session cookies are not. Session cookies are cookies without expiry date and they are meant to be alive and existing for this "session" only.	Available since PHP 5.1.0.
CURLOPT_CRLF	TRUE to convert Unix newlines to CRLF newlines on transfers.	
CURLOPT_DNS_USE_GLOBAL_CACHE	TRUE to use a global DNS cache. This option is not thread-safe and is enabled by default.	
CURLOPT_FAILONERROR	TRUE to fail silently if the HTTP code returned is greater than 300. The default behavior is to return the page normally, ignoring the code.	
CURLOPT_FILETIME	TRUE to attempt to retrieve the modification date of the remote document. You can then retrieve this value using the CURLINFO_FILETIME option with curl_getinfo() .	
CURLOPT_FOLLOWLOCATION	TRUE to follow any " <i>Location:</i> " header that the server sends as part of the HTTP header (note this is recursive, PHP will follow as many " <i>Location:</i> " headers that it is sent, unless CURLOPT_MAXREDIRS is set).	
CURLOPT_FORBID_REUSE	TRUE to force the connection to explicitly close when it has finished processing, and not be pooled for reuse.	
CURLOPT_FRESH_CONNECT	TRUE to force the use of a new connection instead of a cached one.	
CURLOPT_FTP_USE_EPRT	TRUE to use EPRT (and LPRT) when doing active FTP downloads. Use FALSE to disable EPRT and LPRT and use PORT only.	Added in PHP 5.0.0.
CURLOPT_FTP_USE_EPSV	TRUE to first try an EPSV command for FTP transfers before reverting back to PASV. Set to FALSE to disable EPSV.	
CURLOPT_FTPAPPEND	TRUE to append to the remote file instead of overwriting it.	
CURLOPT_FTPASCII	An alias of CURLOPT_TRANSFERTEXT . Use that instead.	

Option	Set value to	Notes
CURLOPT_FTPLISTONLY	TRUE to only list the names of an FTP directory.	
CURLOPT_HEADER	TRUE to include the header in the output.	
CURLOPT_HTTPGET	TRUE to reset the HTTP request method to GET. Since GET is the default, this is only necessary if the request method has been changed.	
CURLOPT_HTTPPROXYTUNNEL	TRUE to tunnel through a given HTTP proxy.	
CURLOPT_MUTE	TRUE to be completely silent with regards to the CURL functions.	
CURLOPT_NETRC	TRUE to scan your <code>~/.netrc</code> file to find your username and password for the remote site that you're establishing a connection with.	
CURLOPT_NOBODY	TRUE to exclude the body from the output.	
CURLOPT_NOPROGRESS	TRUE to disable the progress meter for CURL transfers. Note: PHP automatically sets this option to TRUE , this should only be changed for debugging purposes.	
CURLOPT_NOSIGNAL	TRUE to ignore any CURL function that causes a signal to be sent to the PHP process. This is turned on by default in multi-threaded SAPIs so timeout options can still be used.	Added in CURL 7.10 and PHP 5.0.0.
CURLOPT_POST	TRUE to do a regular HTTP POST. This POST is the normal <i>application/x-www-form-urlencoded</i> kind, most commonly used by HTML forms.	
CURLOPT_PUT	TRUE to HTTP PUT a file. The file to PUT must be set with CURLOPT_INFILE and CURLOPT_INFILESIZE .	
CURLOPT_RETURNTRANSFER	TRUE to return the transfer as a string of the return value of curl_exec() instead of outputting it out directly.	
CURLOPT_SSL_VERIFYPEER	FALSE to stop CURL from verifying the peer's certificate. Alternate certificates to verify against can be specified with the CURLOPT_CAINFO option or a certificate directory can be specified with the CURLOPT_CAPATH option. CURLOPT_SSL_VERIFYHOST may also need to be TRUE or FALSE if CURLOPT_SSL_VERIFYPEER is disabled (it defaults to 2).	TRUE by default as of CURL 7.10. Default bundle installed as of CURL 7.10.
CURLOPT_TRANSFERTEXT	TRUE to use ASCII mode for FTP transfers. For LDAP, it retrieves data in plain text instead of HTML. On Windows systems, it will not set <i>STDOUT</i> to binary mode.	
CURLOPT_UNRESTRICTED_AUTH	TRUE to keep sending the username and password when following locations (using CURLOPT_FOLLOWLOCATION), even when the hostname has changed.	Added in PHP 5.0.0.
CURLOPT_UPLOAD	TRUE to prepare for an upload.	
CURLOPT_VERBOSE	TRUE to output verbose information. Writes output to <i>STDERR</i> , or the file specified using CURLOPT_STDERR .	

value should be an integer for the following values of the *option* parameter:

Option	Set <i>value</i> to	Notes
CURLOPT_BUFFERSIZE	The size of the buffer to use for each read. There is no guarantee this request will be fulfilled, however.	Added in CURL 7.10 and PHP 5.0.0.
CURLOPT_CLOSEPOLICY	Either <i>CURLCLOSEPOLICY_LEAST_RECENTLY_USED</i> or <i>CURLCLOSEPOLICY_OLDEST</i> . There are three other <i>CURLCLOSEPOLICY_</i> constants, but CURL does not support them yet.	
CURLOPT_CONNECTTIMEOUT	The number of seconds to wait whilst trying to connect. Use 0 to wait indefinitely.	
CURLOPT_DNS_CACHE_TIMEOUT	The number of seconds to keep DNS entries in memory. This option is set to 120 (2 minutes) by default.	
CURLOPT_FTPSSLAUTH	The FTP authentication method (when is activated): <i>CURLFTPAUTH_SSL</i> (try SSL first), <i>CURLFTPAUTH_TLS</i> (try TLS first), or <i>CURLFTPAUTH_DEFAULT</i> (let CURL decide).	Added in CURL 7.12.2 and PHP 5.1.0.
CURLOPT_HTTP_VERSION	<i>CURL_HTTP_VERSION_NONE</i> (default, lets CURL decide which version to use), <i>CURL_HTTP_VERSION_1_0</i> (forces HTTP/1.0), or <i>CURL_HTTP_VERSION_1_1</i> (forces HTTP/1.1).	
CURLOPT_HTTPAUTH	The HTTP authentication method(s) to use. The options are: <i>CURLAUTH_BASIC</i> , <i>CURLAUTH_DIGEST</i> , <i>CURLAUTH_GSSNEGOTIATE</i> , <i>CURLAUTH_NTLM</i> , <i>CURLAUTH_ANY</i> , and <i>CURLAUTH_ANYSAFE</i> . You can use the bitwise (or) operator to combine more than one method. If you do this, CURL will poll the server to see what methods it supports and pick the best one. <i>CURLAUTH_ANY</i> is an alias for <i>CURLAUTH_BASIC</i> <i>CURLAUTH_DIGEST</i> <i>CURLAUTH_GSSNEGOTIATE</i> <i>CURLAUTH_NTLM</i> . <i>CURLAUTH_ANYSAFE</i> is an alias for <i>CURLAUTH_DIGEST</i> <i>CURLAUTH_GSSNEGOTIATE</i> <i>CURLAUTH_NTLM</i> .	Added in PHP 5.0.0.
CURLOPT_INFILESIZE	The expected size, in bytes, of the file when uploading a file to a remote site.	
CURLOPT_LOW_SPEED_LIMIT	The transfer speed, in bytes per second, that the transfer should be below during CURLOPT_LOW_SPEED_TIME seconds for PHP to consider the transfer too slow and abort.	
CURLOPT_LOW_SPEED_TIME	The number of seconds the transfer should be below CURLOPT_LOW_SPEED_LIMIT for PHP to consider the transfer too slow and abort.	
CURLOPT_MAXCONNECTS	The maximum amount of persistent connections that are allowed. When the limit is reached, CURLOPT_CLOSEPOLICY is used to determine which connection to close.	
CURLOPT_MAXREDIRS	The maximum amount of HTTP redirections to follow. Use this option alongside CURLOPT_FOLLOWLOCATION .	
CURLOPT_PORT	An alternative port number to connect to.	
CURLOPT_PROXYAUTH	The HTTP authentication method(s) to use for the proxy connection. Use the same bitmasks as described in CURLOPT_HTTPAUTH . For proxy authentication, only	Added in CURL 7.10.7

Option	Set value to	Notes
	<i>CURLAUTH_BASIC</i> and <i>CURLAUTH_NTLM</i> are currently supported.	and PHP 5.1.0.
CURLOPT_PROXYPORT	The port number of the proxy to connect to. This port number can also be set in CURLOPT_PROXY .	Added in PHP 5.0.0.
CURLOPT_PROXYTYPE	Either <i>CURLPROXY_HTTP</i> (default) or <i>CURLPROXY_SOCKS5</i> .	Added in CURL 7.10 and PHP 5.0.0.
CURLOPT_RESUME_FROM	The offset, in bytes, to resume a transfer from.	
CURLOPT_SSL_VERIFYHOST	1 to check the existence of a common name in the SSL peer certificate. 2 to check the existence of a common name and also verify that it matches the hostname provided.	
CURLOPT_SSLVERSION	The SSL version (2 or 3) to use. By default PHP will try to determine this itself, although in some cases you must set this manually.	
CURLOPT_TIMECONDITION	How CURLOPT_TIMEVALUE is treated. Use <i>CURL_TIMECOND_IFMODSINCE</i> to return the page only if it has been modified since the time specified in CURLOPT_TIMEVALUE . If it hasn't been modified, a "304 Not Modified" header will be returned assuming CURLOPT_HEADER is TRUE . Use <i>CURL_TIMECOND_ISUNMODSINCE</i> for the reverse effect. <i>CURL_TIMECOND_IFMODSINCE</i> is the default.	Added in PHP 5.1.0.
CURLOPT_TIMEOUT	The maximum number of seconds to allow CURL functions to execute.	
CURLOPT_TIMEVALUE	The time in seconds since January 1st, 1970. The time will be used by CURLOPT_TIMECONDITION . By default, <i>CURL_TIMECOND_IFMODSINCE</i> is used.	

value should be a string for the following values of the *option* parameter:

Option	Set value to	Notes
CURLOPT_CAINFO	The name of a file holding one or more certificates to verify the peer with. This only makes sense when used in combination with CURLOPT_SSL_VERIFYPEER .	
CURLOPT_CAPATH	A directory that holds multiple CA certificates. Use this option alongside CURLOPT_SSL_VERIFYPEER .	
CURLOPT_COOKIE	The contents of the "Set-Cookie: " header to be used in the HTTP request.	
CURLOPT_COOKIEFILE	The name of the file containing the cookie data. The cookie file can be in Netscape format, or just plain HTTP-style headers dumped into a file.	
CURLOPT_COOKIEJAR	The name of a file to save all internal cookies to when the connection closes.	
CURLOPT_CUSTOMREQUEST	A custom request method to use instead of "GET" or "HEAD" when doing a HTTP request. This is useful for doing "DELETE" or other, more obscure HTTP requests. Valid values are things like "GET", "POST", "CONNECT" and so on; i.e. Do not enter a whole HTTP request line here. For instance. enterina "GET /index.html"	

Option	Set value to	Notes
	<i>HTTP/1.0\r\n\r\n</i> would be incorrect. Note: Don't do this without making sure your server supports the custom request method first.	
CURLOPT_EGSOCKET	Like CURLOPT_RANDOM_FILE , except a filename to an Entropy Gathering Daemon socket.	
CURLOPT_ENCODING	The contents of the <i>"Accept-Encoding: "</i> header. This enables decoding of the response. Supported encodings are <i>"identity"</i> , <i>"deflate"</i> , and <i>"gzip"</i> . If an empty string, <i>""</i> , is set, a header containing all supported encoding types is sent.	Added in CURL 7.10.
CURLOPT_FTPPORT	The value which will be used to get the IP address to use for the FTP "POST" instruction. The "POST" instruction tells the remote server to connect to our specified IP address. The string may be a plain IP address, a hostname, a network interface name (under Unix), or just a plain '-' to use the systems default IP address.	
CURLOPT_INTERFACE	The name of the outgoing network interface to use. This can be an interface name, an IP address or a host name.	
CURLOPT_KRB4LEVEL	The KRB4 (Kerberos 4) security level. Any of the following values (in order from least to most powerful) are valid: <i>"clear"</i> , <i>"safe"</i> , <i>"confidential"</i> , <i>"private"</i> .. If the string does not match one of these, <i>"private"</i> is used. Setting this option to NULL will disable KRB4 security. Currently KRB4 security only works with FTP transactions.	
CURLOPT_POSTFIELDS	The full data to post in a HTTP "POST" operation.	
CURLOPT_PROXY	The HTTP proxy to tunnel requests through.	
CURLOPT_PROXYUSERPWD	A username and password formatted as <i>"[username]:[password]"</i> to use for the connection to the proxy.	
CURLOPT_RANDOM_FILE	A filename to be used to seed the random number generator for SSL.	
CURLOPT_RANGE	Range(s) of data to retrieve in the format <i>"X-Y"</i> where X or Y are optional. HTTP transfers also support several intervals, separated with commas in the format <i>"X-Y,N-M"</i> .	
CURLOPT_REFERER	The contents of the <i>"Referer: "</i> header to be used in a HTTP request.	
CURLOPT_SSL_CIPHER_LIST	A list of ciphers to use for SSL. For example, <i>RC4-SHA</i> and <i>TLSv1</i> are valid cipher lists.	
CURLOPT_SSLCERT	The name of a file containing a PEM formatted certificate.	
CURLOPT_SSLCERTPASSWD	The password required to use the CURLOPT_SSLCERT certificate.	
CURLOPT_SSLCERTTYPE	The format of the certificate. Supported formats are <i>"PEM"</i> (default), <i>"DER"</i> , and <i>"ENG"</i> .	Added in CURL 7.9.3 and PHP 5.0.0.
CURLOPT_SSLENGINE	The identifier for the crypto engine of the private SSL key specified in CURLOPT_SSLKEY .	
CURLOPT_SSLENGINE_DEFAULT	The identifier for the crypto engine used for asymmetric crypto operations.	
CURLOPT_SSLKEY	The name of a file containing a private SSL key.	

PHP/CURL Book with Examples

Option	Set value to	Notes
CURLOPT_SSLKEYPASSWD	The secret password needed to use the private SSL key specified in CURLOPT_SSLKEY . Note: Since this option contains a sensitive password, remember to keep the PHP script it is contained within safe.	
CURLOPT_SSLKEYTYPE	The key type of the private SSL key specified in CURLOPT_SSLKEY . Supported key types are "PEM" (default), "DER", and "ENG".	
CURLOPT_URL	The URL to fetch. You can also set this when initializing a session with curl_init() .	
CURLOPT_USERAGENT	The contents of the "User-Agent: " header to be used in a HTTP request.	
CURLOPT_USERPWD	A username and password formatted as "[username]:[password]" to use for the connection.	

value should be an array for the following values of the *option* parameter:

Option	Set value to	Notes
CURLOPT_HTTP200ALIASES	An array of HTTP 200 responses that will be treated as valid responses and not as errors.	Added in CURL 7.10.3 and PHP 5.0.0.
CURLOPT_HTTPHEADER	An array of HTTP header fields to set.	
CURLOPT_POSTQUOTE	An array of FTP commands to execute on the server after the FTP request has been performed.	
CURLOPT_QUOTE	An array of FTP commands to execute on the server prior to the FTP request.	

value should be a stream resource (using [fopen\(\)](#), for example) for the following values of the *option* parameter:

Option	Set value to	Notes
CURLOPT_FILE	The file that the transfer should be written to. The default is <i>STDOUT</i> (the browser window).	
CURLOPT_INFILE	The file that the transfer should be read from when uploading.	
CURLOPT_STDERR	An alternative location to output errors to instead of <i>STDERR</i> .	
CURLOPT_WRITEHEADER	The file that the header part of the transfer is written to.	

value should be a string *t*

that is the name of a valid callback function for the following values of the *option* parameter:

Option	Set value to	Notes
CURLOPT_HEADERFUNCTION	The name of a callback function where the callback function takes two parameters. The first is the CURL resource, the second is a string with the header data to be written. Using this callback function, it becomes your responsibility to write the header data. Return the number of bytes written.	
CURLOPT_PASSWDFUNCTION	The name of a callback function where the callback function takes three parameters. The first is the CURL resource, the second is a string containing a password prompt, and the third is the maximum password length. Return the string containing the password.	
CURLOPT_READFUNCTION	The name of a callback function where the callback function takes two parameters. The first is the CURL resource, and the second is a string with the data to be read. Using this callback function, it becomes your responsibility to read the data. Return the number of bytes read. Return 0 to signal <i>EOF</i> .	
CURLOPT_WRITEFUNCTION	The name of a callback function where the callback function takes two parameters. The first is the CURL resource, and the second is a string with the data to be written. Using this callback function, it becomes your responsibility to write the data. Must return the exact number of bytes written or this will fail.	

6.0 Header Tracking Tools

6.1 LiveHTTPHeader

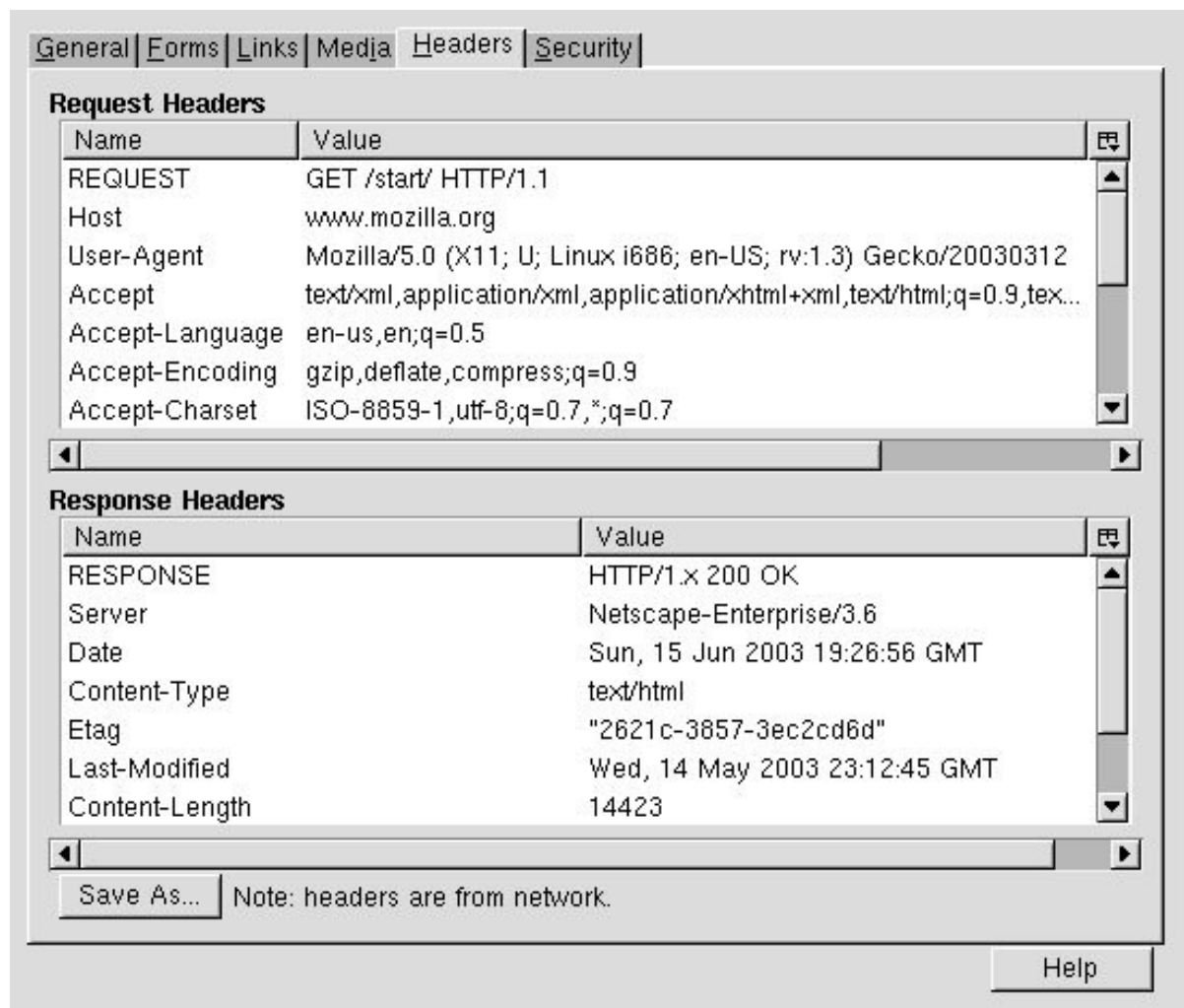
A very good helper to make sure you do this right, is the LiveHTTPHeader tool that lets you view all headers you send and receive with Mozilla/Firefox (even when using HTTPS).

I have first installed **Netscape 7.1**

Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4) Gecko/20030624 Netscape/7.1 (ax)

On my machine and then **livehttpheaders version 0.9** from site <http://livehttpheaders.mozdev.org>

Here are is screen shots of LiveHTTPHeaders 0.9:



6.2 EtheReal

A more raw approach is to capture the HTTP traffic on the network with tools such as ethereal or tcpdump and check what headers that were sent and received by the browser. (HTTPS makes this technique inefficient.) Check the <http://www.ethereal.com> for more information.

6.3 depth understanding

RFC 2616 is a must to read if you want in-depth understanding of the HTTP protocol.

RFC 2396 explains the URL syntax.

RFC 2109 defines how cookies are supposed to work.

RFC 1867 defines the HTTP post upload format.

7.0 RealTime Examples

7.1 Ebay Login.

This Code Let you Login on Ebay Website by Using CURL + PHP.

```
<?php
/*
.....
'   File:                ebay_login.php
'
'   Description:         This script Login you on Ebay.com website with SSL using curl in
php.
'
'   Written by:          Imran Khalid imranlink@hotmail.com
'
'   Languages:           PHP + CURL
'
'   Date Written:        January 08, 2005
'
'   Version:             V.2.0
'
'   Platform:            Windows 2000 / IIS / Netscape 7.1
'
'   Copyright:           Imran Khalid imranlink@hotmail.com
'
.....
*/

// 1-Get First Login Page http://signin.ebay.com/ws/eBayISAPI.dll?SignIn
// This page will set some cookies and we will use them for Posting in Form data.
if($_POST['ebay_user_id'])
{
    $ebay_user_id = $_POST['ebay_user_id']; // Please set your Ebay ID
    $ebay_user_password = $_POST['ebay_user_password']; // Please set your Ebay Password
    $cookie_file_path = "C:/Inetpub/wwwroot/spiders/cookie/cook"; // Please set your Cookie
File path

    // log out.
    $LOGINURL = "http://signin.ebay.com/ws/eBayISAPI.dll?SignIn";
    $agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4) Gecko/20030624
Netscape/7.1 (ax)";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $LOGINURL);
    curl_setopt($ch, CURLOPT_USERAGENT, $agent);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
    $result = curl_exec ($ch);
    curl_close ($ch);

    $LOGINURL = "http://signin.ebay.com/ws2/eBayISAPI.dll?SignIn";
    $agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4) Gecko/20030624
Netscape/7.1 (ax)";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $LOGINURL);
```

```
curl_setopt($ch, CURLOPT_USERAGENT, $agent);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
$result = curl_exec ($ch);
curl_close ($ch);

// 2- Post Login Data to Page
https://signin.ebay.com/ws/eBayISAPI.dll?co_partnerid=2&siteid=0&UsingSSL=1

$LOGINURL = "https://signin.ebay.com/ws/eBayISAPI.dll?co_partnerid=2&siteid=0&UsingSSL=1";
$POSTFIELDS =
'MfcISAPICommand=SignInWelcome&siteid=0&co_partnerId=2&UsingSSL=1&ru=&pp=&pal=&pa2=&pa3=&i1=-
1&pageType=-1&userid='. $ebay_user_id .'&pass='. $ebay_user_password .'&keepMeSignInOption=1';
$referrer =
"https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&favoritenav=&sid=&ruproduct=&pp=&co_partnerId
=2&ru=&i1=&ruparams=&pageType=&pa2=&bshowgif=&pal=&pUserId=&errmsg=&UsingSSL=&runame=&siteid=0
";

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,$LOGINURL);
curl_setopt($ch, CURLOPT_USERAGENT, $agent);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,$POSTFIELDS);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_REFERER, $referrer);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
$result = curl_exec ($ch);
curl_close ($ch);

if (strpos($result, "Your sign in information is not valid"))
{
    echo "\r\n<br><p>Your sign in information is not valid.</p>";
    login_form();
}
else
{

    $LOGINURL = "https://arribada.ebay.com/saw-cgi/eBayISAPI.dll?PlaceCCInfo";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,$LOGINURL);
    curl_setopt($ch, CURLOPT_USERAGENT, $agent);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
    $result = curl_exec ($ch);
    curl_close ($ch);
    //print $result;
    $LOGINURL =
"https://signin.ebay.com/ws/eBayISAPI.dll?co_partnerid=2&siteid=0&UsingSSL=1";
    $POSTFIELDS =
'MfcISAPICommand=SignInWelcome&siteid=0&co_partnerId=2&UsingSSL=1&ru=https%3A%2F%2Farribada.eb
ay.com%2Fsaw-
cgi%2FeBayISAPI.dll%3FPlaceCCInfo%26page%3D0%26adult%3D0%26ru%3Ddefault%26BillingAccountType%3
Ddefault%26pass%3D%7B_pass_%7D%26fromsyi%3D0%26reporting%3D0%26userid%3D&pp=pass&pal=&pa2=&pa3
=&i1=0&pageType=955&userid='. $ebay_user_id .'&pass='. $ebay_user_password;
```

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,$LOGINURL);
curl_setopt($ch, CURLOPT_USERAGENT, $agent);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,$POSTFIELDS);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
$address = curl_exec ($ch);
curl_close ($ch);

    print $address;
} // if login success
} // if form post
else
{
    login_form();
}
////////////////////////////////////
function filter_text( $start, $end, $str_page ){

    $pos = strpos ( $str_page, $start );
    if ( $pos != false ) {
        $pos = $pos + strlen($start);
        $field_value = substr( $str_page, $pos );
        $pos = strpos ( $field_value, $end );
        $field_value = substr( $field_value, 0, $pos );
        $field_value = trim($field_value);
    } // if
    else {
        $field_value = "";
    }
    return $field_value;
} // function
function login_form()
{
    ?>
<form method="post" name="SignInForm" action="ebay_account.php">

<table border="0" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td></td>
</tr>
<tr>
<td valign="top"> <font color="#ff0000" size="4">&nbsp;</font><font size="4">eBay
members, sign in to save time for bidding, selling, and other activities.
</font><br></td>
</tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td></td>
</tr>
<tr>
<td valign="top"><b>eBay User ID</b><br><input type="text" name="ebay_user_id" maxlength="64"
tabindex="1" value="" size="27"><br>
</td>
</tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
```

```
<tr>
<td></td>
</tr>
<tr>
<td valign="top"><b>Password</b><br><input type="password" name="ebay_user_password"
maxlength="64" value="" tabindex="2" size="27"><br>
</td>

</tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="350">
<tr>
<td colspan="2"></td>
</tr>
<tr>
<td width="35%"><input type="submit" tabindex="3" value="Sign In Securely "></td>
</tr>
</table>

</form>

<?
}
?>
```

7.2 Ebay Summary Page.

```
<?php
set_time_limit(0);
/*
.....
'   File:                ebay_Account.php
'   Description:         This script Login you on Ebay.com website with SSL using
curl in php and then take to summary page.
'   Written by:          Imran Khalid imranlink@hotmail.com
'   Languages:           PHP + CURL
'   Date Written:        January 08, 2005
'   Version:             V.2.0
'   Platform:            Windows 2000 / IIS / Netscape 7.1
'   Copyright:           Imran Khalid imranlink@hotmail.com
.....
*/

// 1-Get First Login Page http://signin.ebay.com/ws2/eBayISAPI.dll?SignIn
// This page will set some cookies and we will use them for Posting in Form data.
if($_POST['ebay_user_id'])
{
    $ebay_user_id = $_POST['ebay_user_id']; // Please set your Ebay ID
    $ebay_user_password = $_POST['ebay_user_password']; // Please set your Ebay
Password
    $cookie_file_path = "C:\Apache\Apache\htdocs\cookie.txt"; // Please set your Cookie
File path

    // log out.
    $LOGINURL = "http://signin.ebay.com/ws/eBayISAPI.dll?SignIn";
    $agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4) Gecko/20030624
Netscape/7.1 (ax)";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,$LOGINURL);
    curl_setopt($ch, CURLOPT_USERAGENT, $agent);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
    $result = curl_exec ($ch);
    curl_close ($ch);

    $LOGINURL = "http://signin.ebay.com/ws2/eBayISAPI.dll?SignIn";
    $agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4) Gecko/20030624
Netscape/7.1 (ax)";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,$LOGINURL);
    curl_setopt($ch, CURLOPT_USERAGENT, $agent);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
    $result = curl_exec ($ch);
    curl_close ($ch);

// 2- Post Login Data to Page
https://signin.ebay.com/ws/eBayISAPI.dll?co_partnerid=2&siteid=0&UsingSSL=1

    $LOGINURL =
"https://signin.ebay.com/ws/eBayISAPI.dll?co_partnerid=2&siteid=0&UsingSSL=1";
    $POSTFIELDS =
```



```
'MfcISAPICommand=SignInWelcome&siteid=0&co_partnerId=2&UsingSSL=1&ru=&pp=&pal=&pa2=&pa3
=&i1=-1&pageType=-1&userid='. $ebay_user_id .'&pass='. $ebay_user_password
.'&keepMeSignInOption=1';
    $referrer =
"https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&favoritenav=&sid=&ruproduct=&pp=&co_pa
rtnerId=2&ru=&i1=&ruparams=&pageType=&pa2=&bshowgif=&pal=&pUserId=&errmsg=&UsingSSL=&ru
name=&siteid=0";

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $LOGINURL);
    curl_setopt($ch, CURLOPT_USERAGENT, $agent);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $POSTFIELDS);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_REFERER, $referrer);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
    $result = curl_exec ($ch);
    curl_close ($ch);

    if (strstr($result, "Your sign in information is not valid"))
    {
        echo "\r\n<br><p>Your sign in information is not valid.</p>";
        login_form();
    }
    else
    {

        $LOGINURL = "http://my.ebay.com/ws/eBayISAPI.dll?MyeBay";
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $LOGINURL);
        curl_setopt($ch, CURLOPT_USERAGENT, $agent);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
        curl_setopt($ch, CURLOPT_REFERER, $referrer);
        curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
        curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
        $result = curl_exec ($ch);
        curl_close ($ch);
        print $result;

    } // if login success
} // if form post
else
{
    login_form();
}
////////////////////////////////////
function filter_text( $start, $end, $str_page ){

    $pos = strpos ( $str_page, $start );
    if ( $pos != false ) {
        $pos = $pos + strlen($start);
        $field_value = substr( $str_page, $pos );
        $pos = strpos ( $field_value, $end );
        $field_value = substr( $field_value, 0, $pos );
        $field_value = trim($field_value);
    } // if
    else {
        $field_value = "";
    }
}
```

```
        return $field_value;
    } // function
function login_form()
{
    ?>
    <form method="post" name="SignInForm" action="ebay_account.php">

    <table border="0" cellpadding="0" cellspacing="0" width="100%">
    <tr>
    <td></td>
    </tr>
    <tr>
        <td valign="top"> <font color="#ff0000" size="4">&nbsp;</font><font size="4">eBay
        members, sign in to save time for bidding, selling, and other activities.
        </font><br></td>
    </tr>
    </table>
    <table border="0" cellpadding="0" cellspacing="0" width="100%">
    <tr>
    <td></td>
    </tr>
    <tr>
    <td valign="top"><b>eBay User ID</b><br><input type="text" name="ebay_user_id"
    maxlength="64" tabindex="1" value="" size="27"><br>
        </td>
    </tr>
    </table>
    <table border="0" cellpadding="0" cellspacing="0" width="100%">
    <tr>
    <td></td>
    </tr>
    <tr>
    <td valign="top"><b>Password</b><br><input type="password" name="ebay_user_password"
    maxlength="64" value="" tabindex="2" size="27"><br>
        </td>
    </tr>
    </table>
    <table border="0" cellpadding="0" cellspacing="0" width="350">
    <tr>
    <td colspan="2"></td>
    </tr>
    <tr>
    <td width="35%"><input type="submit" tabindex="3" value="Sign In Securely "></td>
    </tr>
    </table>

    </form>

    <?
    }
    ?>
```

7.3 Calculate Fedex Shipping Charges with XML CURL PHP support.

First need is to Register for Fedex XML API. Please use the sample code to first register.

```
<?php
// fedex_register.php
// Register XML API request on Fedex Server
// http://curl.phptrack.com
// Copyright imran@phptrack.com
// set your AccountNumber 312397xxx
// set your account info etc.

set_time_limit(0);

$xml = '<?xml version="1.0" encoding="UTF-8" ?>
<FDXSubscriptionRequest xmlns:api="http://www.fedex.com/fsmapi" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="FDXSubscriptionRequest.xsd">
<RequestHeader>
<CustomerTransactionIdentifier>String</CustomerTransactionIdentifier>
<AccountNumber>312397xxx</AccountNumber>
</RequestHeader>
<Contact>
<PersonName>Imran Khalid</PersonName>
<CompanyName>PHP computers</CompanyName>
<Department>IT</Department>
<PhoneNumber>8006117725</PhoneNumber>
<PagerNumber>4595746241</PagerNumber>
<FaxNumber>4655750245</FaxNumber>
<E-MailAddress>info@phptrack.com</E-MailAddress>
</Contact>
<Address>
<Line1>318 S</Line1>
<Line2>INTERSTATE 55E</Line2>
<City>CARROLLTON</City>
<StateOrProvinceCode>TX</StateOrProvinceCode>
<PostalCode>75006</PostalCode>
<CountryCode>US</CountryCode>
</Address>
</FDXSubscriptionRequest>';
```

```
$LOGINURL = "https://gatewaybeta.fedex.com:443/GatewayDC";
$cookie_file_path = "C:/Inetpub/wwwroot/sept2005/phptrack/curl/forum_h
elp_codes/hotmail/cookie.php";
// Please set your Cookie File path
$agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-
US; rv:1.4) Gecko/20030624 Netscape/7.1 (ax)";
$reffer = "https://gatewaybeta.fedex.com";

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $LOGINURL);
curl_setopt($ch, CURLOPT_USERAGENT, $agent);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $xml);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_REFERER, $reffer);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
$result = curl_exec ($ch);
curl_close ($ch);
$result = str_replace('><', ">\r\n<", $result);
echo "<textarea rows=30 cols=130>".$result."</textarea>";

?>
```

Output of the above page:

```
<?xml version="1.0" encoding="UTF-8" ?>
=<FDXSubscriptionReply
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FDXSubscriptionReply.xsd">
=<ReplyHeader>

  <CustomerTransactionIdentifier>String</CustomerTransactionIden
  tifier>
</ReplyHeader>
<MeterNumber>1154634</MeterNumber>
<SubscribedService>FedEx Express Shipping</SubscribedService>
<SubscribedService>FedEx Ground Shipping</SubscribedService>
</FDXSubscriptionReply>
```

After getting the MeterNumber, We will send the request for shipping calculation.

```
<?php
// fedex_rates.php
// XML API to request Fedex shipping charges
// http://curl.phptrack.com
// Copyright imran@phptrack.com
// set your AccountNumber 312397xxx
// set your MeterNumber 1154634xxx
// set your account info etc.
    set_time_limit(0);

    $xml = '<?xml version="1.0" encoding="UTF-8" ?>
<FDXRateRequest xmlns:api="http://www.fedex.com/fsmapi" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FDXRateRequest.xsd">
<RequestHeader>
<CustomerTransactionIdentifier>CTIString</CustomerTransactionIdentifier>
<AccountNumber>312397xxx</AccountNumber>
<MeterNumber>1154634</MeterNumber>
<CarrierCode>FDXE</CarrierCode>
</RequestHeader>
<ShipDate>2006-03-13</ShipDate>
<DropoffType>REGULARPICKUP</DropoffType>
<Service>PRIORITYOVERNIGHT</Service>
<Packaging>FEDEXBOX</Packaging>
<WeightUnits>LBS</WeightUnits>
<Weight>10.0</Weight>
<OriginAddress>
<StateOrProvinceCode>TN</StateOrProvinceCode>
<PostalCode>37115</PostalCode>
<CountryCode>US</CountryCode>
</OriginAddress>
<DestinationAddress>
<StateOrProvinceCode>TX</StateOrProvinceCode>
<PostalCode>73301</PostalCode>
<CountryCode>US</CountryCode>
</DestinationAddress>
<Payment>
<PayorType>SENDER</PayorType>
</Payment>
<PackageCount>1</PackageCount>
</FDXRateRequest>';
```

```
$LOGINURL = "https://gatewaybeta.fedex.com:443/GatewayDC";
$cookie_file_path = "C:/inetpub/wwwroot/sept2005/phptrack/curl/forum_h
elp_codes/hotmail/cookie.php"; // Please set your Cookie File path
$agent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-
US; rv:1.4) Gecko/20030624 Netscape/7.1 (ax)";
$referrer = "https://gatewaybeta.fedex.com";

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $LOGINURL);
curl_setopt($ch, CURLOPT_USERAGENT, $agent);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $xml);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_REFERER, $referrer);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);
$result = curl_exec ($ch);
curl_close ($ch);
$result = str_replace('><', ">\r\n<", $result);
echo "<textarea rows=30 cols=130>".$result."</textarea>";

?>
```

Output of the above page:

```
<?xml version="1.0" encoding="UTF-8" ?>
=<FDXRateReply xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
=<ReplyHeader>
  <CustomerTransactionIdentifier>CTIString</CustomerTransactionIdent
ifier>
</ReplyHeader>
=<EstimatedCharges>
  <RateScale>01552</RateScale>
  <RateZone>5</RateZone>
  <CurrencyCode>USD</CurrencyCode>
  <BilledWeight>10.0</BilledWeight>
=<DiscountedCharges>
  <BaseCharge>57.45</BaseCharge>
  <TotalDiscount>0.00</TotalDiscount>
=<Surcharges>
  <Fuel>7.18</Fuel>
  <Other>0.00</Other>
</Surcharges>
  <TotalSurcharge>7.18</TotalSurcharge>
  <NetCharge>64.63</NetCharge>
  <TotalRebate>0.00</TotalRebate>
</DiscountedCharges>
</EstimatedCharges>
  <SignatureOption>NONE</SignatureOption>
</FDXRateReply>
```

7.4 Yahoo Email login and download Address Book.

This PHP/cURL script let you automatically login on Yahoo email site (<http://mail.yahoo.com>) to download address Book in CSV format. Please view attachment file **examples/yahoomail.php** for source code.

7.5 PayPal Login and download Transaction History.

This PHP/cURL script let you automatically login on PayPal site (<http://www.paypal.com>) to download address Transaction History in CSV format. Please view attachment file **examples/paypal.php** for source code.

7.6 Hotmail Email login and download Contacts List.

This PHP/cURL script let you automatically login on Hotmail email site (<http://hotmail.com>) to download address Book in CSV format. Please view attachment file **examples/hotmail.php** for source code.

7.7 Gmail download Address Book.

This PHP/cURL script let you automatically login on gmail.google.com site ([http:// gmail.google.com](http://gmail.google.com)) to download address Book in CSV format. Please view attachment file **examples/gmail.php** for source code.

8.0 Regular Expressions

8.1 What are Regular Expressions?

Regular expressions were designed to make it easier to find, replace and work with strings. You could also use them to complete complex search and replace operations within a given body of text.

A regular expression (regex or regexp for short) is a special text string for describing a search pattern. You can think of regular expressions as wildcards on steroids. You are probably familiar with wildcard notations such as *.txt to find all text files in a file manager. The regex equivalent is `.*\.txt`

Regular expressions can provide a powerful way to work with text. Using regular expressions, you can do complex validation of user input, parse user input and file contents, and reformat strings

8.2 What exactly is possible with Regular Expressions?

Well, Regular Expressions can be used for many different things, namely:

1. Searching for string in a text
2. validating a string
3. pulling out text from string
4. parsing data from a file
5. And your needs

Regular expressions are actually one of the most powerful tools in PHP, or any other language for that matter (you can use it in your `mod_rewrite` rules as well!). There is so much you can do with regex, and we've only scratched the surface in this tutorial with some very basic examples.

8.3 Regular Expressions in PHP

Using regex (regular expressions) is really easy in PHP, and there are several functions that exist to do regex finding and replacing. Let's start with a simple regex find.

Regular Expressions Functions in PHP

Firstly, there are for PHP, 4 types of operations which use regular expressions:

1. `ereg` : performs regular expression operation

`bool ereg (string pattern, string string [, array regs])`

Searches a string for matches to the regular expression given in pattern in a case-sensitive way.

If matches are found for parenthesized substrings of pattern and the function is called with the third argument `regs`, the matches will be stored in the elements of the array `regs`. `$regs[1]` will contain the substring which starts at the first left parenthesis; `$regs[2]` will contain the substring starting at the second, and so on. `$regs[0]` will contain a copy of the complete string matched.

2. `eregi` : performs case-insensitive regular expression operation

`bool eregi (string pattern, string string [, array regs])`

This function is identical to `ereg()` except that this ignores case distinction when matching alphabetic characters.

3. `preg_match` : performs perl-compatible regular expression operation

`int preg_match (string pattern, string subject [, array matches [, int flags]])`

Searches subject for a match to the regular expression given in pattern.

If `matches` is provided, then it is filled with the results of search. `$matches[0]` will contain the text that matched the full pattern, `$matches[1]` will have the text that matched the first captured parenthesized subpattern, and so on.

`flags` can be the following flag:

`PREG_OFFSET_CAPTURE`

If this flag is set, for every occurring match the appendant string offset will also be returned. Note that this changes the return value in an array where every element is an array consisting of the matched string at offset 0 and it's string offset into subject at offset 1. This flag is available since PHP 4.3.0 .

`preg_match()` returns the number of times pattern matches. That will be either 0 times (no match) or 1 time because `preg_match()` will stop searching after the first match. **`preg_match_all()`** on the contrary will continue until it reaches the end of subject. `preg_match()` returns FALSE if an error occurred.

4. `preg_replace` : performs perl-compatible regular expression REPLACE operation

`mixed preg_replace (mixed pattern, mixed replacement, mixed subject [, int limit])`

Searches subject for matches to pattern and replaces them with replacement. If limit is specified, then only limit matches will be replaced; if limit is omitted or is -1, then all matches are replaced.

Replacement may contain references of the form `\n` or (since PHP 4.0.4) `$n`, with the latter form being the preferred one. Every such reference will be replaced by the text captured by the n'th parenthesized pattern. n can be from 0 to 99, and `\0` or `$0` refers to the text matched by the whole pattern. Opening parentheses are counted from left to right (starting from 1) to obtain the number of the capturing subpattern.

When working with a replacement pattern where a backreference is immediately followed by another number (i.e.: placing a literal number immediately after a matched pattern), you cannot use the familiar `\1` notation for your backreference. `\11`, for example, would confuse `preg_replace()` since it does not know whether you want the `\1` backreference followed by a literal 1, or the `\11` backreference followed by nothing. In this case the solution is to use `\${1}1`. This creates an isolated `$1` backreference, leaving the 1 as a literal.

5. `ereg_replace` : performs regular expression REPLACE operation

string **`ereg_replace`** (string pattern, string replacement, string string)

This function scans string for matches to pattern, then replaces the matched text with replacement.

The modified string is returned. (Which may mean that the original string is returned if there are no matches to be replaced.)

If pattern contains parenthesized substrings, replacement may contain substrings

of the form `\digit`, which will be replaced by the text matching the digit'th

parenthesized substring; `\0` will produce the entire contents of string. Up to nine

substrings may be used. Parentheses may be nested, in which case they are

counted by the opening parenthesis.

If no matches are found in string, then string will be returned unchanged.

6. `eregi_replace` : same as above; case-insensitive

string **`eregi_replace`** (string pattern, string replacement, string string)

8.4 Regular Expressions Examples

preg_match function

Have a look at the documentation of the preg_match function (http://php.net/preg_match). As you can see from the documentation, preg_match is used to perform a regular expression. In this case no replacing is done, only a simple find. Copy the code below to give it a try.

```
<?php
// Example reg001.php
// Simple Regular Expressions in PHP
// Copyright http://curl.phptrack.com
// Match some sub text inside a text string.

// Example string
$str = "Let's find the Name :<B>Muhammad Imran</B><table> some table</table>and
some fonts tags of </html>";

// Let's perform the regex
$flag = preg_match("/<B>(.*?)<\B>/", $str, $matches);

// Check if regex was successful
if ($flag = true) {
    // Matched something, show the matched string
    echo htmlentities($matches['0']);

    // Also how the text in between the tags
    echo '<br />' . $matches['1'];
} else {
    // No Match
    echo "Couldn't find a match";
}

?>
```

After having run the code, it's probably a good idea if I do a quick run through the code. Basically, the whole core of the above code is the line that contains the preg_match. The first argument is your regex pattern. This is probably the most important.

The second argument is the subject string. I assume that needs no explaining. Finally, the third argument can be optional, but if you want to get the matched text, or the text in between something, it's a good idea to use it.

```
<?php
// Example reg002.php
// Simple Regular Expressions in PHP
// Copyright http://curl.phptrack.com
// parsing name, email from hotmail contact html page.
// using php function preg_match_all
$str = '
    <html>
    <body>
    <table border=0 cellpadding=0 cellspacing=0 width=100% class="EE"
id="ListTable">
    <form name=doaddy action="/cgi-bin/doaddresses" method=POST>
    <input type=hidden name="" value="">
    <input type=hidden name=_HMaction value="">
    <input type=hidden name=i>
    <input type=hidden name=IsGroup>
    <input type=hidden name=strUsrFltr value="">
    <input type=hidden name=strUsrView value="">
    <input type=hidden name=strAlphNav value="">
    <input type=hidden name=a value=0aee4eac7ebe5d67fa50eb5267e959b02e
b61cb9ba63cb9598857c5a311822e5>
    <tr height=26>
    <td colspan=6 align=right style="BORDER-TOP:none;COLOR:#8D8D8D">
    <font class="K">ALL</font>&nbsp;&#&nbsp;&A&nbsp;&B&nbsp;&<a href="java
script:AN("", "", "C")">C</a>&nbsp;&D&nbsp;&E&nbsp;&F&nbsp;&G&nbsp;&H&nbsp;&I&nbsp;&
J&nbsp;&<a href="javascript:AN("", "", "K")">K</a>&nbsp;&L&nbsp;&M&nbsp;&N&nbsp;&
O&nbsp;&P&nbsp;&Q&nbsp;&R&nbsp;&S&nbsp;&T&nbsp;&U&nbsp;&V&nbsp;&W&nbsp;&X&nbsp;&Y&nbsp;&n
bsp;Z&nbsp;&&nbsp;&</td>
    </tr>
    <tr id="messPrompt">
    </tr>
    <tr bgcolor=#DBEAF5>
    <td width=1% height=24 align=center>&nbsp;&<input name=allbox type=
checkbox onClick="CA()">&nbsp;&</td>
    <td width=1%>&nbsp;&</td>
    <td bgcolor=#A0C6E5>
    <a href="javascript:AD("addrrev=1&addrsort=nick&strUsrFltr=&strUsr
View=&strAlphNav=")" title="Sort by Name" class="FF">
    Name</a>
    </td>
    <td >
    <a href="javascript:AD("addrrev=1&addrsort=email&strUsrFltr=&strUs
rView=&strAlphNav=")" title="Sort by E-Mail" class="FF">E-Mail</a>
    </td>
    <td>
    <font class="FF">Phone</font>
    </td>
    </tr>
    <tr name="" id="13bf6320-da8e-4d40-8759-3c4ab27e36a0">
    <td align=center>
```

```

        <input type=checkbox name="ADDR13bf6320-da8e-4d40-8759-
3c4ab27e36a0" onClick="CCA(this)" id="hotmail">
    </td>
    <td width=1%>
    </td>
    <td nowrap>
    <a href="#" onclick="javascript:DoAD("", "&strUsrView=", event);retu
rn false;">cheema</a>
    </td>
    <td>
    <a href="#" onclick="javascript:DC(event);return false;">cheema@gm
ail.com</a>
    </td>
    <td>
    </td>
    </td>
    </tr>
    <tr name="" id="2b151594-b7ed-4994-8953-44ace5bfc483">
    <td align=center>
    <input type=checkbox name="ADDR2b151594-b7ed-4994-8953-
44ace5bfc483" onClick="CCA(this)" id="hotmail">
    </td>
    <td width=1%>
    </td>
    <td nowrap>
    <a href="#" onclick="javascript:DoAD("", "&strUsrView=", event);retu
rn false;">kahlid</a>
    </td>
    <td>
    <a href="#" onclick="javascript:DC(event);return false;">khalid@ya
hoo.com</a>
    </td>
    <td>
    </td>
    </td>
    </tr>
    <tr name="" id="e7615392-aac4-45e2-87e8-96afa21eb928">
    <td align=center>
    <input type=checkbox name="ADDRe7615392-aac4-45e2-87e8-
96afa21eb928" onClick="CCA(this)" id="hotmail">
    </td>
    <td width=1%>
    </td>
    <td nowrap>
    <a href="#" onclick="javascript:DoAD("", "&strUsrView=", event);retu
rn false;">khalid, imran</a>
    </td>
    <td>
    <a href="#" onclick="javascript:DC(event);return false;">imran@hot
mail.com</a>
    </td>
    <td>
    </td>
    </td>
    </tr>
    <td colspan=5>&nbsp;</td>
    </table>
    </td>
    <tr>
    </table>
```

```
<br>
</form>
<br>
</td>
</tr>
</table>
</body>
</html>
';

// Let's perform the regex
$flag = preg_match_all("/event\);return false;\>(.*)\</a/", $str, $matches);

// Check if regex was successful
if ($flag = true)
{
    // Matched something, show the matched string
    print_r($matches['1']);
}

else
{
    // No Match
    echo "Couldn't find a match";
}

?>
```

preg_replace function

Now that we've had finding, let's do a find-and-replace, with the preg_replace function (http://www.php.net/preg_replace). The preg_replace function works pretty similar to the preg_match function, but instead there is another argument for the replacement string. Copy the code below, and run it.

```
<?php
// Example reg003.php
// Simple Regular Expressions in PHP
// Copyright http://curl.phptrack.com
// Match some text and replace it with a new string.

// Example string
$str = "We want to change amount $<h3>25.09</h3> of the item";

// Perform the preg replace
$result = preg_replace ("/<h3>(.*?)</h3>/", '<h4>20.36</h4>', $str);

echo htmlentities($result);
?>
```

The result would then be the same string, except it would now say '<h4>20.36</h4>' between the h3 tags. This is of course just a simple example, and more advanced replacements can be done.

You can also use keys in the replacement string. Say you still want the text between the brackets, and just add something? You use the \$1, \$2, etc keys for those. For example:

```
<?php
// Example reg004.php
// Simple Regular Expressions in PHP
// Copyright http://curl.phptrack.com
// Match some text inside a text string and
//replace it with new string and also add the old string in result.

// Example string
$str = "We want to change amount $<h3>25.09</h3> of the item";

// Do the preg replace
$result = preg_replace ("/<h3>(.*?)</h3>/", "<h3>new Price 20.36 (the old:
    $1)</h3>", $str);

echo htmlentities($result);
?>
```

This would then print "We want to change amount \$<bla>new Price 20.36 (the old: 25.09)</bla> of the item". \$2 is for the second "catch-all", \$3 for the third, etc.

Use of php functions 'ereg','eregi','ereg_replace'

```
<?php
// Example reg005.php
// Simple Regular Expressions in PHP
// Copyright http://curl.phptrack.com
// Use of php functions 'ereg','eregi','ereg_replace'
?>
<!DOCTYPE HTML PUBLIC "-//
//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.d
td">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Regular expressions</title>
</head>

<body>

    <?php
        $search = "Time of your PC";
        print( "Testing string is: '$search'<br /><br />" );

        // call function ereg to search for pattern 'Now'
        // in variable search
        if ( ereg( "Now", $search ) )
            print( "String 'Now' was found.<br />" );

        // search for pattern 'Now' in the beginning of
```

```
// the string
if ( ereg( "^Now", $search ) )
    print( "String 'Now' found at beginning
           of the line.<br />" );

// search for pattern 'Now' at the end of the string
if ( ereg( "Now$", $search ) )
    print( "String 'Now' was found at the end
           of the line.<br />" );

// search for any word ending in 'ow'
if ( ereg( "[[:<:]]([a-zA-Z]*ow)[[:>:]]", $search,
          $match ) )
    print( "Word found ending in 'ow': " .
           $match[ 1 ] . "<br />" );

// search for any words beginning with 't'
print( "Words beginning with 't' found: " );

while ( eregi( "[[:<:]](t[[:alpha:]]+)[[:>:]]",
              $search, $match ) ) {
    print( $match[ 1 ] . " " );

    // remove the first occurrence of a word beginning
    // with 't' to find other instances in the string
    $search = ereg_replace( $match[ 1 ], "", $search );
}

print( "<br />" );
?>
</body>
</html>
```

Output

Test string is: 'Now is the time'

String 'Now' was found.

String 'Now' found at beginning of the line.

Word found ending in 'ow': Now

Words beginning with 't' found: the time

Parsing Form for hidden fields.

```
<?php
// Example reg006.php
// Simple Regular Expressions in PHP
// Copyright http://curl.phptrack.com
// This example will match a string with
// some text and ending with some text
```



```
//starting from = <input type="hidden" name="__VIEWSTATE" value="
// ending with = "

$str = '
    <html>
    goes some txt etc <b>some thsm tags</b>etc
    and here is the price $25.66 of tie tem
    <input type="hidden" name="__VIEWSTATE" value="55555">
    goes some txt etc <b>some thsm tags</b>etc
    </html>
';

preg_match_all("/input type=\"hidden\" name=\"__VIEWSTATE\" value=\".*?\"/i", $str, $out);
//print_r($out);

$viewstate=str_replace("input type=\"hidden\" name=\"__VIEWSTATE\" value=\"", "", $out[0][0]);
$viewstate=trim($viewstate, ' ');
print $viewstate;
?>
```

Parsing Images links from HTML page.

```
<?php
// Example reg007.php
// Simple Regular Expressions in PHP
// Copyright http://curl.phptrack.com
//Below Code also give me the Pretty good output for
// parsing the images links from html strings.

$str = '<P>Maximum runnSpeed: 2000 1/6 </P>
    <br clear="all">
    <a name="image_1"></a>
    
    <p>
    <p>
    <br clear="all">
    
    </div>
    <p>
    </body></html> ';

preg_match_all('/"; print_r ($matches[0]); echo "</pre>";

$str = '<P>Maximum runnSpeed: 2000 1/6 </P>
    <br clear="all">
    <a name="image_1"></a>
    
    <p>
    <p>
    <br clear="all">
    
    </div>
    <p>
    </body></html> ';
```

```
preg_match_all("#<img.*?src=\"?([^\s>\"]+)[^>]+?>#is", $str, $matches);
echo "<pre>"; print_r ($matches); echo "</pre>";

/*
=====
OUTPUT
=====
<pre>Array
(
  [0] => Array
  (
    [0] => 
    [1] => 
  )

  [1] => Array
  (
    [0] => /i/lkee_FR._ANGLE.jpg
    [1] => /i/50__TOP__BESTBUY.jpg
  )

)
</pre>
*/
?>
```

Further Help:

Please visit the site to get more grip in regular expressions.

<http://www.php.net/manual/en/function.preg-match.php>

Appendix A

Installing the PHP binding

You might also find it helpful to install Apache + PHP + CURL + SSL using the Apache Toolbox.

Original instruction was contributed by Lim Chung Keng. Kudos!

My purpose is to have : PHP code with curl functions (using curl_init etc.) calling HTTPS server.

Build OpenSSL (In openssl-0.9.x directory)

1. ./config
2. make
3. make test
4. make install

Build CURL (In curl-7.x.x directory)

1. ./configure --disable-ipv6 --with-ssl=/usr/local/ssl (if only ssl was built here)
2. make
3. make install (libcurl.so is now in /usr/local/lib)

Build PHP (In php-4.0.xpl1 directory)

1. To make -lssl comes before -lcrypto (as advised), search for "-lcrypto" in ./configure and put "-lssl" before it (*).

2. ./configure --with-mysql --with-openssl=/usr/local/ssl --with-curl=/usr/local/lib --with-apache=../apache_1.3.x

(In most cases of linking problem, it should complain here in debug.log)

3. make
4. make install (If --with-apache option is omitted, it should generate php executable in /usr/local/bin)

5. Library for modphp4 is now in apache_1.3.x/src/modules/php4

Build and install SSL-aware Apache (In mod_ssl-2.8.xxxx directory)

1. ./configure --with-apache=../apache1.3.x --with-ssl=/usr/local/ssl --prefix=/usr/local/apache --activate-module=src/modules/php4/libphp4.a

2. Follow remaining steps the output tells.

3. Restart your apache... BINGO!

(*) = You can optionally do the replacing with this set of commands:

1. sed 's/-lcrypto/-lssl -lcrypto/g' configure > c2
2. rm -f configure
3. cp c2 configure
4. chmod 755 configure

Installing PHP/CURL on IIS

Most of this description was provided by David Withnall.

to test your installation. in the php install directory run the following command:

php.exe -i

the output of this command should be the same as the phpinfo() method - which is a big HTML listing with all the current settings php is running on.

if you get any warning messages check the following:

1. The extensions directory has not been set correctly in the php.ini file to fix it - make sure there is the following line in your php.ini file extension_dir="c:\php\extensions\" (or the relevant directory string) (I put the \ on the end because php automatically puts in a / if its not there. I'm not sure if it has any effect tho. but it did change the warning messages I was getting.)
2. Make sure the php_curl.dll file is in that directory.
3. Also make sure that the files necessary for curl to run are in the system[32] directory [32] for non 9x machines. libeay32.dll ssleay32.dll

4. make sure that the version of php_curl.dll, php.exe & php4ts.dll is the same (I found this out by running the installer. and then copying the php_curl.dll file from the zip package) to fix this replace all the files in the php install directory with those from the zip package. (theres only 3 or so) (stolen directly from the php install.txt)

Windows and PWS/IIS 3 - including PWS on Win 9x/ME

The recommended method for configuring these servers is to use the REG file incuded with the distribution (pws-php4cgi.reg). You may want to edit this file and make sure the extensions and PHP install directories match your configuration - once you have done this, just double click on the file and it will update your registry. Alternatively, you can follow the steps below to do it manually.

WARNING:

These steps involve working directly with the Windows registry. One error here can leave your system in an unstable state. We highly recommend that you back up your registry first. The PHP Development team will not be held responsible if you damage your registry.

Run Regedit.

Navigate to:

HKEY_LOCAL_MACHINE/System/CurrentControlSet/Services/W3Svc/Parameters/ScriptMap

On the edit menu select: New->String Value. Type in the extension you wish to use for your php scripts. ex: .php Double click on the new string value and enter the path to php.exe in the value data field. ex: c:\php\php.exe Repeat these steps for each extension you wish to associate with PHP scripts.

Windows NT/2000 and IIS 4 or newer and PWS 4 on NT Workstation or W2K non server editions

To install PHP on an NT/2000 Server running IIS 4 or newer, follow these instructions. You have two options to set up PHP, using the CGI binary (php.exe) or with the ISAPI module.

In either case, you need to start the Microsoft Management Console (may appear as 'Internet Services Manager', either in your Windows NT 4.0 Option Pack branch or the Control Panel=>Administrative Tools under Windows 2000). Then right click on your Web server node (this will most probably appear as 'Default Web Server'), and select 'Properties'.

If you want to use the CGI binary, do the following: Under 'Home Directory', 'Virtual Directory', or 'Directory', click on the 'Configuration' button, and then enter the App Mappings tab.

Click Add, and in the Executable box, type: c:\php\php.exe (assuming that you have unzipped PHP in c:\php\).

In the Extension box, type the file name extension you want associated with PHP scripts. Leave 'Method exclusions' blank, and check the Script engine checkbox. You may also like to check the 'check that file exists' box - for a small performance penalty, IIS (or PWS) will check that the script file exists and sort out authentication before firing up php. This means that you will get sensible 404 style error messages instead of cgi errors complaining that php did not output any data.

You must repeat from 'Click Add...' for each extension you want associated with PHP scripts. (.php is recommended. although .phtml and .php3 may be required for legacy applications.)

Set up the appropriate security. (This is done in Internet Service Manager), and if your NT Server uses NTFS file system, add execute rights for I_USR_ to the directory that contains php.exe

To use the ISAPI module do the following

If you don't want to perform HTTP Authentication using PHP, you can (and should) skip this step.

Under ISAPI Filters, add a new ISAPI filter. Use PHP as the filter name, and supply a path to the php4isapi.dll.

Under 'Home Directory', click on the 'Configuration' button. Add a new entry to the Application Mappings. Use the path to the php4isapi.dll as the Executable, supply .php as the extension, leave Method exclusions blank, and check the Script engine checkbox.

Stop IIS completely (net stop iisadmin)

Start IIS again (net start w3svc)

Installing PHP/CURL on IIS

Apache Toolbox provides a means to easily compile Apache with SSL, PHP (v4 or v3), MySQL, APC (Alternative PHP Cache), mod_auth_nds, mod_dynvhost, WebDAV, mod_fastcgi, mod_gzip, mod_layout, mod_throttle, mod_accessref, mod_auth_sys, mod_bandwidth, mod_auth_ldap, mod_roaming, mod_perl, Jakarta, OpenLDAP, mod_dtcl, mod_python, mod_frontpage, mod_relocate, mod_backhand, mod_trigger, mod_watch, mod_filter, libcurl for PHP, MING for PHP, mod_auth_mysql, mod_auth_samba, mod_index_rss, mod_random [by Brian Aker], mod_allowdev, mod_auth_cookie, mod_auth_cookie_file, mod_auth_external, mod_auth_inst, mod_auth_system, mod_eaccess, mod_bandwidth, mod_cgisock, mod_urlcount, mod_disallow_id, mod_peekhole, mod_put, mod_qs2ssi, mod_session, mod_cvs, mod_macro, mod_random, mod_ip_forwarding, mod_ticket, mod_monitor, jakarta-tomcat, jakarta-ant, jakarta-servletapi, Java Development Kit, Java API for XML Parsing, Java Secure Sockets Extension, mod_pcgi2, apache IPv6 patch, and the latest gd libraries with PNG/JPEG/Freetype2/Zlib support.

You can also hack apache's internal settings so it will report a different version of httpd all together, improve your security.

Includes support for mod_access, mod_asis, mod_auth_db, mod_autoindex, mod_digest, mod_example, mod_imap, mod_log_agent, mod_mime, mod_negotiation, mod_setenvif, mod_status, mod_usertrack, mod_actions, mod_auth, mod_auth_dbm, mod_cern_meta, mod_dir, mod_expires, mod_include, mod_log_config, mod_mime_magic, mod_proxy, mod_unique_id, mod_vhost_alias, mod_alias, mod_auth_anon, mod_auth_digest, mod_cgi, mod_env, mod_headers, mod_info, mod_log_referer, mod_mmap_static, mod_rewrite, and mod_speling, which come default with apache.

It is fully customizable and menu driven. Everything is compiled from source. It checks for RPMs that might cause problems and uses wget to automatically download the source if it's missing.

That's 63 3rd party modules plus the 36 modules that come with apache.

Appendix B

Installing cURL

From PHP version 4.2.3 on, you need a cURL version of at least 7.9.0. From PHP version 4.3.0 on, you need a cURL version of at least 7.9.8.

Windows:

As with any PHP extension in Windows, you will need the PHP distribution that includes external extensions. Once PHP is installed, you will need to copy the files `php4ts.dll`, `ssleay32.dll`, `php_curl.dll`, `msvcrt.dll` from the 'DLLs' folder to your Windows PATH, i.e.:

```
c:\windows\system for Windows 9x/Me  
c:\winnt\system32 for Windows NT/2000  
c:\windows\system32 for Windows XP
```

cURL can then be enabled by uncommenting the line `'extension=php_curl.dll'` in the `php.ini` file. Alternatively you can load the module dynamically in your script using:

```
<?php  
dl("php_curl.dll");  
?>
```

UNIX:

Your local mirror for downloading cURL can be found at <http://curl.haxx.se/>. Precompiled binaries are also available for a wide range of operating systems.

Because cURL relies on the `openssl` library for SSL connections, `openssl` must be installed first. If `openssl` is not installed, SSL support will be omitted from the cURL build. After installing cURL (`./configure`, `make`, `make install`), PHP must be recompiled to include cURL support (`--with-curl`).

If cURL support is enabled, the `phpinfo()` function will display it in its output.

References

We have used the following sites to complete this book:

<http://www.openssl.org> is the home of the OpenSSL project.

<http://curl.haxx.se> is the home of the cURL project.

<http://www.php.net> is the home of the PHP project.

<http://livehttpheaders.mozdev.org> is the home of the livehttpheaders.

<http://www.ethereal.com> is the home of the ethereal.

We are grateful to the creators of the respective sites for providing such a beautiful help.