

Sonic PayPal Applications

Website Payments Pro – MicroCart

Introduction

Sonic PayPal Applications are developed in PHP so that you can install them on any hosting service that supports PHP v5.x or greater. (See “Requirements”). The application is embedded in a series of php files that is wrapped with an HTML template for demonstration purposes. You are welcome to use that template or use your own (See “Customization of User Interface). Applications come in two forms: with source code and without source code. Note that there are NO differences between these two forms in the instructions presented in this document. This manual deals with the major configuration and setup issues that will be needed with both versions. The source code version comes with extensive comments within the source code.

Table of Contents

Introduction.....	1
Skill Level Requirements.....	2
System Requirements.....	2
SSL Certificate.....	2
Installation.....	2
General System Configuration.....	3
General Settings.....	3
Debug.....	4
Company.....	4
Database Settings.....	4
Paths and URLs.....	5
Email Settings and Defaults.....	5
Tax and Shipping Settings.....	6
Cart Status and Locale.....	7
Newsletter Settings (optional).....	7
Other Defaults.....	8
Products Configuration.....	8
Field Naming Convention.....	9
PayPal Configuration.....	9
PayPal Configuration Instructions.....	10
Other Settings.....	11
Customization of User Interface.....	12

Skill Level Requirements

Installing and configuring this application requires understanding and with minimal experience in, the following areas:

- Basic PHP coding and syntax
- Hosting configuration
- SSL configuration
- Database configuration and setup.

We have made considerable effort in this installation and configuration manual to help you work through a number of these details. We have chosen to “keep it simple” and therefore minimize the cost of these applications such that they can be affordable for those that have the needed skills or have access to someone that does have these skills. SonicWebTech has installation services and we would be happy to provide complete installation and configuration for these applications.

System Requirements

Sonic PayPal Applications have the following general requirements:

- PHP 5.x or greater with MySQL extension.
- SendMail for outgoing email or equivalent configuration.
- MySQL 5.x or greater.
- SSL Certificate

If you have purchased the source code you can alter a number of these requirements to your needs.

SSL Certificate

This application requires that an SSL certificate be installed on your target domain. The details of this process are beyond the scope of this installation manual. It will be assumed in the following installation instructions that you have a folder location on your hosting site that is configured for SSL. Contact your hosting provider for further details.

Installation

Installation involves three areas: creating the database, copying the application files to your server, and configuring the system. Following is the basic procedure:

1. Unzip the downloaded file onto your computer.
2. Log into your website administration console and open your database administration tool (typically phpMyAdmin this will be assumed in these directions)
 1. Create a database.
 2. Import the sql script “wppro_microcart.sql” found in the folder “database” in the unzipped

- download.
3. Setup the SSL Certificate (per your hosting providers instructions)
 4. Copy all of the remaining files to the folder that is configured for SSL protection. It is important that you maintain the exact same folder structure as produced by unzipping the downloaded file.

At this point you should be able to browse to the installation folder using the following URL template:

<http://host.domain.com/installfolder/purchase.php>

You cannot proceed beyond this first page without doing the configuration. It is advisable that you review all of the configuration settings. Some will be labeled optional, but you should review them all so that you are aware of these settings for future reference.

General System Configuration

Open a copy of the configuration file “/includes/config_inc.php”. Review the following configuration options and determine which options you will be required to update.

General Settings

The following settings are discussed in the section:

```
// system debug setting
$g_isDebug = false;

$g_sCompanyName = "SonicWebTech";
$g_sCompanyPhone = "619.955.6380";
$g_sCompanyHours = "8:00 am to 4:00 pm PST, Monday through Friday";
```

Record Your Settings

Debug	
Name	
Phone	
Hours	

Debug

The debug setting triggers the output of various session and posting variables at the bottom of each webpage in the system. On the final payment page it will also alter the behavior of the “paypal_api_error.php” page in that it will provide more detailed error messages. These details are hidden when debug is set to “false”. It is set to “false” by default.

Company

Fill in the appropriate settings as they apply to your company. These setting will appear in various emails that the system produces, which are sent to your customers.

Database Settings

These settings should reflect how you configured your database on your hosting service system. If you are using phpMyAdmin, these setting will be displayed for you after you create the database. Make note of these settings and record them here. The server can take an IP address or a FQDN (Fully qualified Domain Name) that will resolve to an IP address via DNS.

```
$g_sServer      = "###.###.###.###";  
$g_sDbuser     = "loginname";  
$g_sDbpassword = "password";  
$g_sDatabase   = "wpro_microcart";
```

Keep in mind that passwords are case sensitive. The database name does not have to be the same as demonstrated, but must match the name you used when you created the database.

Record Your Settings

Server	
User	
Password	
Database Name	

Paths and URLs

These settings allow the system to adapt to the unique environment on your hosting servers. Notes on each follows:

```
// additional path - possibly blank
$g_sUrlpath = '/PayPal_WPPro_MicroCart';
// home page (optional)
$g_sUrlhome = 'purchase.php';
```

The Urlpath is often blank, but might be needed if you have the application installed in a “sub folder” on your system. (Note: Always pre-pend it with a forward slash as demonstrated, if not leave it blank) The Urlhome can be changed as needed, so that you can redirect where your customer goes when they click “home”. For example: you might want to send them to some other page on your website. It does not have to be “purchase.php”. That is only there for demo purposes.

Email Settings and Defaults

```
$g_sEmailSubject = 'SonicWebTech - Thank you for your Donation';
$g_sEmailFromName = 'SonicWebTech Staff';
$g_sEmailFromEmail = 'sales@sonicspider.com';
// comma separated list of emails
$g_sEmailCCList = 'jbm@sonicspider.com, 'tigger@microps.com';
```

These settings should be self explanatory. The CCList can be blank or contain as many emails as you require. They must be separated by a comma. Generally the CCList will point to members of your company that need to be alerted to the sell.

Record Your Settings

Subject	
From Name	
From Email	
CC List	

Tax and Shipping Settings

These settings control how the shipping and tax are handled. There is two basic modes of operation for each (Shipping and Tax).

1. Default global settings
2. By State settings

To a large degree these setting are mutually exclusive and are controlled by the following logical variables:

```
$g_doShipByState = true;  
$g_doTaxByState = true;
```

doShipByState = true will route to a shipping state selection page and use the array `g_arShipList`. If a state is zero (0) it will use the default value found in `g_dEachShip`. In this way you only have to list the "exceptions" in the array.

doShipByState = false will use `BaseShip` and `EachShip` and ignore the array

doTaxByState = true will also route user to the shipping state selection page and use the array `g_arTaxList`. Default value is zero (0).

doTaxByState = false will use the setting `g_dTaxRate` for all orders.

The default global settings are:

```
// used if doTaxByState is false  
$g_dTaxRate = 0;  
  
// used if doShipByState is false  
$g_dBaseShip = 1.0;  
$g_dEachShip = 6.0;
```

The "by State" array settings are controlled by two arrays: `$g_arShipList` and `$g_arTaxList`. Replace numbers for each state that you want a shipping value or a Tax rate (Example provided for California, CA)

```
// Shipping and Tax configurations  
// used if doShipByState is set to true  
$g_arShiplist = array(  
    'AL' => 0,  
    'AK' => 0,  
    'AZ' => 0,  
    'AR' => 0,  
    'CA' => 10.00,
```

```

... <snip> ...
    'VA' => 0,
    'WA' => 0,
    'WV' => 0,
    'WI' => 0,
    'WY' => 0);

// used if doTaxByState is true
$g_arTaxlist = array(
    'AL' => 0,
    'AK' => 0,
    'AZ' => 0,
    'AR' => 0,
    'CA' => 0.0725,
    ... <snip> ...
    'VA' => 0,
    'WA' => 0,
    'WV' => 0,
    'WI' => 0,
    'WY' => 0);

```

Cart Status and Locale

The cart status and Locale settings should not be altered for normal use. Their default values are:

```

// Cart Status
$g_sStatus = "new"; // new, order, purchase
$g_sLocale = "en_US";

```

Newsletter Settings (optional)

If you are interested in enrolling your customers in your newsletter, set this to “true”. Update the text accordingly. It is set to “true” by default so that its location is demonstrated. Set it to “false” to remove it from the webpage.

```

$g_doNewsletter = true;
$g_sNewsletterText = "Check if you would like to have us send you our
monthly newsletter";

```

Other Defaults

Miscellaneous defaults settings. Currency code should reflect the default currency that you have your PayPal account configured. Payment type for donations generally is immediate, which is the “Sale” setting. You might change this if you are promising to send some materials as part of their donation. In that case you would set it to “Authorization” and then do a “delayed capture” once the additional materials have been readied for shipping.

```
$g_sCurrencyCode = "USD";  
  
$g_sPaymentType = "Sale"; // Sale or Authorization
```

Products Configuration

All products are configured on the “purchase.php” page. There is a very specific naming convention that is required so that the rest of the system will pickup the values and pass them on to the database. Each product must have it's own version of this code block

```
<input type="text" name="1_prodqty" id="qty_01" size="10"  
class="center" value="<?php echo($_SESSION['qty_01']); ?>" />  
<input type="hidden" name="1_prodid" id="prodid_01" value="GB1234" />  
<input type="hidden" name="1_prodname" id="prodname_01" value="Holiday  
Gift Basket" />  
<input type="hidden" name="1_prodprice" id="prodprice_01" value="39.95"  
</input  
<textarea rows="5" name="1_instructions" cols="60"></textarea>
```

The <input> tags do not have to be in a table but each group of tags much be in a unique “form” tag. The example provided is named “productForm” but the name is completely optional. It is critical that each form has a unique name. The form might be named after the product as an example.

Field Naming Convention

There are five required fields with the name attribute: #_prodqty, #_prodid, #_prodname, #_prodprice, and #_instructions. Each group of five fields must be prepended with a number starting with one (1).

Example:

- Spring Basket
 - 1_prodqty
 - 1_prodid
 - 1_prodname
 - 1_prodprice
 - 1_instructions
- Summer Basket
 - 2_prodqty
 - 2_prodid
 - 2_prodname
 - 2_prodprice
 - 2_instructions

The “id” attribute must also be unique but does not require a number, You will use the “id” attribute for your JavaScript validation code as demonstrated for you in the “checkForm()” function.

PayPal Configuration

In addition to the general configuration, you will need to configure the system with your PayPal settings. It is assumed that you already have a Business or Premier account and are familiar with the PayPal system. PayPal has a large number of help documents to assist you. SonicWebTech has a number of consulting packages that you can purchase so that we can assist you in understanding your choices and how to configure and utilize your PayPal account. The file that controls these settings is: “/includes/pp_constants.php”.

There are two types of settings: Sandbox and Live. The application, as delivered, is configured for the Sandbox. This allows you to test your system and experience its operation exactly like your customers will experience it. It is advised that you setup Sandbox accounts and do this testing before going live. To apply for a Sandbox account go to:

<https://developer.paypal.com/>

.. and click “Sign Up Now”. After you sign up there will be a “Documentation” link. Click that link and download the Sandbox documentation.

API Credentials

To get the information required for the API credentials, login to your account and click “Profile”. Then under “Account Information” select “API Access”. PayPal does require your account to be “verified” and if that is NOT the case it will return some error messages to that effect. Look

under “Notifications” if there is a problem and follow those instructions.

Under “API Access” select “Option 2 – Request API credentials”. Then select “Request API Signature”. The information returned should look like the following:

```
API Username    tigger_api1.microps.com
API Password    8FARJ657QLMHAUEB
Signature       An5ns1Kso7MWUdW4ErQKJJJ4qi4-Asv.aGhBVJtbDXbJYI62718If90I
```

Record Your Settings

UserName	
Password	
Signature	

PayPal Configuration Instructions

Open the file “pp_constants.php” and using the above API credentials information, edit the following settings:

API User Name

```
define('API_USERNAME', 'tigger_api1.microps.com');
```

Input your user name, replacing the sample provided.

API Password

```
define('API_PASSWORD', '8FARJ657QLMHAUEB');
```

Input your password, replacing the sample provided.

Signature

```
define('API_SIGNATURE', 'An5ns1Kso7MWUdW4ErQKJJJ4qi4-Asv.aGhBVJtbDXbJYI62718If90I');
```

Input your signature (cut and paste is best to avoid typo errors), replacing the sample provided.

Other Settings

The Sonic PayPal Application comes configured for testing in the sandbox. If you are going to immediately go live you need to update the following settings:

API Endpoint

Live

```
=====
API Certificate      https://api.paypal.com/nvp
API Signature       https://api-3t.paypal.com/nvp
```

SandBox

```
=====
API Certificate      https://api.sandbox.paypal.com/nvp
API Signature       https://api-3t.sandbox.paypal.com/nvp
```

Update this when going live.

```
*/
```

```
define('API_ENDPOINT', 'https://api-3t.sandbox.paypal.com/nvp');
```

Update the API_ENDPOINT to reflect your requirements. This application uses the “API Signature” endpoints. (Note: There is some inconsistency in the terminology on PayPal. Some documentation refers to this setting as “API Endpoint” and other documentation refers to this setting as the “API Server”. They are one and the same.)

Proxy Settings

If you required proxy settings it is because you are hosting your application on a private hosting system (at home or within a company). You will require the assistance of an IT professional to insure these settings are correct. This is beyond the scope of this manual.

PayPal Express URL Settings

This is the URL that the buyer is first sent in order to authorize payment with their PayPal account. Change the URL depending on whether you are testing in the sandbox or going to the live PayPal site. The default setting is for the sandbox. Therefore, make sure you update this setting when going live.

For the sandbox, the URL is

```
https://www.sandbox.paypal.com/cgi-bin/webscr&cmd=_express-checkout&token=
```

For the live site, the URL is

```
https://www.paypal.com/webscr&cmd=_express-checkout&token=
```

```
define('PAYPAL_URL', 'https://www.sandbox.paypal.com/cgi-bin/webscr&cmd=_express-checkout&token=');
```

Customization of User Interface

The web pages that the user will interact with are fully customizable. These pages are mostly just HTML pages with some bits of PHP. If you are not familiar with PHP you should be careful to avoid altering those tags (<?php ?>). The customer interface pages are:

- purchase.php – Main product page. On this page you place all of your products. See “Products Configuration”
- shipchoice.php – Provides a list of states for the customer to select for shipping.
- purchase_problem.php – General problem page
- direct_checkout_process_form.php – Direct checkout form
- direct_checkout_process_payment.php – Direct checkout payment receipt
- express_checkout_process_details.php – Express checkout review
- express_checkout_process_payment.php – Express checkout receipt
- paypal_api_error.php – PayPal API error page

Each page is made up of four sections:

1. PHP setup (do not touch)
2. HTML header (fully customizable)
3. PayPal body (limited adjustment, ONLY by an experienced programmer)
4. HTML footer (fully customizable)

The **PHP setup** is demarcated by the tag:

```
<?php ?>
```

You should avoid editing anything within those tags as it could break the functionality of the application.

The **HTML header and footer** are “wrappers” around the **PayPal body** section. Both the header and footer are fully customizable.

The **PayPal body** section is demarcated by a comment tag as follows:

```
<!-- PayPal Content Block  
Copy this div block to our website template.  
Use paypal.css style sheet that controls all settings within this  
block  
-->
```

..and ends with the comment tag:

```
<!-- END - PayPal Content Block -->
```

Unless you have PHP experience you should not edit the code or tags found between these two blocks. After this comment tag, the remainder of the code is HTML and is the footer of the webpage. This is fully editable by you.

Generally, Dreamweaver will handle displaying the HTML in a way that will allow you to make edits. Some care should be taken because even Dreamweaver will, on occasion, add code that can alter the functionality of the **PayPal body** section.

There are two ways to customize your application:

1. Enhance the existing **HTML header and footer** blocks using CSS and minor adjustments to the HTML code.
2. Cut the **PHP setup** and **PayPal body** sections and insert these into templates of your current website. **CAUTION: It is important that you use the EXACT same file names.**

The application comes with three style sheets:

- basic.css – This style sheet controls the **HTML header and footer** sections.
- paypal.css – This style sheet controls the **PayPal body** section.
- cssreset.css – This style sheet controls the entire page and removes most browser specific styles so that the page displays the content in a more consistent manner. This style sheet is optional but, be forewarned, that the paypal.css style sheet will need some adjustments if you are not using the “reset” stylesheet.

If you choose to use option #1, then you will mostly edit the basic.css style sheet with only minor adjustments to the paypal.css. Make sure you are careful to keep them separate with regard to the sections that they control.

If you choose to use option #2, then you would link in the paypal.css style sheet and then make changes such that it then matches the styles in your main website template.

IMPORTANT: YOU MUST REMEMBER TO ALWAYS USE THE EXACT SAME FILE NAMES. THE SYSTEM IS HARD WIRED TO EXPECT THOSE FILES.