



Stellar Web Solutions PHP IPN System V2.1 (for PayPal)

Issued: 2007 08 04

Overview

Congratulations on the purchase of the Stellar Web Solutions IPN System for PayPal. The Stellar IPN system records all PayPal transactions into a MySQL database and optionally allows confirmation/notification emails to be sent. Use the Stellar IPN System as a foundation to integrate PayPal sales into your own web applications or implement your own server-side electronic goods delivery or sales tracking system.

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.

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.

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 for use with your web applications or back-end sales systems. 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. Support for PayPal IPN variables: Basic, Advanced, Custom, Website Payments Standard, Website Payments Pro, Refunds, Currency, Currency Exchange, Subscriptions, Auctions, and Dispute Notification. Only Masspay variables are not supported in this version. Optional email confirmation capability. Both single item and multi-item shopping cart transactions are supported. Refunds and reversed/disputed transactions are recorded in the 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.

Installation Procedure

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

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

Install Step 1: 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 2: 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, if you cannot edit files right on your webserver, FTP the file to a PC where you can use Notepad/Wordpad (or equivalent). 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'; // your PayPal primary email address

Advanced Settings

- \$receiver_id = 'XXXXXXXXXXXX'; //optional – validate transactions based on receiver ID instead of email address (not recommended)
- \$db_table_sold_trans = 'stellar_ipn_trans'; //database table for transactions
- \$db_table_sold_item = 'stellar_ipn_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

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/ipn/install.php>

Install Step 4: Using IPN Transaction Data.

Transaction specific data is stored in the MySQL table defined with the name set in the stellar-settings.php variable \$db_table_sold_trans, by default this table is called stellar_ipn_trans.

Item specific data is stored in the MySQL table defined with the name set in the stellar-settings.php variable \$db_table_sold_item, by default this table is called stellar_ipn_item.

The transaction and items tables can be joined based on the condition “\$db_table_sold_trans.id=\$db_table_sold_item.parent_id”:

Example:

```
select * from $db_table_sold_trans LEFT JOIN $db_table_sold_item ON
$db_table_sold_trans.id=$db_table_sold_item.parent_id;
```

Refunded or reversed transactions can be located based on the parent_txn_id of refund/reversal transactions matching up with the original transaction's txn_id.

Example (locating a refunded transaction):

```
SELECT * FROM $db_table_sold_trans where parent_txn_id = 'query_txn_id';
```

Example (joining items and refunded transactions together):

```
select * from $db_table_sold_trans as a LEFT JOIN $db_table_sold_item as b
ON a.id=b.parent_id LEFT join $db_table_sold_trans as c ON a.txn_id =
c.parent_txn_id;
```

Example (selecting all completed transactions without disputes/refunds/reversals):

```
select COUNT(*) from $db_table_sold_trans as a LEFT JOIN
$db_table_sold_item as b ON a.id=b.parent_id LEFT join
$db_table_sold_trans as c ON a.txn_id = c.parent_txn_id where
a.payment_status = 'Completed' AND b.item_number='$item_number'
AND c.payment_status is NULL;
```

Any PayPal provided variable can be accessed through the MySQL database created by the Stellar IPN System.

Example (Calculate the sum of all sales):

```
select sum(b.mc_gross) from $db_table_sold_trans as a LEFT JOIN
$db_table_sold_item as b ON a.id=b.parent_id LEFT join
$db_table_sold_trans as c ON a.txn_id = c.parent_txn_id where
a.payment_status = 'Completed' AND c.payment_status is NULL;
(the result will be in the column named “sum(b.mc_gross)”)
```

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.

Using PayPal IPN On Your eCommerce Site

You can start using PayPal 's Instant Payment Notification (IPN) in two ways. Either by setting up IPN in your PayPal profile for all transactions, 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">
```

Appendix A: 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;
```