<MAGENTO/>

A practical userguide to integrate Magento ecommerce and Stanley/Stella API

Prerequisites

This guide will explain how to connect a Magento 2.4 webshop with Stanley/Stella API Framework to periodically synchronize the product referential.

The guide assumes that you are familiar with Magento and that you already have your Magento running.

It also requires a little bit of development knowledge in order to transform the CSV files.

In this guide, we will illustrate the integration in the PHP development environment but this could also be realized in any other modern development environment.

Please note that the PHP code provided in this document is for illustration purpose only.

Further technical explanation on the API can be found at https://api.stanleystella.com

The demo website build with these instructions can be found at https://testmagento.stanleystella.com/

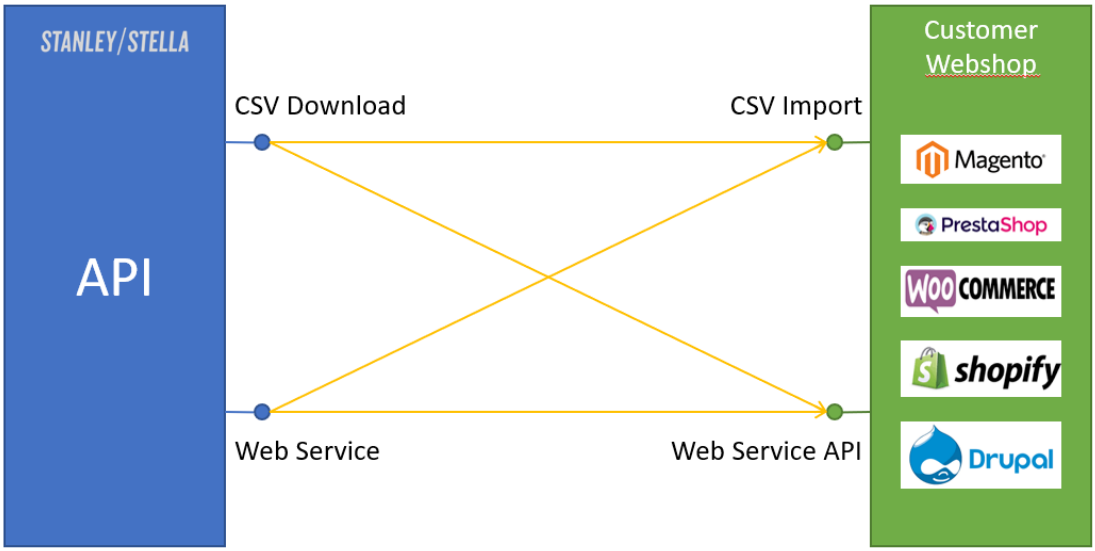
Table of Contents

Integration scenario Stanley/Stella API → Magento

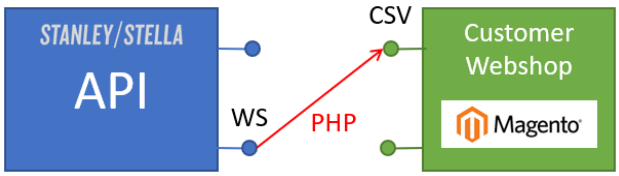
Like a lot of ecommerce solutions, Magento provides 2 ways to import product information:

- Manual import via CSV file (can be scheduled)
- Programmatic import via REST API or SOAP API

At Stanley/Stella, our API provides these 2 capabilities as well.



In this scenario, we chose to use the REST API on Stanley/Stella side and the CSV import on Magento side.



The integration logic in the middle has been developed in PHP.

Connect to Stanley/Stella Product API in PHP

The first step of the integration is to call the Stanley/Stella Product REST API from the PHP code.

The standard output of the Stanley/Stella API is a “flat” list of product variants (SKU’s).

In our PHP code, we added the grouping of the variants at style level and the sorting of the variants by colors and sizes.

For simplicity, we only download the product data in 1 language (English).

See the following PHP code as inspiration. Feel free to extend this code with additional data coming from the API.

```php
<?php

function getAllSTSTProducts() {
    $url = "https://api.stanleystella.com/webrequest/products/get_json";
    $jsonData = array(
```

```
        'jsonrpc' => '2.0',
        'method' => 'call',
        'params' => array(
            'db_name' => "production_api",
            'password' => "Test0220",
            'user' => "test@stanleystella.com",
            "LanguageCode" => "en_US"
        ),
        'id' => 0
    );

    $ch = curl_init($url);
    $jsonDataEncoded = json_encode($jsonData);

    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonDataEncoded);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
    $result = curl_exec($ch);

    $jsonDataDecoded = json_decode(json_decode($result)->result, true);
    curl_close($ch);
    // Sort the data by style, color and size
    foreach ($jsonDataDecoded as $key => $row) {
        $styleorder[$key] = $row['SequenceStyle'];
        $sizeorder[$key]  = $row['SizeSequence'];
        $colororder[$key] = $row['ColorSequence'];
    }
    array multisort($styleorder, SORT ASC, $colororder, SORT ASC, $sizeorder, SORT ASC,
$jsonDataDecoded);

    return $jsonDataDecoded;
}

function getAllSTSTProductsGrouped() {
    $stpm = getAllSTSTProducts();

    $allststyles = array();
    foreach ($stpm as $key => $variant) {
        $stylecode = $variant["StyleCode"];
        if (!isset($allststyles[$stylecode])) {
            $allststyles[$stylecode] = array();
            $allststyles[$stylecode]["StyleCode"] = $variant["StyleCode"];
            $allststyles[$stylecode]["StyleName"] = $variant["StyleName"];
            $allststyles[$stylecode]["Type"] = $variant["Type"];
            $allststyles[$stylecode]["Category"] = $variant["Category"];
            $allststyles[$stylecode]["Gender"] = $variant["Gender"];
            $allststyles[$stylecode]["Fit"] = $variant["Fit"];
            $allststyles[$stylecode]["Neckline"] = $variant["Neckline"];
            $allststyles[$stylecode]["Sleeve"] = $variant["Sleeve"];
            $allststyles[$stylecode]["ShortDescription"] = $variant["ShortDescription"];
            $allststyles[$stylecode]["LongDescription"] = $variant["LongDescription"];
            $allststyles[$stylecode]["SequenceStyle"] = $variant["SequenceStyle"];
        }

        if (!isset($allststyles[$stylecode]["variants"])) $allststyles[$stylecode]["variants"]
= array();
        $var = array();
        $var["B2BSKUREF"] = $variant["B2BSKUREF"];
        $var["ColorCode"] = $variant["ColorCode"];
        $var["Color"] = $variant["Color"];
        $var["ColorSequence"] = $variant["ColorSequence"];
        $var["SizeCode"] = $variant["SizeCode"];
        $var["SizeCodeNavision"] = $variant["SizeCodeNavision"];
        $var["SizeSequence"] = $variant["SizeSequence"];
        $var["Stock"] = $variant["Stock"];
        $var["Price<10 EUR"] = $variant["Price<10 EUR"];
        $var["SKU_Start_Date"] = $variant["SKU_Start_Date"];
        $var["Published"] = $variant["Published"];
        $allststyles[$stylecode]["variants"][$variant["B2BSKUREF"]] = $var;

    }

    return($allststyles);
```

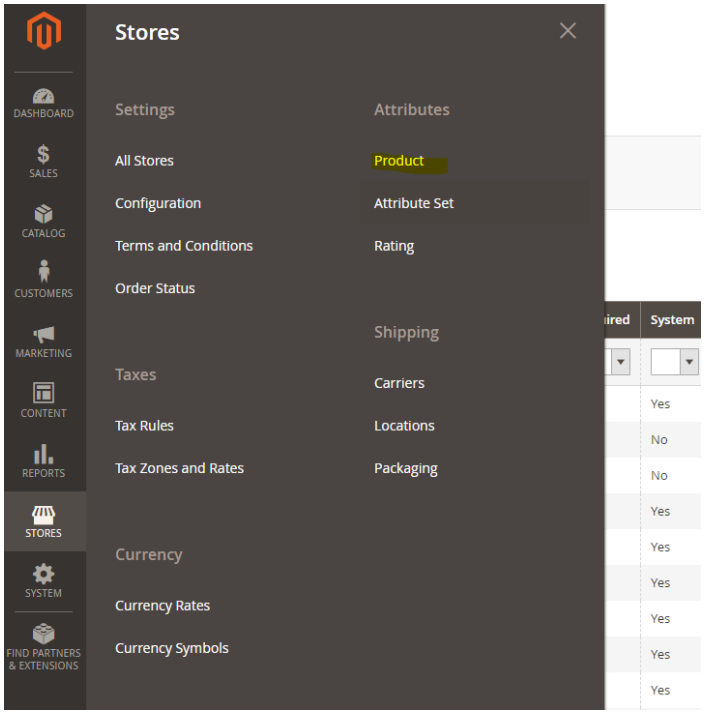
```
}
```

Import CSV's in Magento

Before we start: Define a new attribute for sizes

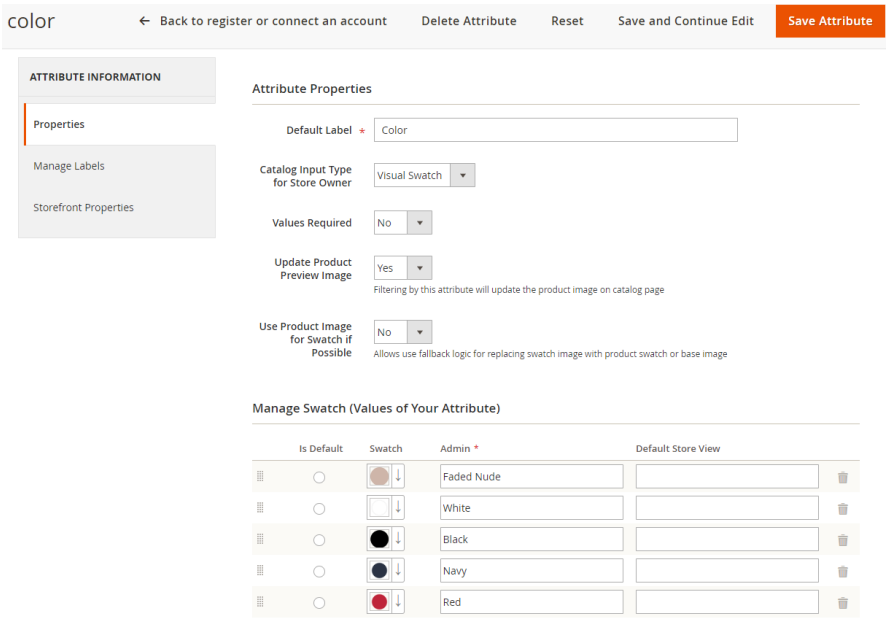
Magento uses attributes to describe the values of variants dimensions like colors and sizes.

The product attributes can be found under "Stores" and then "Product".



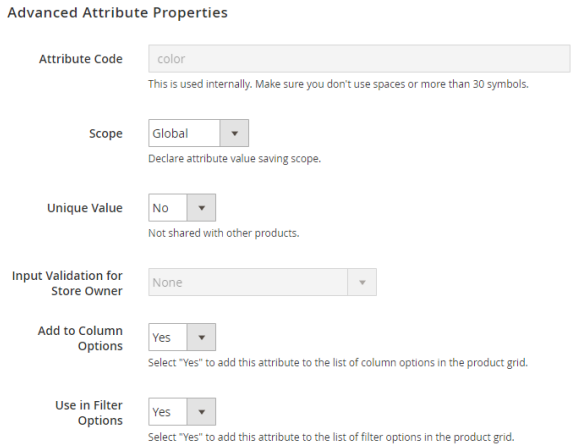
Colors

In a standard Magento, the attribute color is already defined. But there are no values. We did not find a way to import values for attributes in Magento. So we defined these values by hand, one by one.



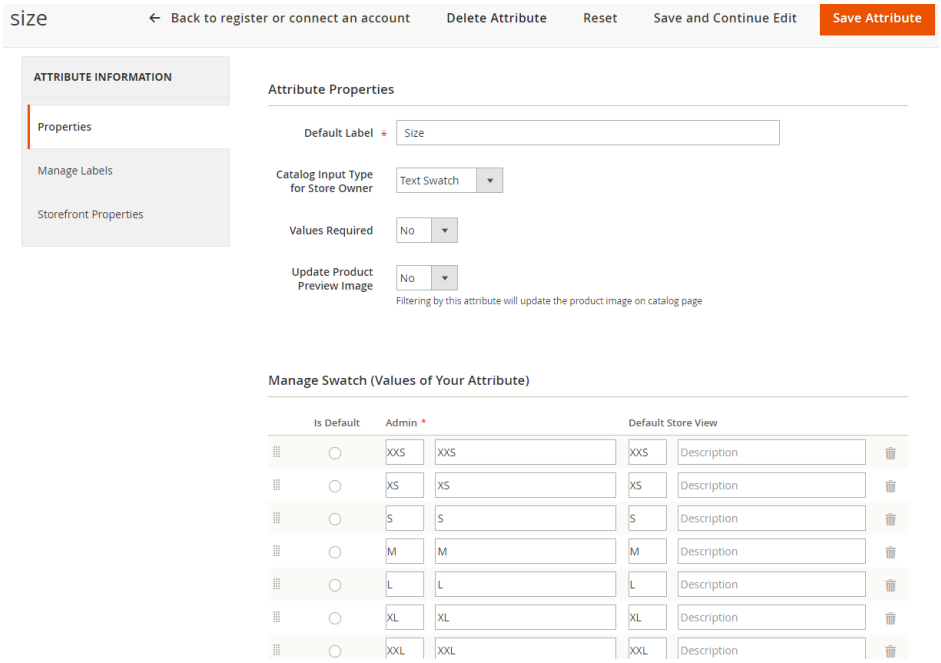
As inspiration, we selected the following settings. This can be different for your own shop.

- Visual swatch
- Values required: No
- Update product preview image: Yes
- Use product image for swatch if possible: No
- Make sure to check the settings under “Advanced Attribute Properties”
    - Scope should be “global”



Sizes

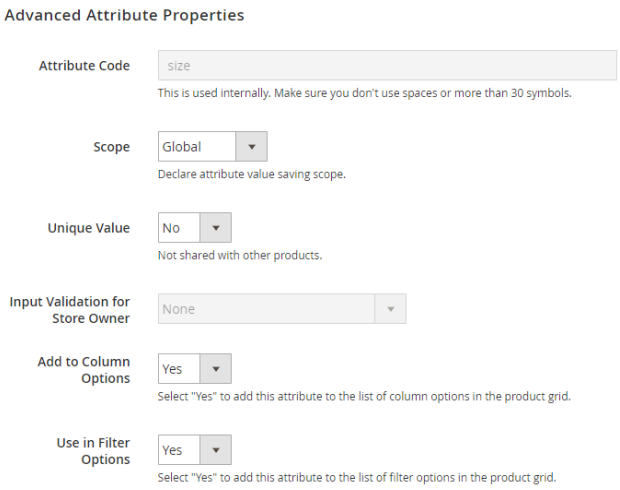
In standard Magento, there is no attribute for sizes. We created a new attribute for the sizes.



For sizes, we selected the following settings:

- Text swatch
- Values required: No
- Update product preview image: No

Make sure to check the Advanced Attribute Properties. The size attribute should be defined as “Global”.

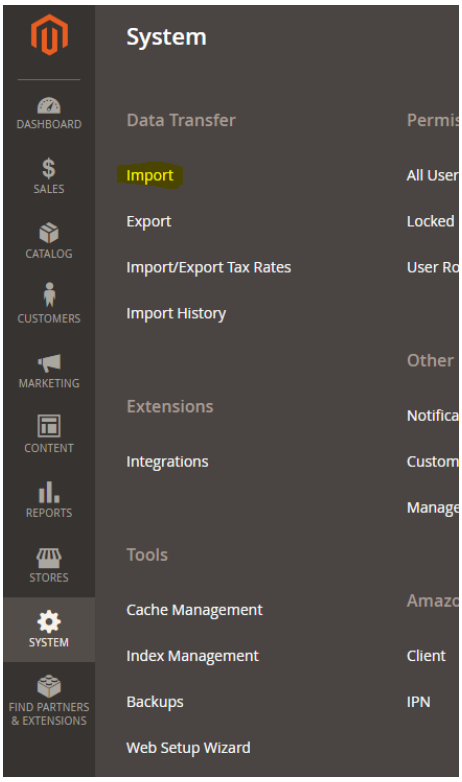


How to import CSV’s in Magento

Magento offers a CSV import functionality to import:

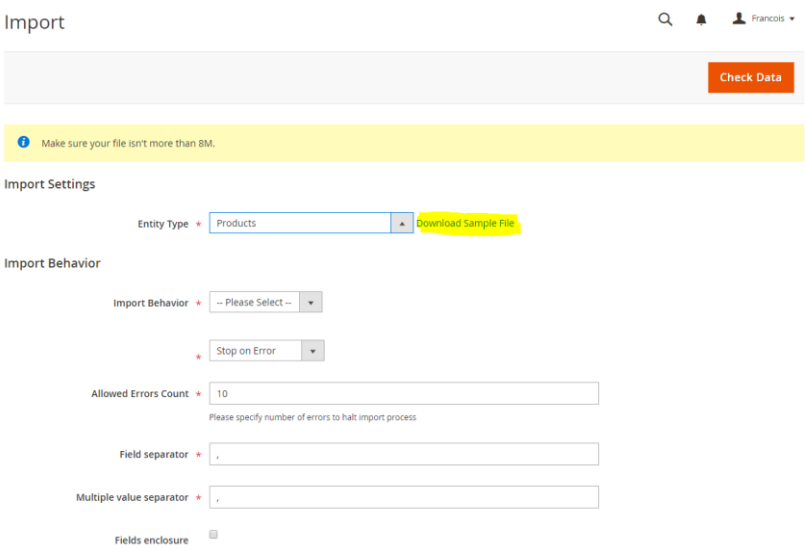
- Products
- Advanced pricing
- Customer data

The import functionality can be found under “System” and Import.



We will mainly use 1 CSV file: a Product import file

It is possible to download sample CSV files for each of these objects.



The Product CSV import file will automatically create the categories and the genders but not the colors and sizes, unfortunately.

In the import form, make sure to select the right settings:

- Import behaviour: Add/Update
- Field separator: We use the semi-colon ‘;’.
- Multiple value separator: We use the comma ‘,’.

Click on “Check Data”. Magento will tell you if there are errors in your CSV file.

Note:

We observed some limitations with the length of the “picture” field. Magento lets you import pictures on products and variants via the same CSV using URL’s of the pictures. Magento then follows the URL’s and downloads the pictures. Multiple pictures can be imported by separating them with a comma ‘,’ inside the corresponding column. But apparently, this field has a maximum length and it was not possible to import all pictures per specific variant when there were more than 5 pictures for the variant.

Generating the right CSV import files

Creating the colors and the sizes

As described earlier, it is needed to create the color and the size values in Magento first. Fortunately, the CSV import can work with the text value of colors and sizes and does not require to work with internal Magento code.

Generating the Product CSV import file

The Product CSV import file contains both lines for the styles as for the variants. So, we first call the Stanley/Stella Product API, with grouped variants per styles. We then loop on all styles and variants.

- The first column contains the sku. We choose to use the style code as sku for the styles and the Stanley/Stella standard sku for the variants.
- Magento works with different types of products. In this scenario, we work with simple product for the variants and configurable products for the styles. We then link the simple products to their corresponding configurable product.
- We do not define prices on the styles, only on the variants.
- For the variants, we have to define the combination of attributes (column additional_attributes). We use there the values that we defined earlier for the color and size attributes.
- For the style, we have to define the related variants (column configurable_variations) by identifying them by their sku, color and size.

See the example of PHP code hereunder for more details.

Next step: Automation

There are still some issues to be solved if we want to automate this.

- Magento provides a possibility to “listen” to a folder and automatically import CSV files that are dropped in this folder.
- We would need to find a way to create new colors and new sizes on the fly when importing the CSV. This issue prevent us momentarily to fully automate the import process of products into Magento.

PHP Code

See the following PHP code as inspiration for generating the CSV files.

```php
<?php
ini_set("memory_limit","128M");
ini_set("max_execution_time",9000);
ini_set('default_charset', 'utf-8');
header('Content-Type: text');

$stpm = getAllSTSTProductsGrouped();

$fp = fopen('magento_product_import.csv', 'w');
addMagentoCSVHeader($fp);

foreach ($stpm as $stylecode => $style) {
    $stylename = $style["StyleName"];
    $gender = $style["Gender"];
    $type = $style["Type"];
    $category = $style["Category"];
    $shortdescription = $style["ShortDescription"];
    $longdescription = str_replace("\n", "<br/>", $style["LongDescription"]);
    $fit = $style["Fit"];
    $neckline = $style["Neckline"];
    $sleeve = $style["Sleeve"];
    $categorystring = "Default Category/".$gender.",Default Category/".$gender."/".$type;
    // Get images for current style
    $picurls = array();
    $mainpic = "";
    $pics = getSTSTProductImages($stylecode);
    foreach ($pics as $key => $pic) {
        $colorcode = $pic["ColorCode"];
        $picname = $pic["FName"];
        $picurl = $pic["HTMLPath"];

        if (!isset($picurls[$colorcode])) $picurls[$colorcode] = array();
        $picurls[$colorcode][] = $picname;

        if (substr($picname,0,3)=="SFM" && $mainpic=="") {
            $mainpic = $picname;
        }
    }

    $variationstring = "";
    $first = true;
    foreach ($style["variants"] as $key => $var) {
        if (!$first) {
            $variationstring.="|";
        }
        $sku = $var["B2BSKUREF"];
        $colorcode = $var["ColorCode"];
        $colorname = $var["Color"];
        $sizecode = $var["SizeCodeNavision"];
        $sizename = $var["SizeCode"];

        $variationstring.="sku=".$sku;
        $variationstring.=",color=".$colorname.",size=".$sizename;

        $first = false;
    }

    $pictures = array();
    $pictures[] = $mainpic;
    addMagentoCSVLine($fp, $stylecode, "configurable", $categorystring, $stylename,
$longdescription, $shortdescription, "", "Catalog, Search", "", "", $variationstring,
"color=Color,size=Size", $pictures);

    // Generate lines for variants
    foreach ($style["variants"] as $key => $var) {
        $sku = $var["B2BSKUREF"];
        $colorcode = $var["ColorCode"];
        $colorname = $var["Color"];
```

```
        $sizecode = $var["SizeCodeNavision"];
        $sizename = $var["SizeCode"];
        $price = $var["Price<10 EUR"];
        $stock = $var["Stock"];

        $pictures = array();
        if (isset($picurls[$colorcode])) {
            $pictures = $picurls[$colorcode];
        }
        addMagentoCSVLine($fp, $sku, "simple", "", $stylename." ".$colorname." ".$sizename, "",
"", $price, "Not Visible Individually", "color=".$colorname.",size=".$sizename, $stock, "", "",
$pictures);

    }

}

fclose($fp);


function addMagentoCSVLine($fp, $sku, $prod_type, $categories, $name, $long_desc, $short_desc,
$price, $visible, $attributes, $qty, $variations, $variation_labels, $pictures) {
    $csvrow = array();
    $csvrow[] = $sku;
    $csvrow[] = "";
    $csvrow[] = "Default";
    $csvrow[] = $prod_type;
    $csvrow[] = $categories;
    $csvrow[] = "base";
    $csvrow[] = $name;
    $csvrow[] = $long_desc;
    $csvrow[] = $short desc;
    $csvrow[] = "";
    $csvrow[] = "1";
    $csvrow[] = "None";
    $csvrow[] = $visible;
    $csvrow[] = $price;
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = strtolower(str_replace(" ", "-", $name));
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $base = "";
    $small = "";
    $first = true;
    $piccount = 0;
    $maxpics = 2;
    foreach ($pictures as $key => $pic) {
        if ($pic!="") {
            if (!$first) {
                $base.=",";
                $small.=",";
            }
            $base.="https://api.stanleystella.com/Pictures/".$pic;
            $small.="https://api.stanleystella.com/Pictures-LD/".$pic;
            $first = false;
            $piccount++;
            if($piccount>$maxpics) break;
        }
    }
    $csvrow[] = $base; // Base Image
    $csvrow[] = "";
    $csvrow[] = $small; // Small Image
    $csvrow[] = "";
    $csvrow[] = $small; // Thumbnail Image
    $csvrow[] = "";
    $csvrow[] = ""; // Swatch Image
    $csvrow[] = "";
    $csvrow[] = date("Y-m-d H:i:s");
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "Block after Info Column";
    $csvrow[] = "";
    $csvrow[] = "";
```

```
    $csvrow[] = "";
    $csvrow[] = "Use config";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = $attributes;
    $csvrow[] = $qty;
    $csvrow[] = "0";
    $csvrow[] = "1";
    $csvrow[] = "0";
    $csvrow[] = "0";
    $csvrow[] = "1";
    $csvrow[] = "0";
    $csvrow[] = "0";
    $csvrow[] = "100000000";
    $csvrow[] = "1";
    $csvrow[] = "1";
    $csvrow[] = "10";
    $csvrow[] = "1";
    $csvrow[] = "1";
    $csvrow[] = "1";
    $csvrow[] = "1";
    $csvrow[] = "1";
    $csvrow[] = "1";
    $csvrow[] = "0";
    $csvrow[] = "0";
    $csvrow[] = "0"; // Website id
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = $variations;
    $csvrow[] = $variation_labels;
    $csvrow[] = "";
    fputcsv($fp, $csvrow, ";");
}

function addMagentoCSVHeader($fp) {
    // CSV Header
    $csvrow = array();
    $csvrow[] = "sku";
    $csvrow[] = "store_view_code";
    $csvrow[] = "attribute_set_code";
    $csvrow[] = "product_type";
    $csvrow[] = "categories";
    $csvrow[] = "product_websites";
    $csvrow[] = "name";
    $csvrow[] = "description";
    $csvrow[] = "short_description";
    $csvrow[] = "weight";
    $csvrow[] = "product_online";
    $csvrow[] = "tax_class_name";
    $csvrow[] = "visibility";
    $csvrow[] = "price";
    $csvrow[] = "special_price";
    $csvrow[] = "special_price_from_date";
    $csvrow[] = "special_price_to_date";
    $csvrow[] = "url_key";
    $csvrow[] = "meta_title";
    $csvrow[] = "meta_keywords";
```

```
    $csvrow[] = "meta_description";
    $csvrow[] = "base_image";
    $csvrow[] = "base_image_label";
    $csvrow[] = "small_image";
    $csvrow[] = "small_image_label";
    $csvrow[] = "thumbnail_image";
    $csvrow[] = "thumbnail_image_label";
    $csvrow[] = "swatch_image";
    $csvrow[] = "swatch_image_label";
    $csvrow[] = "created_at";
    $csvrow[] = "updated_at";
    $csvrow[] = "new_from_date";
    $csvrow[] = "new_to_date";
    $csvrow[] = "display_product_options_in";
    $csvrow[] = "map_price";
    $csvrow[] = "msrp_price";
    $csvrow[] = "map_enabled";
    $csvrow[] = "gift_message_available";
    $csvrow[] = "custom_design";
    $csvrow[] = "custom_design_from";
    $csvrow[] = "custom_design_to";
    $csvrow[] = "custom_layout_update";
    $csvrow[] = "page_layout";
    $csvrow[] = "product_options_container";
    $csvrow[] = "msrp_display_actual_price_type";
    $csvrow[] = "country of manufacture";
    $csvrow[] = "additional_attributes";
    $csvrow[] = "qty";
    $csvrow[] = "out_of_stock_qty";
    $csvrow[] = "use_config_min_qty";
    $csvrow[] = "is_qty_decimal";
    $csvrow[] = "allow backorders";
    $csvrow[] = "use_config_backorders";
    $csvrow[] = "min_cart_qty";
    $csvrow[] = "use_config_min_sale_qty";
    $csvrow[] = "max_cart_qty";
    $csvrow[] = "use_config_max_sale_qty";
    $csvrow[] = "is in stock";
    $csvrow[] = "notify_on_stock_below";
    $csvrow[] = "use_config_notify_stock_qty";
    $csvrow[] = "manage_stock";
    $csvrow[] = "use_config_manage_stock";
    $csvrow[] = "use_config_qty_increments";
    $csvrow[] = "qty_increments";
    $csvrow[] = "use_config_enable_qty_inc";
    $csvrow[] = "enable_qty_increments";
    $csvrow[] = "is_decimal_divided";
    $csvrow[] = "website_id";
    $csvrow[] = "related_skus";
    $csvrow[] = "related_position";
    $csvrow[] = "crosssell_skus";
    $csvrow[] = "crosssell_position";
    $csvrow[] = "upsell_skus";
    $csvrow[] = "upsell_position";
    $csvrow[] = "additional_images";
    $csvrow[] = "additional_image_labels";
    $csvrow[] = "hide_from_product_page";
    $csvrow[] = "custom_options";
    $csvrow[] = "bundle_price_type";
    $csvrow[] = "bundle_sku_type";
    $csvrow[] = "bundle_price_view";
    $csvrow[] = "bundle_weight_type";
    $csvrow[] = "bundle_values";
    $csvrow[] = "bundle_shipment_type";
    $csvrow[] = "configurable_variations";
    $csvrow[] = "configurable_variation_labels";
    $csvrow[] = "associated_skus";
    fputcsv($fp, $csvrow, ";");
}
```