



# User Manual

## Shopping Cart Module v1.x

Created by:  
Envex Developments Inc.  
<http://www.envex.com/>

Version 1.2  
April 25<sup>th</sup>, 2005

# Table of Contents

1. [Introduction](#)
2. [Getting Started](#)
  - 2.1 [Reload Crontab File](#)
  - 2.2 [Move order.cgi Script](#)
  - 2.3 [First Time Setup](#)
  - 2.4 [Create Payment Methods](#)
  - 2.5 [Tax Settings](#)
  - 2.6 [Privilege Settings](#)
  - 2.7 [Manage Customer Area](#)
  - 2.8 [Notification Settings](#)
  - 2.9 [Create Product Categories](#)
  - 2.10 [Create Products](#)
  - 2.11 [Create Product Commissions](#)
  - 2.12 [Create HTML Links](#)
3. [Processing Orders](#)
  - 3.1 [Customer's Shopping Cart](#)
  - 3.2 [Order Submissions](#)
  - 3.3 [Process Pending Orders](#)
  - 3.4 [Pending Payment Methods](#)
  - 3.5 [Process Refund](#)
  - 3.6 [Recurring Orders](#)
4. [HTML Templates](#)
  - 4.1 [HTML Template Descriptions](#)
5. [Integrate Third Party Scripts](#)
  - 5.1 [File / Directory Structure](#)
  - 5.2 [Available Variables](#)
  - 5.3 [Important Notes](#)

# 1.) Introduction

The Shopping Cart module allows you to easily sell products through your member's self replicating web sites. You can have commissions automatically added to the members, when an order is placed from their self replicating web site. Many features are included such as, product categories, tax settings, password protected Customer Area, plus more.

## 2.) Getting Started

This chapter explains how to get started using the Shopping Cart module, once it has already been installed on your web server. If the module has not yet been installed on your server, you may do so through the Modules section of the Envex Developments Customer Area.

### 2.1 Reload Crontab File

Once the Shopping Cart has been installed on your web server, you must first reload your Crontab file to ensure everything works correctly. This can be done by logging into your web server via SSH, and changing directories to the `/xflow_data/` directory on your server. Once there, type the following at the SSH prompt:

```
crontab xflow.crontab
```

If you do not know how, or do not feel comfortable reloading your Crontab file, please submit a support request to Envex Developments, asking them to reload your crontab file for you.

### 2.2 Move order.cgi Script

This step is completely optional, and only required if you have access to an SSL server, and want to make sure all orders are placed on the SSL server. To do this, you need to move the `order.cgi` script on your server. FTP into your server, and change to the `/cgi-bin/xflow/modules/shop/bin/` directory on your server. You will see an `order.cgi` script in this directory, which you can download, then re-upload to your SSL server. Please make note of the URL of the `order.cgi` script after you've moved, as it's required during First Time Setup.

### 2.3 First Time Setup

The first thing you need to do is run First Time Setup, which can be done through the *Setup->First Time Setup* menu of the Admin Control Panel for the Shopping Cart. The questions should be fairly straight forward. Just complete the forms, until First Time Setup is complete.

### 2.4 Create Payment Methods

Next, you need to create all payments methods which customers will be able to pay for their order with. This can be done through the *Setup->Payment Settings* menu, and works exactly the same as creating payment methods through the main menu of the Admin Control Panel.

### 2.5 Tax Settings

You can manage your tax settings through the *Setup->Tax Settings* menu. When an order is submitted, the shopping cart will go through the tax settings, and see if the customer's profile matches any taxes. The needed amount is then added to the order total. At the bottom of the page, there is a section titled Add New Page, which looks something like:

### Add New Tax

To add a new tax, complete the below form. Only the first three fields are required to be automatically added to all orders who match the criteria below.

|                 |   |
|-----------------|---|
| Tax Name:       | <input type="text" value="GST"/>        |
| Amount Type:    | <input type="text" value="Percentage"/> |
| Amount:         | \$ <input type="text" value="7"/>       |
| Customer Field: | <input type="text" value="Country"/>    |
| Matching Text:  | <input type="text" value="canada"/>     |

The above example would automatically add 7% GST to all orders where the customer's Country fields equals 'canada'. Continue making all needed taxes, using the same method.

## 2.6 Privilege Settings

Next, you need to set all privilege settings, which can be done through the *Setup->Privilege Settings* menu. This page contains two sections. The first section lists all administrators (except the first) in your database, allowing you to specify exactly which action each administrator can perform. The second section shows a list of all actions, which can be performed through the Customer Area.

## 2.7 Manage Customer Area

This step is only required if you plan to use the password protected Customer Area. You can manage the web pages in your Customer Area through the *Setup->Manage Customer Area* menu. This is exactly the same as managing web sites / Member Only Areas through the main Admin Control Panel.

You will also need to generate the HTML login form for the Customer Area, which can be done through the *Setup->Generate HTML Forms*.

## 2.8 Notification Settings

Next, you need to setup your notification settings, which can be done through the *Communicate->Notification Settings* menu. This basically works exactly the same as creating notifications through the main menu of the Admin Control Panel.

## 2.9 Create Product Categories

The Shopping Cart fully supports an unlimited number of product categories. You can create all needed categories through the *Products->Manage Categories* menu. Simply type the

category name into the provided text box, and continue until you've created all needed categories. You can create sub-categories within a category, by clicking on the category name link, and entering the category. Please note, this step is optional, and product categories are not required.

## 2.10 Create Products

Next, you need to create all products you will be selling. This can be done through the *Products->Add New Product* menu. The form itself should be fairly straight forward. The Product ID field can be anything you wish, and will be used to reference the product throughout the script. Continue filling out the form, until you've created all needed products.

## 2.11 Create Product Commissions

You can have commission automatically added to members, for orders placed from their self replicating web site. This can be done by creating all needed product commissions, through the *Products->Manage Commissions* menu. This is very similar to creating commission transactions through the main menu of the Admin Control Panel. Continue creating commission transactions through this menu, for all needed products in your database.

## 2.12 Generate HTML Links

Last, you need to create all needed HTML links, and place them in your self replicating web site. This can be done through the *Setup->Generate HTML Forms* menu. The second section on the page titled Generate HTML Links, allows you to generate all needed links to place in your self replicating web site. You can specify up to four products when creating a link to add products to the cart, or purchase from the Customer Area. The quantity fields just below the product ID fields, allow you to specify the quantity of each product to add / purchase.

Below explains how to manually create the links. All scripts for these links are located in the `/cgi-bin/xflow/modules/shop/bin/` directory of your server.

### 2.12.1 Add Product to Cart

You can allow people to add products to their cart, by creating a link to the following URL:

[http://www.domain.com/cgi-bin/xflow/modules/shop/bin/add\\_cart.cgi](http://www.domain.com/cgi-bin/xflow/modules/shop/bin/add_cart.cgi)

The following elements are allowed in the query string:

|                   |   |
|-------------------|---|
| <b>product_id</b> | The ID# of the product(s) you would like to add. You can add multiple products to the cart, by specifying a comma delimited list of all product ID#'s. This element is required.  |
| <b>quantity</b>   | The quantity of all products being added. If adding multiple programs, a comma delimited list, specifying the quantity of each product respectively. For example, the second product ID# in the list, would correspond with the second quantity value. This element is optional, and if not present, will default to 1. |
| <b>reseller</b>   | The username of the reseller for this order. Almost always, the value will be <code>~username~</code> . Then when someone views a   |

|  |  |
|--|--|
|  | member's self replicating web site, the ~username~ merge field will be replaced with their username, making them the reseller for the order. This field is optional, and if not present, will default to the Default Reseller you specified during First Time Setup. |
|--|--|

For example, to add the product ID# CD029, you would use the following link:

```
add_cart.cgi?product_id=CD029&reseller=~username~
```

Or to add the product ID# SER032, and three of the product ID# CD318, you would use the following link:

```
add_cart.cgi?product_id=SER032,CD318&quantity=1,3&reseller=~username~
```

### 2.12.2 Browse Category

If you're using product categories, you can allow people to browse all products in your database by category. Each page shows a list of sub-categories and products within the selected category they are browsing. You can allow people to browse categories by linking to the following URL:

```
http://www.domain.com/cgi-bin/xflow/modules/shop/bin/browse.cgi
```

The only element in the query string is **category**. The value is the ID# of the product category you would like to browse. You can find the ID# by going to the *Products->Manage Categories* menu. For example, if you would like to browse the category ID# 13, you would use the following link:

```
browse.cgi?category=13
```

Alternatively, you can browse all categories in your database, starting from the root categories, by simply going to the browse.cgi script, without any query string.

### 2.12.3 View Shopping Cart

You can allow people to view and manage the contents of their shopping cart, by going to the following URL:

```
http://www.domain.com/cgi-bin/xflow/modules/shop/xflow/view_cart.cgi?reseller=~username~
```

### 2.12.4 Checkout Link

You can allow people to check out and place their order, by creating a link to the following URL:

```
http://www.domain.com/cgi-bin/xflow/modules/shop/bin/checkout.cgi
```

No query string is required. By going to the URL, the customer will be automatically redirected to the order form, to place their order. However, you can allow a customer to checkout with specific product(s), instead of making the customer add products to their cart first. This is good if, for example, the customer will only ever order a certain product. You

can create one link to the checkout.cgi script, which will take the customer directly to the order form to purchase the specified products, instead of making the customer first add them to their cart.

To create this link, add a query string to the checkout.cgi script, using the following elements:

|                   |   |
|-------------------|---|
| <b>override</b>   | Set this to <b>1</b>  |
| <b>product_id</b> | The ID# of the product(s) you would like to purchase. You can add multiple products to the cart, by specifying a comma delimited list of all product ID#'s. This element is required.   |
| <b>quantity</b>   | The quantity of all products being purchased. If purchasing multiple products, a comma delimited list, specifying the quantity of each product respectively. For example, the second product ID# in the list, would correspond with the second quantity value. This element is optional, and if not present, will default to 1.   |
| <b>reseller</b>   | The username of the reseller for this order. Almost always, the value will be <b>~username~</b> . Then when someone views a member's self replicating web site, the <b>~username~</b> merge field will be replaced with their username, making them the reseller for the order. This field is optional, and if not present, will default to the Default Reseller you specified during First Time Setup. |

For example, if you would like to checkout with the product ID# GA482, you would use the following link:

```
checkout.cgi?override=1&product_id=GA482
```

Or if you would like to checkout with the product ID# ML229, and two of the product ID# PW418, you could use the following link:

```
checkout.cgi?override=1&product_id=ML229,PW418&quantity=1,2
```

### 2.12.5 Purchase product from Customer Area

You can allow customers to easily purchase products through the Customer Area, allowing them to use any existing payment methods on their account. This saves your customers from having to enter any contact or billing information. Instead, they can simply select which payment method to use from a list, and that's it.

To allow customers to purchase products from within the Customer Area, you need to create the following link:

```
index.cgi?~authtext~&action=purchase&product_id=[PRODUCT]&quantity=[QUANTITY]
```

Replace the **[PRODUCT]** and **[QUANTITY]** text with their correct values, as explained above when creating a link to add products to the shopping cart, or checkout.

### 2.13 Setup Complete!

If you have successfully completed all above steps, the Shopping Cart module has now been fully setup, and is ready to take orders. All needed links should be on your self replicating web site, allowing customers to add products and checkout. When a customer checks out, they will be automatically redirected to an order form, where their order will be processed accordingly.

## 3.) Processing Orders

This chapter explains how orders are processed and handled by the Shopping Cart. Everything is explained, including recurring transactions, coupon codes, processing refunds, plus more.

### 3.1 Customer's Shopping Cart

This is what allows customers to add products to their shopping cart, view / manage their cart, and so on. This process does use cookies to store the needed reference information for the customer. If a customer's web browser does not allow cookies, they will not be able to add any products, checkout, etc.

The first time a customer either, adds a product to their cart, or views the contents of their cart, the script will assign the reseller for the order. Once the reseller has first been assigned, it will not change. This allows the customer to stray off the member's web site, but the member will still receive credit for the order.

### 3.2 Order Submissions

When you checkout, an order form is displayed where you can fill out your contact and payment information. If you specified to ask for returning customers during First Time Setup, the script will first ask the customer if they've purchased before, and for their Customer ID#. If they enter a valid customer ID#, they will be prompted to confirm the information, then brought to the order form. This way, the customer does not have to enter their contact information again, and instead, only has to enter their payment information.

Once an order is submitted, the shopping cart will first determine the total amount to charge the customer. First, all products in the customer's cart are added up. Next, all taxes are checked, and if any match the customer, they will be added to the total amount. Last, the shopping cart checks for a coupon code, and if needed, subtracts the needed amount.

The payment is then physically processed. If the customer is paying through a third party payment processor, such as PayPal or 2CheckOut, the order will be added to the database with the status of pending, and the customer will be automatically redirected to the needed payment server. Once the customer has submitted payment, the third party payment processor will contact the shopping cart on your server. The shopping cart will then automatically approve the order, without any intervention from you.

Otherwise, if you have a real merchant account, and accept credit cards on your server, the credit card will be authorized / charged behind the scenes, in real time. If declined, the customer will receive an error saying their credit card has been declined. If approved, the order will be added to the database with a status of either approved or pending, depending on your settings.

### 3.3 Process Pending Orders

If during First Time Setup, you setup the shopping cart to only authorize credit cards for new orders, then all new orders submitted will have a status of pending. Otherwise, the only orders that will appear as pending are those paid through an unsupported / manual payment processor, and those paid through a third party payment processor, but haven't submitted payment yet.

You can manually process all pending orders through the *Orders->Process Pending Orders* menu of the Admin Control Panel. This page contains two tables, listing all new orders and recurring payments, which are pending. Simply select all orders you would like to process, and complete the form. Selecting Yes to the question asking to process through your merchant account, the script will automatically contact your merchant account, and capture the previously taken authorization on the credit card.

### **3.4 Pending Payment Methods**

When a customer updates their payment method for a recurring subscription through the Customer Area, the payment method might first be pending, depending on your settings. You can process all pending payment methods through the *Orders->Pending Payment Methods* menu. Simply select which payment methods to process, and select the status. If approved, the payment method will be automatically assigned to the needed recurring subscription, and will be charged from now on.

### **3.5 Process Refund**

You can process a refund through the *Orders->Process Refund* menu. This page contains a table, which lists all payments processed on this order. The first payment is always the initial payment, which the customer was charged when the order was first placed. All other payments are recurring payments for the order.

Simply select all payments you would like to refund, and complete the form. If this order was paid with by a credit card, through a real merchant account, there'll be a question asking if you would like to process the refund through the merchant account. If Yes, the shopping cart will contact your merchant account behind the scenes, and physically credit the customer's credit card for the needed amount.

### **3.6 Recurring Orders**

The Shopping Cart fully supports recurring orders, such as monthly payments for a service. When a new order is submitted, all recurring subscriptions will be created. Once every night a script will automatically run via Crontab, and process all needed recurring payments. If the customer is paying through a third party payment processor, such as PayPal or 2Checkout, the payment will be added to the database as pending. However, every time the member is charged by the third party payment processor, the next payment date for the recurring subscription is automatically updated accordingly, so the script may never add the recurring payment as pending.

## 4.) HTML Templates

This chapter gives a description of all HTML templates used by the Shopping Cart. If you do not already know the basics of how xFlow uses HTML templates, such as the special HTML tags used, please consult the Developer's Manual which can be found on the online documentation site at <http://docs.envex.com/>

### 4.1 HTML Template Descriptions

The below tables list the location and description of every HTML template used by the Shopping Cart module.

**Directory:** */cgi-bin/xflow/modules/shop/htmlt/*

| HTML Template          | Description  |
|------------------------|--|
| reminder.htmlt         | Displayed when a customer clicks on the <i>Forgot Password</i> link of the login form. Asks for the e-mail address, to e-mail the password to. |
| reminder_success.htmlt | Displayed after a customer submits the reminder.htmlt template, and their password has been e-mailed to them.                                  |

**Directory:** */cgi-bin/xflow/modules/shop/htmlt/carea/*

| HTML Template        | Description   |
|----------------------|---|
| edit_profile.htmlt   | Displayed when a customer goes to edit their profile through the Customer Area.                                       |
| login_expired.htmlt  | When a customer has been automatically logged out of the Customer Area, due to too many minutes of inactivity.        |
| login_inactive.htmlt | Displayed when a customer, who's account is inactive, tries to login to the Customer Area.                            |
| login_invalid.htmlt  | Displayed when a customer enters an invalid username or password when logging into the Customer Area.                 |
| logout.htmlt         | Small template displayed after a customer successfully logs out of the Customer Area.                                 |
| order_approved.htmlt | Displayed when a customer purchases a product through the Customer Area, and the payment is approved.                 |
| order_declined.htmlt | Displayed when a customer purchases a product through the Customer Area, and the payment is declined.                 |
| order_error.htmlt    | Displayed when a customer purchases a product through the Customer Area, and the payment processor returned an error. |

|                               |   |
|-------------------------------|---|
|                               |   |
| order_history.htmlt           | Shows a list of all orders submitted on the customer's account. Displayed when a customer views their order history.  |
| order_pending.htmlt           | Displayed when a customer purchases a product through the Customer Area, and the payment is pending.  |
| order_view.htmlt              | Displayed when a customer views information on an order submitted on their account. Shows order details, payment information, products purchased, etc.                                      |
| purchase.htmlt                | Displayed when a customer wants to purchase additional products through the Customer Area. Asks the customer which payment method they would like to use.                                   |
| subscriptions.htmlt           | When a customer views all recurring subscriptions on their account, through the Customer Area.  |
| subscriptions_manage.htmlt    | Displayed when a customer manages a recurring subscription on their account. Shows all details on the recurring subscription, and allows the customer to update the payment method charged. |
| support_reply_thankyou.htmlt  | Only applicable if you have the Support System module installed. Displayed after a customer submits a reply to a support request through the Customer Area.                                 |
| support_submit.htmlt          | Only applicable if you have the Support System module installed. Displayed when a customer goes to submit a new support request through the Customer Area.                                  |
| support_submit_thankyou.htmlt | Only applicable if you have the Support System module installed. Displayed after a member successfully submits a new support request through the Customer Area.                             |
| support_view.htmlt            | Displayed when a customer views / manages a support request on their account, through the Customer Area.  |
| support_view_all.htmlt        | Only applicable if you have the Support System module installed. Displayed when the customers views a list of all support requests on their account, including closed requests.             |
| support_view_open.htmlt       | Only applicable if you have the Support System module installed. Displayed when a customer views a list of all open support requests currently on their account.                            |
| view_payment.htmlt            | Displayed when a member views details of a payment, through the subscriptions_manage.htmlt template, when managing a recurring subscription.  |

**Directory:** */cgi-bin/xflow/modules/shop/htmlt/cart/*

| <b>HTML Template</b> | <b>Description</b>  |
|----------------------|---|
| add_success.htmlt    | Displayed after someone successfully adds a new product to their cart through the add_cart.cgi script.        |
| browse.htmlt         | Displayed when someone browses all categories and products through the browse.cgi script.                     |
| view.htmlt           | Displayed when someone views details on a specific product.   |
| view_cart.htmlt      | Displayed when someone views / manages the contents of their shopping cart, through the view_cart.cgi script. |

**Directory:** */cgi-bin/xflow/modules/shop/htmlt/order/*

| <b>HTML Template</b>    | <b>Description</b>  |
|-------------------------|---|
| order_approved.htmlt    | Displayed when someone submits a new order, which is approved.  |
| order_declined.htmlt    | Displayed when someone submits a new order, which is declined.  |
| order_error.htmlt       | Displayed when someone submits a new order, and the payment processor returned an error while processing payment.   |
| order_form_new.htmlt    | The order form displayed when a new customer submits an order. Asks for the customer's contact information, payment information, and so on.                                       |
| order_form_return.htmlt | The order form displayed when a returning customer submits an order. This template doesn't ask for the customer's contact information, but instead, only the payment information. |
| order_pending.htmlt     | Displayed when someone submits a new order, which has a status of pending.  |
| return_ask.htmlt        | Displayed when someone first checks out. Asks if the customer is a returning customer, and if so, the Customer ID#.   |
| return_found.htmlt      | Displayed if the Customer ID# specified in the return_ask.htmlt template exists in the database. Asks the returning customer to confirm their contact info.                       |
| return_notfound.htmlt   | Displayed if the Customer ID# specified in the return_ask.htmlt template does not exist in database. Asks the customer for a new Customer ID#, or to continue as a new customer.  |

# 5.) Integrate Third Party Scripts

This chapter is only meant for experienced Perl programmers, who have a need to integrate third party scripts into the Shopping Cart. Every time one of many actions occur, such as an order being approved, or a certain product being ordered, the Shopping Cart will check to see if a Perl script exists in the needed location. If one does, the Shopping Cart will automatically execute the script.

Please note, all third party scripts must be written in Perl. xFlow places no restrictions on the third party scripts, and executes them within an "eval {}" block, allowing for full usage of Perl.

**NOTE:** Envex Developments does not provide any technical support for any problems in creating or maintaining third party scripts.

## 5.1 File / Directory Structure

All third party scripts must be uploaded into the `/cgi-bin/xflow/modules/shop/scripts/` directory of your web server. Every time one of many actions occur, the Shopping Cart will check to see if a script exists at the needed location, and if exists, will automatically execute that script. The below table lists the exact location where all scripts must be placed, and the action which will execute them.

All locations listed in the below table are relative to the `/cgi-bin/xflow/modules/shop/scripts/` directory. For example, if the location is `"/customer/create.pl"`, the actual location of the script is `/cgi-bin/xflow/modules/shop/scripts/customer/create.pl`.

| Action               | Script Location                      | Description   |
|----------------------|--------------------------------------|---|
| New customer created | <code>/customer/create.pl</code>     | When a new customer is added to the database  |
| Customer activated   | <code>/customer/activate.pl</code>   | When a customer's account is activated  |
| Customer deactivated | <code>/customer/deactivate.pl</code> | When a customer's account is deactivated  |
| Customer expired     | <code>/customer/expired.pl</code>    | When a customer's account has expired.  |
| Order processed      | <code>/order/[STATUS].pl</code>      | When an order is submitted / processed. The <code>[STATUS]</code> represents the status of the order, and can be: <ul style="list-style-type: none"><li>• approved</li><li>• declined</li><li>• error</li><li>• pending</li><li>• refunded</li><li>• expired</li></ul> For example, <code>/order/approved.pl</code> would be executed every time an order |

|                   |                                   |   |
|-------------------|-----------------------------------|---|
|                   |                                   | is approved.  |
| Product purchased | /product/[PRODUCT_ID]_[STATUS].pl | <p>When a specific product is purchased. The [PRODUCT_ID] is the ID# of the product in your database, and the [STATUS] is the status of the order, and can be:</p> <ul style="list-style-type: none"> <li>• <b>approved</b></li> <li>• <b>declined</b></li> <li>• <b>error</b></li> <li>• <b>pending</b></li> <li>• <b>refunded</b></li> <li>• <b>expired</b></li> </ul> <p>For example, /product/CD427_approved.pl would be executed every time someone purchased the product ID# CD427, and the order was approved.</p> |

## 5.2 Available Variables

There are several variables available to the third party scripts, given different information such as the customer's profile, order information, and so on. Below explains all variables available:

### 5.2.1 %CUSTOMER Hash

The %CUSTOMER has is available to all third party scripts. The keys are the fields of your customer database, and the values are the customer's profile. For example, to get the customer's ID#, you would use `$CUSTOMER{customer_id}`, or to get the customer's first\_name, you would use `$CUSTOMER{first_name}`. Please note, all keys in the hash are lowercase.

### 5.2.2 %RESELLER Hash

The %RESELLER hash is available to all third party scripts, except those located in the /scripts/customer/ directory. This hash contains the reseller's profile, and is similar to the %CUSTOMER hash. The keys of the hash are the fields in your member database, and the values are the reseller's profile. For example, `$RESELLER{username}`, and `$RESELLER{email}`, would give you the reseller's username and e-mail.

### 5.2.3 %ORDER Hash

The %ORDER hash is available to all third party scripts, except those located in the /scripts/customer/ directory. This hash contains information on the order. The following variables are contained within the %ORDER hash:

| <b>Variable</b>           | <b>Name</b>          | <b>Description</b>  |
|---------------------------|----------------------|---|
| \$ORDER{'order_id'}       | Order ID#            | The ID# of the order being processed  |
| \$ORDER{'transaction_id'} | Transaction ID#      | The transaction ID# of the payment being processed.                                       |
| \$ORDER{'amount'}         | Amount               | The total amount being charged  |
| \$ORDER{'coupon'}         | Coupon Amount        | If a coupon code was used, the amount deducted from the order. This amount is positive.   |
| \$ORDER{'tax'}            | Tax Amount           | The total amount of tax added to the order.   |
| \$ORDER{'merchant_ref'}   | Merchant Reference # | The reference number assigned by your merchant account / payment processor for the order. |
| \$ORDER{'merchant_text'}  | Merchant Text        | A small text message given off by the merchant account / payment processor.               |
| \$ORDER{'ip_address'}     | IP Address           | The IP address used to submit the order.  |
| \$ORDER{'payment_date'}   | Payment Date         | The date the payment was processed. Formatted in YYYY-MM-DD                               |
| \$ORDER{'payment_time'}   | Payment Time         | The exact time the payment was processed. Formatted in HH:MM:SS                           |

If the third party script is located in the /product/ directory, and is executed when a specific product is purchased, the following variables are also available:

| <b>Variable</b>         | <b>Name</b>  | <b>Description</b>   |
|-------------------------|--------------|--|
| \$ORDER{'product_id'}   | Product ID#  | The ID# of the product being purchased.  |
| \$ORDER{'product_name'} | Product Name | The name of the product being purchased.   |
| \$ORDER{'quantity'}     | Quantity     | The quantity of the product being purchased.   |
| \$ORDER{'unit_amount'}  | Unit Amount  | The amount per unit (quantity) that was charged.   |
| \$ORDER{'total_amount'} | Total Amount | The total amount charged for this product. Basically, this is the quantity and unit amount multiplied. |

### 5.3 Important Notes

- To ensure no server errors occur, make sure all third party scripts have **1**; at the bottom.