



Stellar Web Solutions eVendor Pro V1.2 (for PayPal)

Issued: 2007 10 03

Overview

Congratulations on the purchase of the Stellar Web Solutions eVendor Pro system. The Stellar Web Solutions eVendor Pro allows you to sell and deliver electronic goods such as eBooks, music, or software from your own website. The eVendor Pro is different from the eVendor Basic in that the eVendor Pro includes a shopping cart to support multiple items in the same transaction.

By using this software, you agree to the conditions of the attached licensing agreement. This software is licensed - not sold, please read the license agreement carefully. This software is protected by international copyright, it cannot be redistributed, resold, or transferred in whole or in part in any way.

Requirements

- This software is designed to work on both Linux/UNIX or Microsoft Windows.
- Web Server with a PHP 4 or higher.
- Database – MySQL 4 or higher is recommended.
See www.mysql.com for more information about MySQL.
- For Encrypted Website Payments (EWP) OpenSSL 0.93 or higher is required.

Package Manifest

The software distribution package contains the following files:

- readme.pdf – this file
- license.pdf – software license.
- install.php – Database installation script.
- ipn.php – PayPal IPN Script.
- stellar-settings.php – Configuration options.
- catalog.php – Sample product catalog.
- stellar-ipnmod.php – IPN code library.
- stellar-evendorpro.php – Sales and electronic delivery library.

- stellar-cartmod.php – Shopping cart script and configuration options.
- stellarcart.php – Sample shopping cart page.
- stellarshop.php – Sample shopping page with cart status code.
- return.php – Sample return/download page.
- index.php – Sample integration page – sample add to cart links.

Support

To obtain installation support, we recommend emailing us at stellar@stellarwebsolutions.com. Otherwise, visit www.stellarwebsolutions.com/en/contact.php for more ways to get support.

Installation Services

Contact Stellar Web Solutions to inquire about our low cost professional installation services available.

Features

Integrates PayPal processed transactions to securely deliver electronic products that can be downloaded by your customers. PayPal IPN data is authenticated, verified, and stored in a MySQL database. The Shopping cart stores items in PHP session variables, preventing the need to conduct database maintenance. Fully supports PayPal website payments standard (WPS) with optional encrypted website payments (EWP).

Secure Anti-Tampering System

- Transaction data is authenticated by retransmitting the received data back to PayPal to verify it's authenticity.
- Verification of the transaction being sent to your PayPal account and not spoofed.
- Transactions are only processed one time, duplicate retransmissions are ignored.
- Pending transactions will not allow a download until the funds are received.
- Access to disputed or refunded orders is instantly removed.
- Download user's IP address is recorded, optional email notification of each download.
- Purchaser's email address is used to access the download link to deter link sharing.

Secure Anti-eShoplifting System

- Optional encrypted websites payments (EWP) to encrypt pricing and options prior to transmission to PayPal – see http://www.stellarwebsolutions.com/en/articles/paypal_button_encryption_php.php for instructions to create the necessary certificates to support encryption and for instructions to add the certificates to your PayPal account.

System Overview

The eVendor Pro system includes a PHP based shopping cart that can be integrated into your website. Cart status view can be included anywhere on your site and the cart and return/download pages can be customized to your website's look and feel. The cart page allows the customer to view and remove items from their order, as well as return to a main shopping page. The shopping cart can be used to sell physical products as well as the digital goods.

begins with the checkout page which formats a purchase form for transmission to PayPal. Purchase buttons can be added anywhere on your website(s) or in emails to point to the stellarcart.php page (Step 4).

The customer proceeds with the purchase on PayPal's website and follows a return link to your website. An IPN post from PayPal to ipn.php uploads the transaction data to your website and the data is stored in your MySQL database.

The return.php page displays a download link for the customer after payment was successful. The IPN processing script also emails an alternate download link in case the customer did not return to your website or paid with a non-instant payment that requires them to check back later. A refund or disputed transaction automatically cancels the associated download links.

Installation Procedure

To install the Stellar Web Solutions eVendor Pro and IPN scripts there is a five step process:

1. Edit the stellar-settings.php PHP file to customize your site settings and password. You will need to know the database settings from your ISP – database hostname/IP, username, password, and the database name – called a schema.
2. Copy the PHP files to your website.
3. In your web browser, visit install.php to build the database tables.
4. Set up your website to call the checkout PHP functions.
5. Customize the look of your cart and checkout/download pages.

Install Step 1: Configuring stellar-settings.php text file.

The stellar-settings.php file is a PHP “include” file with the common variables used by the other PHP scripts. This file can be edited with any text editor such as Notepad. Some of the example paths use the location /evendor – you can replace this with any web accessible path. Please set values to the following system variables:

System Variables Configuration

The format of the each variable is “\$variable_name = 'value'; //comment”:

- \$db_host = 'localhost'; // database host, if local enter localhost
- \$db_schema = 'databasename'; // database name or schema

- \$db_user = 'user'; // database username to add, update, remove data, add tables.
- \$db_password = 'password'; // database password for the above user
- \$business = 'email@website.com'; // your PayPal primary email address
- \$currency_code = 'USD'; //PayPal currency code for buttons - United States=USD, Canada=CAD, Great Britain=GBP, Euros=EUR etc.
- \$global_currency= '\$'; //currency symbol for the cart page € or \$ etc
- \$continue_url = '/evendor/stellarshop.php'; //continue shopping button in cart
- \$cart_url = '/evendor/stellarcart.php'; //cart link in cart status blob

The \$paypal_cart array contains fields passed directly to PayPal – the rm, notify_url and return page addresses are required:

```
//settings that will be sent directly to PayPal
$paypal_cart = array(
    'cbt' => 'Get the Instant Download Now',
    'notify_url' => 'http://www.yoursite.com/shop/ipn.php',
    'return' => 'http://www.yoursite.com/shop/return.php',
    'rm' => '2',
);
```

Advanced Settings

- \$db_table_sold_trans = 'stellar_evendorpro_trans'; //database table for transactions
- \$db_table_sold_item = 'stellar_evendorpro_item'; //database table for items
- \$db_table_sold_download = 'stellar_evendorpro_download'; //database table for download tracking
- \$receiver_id = 'XXXXXXXXXXXX'; //optional – validate transactions based on receiver ID instead of email address (not recommended)
- \$stellarShop = 'stellareVendorProV10'; //unique ID of the cart in PHP sessions
- \$display_coupon_entry = 0; //1 - show coupon entry box/ 0=hide 1=show
-
- \$global_do_email=1; //email a confirmation 1=on 0=off
- \$global_do_alerts=1; //send email alert on download activity 1=on 0=off
- \$global_debug = 0; //debug - log transactions to file 1=on 0=off
- \$global_logfile = 'stellarlog.txt'; //file to log transactions to
- \$global_disable_test = 0; //security feature - disable sandbox transactions from processing 1=on 0=off
- \$global_pending = 0; //process pending transactions as if they are paid 1=on 0=off

Encryption Settings (only required to use PayPal button encryption)

- \$use_encryption = 0;
- \$site_key = '/usr/home/stellar/paypal/prvkey.pem';
- \$site_cert = '/usr/home/stellar/paypal/pubcert.pem';
- \$site_cert_id = '1234567890HSU';

- \$paypal_cert = '/usr/home/stellar/paypal/paypal_cert.pem';
- \$sandbox_cert = '/usr/home/stellar/paypal/paypal_sandbox.pem';
- \$site_sandbox_key = \$site_key;
- \$site_sandbox_cert = \$site_cert;
- \$site_sandbox_cert_id = '12345678906EA';
- \$OPENSSL = "/usr/bin/openssl";

Install Step 2: Installing the PHP Files

Prior to this step, you should have a webserver which can execute PHP code, you should have at least PHP Version 4 or higher. Most commercial hosting services provide PHP already installed.

- Place the provided .php files on your webserver in a web accessible directory. You can use FTP or a method specified by your internet service provider to upload files to your webserver.

Install Step 3: Building the Database Table.

To install the tables into your database you will need to open your web browser to your website and the path of the install.php file. The database will be automatically be installed if your settings in stellar-settings.php were correct.

Example: <http://www.stellarwebsolutions.com/evendor/install.php>

If you receive an error message, you are likely missing the correct, hostname, username, password, or database name (database schema).

Install Step 4: Setup and use eVendor Pro Digital Goods delivery system.

PHP Item Catalog Setup

The PHP catalog defines both pricing and product information for the cart page and checkout to PayPal, as well as the download filename for electronic goods delivery once the customer has paid.

Items in the catalog.php are stored in a PHP array format, the provided catalog.php file contains sample item entries in an array called \$items which is an array of associative arrays.

```
array(
  'item_name' => 'Item Name',
  'item_number' => 'Item Number',
  'amount' => '123.45',
  'on0' => 'First Option Name',
  'os0' => 'First Option Selection',
  'on1' => 'Second Option Name',
  'os1' => 'Second Option Selection',
  'file' => 'package.zip',//filename of product – can be any format
```

```

),
array(
'item_name' => 'Item Name 2',
'item_number' => 'Item Number 2',
'amount' => '123.45',
'on0' => 'First Option Name',
'os0' => 'First Option Selection',
'on1' => 'Second Option Name',
'os1' => 'Second Option Selection',
'shipping' => '1.23', //currency amount, first item shipping charge
'shipping2' => '1.23', //currency amount, additional items charge
'tax' => '1.23', //amount charged for tax
'handling' => '1.23', //handling charge
'img_sml' => '/images/cart_hat.jpg', //image to display in cart – leave blank for
none
'file' => 'package2.mp3', //filename of product – can be any format
),

```

You can optionally use dynamic PHP code to create the list of items from calculated values or from a database.

All of the above variables are standard PayPal variable names.

Email Confirmation

If the setting `$global_do_email=1;` in `stellar-settings.php`, a confirmation email will be sent to the purchaser on successful receipt of a IPN transmission from PayPal. The contents of the email should be set in the `sendConfirmation` function in `stellar-settings.php`.

Example:

```

in stellar-settings.php:
    $global_do_email=1;

```

```

function sendConfirmation($to, $from, $txn_id, $custom, $invoice)
{
    #email confirmation
    $subject = "Purchase Download Instructions - StellarWebSolutions.com";
    $content = "Thank you for you purchase on StellarWebSolutions.com. \n"
        . "In case you were not automatically redirected to the\n"
        . "instant download page after completing the purchase,\n"
        . "you may access your downloads from the following page:\n\n"
        . "http://www.stellarwebsolutions.com/shop/return.php?payer_"
        . "email=$to&txn_id=$txn_id\n"
        . "Purchaser Email: $to\nTransaction ID: $txn_id\n\n"
        . listItems($txn_id)
        . "For technical support questions, please email “

```

```

        .”stellar@stellarwebsolutions.com.\n\n”
        .”www.stellarwebsolutions.com”];
    emailMessage($to,$from,$subject,$content);
};

```

Note: The listItems(\$txn_id) function will optionally list all the items within the order.

Purchase Links and Buttons for the Checkout Page

Items can be added to the Stellar eVendor Pro built-in Cart with an HTML form “POST” or added to the URL (GET request). Buttons or get requests should point to your cart page, ie the stellarcart.php. You do not need to use the included cart page for integration into your own shopping cart system – just remember to use the eVendor Pro's return URL and notify_url variables to ensure the transaction data is posted to the eVendor Pro's IPN script and that the product item_number and file are defined in the catalog.php file.

Fields accepted by the shopping cart page are:

- item_number – the item number
- quantity – number of items – if left blank defaults to 1
- on0 and on1 – option names
- os0 and os1 – option choice

Basic add to cart button example:

```

<form action="stellarcart.php" method="POST">
<input type="hidden" name="item_number" value="widgetproduct-02">
<input type="submit" name="Submit" value="Buy Our Widget eBook $29.99">
</form>

```

Example GET requests:

Buy an item:

http://yoursite.com/evendor/stellarcart.php?item_number=widgetproduct-07

Install Step 5: Integrate the eVendor Pro Cart into your website's content.

The stellarcart.php and stellarshop.php files included in this package contains sample code to display the cart contents and shopping status respectively.

Shopping Cart Page

You can use the provided stellarcart.php page to build your own shopping cart page. Edit the path to your copy of stellarcart.php to provide to correct path to stellar-cartmod.php:

```

<?php include_once("<relative path>/stellar-cartmod.php"); ?>

```

Your content can be added between the lines:

```
<?PHP
    require_once ('<path to>/stellar-cartmod.php');
    //$test_ipn = 1; //set test_ipn to 1 to use the Sandbox for processing
    stellarcart_open();
?>
```

and to display the cart blob, execute this function:

```
<?PHP
    stellarcart_action();
```

```
?>
```

and your content through until the end of the file. You can use the SandBox for testing payment processing by uncommenting the line \$test_ipn=1;

Cart/Status Integration into your PHP shopping pages

To allow the cart status line to be displayed on any PHP page on your site, the following pre-header lines (before your first <HTML> tag) need to be added:

```
<?PHP
    require_once ('<path to>/stellar-cartmod.php');
    stellarcart_open();
?>
```

Then to actually display the cart status line, add the following into the body of the PHP/HTML portion of the page:

```
<?PHP stellarcart_status(); ?>
```

See the stellarshop.php for a sample of the above integration to a sample shopping page.

Look and Feel

Edit the printCart() function in stellar-cartmod.php to change the CSS definitions to your own look and feel.

Edit the stellarcart_status() in stellar-cartmod.php to change the text/shopping cart icon image for the Cart/Status (quick status display).

Advanced Catalog.php Settings:

Discounts

Discounts are similarly defined as items are, but in the \$discounts array. Coupons require the “id” field while volume discounts which will apply to all customers require the “volume” variable to define the quantity for which the new price takes effect. Either the

“percent” or “amount” variable can be used to define the discount. To apply a coupon to all items – leave the “item_number” field blank and it will apply to all items.

```
array(  
    array(  
        'item_number' => 'Item Number',  
        'id' => 'Coupon code',  
        'percent' => '% discount for item',  
    ),  
    array(  
        'item_number' => 'Item Number',  
        'id' => 'Coupon Code',  
        'amount' => 'new lower item price',  
    ),  
    array(  
        'id' => 'Coupon code',  
        'percent' => 'cart wide discount percentage',  
    ),  
    array(  
        'item_number' => 'Item Number',  
        'volume' => 'minimum quantity for new price',  
        'percent' => '% discount for item',  
    ),  
    array(  
        'item_number' => 'Item Number',  
        'volume' => 'minimum quantity for new price',  
        'amount' => 'new lower item price',  
    ),  
);
```

Advanced Add to Cart Buttons

Items can be added to the Stellar eVendor Pro Cart with an HTML form “POST” or added to the URL (GET request). Buttons or get requests should point to your cart page, ie the stellarcart.php.

Fields accepted by the shopping cart page are:

- item_number – the item number
- coupon – coupon ID code
- on0 and on1 – option names
- os0 and os1 – option choice

Basic add to cart button example:

```
<form action="stellarcart.php" method="POST">  
<input type="hidden" name="item_number" value="stellar-shirt-02">  
<input type="submit" name="Submit" value="Buy Golf Shirt $29.99">  
</form>
```

Example add to cart button with options:

```
<form action="stellarcart.php" method="POST">
  <input type="hidden" name="item_number" value="stellar-shirt-01">
  Size: <select name="os0" >
    <option value="Small">Small</option>
    <option value="Medium">Medium</option>
    <option value="Large" selected>Large</option>
  </select>
  Color:
  <select name="os1" >
    <option value="White" selected>White</option>
    <option value="Black">Black</option>
  </select>
  <input type="submit" name="Submit" value="Buy T Shirt $19.99">
</form>
```

Example GET requests:

Add a discount:

<http://yoursite.com/shop/stellarcart.php?coupon=savings1>

Add an item:

http://yoursite.com/shop/stellarcart.php?item_number=stellar-hat-07

See the example code in stellarshop.php for more sample add to cart buttons.

Clearing the Cart

You can optionally clear the shopping cart contents after a purchase by adding the following to a return page (you can specify the return URL by adding the line “return’ => <http://www.yoursite.com/shop/return.php>,” to the \$paypal_cart array in stellar-cartmod.php to automatically redirect users to the return page after a purchase – the 'cbt' variable defines the text for the return to merchant link).

Before the first <HTML> tag, add the following code to a return page (ie return.php):

```
<?PHP
require_once ('stellar-cartmod.php');
stellarcart_close();
?>
```

Congratulations, you have now installed the Stellar Web Solutions eVendor Pro system.

Appendix A: PHP Primer

If you are not completely familiar with the PHP programming language, the website php.org is a valuable reference. While our software is designed to not require extensive PHP experience, some basic knowledge is necessary.

Errors

A blank page that should have content usually means that an error occurred and that error displaying is disabled. The `ipn.php` or `stellar-settings.php` pages are supposed to be blank. To turn on error displaying for a particular page to debug a problem, add the following lines to the very top of the page:

```
<?PHP
    error_reporting(E_ALL);
    ini_set('display_errors','On');
?>
```

Syntax

PHP code is embedded into HTML, written within “`<?PHP`” and “`?>`” tags, PHP code is executed by the webserver prior to sending the result to a user. This allows passwords and confidential information to be used – such as database connection settings – as this “code” content cannot be seen by users visiting your website. Each line of PHP code must end with a semi-colon “`;`” - the exception to this is that strings can run over multiple lines. Normal strings are enclosed within single quotes (apostrophes) `'string'` and strings which may contain variables must be enclosed within double quotes “string with a `$variable`”.

Comments

Text that follows slashes “`//`” on a line are not executed. We typically comment areas of the code where an advanced user might wish to change to the look and feel of the output of the scripts.

Variables

Variables are defined with a dollar sign, such as `$variable`. A variable followed by an equals sign assigns a value to the variable. When configuring the `stellar-settings.php` file you will assign variables to values. Values within single quotes (apostrophes) `'value'` or double quotes “value” are strings. Quotes are not required when setting a value to the value returned from a function.

Sample variable assignment:
`$variable = 'text value';`

Arrays

Similar to normal variables, arrays add an extra level of complexity and allow multiple items to be stored within the same variable. Most of the Stellar Web Solutions code uses associative arrays which use two entries – a name and value – for each item in the array. This allows the value that corresponds to a name to be found easily. Items are assigned within the “`array()`” function with the “`=>`” operator and separated with commas “`,`”.

Sample array definition:
`$array = array('name1' => 'value1',
 'name2' => 'value2',
);`

You can add items to an already defined arrays as follows:

```
$array['name3'] = 'value3';
```

Displaying Text

To display text output on the webpage – use the “echo” function. To include a new line, use “\n”.

Example:

```
echo “this is a string\n”;
```

Block of text spanning multiple lines:

```
echo “<P>block of text  
spanning multiple  
lines</P>”;
```