

# Ayopa Module

## Installation

Clear the store cache under var/cache and all cookies for your store domain. Disable compilation for Magento 1.4+. This step eliminates almost all potential problems. It's necessary since Magento uses cache heavily.

### Backup Your Data

Backup your store database and web directory.

### Download and Extract

Download and unzip extension contents on your computer and navigate inside the extracted folder.

### Step 1 - Upload Files

Navigate inside *step\_1* directory. Using your FTP client upload content of *step\_1* directory to your store root. This will not overwrite your app directory, it will add the new files to this directory.

### Edit shipping\_method.phtml Template File

In your web directory, check your theme for /app/design/frontend/*yourtheme*/template/checkout/onepage/shipping\_method.phtml. If this file does not exist in your theme, navigate to app/design/frontend/base/template/checkout/onepage.

Edit shipping\_method.phtml as indicated:

Change this code:

```
<button type="button" class="button" onclick="shippingMethod.save();">
```

To this:

```
<button type="button" class="button" onclick="ayopaSubmit();">
```

### Edit text.phtml Template File

In your web directory, check your theme for /app/design/frontend/*yourtheme*/template/catalog/product/view/options/type/text.phtml. If this file does not exist, navigate to app/design/frontend/base/template/catalog/product/view/options/type.

Edit text.phtml as indicated:

Change this code:

```
<dt><label<?php if ($_option->getIsRequire()) echo ' class="required"' ?><?php  
if ($_option->getIsRequire()) echo '<em>*</em>' ?><?php echo $this->  
htmlEscape($_option->getTitle()) ?></label>  
    <?php echo $this->getFormattedPrice() ?></dt>
```

To this:

```
<?php if ($_option->getTitle() != 'GroupBuyNSave'): ?>  
<dt><label<?php if ($_option->getIsRequire()) echo ' class="required"' ?><?php  
if ($_option->getIsRequire()) echo '<em>*</em>' ?><?php echo $this->  
htmlEscape($_option->getTitle()) ?></label>  
    <?php echo $this->getFormattedPrice() ?></dt>  
<?php endif; ?>
```

This will hide the Ayopa label in the Custom Options display.

### Edit default.phtml file (Optional)

This change gives your customer's an explanation of the GroupBuyNSave item in their cart. In your web directory, check your theme for `/app/design/frontend/yourtheme/template/checkout/cart/item/default.phtml`. If this file does not exist, navigate to `app/design/frontend/base/template/checkout/cart/item/`.

Edit text.phtml as indicated:

Change this code:

```
<dt><?php echo $this->htmlEscape($_option['label']) ?></dt>  
<dd><?php if (isset($_formattedOptionValue['full_view'])): ?>  
class="truncated"<?php endif; ?><?php echo $_formattedOptionValue['value'] ?>
```

To this:

```
<?php if($_option['label'] == "GroupBuyNSave"): ?>  
    <dl class="item-options">  
        <dt>GroupBuyNSave</dt>  
        <dd>You are buying at the published price but will be rebated the  
        difference via paypal when the GroupBuy ends. New users will  
        receive an email showing how to easily track this purchase and  
        rebate via facebook. The current GroupBuy price is <?php echo  
        Mage::helper('ayopamodule')->getCurrentPrice($_option['value']);  
        ?>. Proceed with your checkout as usual and you will be prompted  
        to register for Ayopa GroupBuy if you are not already  
        registered.</dd>  
    </dl>  
<?php else: ?>  
    <dt><?php echo $this->htmlEscape($_option['label']) ?></dt>  
    <dd><?php if (isset($_formattedOptionValue['full_view'])): ?>  
        class="truncated"<?php endif; ?><?php echo  
        $_formattedOptionValue['value'] ?>  
<?php endif; ?>
```

## Step 2 – Upload or Modify local.xml File

Navigate inside *step\_2* directory in the extracted module file. If you use a different from default theme - be sure to rename *step\_2/app/design/frontend/default/default* folder to your store's values. If you already have a *local.xml* file in your */app/design/frontend/default/default* folder, copy the contents of *local.xml* between the *<layout>* tags of the module to your *local.xml* file and save the file to your store. Otherwise, use your FTP client upload content of *step\_2* directory to your store root.

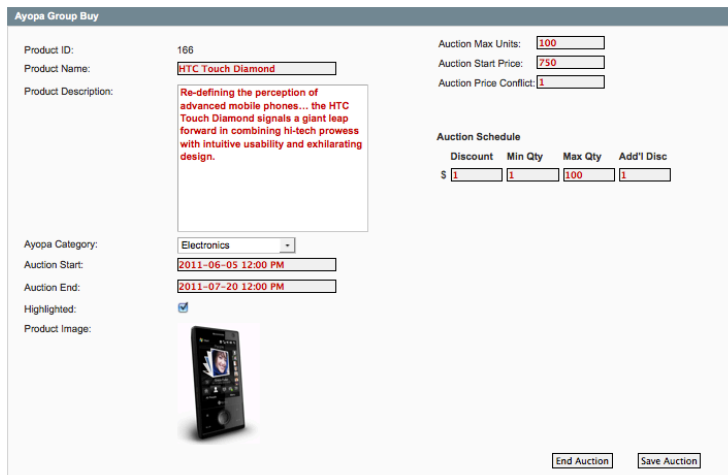
# Configuration

## Enter Ayopa Credentials

In the admin, go to System → Configuration. Under Ayopa Settings, click Configuration. Enter your Ayopa username (email) and password. Save Config.

## Create GroupBuy for Product

In the admin, go to Catalog → Manage Products → Edit a product → Ayopa. Click the New Auction button. Enter the auction information. Click Save Auction.



The screenshot shows the 'Ayopa Group Buy' configuration form. It includes fields for Product ID (166), Product Name (HTC Touch Diamond), and Product Description (Re-defining the perception of advanced mobile phones...). The Auction Max Units is set to 100, Auction Start Price is 750, and Auction Price Conflict is 1. The Auction Schedule table shows a discount of \$1, a minimum quantity of 1, and a maximum quantity of 100. The Ayopa Category is set to Electronics, the Auction Start is 2011-06-05 12:00 PM, and the Auction End is 2011-07-20 12:00 PM. The Highlighted checkbox is checked. The Product Image is a small image of the HTC Touch Diamond phone. At the bottom right, there are 'End Auction' and 'Save Auction' buttons.

Discount	Min Qty	Max Qty	Add'l Disc
\$ 1	1	100	1