



Stellar Web Solutions Order Tracker V1.2 (for PayPal)

Issued: 2007 07 31

Overview

Congratulations on the purchase of the Stellar Web Solutions Order Tracker for PayPal. The Stellar Order Tracker system stores PayPal transactions and provides a feature rich view of your transactions – order information, shipping label template, invoice template, record shipping tracking numbers, and email customers the shipping tracking details. Customize the included PHP/HTML templates for your own look and feel.

By installing and 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.

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.
- stellar-ipnmod.php – IPN code library.
- stellar-ordertracker.php – Order tracker functions.
- index.php – Main Order Tracker page.
- ship.php – Shipping information page – email template.

- label.php – Shipping label template.
- invoice.php – Invoice template
- style.css – default style sheet which can be edited to use your own colors.
- Icons, example 598 pixel wide label banner.

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 directly into your website with a PHP-based order management system to track your recent orders. Orders can be archived, put on hold, or deleted from the database. PayPal IPN data, which is posted to this script over the web, is authenticated, verified, and stored in a MySQL database. All standard PayPal transaction and item variables are supported and stored in your own MySQL database and available to your label/invoice/email templates. Optional email confirmation capability. Both single item and multi-item shopping cart transactions are supported. Proper handling of refunds and reversed/disputed transactions. Auction user IDs supported.

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 can be handled separately from completed transactions.

Installation Procedure

To install the Stellar Web Solutions IPN System there is a four step process:

1. Edit the stellar-settings.php PHP file to customize your site settings and password.
2. Copy the PHP files to your website.
3. In your web browser, visit install.php to build the database tables.
4. Set up email notifications and personalize the include templates.

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. If you use a special editor such as WordPad or FrontPage remember to save the file back as “Text”. You may need to contact your ISP to confirm the database name-called a “Schema”, and your username/password.

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 = 'mycorp'; // database schema ie. The database name
- \$db_user = 'user1'; // database username to add, update, remove data, add tables.
- \$db_password = 'pAssWorD1'; // database password for the above user
- \$business = 'paypal@domain.com'; // your PayPal primary email address
- \$accounts = array ('admin' => 'password1'); // web interface login – you can use anything as a username and password

Advanced Settings

- \$receiver_id = 'XXXXXXXXXXXX'; //optional – validate transactions based on receiver ID instead of email address (not recommended)
- \$db_table_sold_trans = 'stellar_otrack_trans'; //database table for transactions
- \$db_table_sold_item = 'stellar_otrack_item'; //database table for items
- \$db_table_tracking = 'stellar_otrack_tracking'; //database table for tracking
- \$global_do_email=0; //email a confirmation
- \$global_debug = 0; //debug - log transactions to file
- \$global_logfile = 'stellarlog.txt'; //file to log transactions to
- \$global_disable_test = 0; //security feature - disable sandbox transactions from processing
- \$global_pending = 0; //process pending transactions as if they are paid

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. We suggest creating a directory for the Stellar Order Tracker, in our examples, the directory “orders” is used.

- A. Place the provided .php, .jpg, .gif, and .css 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/orders/install.php>

Install Step 4: Using the Order Tracker and personalizing the templates.

Getting Transaction Data On Your eCommerce Site

You can start receiving PayPal 's Instant Payment Notification (IPN) in two ways. Either by setting up IPN in your PayPal profile for all transactions (including auctions), or by adding the notify_url variable to your button code as follows:

Example:

```
<input type="hidden" name="notify_url"
value="http://www.yoursite.com/ipn/ipn.php">
```

Using the Stellar Order Tracker

To start using the Stellar Order Tracker, browse to the index.php file that was included in your package and uploaded to your webserver in step 2.

Example: <http://www.stellarwebsolutions.com/orders/index.php>
or <http://www.stellarwebsolutions.com/orders/>

There are 3 “folders” of transactions – new orders, archived items, and on hold. All transactions initially start in the new orders area and are color coded based on the payment status – by default, green is paid, yellow is pending, blue was refunded, and red is failed or disputed.

The “View New”, “View Archive”, and “View On Hold” buttons allow you to view each area. The top status area will display the current number of paid orders, gross income, PayPal fees, and net income for the selected date range.

The “Add To New”, “Add To Archive” and “Add To On Hold” buttons will move selected items to each of the areas and return you to viewing new orders.

Archived orders can be purged/permanently deleted from the database by selecting some orders and clicking the delete button.

Personalizing The Templates

Quick Start option – simply create your own 598 x 53 pixel banner.jpg with your own logo and address – this will appear on the shipping label and invoice.

ship.php: Insert your own shipping notification email generic content after the line “Thank you for your order.”.

label.php: Edit the template for your own shipping label after the line “<HTML>”. Any PayPal variable can be accessed as \$t['variablename'].

invoice.php: Edit the template for your own invoice after the line “<HTML>”. Any PayPal variable can be accessed as \$t['variablename'].

Email Confirmation Of Purchase/Order

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 completely paid 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.

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.

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>”;
```

Appendix B: Database MySQL Reference

See https://www.paypal.com/IntegrationCenter/ic_ipn-pdt-variable-reference.html for definitions of the database fields below.

Transactions:

```
CREATE TABLE `$_db_table_sold_trans` (  
  `id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  `invoice` VARCHAR(127),  
  `custom` VARCHAR(255),  
  `test_ipn` INTEGER UNSIGNED DEFAULT 0,  
  `memo` VARCHAR(255),  
  `business` VARCHAR(127),  
  `receiver_id` VARCHAR(13),  
  `receiver_email` VARCHAR(127),  
  `first_name` VARCHAR(64),  
  `last_name` VARCHAR(64),  
  `contact_phone` VARCHAR(24),  
  `address_city` VARCHAR(40),  
  `address_country` VARCHAR(64),  
  `address_country_code` VARCHAR(2),  
  `address_name` VARCHAR(128),  
  `address_state` VARCHAR(40),  
  `address_status` VARCHAR(12),  
  `address_street` VARCHAR(200),  
  `address_zip` VARCHAR(20),  
  `payer_business_name` VARCHAR(127),  
  `payer_email` VARCHAR(127),  
  `payer_id` VARCHAR(13),  
  `payer_status` VARCHAR(12),  
  `residence_country` VARCHAR(2),  
  `txn_id` VARCHAR(17),  
  `txn_type` VARCHAR(24),  
  `transaction_entity` VARCHAR(10),  
  `tax` DECIMAL(6,4),  
  `auth_id` VARCHAR(64),  
  `auth_exp` VARCHAR(64),  
  `auth_status` VARCHAR(12),  
  `auth_amount` DECIMAL(10,2),  
  `num_cart_items` INTEGER,  
  `mc_currency` VARCHAR(3),  
  `exchange_rate` DECIMAL(10,2),  
  `mc_fee` DECIMAL(10,2),  
  `mc_gross` DECIMAL(10,2),  
  `parent_txn_id` VARCHAR(17),  
  `payment_date` VARCHAR(64),  
  `payment_status` VARCHAR(24),
```

```

`payment_type` VARCHAR(12),
`pending_reason` VARCHAR(16),
`reason_code` VARCHAR(16),
`remaining_settle` DECIMAL(10,2),
`mc_handling` DECIMAL(10,2),
`mc_shipping` DECIMAL(10,2),
`settle_currency` VARCHAR(3),
`case_id` VARCHAR(64),
`case_type` VARCHAR(12),
`case_creation_date` VARCHAR(64),
`handling` DECIMAL(10,2),
`shipping` DECIMAL(10,2),
`settle_amount` DECIMAL(10,2),
`auction_buyer_id` VARCHAR(64),
`auction_closing_date` VARCHAR(64),
`auction_multi_item` INTEGER,
`for_auction` VARCHAR(10),
`subscr_date` VARCHAR(64),
`subscr_effective` VARCHAR(64),
`period1` VARCHAR(12),
`period2` VARCHAR(12),
`period3` VARCHAR(12),
`amount1` DECIMAL(10,2),
`amount2` DECIMAL(10,2),
`amount3` DECIMAL(10,2),
`mc_amount1` DECIMAL(10,2),
`mc_amount2` DECIMAL(10,2),
`mc_amount3` DECIMAL(10,2),
`recurring` VARCHAR(1),
`reattempt` VARCHAR(1),
`retry_at` VARCHAR(64),
`recur_times` INTEGER,
`username` VARCHAR(64),
`password` VARCHAR(64),
`subscr_id` VARCHAR(19),
`receipt_id` VARCHAR(24),
`time` DATETIME,
PRIMARY KEY(`id`)
)
TYPE = InnoDB;";

```

Items:

```

CREATE TABLE `$_db_table_sold_item` (
  `id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  `parent_id` INTEGER UNSIGNED,
  `sequence_id` INTEGER UNSIGNED,

```

```
`test_ipn` INTEGER UNSIGNED DEFAULT 0,  
`item_name` VARCHAR(127) NOT NULL,  
`item_number` VARCHAR(127),  
`mc_gross` DECIMAL(10,2),  
`option_name1` VARCHAR(64),  
`option_selection1` VARCHAR(200),  
`option_name2` VARCHAR(64),  
`option_selection2` VARCHAR(200),  
`quantity` INTEGER,  
`mc_shipping` DECIMAL(10,2),  
`mc_handling` DECIMAL(10,2),  
`shipping` DECIMAL(10,2),  
`tax` DECIMAL(6,4),  
`time` DATETIME,  
PRIMARY KEY(`id`)  
)  
TYPE = InnoDB;
```