PayPal Password Management Installation Guide

For Professional Use Only
Currently only available in English.

A usage Professional Uniquement
Disponible en Anglais uniquement pour l'instant.

Last Updated: April 2002

PayPal Password Management Installation Guide

Notice of Non-Liability

PayPal, Inc. and the authors assume no liability for errors or omissions, or for damages, resulting from the use of this Manual or the information contained in this Manual.

Contents

List of Tables

Preface

This Document

This is the original release for *Password Management Installation Guide,* a document that describes installing and configuring the PayPal password management script for use by merchants who use PayPal's subscription management.

Intended Audience

This document is written for users of PayPal subscription management feature.

Organization of This Document

Chapter 1, "Introduction," gives some background information about this product.

Chapter 2, "Apache Basic Authentication," gives some background information about Apache web server basic authentication and how to configure it.

Chapter 3, "Configuring the PayPal Password Management Script," details how to configure PayPal's Perl script for password management.

Chapter 4, "Error Handling and Reporting ," details errors that might occur, their potential causes, and possible solutions for them.

Notational Conventions

This document uses typefaces to identify the characteristics of text. These typefaces and the characteristics they imply are described below:

Typeface	How Used
serif italics	A document title.
	A term being discussed or defined. For example: A *file* is a readable or writable stream of characters …
	Boolean values (not keywords). For example: The function returns *true* if it encounters an error.
`monospaced`	Pathnames or file names that appear in body text frames.
	Code-related names that appear in body text frames. Such names are used for functions, callbacks, arguments, data structures, and fields. For example: `AbstractResponseType` is the SOAP response type definition on which all PayPal API response methods are based.
	Components of Internet protocol requests and responses, such as HTTPS and `FORM` variables. For example: The PayPal system uses a `method=POST` request to return IPN status variables related to subscriptions, such as `txn_type`.

Typeface	How Used
Serif bold	User interface names, such as window names or menu selections. For example: On the **Profile** page, click **Email** to confirm your email address.
San-serif oblique	Placeholders used in the context of a format or programming standard or formal descriptions of PayPal system syntax. Placeholders indicate values or names that the reader should provide. Example: For example, `amount` is the variable for a single-item shopping cart, but `amount_`*X* is the name of the variable for a multi-item shopping cart. `amount_3` is the item amount for the third item in a multiple-item shopping cart.

To convey additional information, this document may also apply color and underlining to words or phrases that use the typefaces described above. Such use is described below:

Text attribute	How Used
xxxxxx	Hypertext link to a page in the current document or to another document in the set.
xxxxxx	Hypertext link to a URL or that initiates a web action, such as sending mail.

Documentation Problems

If you discover any errors in or have any problems with this documentation, please email us by following the instructions below. Describe the error or problem as completely as possible and give us the document title, the date of the document (located at the foot of every page), and the page number or page range.

To contact Developer Technical Support about documentation problems:

1. Log in to your account at https://developer.paypal.com/ by entering your email address and password in the **Member Log In** box

2. Click **Help Center** at the bottom of the box on the right side of the page.

3. Click **Email PayPal Technical Support**.

4. Complete the form.

1 Introduction

Subscription services often give their subscribers unique usernames and passwords in order that they may access "member only" sections of their sites. These merchants have the option to have PayPal generate unique usernames and passwords for their subscribers.

If a merchant chooses this option when creating a subscription button, a username and password pair generated by PayPal will be presented to their subscribers on the PayPal completion page after they successfully sign-up.

A merchant must then activate the username and password pair on his server so that the pair is valid. To do this, merchants must integrate the instant server communications PayPal will send them.

Merchants can either write their own code on their server to integrate our real-time communications.

Or, merchants who are password protecting their site through Basic Authentication (using an Apache web server on Linux) can download from PayPal a Perl module developed by PayPal that is capable of providing complete subscription management. The Perl script is easy to install and interacts with PayPal's Instant Payment Notification interface to automatically activate and deactivate subscription accounts.

This document details how to integrate your server with this Perl script.

System Requirements

Password Management is available for merchants running an Apache web server on Linux and securing access to a portion of their site, such as content, using "Basic Authentication." Although not tested, this script should work across most Unix installations. The Perl script, as is, does not work on NT.

2 Apache Basic Authentication

Basic Authentication is a way to password-protect your web content. Access is granted only when a valid username and password is presented to your webserver.

Apache provides a simple way to use Basic Authentication. By creating an `.htaccess` file you can protect an entire directory of content. The `.htaccess` file tells your web server where to find a list of authorized usernames and passwords: a file called `.htpasswd`.

Setting Up Basic Authentication with Apache

If you need to set up Basic Authentication, follow these steps:

1. Identify the directory you want to password protect.
2. Create an `.htaccess` file. To create a password protected or "members only" section of your website you need to create a `.htaccess` file. Change directories to the directory you want to protect, for example:

   ```
   cd /var/www/html/members
   ```

   Create a new text file using any text editor; for example:

   ```
   pico .htaccess
   ```

3. Add the following lines to the top of the file.

   ```
   AuthName "Your Website Name"
   AuthType Basic
   AuthUserFile /var/www/files/.htpasswd
   AuthGroupFile /dev/null

   require valid-user
   ```

4. Change "*Your Website Name*" to the name of your website. You'll also want to change the path specified by the line `AuthUserFile` to any location that is not in your document root (the directory where HTML pages reside).
5. Create the `.htpasswd` file. The `AuthUserFile` specifies the file that maintains valid username and password combinations to the protected area of your website. Create this file using the following command (substitute `/var/www/files/.htpasswd` with your own path)

   ```
   touch /var/www/files/.htpasswd
   ```

Because this file needs to be accessed and written to by the Password Management script, the owner must be the Apache user that the script will run as. For example if your Apache username is apache:

```
   chown /var/www/files/.htpassword apache
```

Apache Configuration

You need to verify that Apache's configuration is ready to support basic authentication. Look in your Apache configuration file, `httpd.conf`, for the following (where `/var/www/html` is your own document root):

```
<Directory "/var/www/html">
```

A few lines below this entry, find the line:

```
AllowOverride None
```

Change this to:

```
AllowOverride AuthConfig
```

After you have saved your changes you need to restart Apache for the new configuration to take effect.

3 Configuring the PayPal Password Management Script

First, install the Perl script. This is a four-step process.

1. Download the package and copy it to a temporary directory on your web server.
2. Uncompress the file with the following command:

```
gzip -dc filename.tar.gz | tar -x
```

The temporary directory should now contain the following files:

```
paypal.pl
README
```

3. Next you need to copy the paypal.pl script to your web server's CGI script directory. For example if your directory is "`/var/www/cgi-bin`" then you would run the following command:

```
cp paypal.pl /var/www/cgi-bin/.
```

4. For the web server to run the Password Management script you'll need to change its owner. The script should be owned by the user Apache runs as. To determine what user Apache runs under either ask your system administrator or look in your Apache configuration file (`httpd.conf`). To change ownership of this file use chown as shown in the following example:

```
chown /var/www/cgi-bin/paypal.pl apache
```

The Perl script is now successfully installed.

Configuring the PayPal Password Management Script

Once the Perl script is installed and Basic Authentication is set up, you will need to configure your server.

Using a text editor you need to fill in a few configuration options in `paypal.pl`.

1. Open the file (it should be in your CGI script directory; this example assumes `/var/www/cgi-bin`):

```
pico /var/www/cgi-bin/paypal.pl
```

2. Near the top of the file are the following lines:

```
# User Constants - customize these to YOUR installation
my $PASSWORD_FILE = '/var/www/files/.htpassword';
my $TRANSACTION_FILE = '/var/www/files/processed_txns';
my $PAYPAL_URL = 'http://www.paypal.com/cgi-bin/webscr';
my $PERIOD1 = '';
my $PERIOD2 = '';
my $PERIOD3 = '';
my $AMOUNT1 = '';
my $AMOUNT2 = '';
my $AMOUNT3 = '';
my $SMTP_SERVER = '192.168.1.10';
my $ADMIN_EMAIL = 'admin@abc123.com';
my @PAYMENT_EMAILS = ('s@a123.com');
```

3. Change the path of `$PASSWORD_FILE` to that of your `.htpassword` file which you created above.
4. You need to specify a path and filename for `$TRANSACTION_FILE`. This file is used to track the transactions the Password Management script has processed. Create this file using the touch command:

   ```
   touch /var/www/files/processed_txns
   ```

5. Then change the owner so the script will be able to write to it (replace our example Apache username with your actual username):

   ```
   chown /var/www/files/processed_txns apache
   ```

6. If you plan on using SSL to secure communications between PayPal and your website then you should change `$PAYPAL_URL` to:

   ```
   my $PAYPAL_URL = 'https://www.paypal.com/cgi-bin/webscr';
   ```

However, be sure to follow the later directions on downloading and installing the SSL modules. Otherwise leave this as is.

7. To ensure that a new subscription sign-up meets your expected terms and conditions you need to fill in each pay period and amount. For example, if you created a subscription button and specified that the first month was $20.00, the second month was $10 and then $5 for each month thereafter the period and amount constants would look like:

   ```
   my $PERIOD1 = '1 M';
   my $PERIOD2 = '1 M';
   my $PERIOD3 = '1 M';
   my $AMOUNT1 = '20.00';
   my $AMOUNT2 = '10.00';
   my $AMOUNT3 = '5.00';
   ```

Each period is expressed as a number followed by a single capital character. Valid characters are D (for day), W (for week), M (for month), Y (for year). Amounts are specified in dollars and correspond with their respective period (period1 goes with amount1, etc.).

8. The PayPal Password Management script features email alerts to notify the website operator of any unexpected errors. Enabling this feature requires you to modify the following constants:

```
my $SENDMAIL_PATH = '/usr/sbin/sendmail';
my $ADMIN_EMAIL = 'admin@abc123.com';
```

The `$SENDMAIL_PATH` should be filled in with the location of sendmail. Your ISP or system administrator should be able to provide this information. `$ADMIN_EMAIL` can be any valid email address. All email notifications are sent to this address.

If you aren't interested in receiving error notification emails you can disable this feature by leaving the `$ADMIN_EMAIL` and `$SENDMAIL_PATH` fields empty.

9. You will need to enter the primary email address of your PayPal account. This is important to validate the notifications sent to you.

The last user constant: `@PAYMENT_EMAILS` is your primary PayPal email address.

```
my @PAYMENT_EMAILS = ('payme@paypal-test-adr.com');
```

NOTE: If you change your primary email address in your PayPal account you will have to change this variable.

Updating your PayPal Account Preferences

Your website is now ready for PayPal Password Management. All that remains is to activate your account and update your subscription signup button.

To activate Password Management:

1. Log in to PayPal with your Business or Premier account. Next click on the "Profile" tab. Click on the Preferences link. And finally click on Instant Payment Notification (IPN) Preferences.

2. Select the checkbox to activate IPN and enter the URL of the paypal.pl script you installed above. For example:

```
http://www.paypal-testing.com/cgi-bin/paypal.pl
```

NOTE: If you intend to secure communications between PayPal and your website make sure the URL you specify is `https://` (this requires the installation of additional modules detailed later).

3. Click on "Save Changes" to update PayPal with your changes.

Now that IPN is enabled all that remains is updating your Subscription buttons on your website.

4. Click on the Sell tab and then the Subscriptions link. Fill in all the applicable information. Be sure to check the box labeled "Advanced Feature: Check here if you would like PayPal

to generate usernames and passwords for your subscribers." The new button works just as before, just cut and paste the HTML.

You have completed the basic installation of PayPal User Password Management.

PayPal Password Management Encryption

At this point communication between your website and PayPal is unencrypted. Adding encryption is recommended but not a necessity.

If you would like to encrypt comunications, follow these steps.

NOTE: The following assumes that your Apache webserver already has support for SSL installed and can therefore accept and respond to https requests.

1. First, make sure that you have completed the two necessary configuration steps (highlighted previously). Specifically, be sure that the URL you entered in PayPal's IPN Preferences page is **`https://`** and not `http://`. Also doublecheck that the following user constant in `paypal.pl` reads:

   ```
   my $PAYPAL_URL = 'https://www.paypal.com/cgi-bin/webscr';
   ```

2. Encryption requires the installation of two free software components. Begin by downloading and uncompressing the latest OpenSSL release at

   ```
   http://www.openssl.org
   ```

3. Uncompress the package with:

   ```
   gzip -dc filename.tar.gz | tar -x
   ```

4. Change to the directory that was created and run the following commands (you must be logged in as root):

   ```
   ./config
   make
   make test
   make install
   ```

5. Next, download the latest version of the Perl module: Crypt::SSLeay available at

   ```
   http://www.cpan.org
   ```

6. Uncompress it:

   ```
   gzip -dc filename.tar.gz | tar -x
   ```

7. Change to the directory that was created and install the module with the following commands (you must be logged in as root):

   ```
   perl Makefile.PL
   make
   ```

```
make test
make install
```

8. When you connect to PayPal's secure server (https://www.paypal.com) it will present you with a digital certificate, signed by a trusted Certificate Authority (Verisign). To be 100% certain of the authenticity of this transaction your code should verify PayPal's certificate using Verisign's public key.

If you would like more information about the verification process, it can be found at http://www.verisign.com/

You can also review this useful FAQ on digital certificates:
http://digitalid.verisign.com/server/about/aboutFAQ.htm

4 Error Handling and Reporting

The PayPal Password Management Script has two mechanisms for reporting errors, the Apache error log (error_log) and notification emails. All exceptions are logged to the error log, however, email notifications are optional. Errors are reported identically in both.

Below is a list of 8 possible error messages, examples on how they may occur, and potential solutions.

TABLE 4.1 Errors, Possible Causes, Potential Solutions

Error	Possible Cause	Potential Solution
Notification received was NOT from PayPal - Message was ignored.	An http or https post was made to your paypal.pl script, however the contents of the message was determined by PayPal to be invalid.	Because the message was determined to be inauthentic (not from PayPal) it was ignored, no action is required.
Notification could not be acknowledged due to a network or PayPal issue. PayPal will retry until it succeeds.	The script was unable to communicate with PayPal's website.	Check that your webserver can reach other websites outside your network. Check that the user constant in paypal.pl: `my $PAYPAL_URL` is a valid, working URL. Check that you can browse to PayPal's website, not other action is required.
Error Message: This customer did not sign-up according to your payment terms. Although payment was accepted the account was not activated.	The user tampered with your PayPal subscription button or you did not properly specify some of the following user constants: `my$PERIOD1, my $PERIOD2, my $PERIOD3, my $AMOUNT1, my $AMOUNT2, my$AMOUNT3.`	Log into your PayPal account and verify that the user's subscription does not match what you expected. If the subscription does match, then check the above user constants and reconfigure as necessary (see configuration section). You may wish to refund the users money or manually add them to your .htpassword file (see manual maintenance below).

TABLE 4.1 Errors, Possible Causes, Potential Solutions

Error	Possible Cause	Potential Solution
An IPN was received that did not match your primary email address - Message was ignored.	A valid notification was received however the recipient email address for the subscription did not match the one specified in `my@PAYMENT_EMAILS` in paypal.pl. Either a malicious person is trying to gain access or `my @PAYMENT_EMAILS` needs to be updated. Solutions: Log into your PayPal account and verify that the subscription exists in your account. If it does, you need to update `my @PAYMENT_EMAILS`with your primary email address in paypal.pl. Additionally you'll need to add this user manually (he will be removed automatically), see manual maintenance for instructions. Otherwise, there is no action necessary.	
An IPN was received that was already processed - Message was ignored.	A malicious user has attempted to fake a legit transaction by capturing and resending a notification after it has already been processed.	Because the paypal.pl script records the transactions that have been processed the attempt was thwarted, nothing to be done.
Unable to access: /path/filename - Some system error message	The script could not access the specified file.	Check the file's permissions and owner. Also make sure the file's path is correct. Password Management will not work until this issue is resolved.
Unable to close: /path/filename - Some system error message	This is unexpected.	Nothing to be done. May require investigation if this error persists.

TABLE 4.1 Errors, Possible Causes, Potential Solutions

Error	Possible Cause	Potential Solution
Username: pp-somename already exists	This is unexpected. PayPal checks each name against previously used names to be certain it has not been used on your site. If your `.htpassword` file is being used by other credit card processors it is theoretically possible a duplicate can occur however this is extremely unlikely. To reduce even further the remote possibility of duplicates we prepend our random usernames with pp-.	This will require manual intervention (see below). You cannot have two users of the same name. The user was not added.

Manual Modifications to .htpasswd

The file that tracks usernames and password can be manually altered to fix exceptional problems that can come up.

The authors of Apache have included in its distribution an executable called `htpasswd` for this task.

Full documentation can be obtained at any UNIX prompt by typing:

```
man htpasswd
```

To add a new user named "admin" with a password of "mygr8tpass" where the password file is located at `/var/www/files/.htpasswd`:

```
htpasswd -b -d /var/www/files/.htpassword admin mygr8tpass
```

Removing a user does not require a special tool. Simply find the line in the `.htpassword` file with the username you'd like to remove and delete the entire line. Example:

```
pp-pianotuple:33z.TrcL5d6Rs
pp-theshade:9Wzsv2TBDkGnw
```

To deactivate the user "`pp-theshade`" remove the line so the file reads:

```
pp-pianotuple:33z.TrcL5d6Rs
```

Tips

If you manually add a username to rectify an error, the username must be identical to the one generated by PayPal. This enables PayPal to deactivate this account when the subscription is canceled or expires.

NOTE: Usernames and passwords for all your subscribers are found in each of your subscriber's subscription creation details within your history log.

If you want to deactivate a username and password, simply canceling a users subscription will not necessarily deactivate the pair. If the cancellation was due to a payment failure, usernames and passwords are immediately canceled when the subscription is canceled. If the cancellation was made by the buyer or seller, then the usernames and passwords will be active until the subscription terms expire (the date of the next scheduled payment). If you wish to deactivate these usernames and passwords when you cancel a subscriber, you will need to manually modify the `.htpassword` file.

Recreating .htpasswd

You can recreate your `.htpasswd` file using the downloadable list of your active subscribers passwords. Here's how:

Obtain the list of all subscription-related (non-payment) entries

1. Log in to your PayPal account
2. Go to the **History** subtab of the **My Account** tab
3. Choose the **Download Log** link
4. Choose the **All transactions** radio button and the **Subscriptions** radio button (which will only appear after someone has subscribed to your service)
5. Click **Download History**.

This file will contain all of your non-payment subscription-related entries (Sign-ups, Cancellations, and Completions).

To get your Subscription Creations , filter out the Subscription Cancellation and Subscription Completion Transactions. This is your list of subscribers.

To view all subscribers with valid passwords, you can then sort by all subscriptions with the status "active", "active-completed", or "activecancelled" (thereby filtering out the inactive-cancelled or inactive-completed).

The passwords of these subscribers make up the `.htpasswd` file.