



## **Stellar Web Solutions Donations Tracker V1.5 (PayPal)**

Issued: February 17, 2010.

### ***Overview***

Congratulations on the purchase of the Stellar Web Solutions Donations Tracking (PayPal IPN Edition) system. The Stellar Web Solutions Donations Tracking interface allows you to integrate PayPal transaction data into your website and MySQL database for use by your web applications. 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](http://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-merge.php – Graphic merge script.
- stellar-therm.php – Thermometer graphic script.
- stellar-donations.php – Donations reporting library.
- index.php – Demo page.
- donorlist.php – Demo of list of donors.

## **Support**

To obtain installation support, we recommend emailing us at [stellar@stellarwebsolutions.com](mailto:stellar@stellarwebsolutions.com). Otherwise, visit [www.stellarwebsolutions.com/en/contact.php](http://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 for real-time up-to-date views of your donations. PayPal IPN data is authenticated, verified, and stored in a MySQL database.

## **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.
- Optional PayPal Encrypted Website Payments support.

## **Installation Procedure**

To install the Stellar Web Solutions IPN scripts 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 your website to call the donations reporting PHP functions.

## **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. 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
- \$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](mailto:paypal@domain.com)'; // your PayPal primary email address

- \$notify\_url = 'http://www.yoursite.com/donations/ipn.php'; //url to IPN for button code
- \$return = 'http://www.yoursite.com/donations/'; // url to return page for button code
- \$global\_currency= '\$'; // url to return page for button code
- \$currency\_code = 'USD'; //PayPal currency code for buttons - United States=USD, Canada=CAD, Great Britain=GBP, Euros=EUR etc.

### Advanced Settings

- \$receiver\_id = 'XXXXXXXXXXXX'; //optional – validate transactions based on receiver ID instead of email address (not recommended)
- \$db\_table\_sold\_trans = 'stellar\_paypal\_trans'; //database table for transactions
- \$db\_table\_sold\_item = 'stellar\_paypal\_item'; //database table for items
- \$global\_pending = 1; //process pending transactions as if they are paid
- \$global\_do\_email=0; //email a confirmation (edit the sendConfirmation() function to contain your email content.
- \$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

### Encryption related settings (only required if you are using PayPal encryption):

- \$use\_encryption = 1; //use encrypted website payments 1 / 0 plain text
- \$site\_key = '/usr/home/paypal/prvkey.pem'; //your private key
- \$site\_cert = '/usr/home/paypal/pubcert.pem'; //your public certificate
- \$site\_cert\_id = 'ABCDEF1234567'; // certificate id provided by PayPal
- \$paypal\_cert = '/usr/home/paypal/paypal\_cert.pem'; //paypal certificate
- \$sandbox\_cert = '/usr/home/paypal/paypal\_sandbox.pem'; //paypal sandbox certificate
- \$site\_sandbox\_key = \$site\_key; //your private key for sandbox
- \$site\_sandbox\_cert = \$site\_cert; //your public certificate for sandbox
- \$site\_sandbox\_cert\_id = 'ABCDEFG123456'; // certificate id provided by PayPal for sandbox
- \$OPENSSL = "/usr/bin/openssl"; // path to 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.

- A. 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/donations/install.php>

You can delete the install.php after it successfully runs, there is no risk to leaving the file on your server.

### **Install Step 4: Setup and use the Donations Tracking system.**

The index.php file included in this package contains sample code on how use create PayPal donation buttons and how to display report text and graphics.

To allow your PHP code to call any of the pre-defined, you must include the following two include statements in your PHP pages:

```
<?php include_once("<relative path>/stellar-settings.php"); ?>  
<?php include_once("<relative path>/stellar-donations.php"); ?>
```

Example:

```
<?php include_once("./donations/stellar-settings.php"); ?>  
<?php include_once("./donations/stellar-donations.php"); ?>
```

## Using the Donations Tracking System

### Buttons

When using the complex status text displays, PayPal payment buttons are automatically generated, you can add a payment button on other pages that use the simple value displays or thermometer as follows:

#### Automatically Generated Paypal Buttons

The Stellar Donations Tracker can automatically create a standard donation payment button with all the necessary settings:

```
<?PHP printDonationButton('donation'); ?>
```

To prompt for a comment in the memo field:

```
<?PHP printDonationButton('donation',1); ?>
```

Monthly rotation donations:

```
<?PHP printDonationButton(date("FY")); ?>
```

#### Standard PayPal Button:

The following HTML code is a sample of a basic PayPal button for Donations.

```
<form action="https://www.paypal.com/cgi-bin/webscr" method="post">
  <P><B>Live Site Sample Button</B></P>
  <input type="hidden" name="cmd" value="_xclick">
  <input type="hidden" name="business" value="<your PayPal email address">
  <input type="hidden" name="item_name" value="Donation">
  <input type="hidden" name="item_number" value="donation">
  <input type="hidden" name="currency_code" value="USD">
  <input type="hidden" name="notify_url" value="<path to
ipn.php>/donations/ipn.php">
  <input type="hidden" name="return" value="<path to return page>/donations/">
  <input type="image" src="https://www.paypal.com/en_US/i/btn/x-click-
but21.gif" border="0" name="submit" alt="Make payments with PayPal - it's fast,
free and secure!">
</form>
```

The item\_number field is used to track your campaign, you can specify different values to allow for separate total reporting, ie. item\_number could be “donations2006” for your 2006 Campaign.

For monthly donations that will automatically reset on the 1<sup>st</sup> of every month – use the following dynamic item\_number:

```
<input type="hidden" name="item_number" value="<?PHP echo date("FY"); ?>
">
```

The notify\_url specifies the IPN script to use to track your donations, this line should be added to any donation buttons generated by PayPal.

You can also specify the donation amount to support pre-set donations amounts:

```
<input type="hidden" name="amount" value="10.00">
```

See [www.paypal.com](http://www.paypal.com) for more Donation Button options.

## Reports

### Individual Donations Totals.

For each item\_number, you can access the total of all donations gross, net, or the PayPal fees charged:

Example:

```
<P>Total Donations: $<?php echo getDonations('donation'); ?></P>
```

```
<P>Total Fees: -$<?php echo getDonationFees('donation'); ?></P>
```

```
<P>Net Donations: $<?php echo getNetDonations('donation'); ?></P>
```

Note: The 'donation' passed to the function should correspond to the item\_number used in your PayPal donation button.

### Thermometer Graphic

The thermometer graphic can be called from your PHP HTML code as follows:

```
<img src=stellar-therm.php?current=<?php echo getNetDonations('donation'); ?>
>&target=200>
```

The options that can be appended to the URL are as follows:

&target=<amount>: Goal for your donations campaign – default 100.

&type=<gif/png/jpg>: Output image type – default is gif

&symbol=<\$/EUR/GBP/YEN>: Currency symbol

&width=<pixel width>: Image width

&height=<pixel height>: Image height

&bar=<pixel width>: thermometer's bar width

&ball=<ball width>: thermometer's ball width – number of multiples of the bar width

&padding=<pixel width>: Padding around the top and bottom of the image

&scaling=<1000/1000000>: Scale the currency to this amount (show thousands etc).

&scalingSymbol=<K/M>: Symbol to used when currency amounts are scaled.

### Merged Graphics

The merged custom graphic can be called from your PHP HTML code as follows:

```
<img src=stellar-merge.php?current=<?php echo getNetDonations('donation'); ?>
>&target=200>
```

You should edit the stellar-merge.php file to store your graphic and settings:

mergeI(\$\_GET['current'], \$\_GET['target'], '<main image path>', '<overlaid image path>', <bottom of area to merge>, <top of area to merge>);

## Complex Reports

Four sample complex reports are defined as functions within stellar-donations.php. donationsMap(<item\_number to report on>,<goal amount>) function prints the donation detail as well as recent donor first names and initials. The HTML and data displayed can be modified to display any way you like. See the Advanced Views section for more information on built in functionality of donationsMap.

The donationsMapComments(<item\_number to report on>,<goal amount>,<minimum donation to show comments for>) function prints the donation detail as well as the top donor's first names and initials. For donors who've donated above a specified amount (which can be 0 to display comments/memos for all donations), the memo field on the PayPal checkout page is printed after their name and donation amount. This can allow higher donors to list their URL or similar advertisement. To have a HTML link shown, a user would have to enter the html in the memo field on PayPal.com: “<a href=<http://www.stellarwebsolutions.com/>>Stellar Web Solutions</a>”. See the Advanced Views section for more information on built in functionality of donationsMap.

The donationsMapTop(<item\_number to report on>,<goal amount>,<optional number of items to display>) function prints the donation detail of either all donors or the top 5 donations as well as donor first names and initials.

The donationsMapRecent(<item\_number to report on>,<goal amount>,<optional number of recent donations to display>) function prints the donation detail of either all donors or the most recent 5 donations as well as donor first names and initials.

The donationsMapMonthly(<item\_number to report on>,<goal amount>,<optional number of recent donations to display>) function prints the donation detail of either all donors or the top 5 donations as well as donor first names and initials. While this function is similar to the donationsMapTop function, the title of the campaign will display the current month and year. When used with a dynamic call to this function with the PHP date, a no maintenance automatically resetting monthly campaign is the result.

Examples:

```
<?PHP donationsMap('donation',100); ?>  
<?PHP donationsMapComments('donation',100,5); ?>  
<?PHP donationsMapTop('donation',100); ?>  
<?PHP donationsMapRecent('donation',100); ?>  
<?PHP donationsMapMonthly(date("FY"),100); ?>
```

## Scrolling Marquee

The `donationsMarquee` function in `stellar-donations.php` prints a text listing of the top donors which can be wrapped in a `<marquee>` `</marquee>` HTML tag to create a scrolling list of donors. Any HTML may be used to create your own look and feel for the text. The first parameter is the `item_number` of the donations campaign.

For example, the following HTML and PHP will display a donations marquee in a black box with white text:

```
<TABLE border=0>
<TR><TD bgcolor=black><font color=white size=2>
<marquee><?PHP donationsMarquee('donation'); ?></marquee>
</TD></TR></TABLE>
```

## List of Donors

The `donationsMarquee` function in `stellar-donations.php` prints a HTML table listing of the donors.

For example, the following HTML and PHP will display a donations marquee in a black box with white text:

```
<?PHP donationsList('donation'); ?>
```

## Advanced Views

The `donationsMap` function in `stellar-donations.php` can be called with a variety of options. All of the other `donationMap` functions are all just shortcuts for the `donationsMap` main function. To edit the look and feel of the HTML or CSS, edit the PHP function `donationsMap` in `stellar-donations.php`. Additionally, all the transaction and item variables provided in the PayPal IPN can be referenced in PHP as “`$row[variable]`” within this function – see the Appendix of this document for the list of variables.

The following are the possible options:

`donationsMap(<item_number>, <target campaign goal as a decimal number 100 or 75.75 etc?>, <view: 'normal' (last 30 days by default), 'top' (top 5 donations by default), 'last' (show the 5 most recent transactions), 'monthly' (Top donations titled with the current month and year)>, <number of items or days to display>, <minimum donation amount to show comments>, <display comments 1=yes, or 0=no>`

Example:

```
donationsMap('donation', 100, 'normal', 30) (displays the past 30 days of donations)
donationsMap('donation', 100, 'normal', 30, 0,1) (displays the past 30 days of donations with comments/memos )
donationsMap('donation', 100, 'top', 10) (displays the top 10 donations)
```

`donationsMap('donation', 100, 'top', 0,1)` (displays the top 10 donations with comments/memos)  
`donationsMap('donation', 100, 'top', 10, 5, 1)` (displays the top 10 donations with comments/memos for donations above 5 currency units – dollars or euros etc)  
`donationsMap('donation', 100, 'last')` (displays the last 10 donations)  
`donationsMap('donation', 100, 'last', 0,1)` (displays the last 10 donations with comments/memos)

The `donationsMapComments` can also be used in a similar way, the following are it's options:

`donationsMapComments(<item_number>, <target campaign goal as a decimal number 100 or 75.75 etc?>, <minimum donation amount to show comments>, <view: 'normal' (last 30 days by default), 'top' (top 5 donations by default), 'last' (show the 5 most recent transactions)>, <number of items or days to display>)`

Example:

`donationsMapComments('donation', 100, 0, 'top', 20)` (displays the top 20 donations with comments/memos for all donations.  
`donationsMapComments('donation', 100, 5, 'top', 20)` (displays the top 20 donations with comments/memos for donations above 5 currency units – dollars/euros etc.

## Displaying Comments

When comments are displayed, the PayPal variable “cbt” is used to display the following label for the memo field on the PayPal payment page “Optional public message to display”.

## Anonymous Complex Display

To anonymize the donors listing (by displaying the memo field or if the memo is left blank simply “Anonymous”) for each donation in the complex status views, include the following setting on the PHP line before the `donationsMap` function is called “`$global_anonymous=1;`”. Additionally the PayPal variable “cbt” is used to display the following label for the memo field on the PayPal payment page “Optional anonymous message/name to display”.

Example:

```
$global_anonymous=1;  
donationsMapComments('donation', 100, 0, 'top', 20)
```

## **Appendix A: PHP Primer**

If you are not completely familiar with the PHP programming language, the website [php.org](http://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>”;
```

## Appendix B: Database MySQL Reference

### Transactions:

```
CREATE TABLE `sdb_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_amount` 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),
`receipt_id` VARCHAR(24),
`time` DATETIME,
PRIMARY KEY(`id`)
)
TYPE = InnoDB;";

```

**Items:**

```

CREATE TABLE `Sdb_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;

```