Bazaarvoice

Bazaarvoice for Magento Extension Implementation Guide v6.1.2.3

Version 6.1.2.3

Bazaarvoice Inc.
8/5/2015

Introduction

Bazaarvoice maintains a pre-built integration into the Magento platform. This pre-built integration is delivered in the form of a Magento Extension.

The extension is supplied “as-is” and is intended to function as a starter kit for clients to implement Bazaarvoice. It includes support for commonly required integration points, but does not include pre-built functionality for the exhaustive list of Bazaarvoice features. This extension may need to be customized by the client in order to implement functionality not included in the extension

System Requirements

http://www.magentocommerce.com/system-requirements

- Supported Operating Systems:
  - Linux x86, x86-64
- Supported Web Servers:
  - Apache 1.3.x
  - Apache 2.0.x
  - Apache 2.2.x
  - Nginx (starting from Magento 1.7 Community and 1.12 Enterprise versions)
- Supported Browsers:
  - Microsoft Internet Explorer 7 and above
  - Mozilla Firefox 3.5 and above
  - Apple Safari 5 and above on Mac only
  - Google Chrome 7 and above
  - Adobe Flash browser plug-in should be installed
- PHP Compatibility:
  - 5.2.13 - 5.3.24
- Required extensions:
  - PDO_MySQL
  - simplexml
  - mcrypt
  - hash
  - GD
  - DOM
  - iconv
  - curl
  - SOAP (if Webservices API is to be used)
- Safe_mode off
- Memory_limit no less than 256Mb (preferably 512)
- MySQL:
  - Magento Enterprise Edition 1.13.0.0 or later: MySQL 5.0.2 or newer
  - Magento Enterprise Edition 1.12.0.2 or earlier: MySQL 4.1.20 or newer
  - Magento Community Edition (all versions): MySQL 4.1.20 or newer
- SSL:

    - If HTTPS is used to work in the admin, SSL certificate should be valid. Self-signed SSL certificates are not supported
- Server - hosting - setup:
    - Ability to run scheduled jobs (crontab) with PHP 5
    - Ability to override options in .htaccess files

Prepare for Installation

In preparation for installation there are several steps that you should take.

1. Ensure you are working on a "Development/Staging" environment. It is not recommended that you install this or any extension directly in your production environment. You will need to take several steps to ensure the functionality contained in this extension has been properly integrated into your theme(s). This will take time and testing GURANTEED!
2. During installation make sure you are the only person making changes to this environment. This will ensure any errors that occur during this process are isolated to the work that you are doing, and not potentially the result of someone else's actions.
3. Ensure you have backups of your code, and database, if you are using source control you can always revert back, but it is more difficult to revert changes to the database.
4. Check to make sure compilation mode is not enabled, and that Magento is not currently compiled. To do this Navigate to System > Tools > Compilation. Confirm that the Compiler Status = Disabled, and Compilation State = Not Compiled. If that is not the case login to the server with SSH, and perform the following commands
   cd /path/to/magento
   php shell/compiler.php disable
   php shell/compiler.php clear

Get a Copy of the Extension Package

Contact Bazaarvoice for the latest copy of the Bazaarvoice for Magento extension.

Install the Extension Package

Now that we are properly prepared we can proceed to install the extension.

1. After logging into your Magento Admin panel Navigate to System > Magento Connect > Magento Conenct Manager, and login with your Magento Admin credentials
2. Install the package using the Direct package file upload step by choosing your package, and using the upload button. The package should be installed & caches refreshed.

The package should install without issue.

Complete the following steps before proceeding to configuration.

1. Navigate to System > Cache Management and
    a. Flush Magento Cache
    b. Flush Cache Storage
    c. Select All Cache Types, and click Submit to Refresh all types
2. Log Out of the Admin panel
3. Log back into the Magento Admin Panel.

Additional Steps When Using a Custom Design:
The extension installs frontend design files into the **enterprise** design package. If you are using a different package for your storefront, please make sure to copy the following files and directories into the default theme of the package you are using:

```
~/app/design/frontend/enterprise/default/layout/bazaarvoice.xml
~/app/design/frontend/enterprise/default/template/bazaarvoice/
```

Prepare for Configuration

Yes that's right, you need to prepare for configuration too! By this time you have probably already discussed the layout of your Magento install, and how that converts to Bazaarvoice Instances, and locales.

Refer to *Appendix A: Mapping your Magento Environment to your Bazaarvoice Environment* for more details.

General Configuration Setup

With the information you obtained from Bazaarvoice it is time to start configuring the extension.

1. Log in to the Magento Admin
2. Navigate to System > Configuration > General > Bazaarvoice
3. General Configuration:
    - **a.** Set **Enable Bazaarvoice Extension** to **Yes**
    - b. Select the desired Bazaarvoice **Environment** (Staging or Production)
    - c. Set the **Client Name** (the name defined in the Bazaarvoice platform for the client instance; typically referred to as "client folder name")
    - **d.** Set the **FTP Password**
    - e. Set the **Deployment Zone** (default is "Main Site")
    - f. Set the **Locale** (e.g en_US, fr_CA, etc.; language-only values are not supported by Bazaarvoice)

See the Appendix: Magento Environment Mapping to understand how to configure these values when multiple Websites, Store (Groups), and/or Store Views are involved.

Ratings and Reviews

Disable Native Magento Ratings & Reviews Display Functionality

If the Magento native reviews module is integrated with your storefront, disable the display before configuring Bazaarvoice Ratings & Reviews.

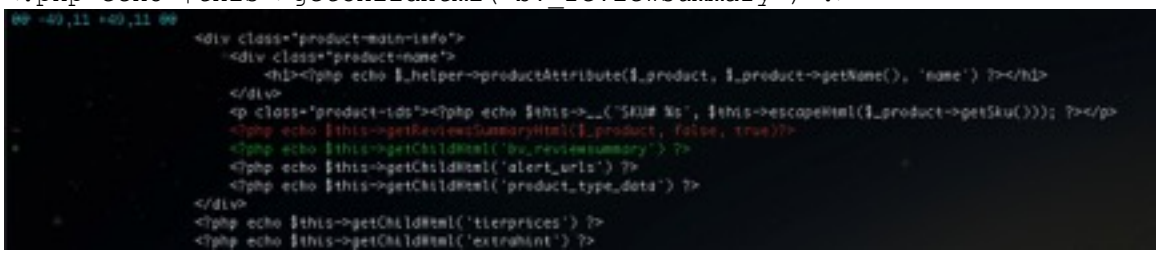
1. Log in to the Magento **Admin Panel**.
2. From the row of menu tabs, choose **System** → **Configuration**.
   The **General** page opens.
3. In the **Advanced** section of the navigation panel, click **Advanced**.
   The **Advanced** page opens.
4. Select `Disable` from the **Mage_Rating** list box.
5. Select `Disable` from the **Mage_Review** list box.
6. Click **Save Config**.
7. Clear your cached Magento pages.

Refer to your Magento documentation for more information on clearing your cached pages.
Note: The above steps will only disable the display of native Magento ratings & reviews functionality. The native Magento reviews module will continue to be enabled, but will not adversely affect the Bazaarvoice functionality.

Adding Ratings & Reviews to the Product Page

1. Ensure that **Enable Ratings & Reviews** is checked in the Magento control panel.
2. Open the following file in your Magneto installation: `app/design/frontend/[your_interface]/[your_theme]/template/catalog/product/view.phtml`
3. Insert the following code where you want the Ratings Summary to appear:

```
<?php echo $this->getChildHtml('bv_reviewsummary') ?>
```



4. Insert the following code where you want the Reviews block to appear:

```
<?php echo $this->getChildHtml('bv_reviews') ?>
```

If you have put the ratings and reviews content behind a tab, see the Advanced Configuration appendix section *Putting Bazaarvoice Content Behind a Tab* for details on implementing the doShowContent() callback function.

Adding Inline Ratings to the Category (and/or Search Results) Pages

Perform the following steps to display an overall star-rating image next to a product, such as a Search page or a Category page that lists your products:

1. Ensure that **Enable Hosted Inline Ratings** is checked in the Magento control panel.
2. Open the following file in your Magento installation: `app/design/frontend/enterprise/default/template/catalog/product/list.phtml`
3. Locate the line that resembles the following example: `$_helper = $this->helper('catalog/output');`
4. Add the following code immediately after this line: `$_bvHelper = $this->helper('bazaarvoice');`
5. In the same file, locate the two occurrences of the following text: `<h2 class="product-name">`
6. Add the following code immediately after the `</h2>` element that closes each occurrence:

```
<div id="BVRRInlineRating-<?php echo $_bvHelper-
>getProductId($_product) ?>"></div>
```

7. Locate this, on approximately line 41:
   <?php else: ?>
   <?php echo $this->getToolbarHtml() ?>
   Add a line in the middle to make this:
   <?php else: ?>
   <?php echo $this->getChildHtml('inline_ratings') ?>
   <?php echo $this->getToolbarHtml() ?>

Once this step has been taken, and your caches are cleared you should see the following in the source of your category page.

In the HEAD ex.

```
<script type="text/javascript" src="http://display-stg.ugc.bazaarvoice.com/static/YOUR_-
CLIENT_NAME/Main+Site/en_US/bvapi.js"></script><link type="text/css"
rel="stylesheet" href="http://display.ugc.bazaarvoice.com/static/YOUR_CLIENT_NAME/
Main Site/18/10605/en_US/stylesheets/screen.css">
<script type="text/javascript">//<![CDATA[
      var Translator = new Translate([]);
      //]]></script><script type="text/javascript" charset="utf-8" async="" data-requirecon-
text="_BV" data-requiremodule="secondary" src="http://display.ugc.bazaarvoice.com/sta-
tic/YOUR_CLIENT_NAME/Main Site/18/10605/en_US/scripts/secondary.js"></script>
```

In the body toward the top:

```
<!-- Begin BV Inline Ratings -->
   <script type="text/javascript">
   $BV.ui( 'rr', 'inline_ratings', {
     productIds : {
                 'Your External ID' : {
        url : 'http://www.yourdomain.com/path-to-product-page.html'
      },
     },
     containerPrefix : 'BVRRInlineRating'
   });
   </script>
<!-- End BV Inine Ratings -->
```

For each product displayed on the category list or grid view:

```
<div id="BVRRInlineRating-YOUR_EXTERNAL_ID">
```

If all of the above is displaying except for the first section in the body containing the external ids and product urls then there is something in your theme keeping the bazaarvoice ratings.phtml template file from loading. Replace the code snippet: <?php echo $this->getChildHtml('inline_ratings') ?> with the following:

```
<!-- Begin BV Inline Ratings -->
<?php if(Mage::getStoreConfig('bazaarvoice/rr/enable_inline_ratings') === '1'
   && Mage::getStoreConfig('bazaarvoice/rr/enable_rr') === '1'
   && Mage::getStoreConfig('bazaarvoice/general/enable_bv') === '1'): ?>
   <script type="text/javascript">
   $BV.ui( 'rr', 'inline_ratings', {
     productIds : {
     <?php foreach($this->getLoadedProductCollection() as $_product): ?>
       <?php $_productId = Mage::helper('bazaarvoice')->getProductId($_product); ?>
       '<?php echo $_productId ?>' : {
```

```
            url : '<?php echo $_product->getProductUrl() ?>'
          },
        <?php endforeach; ?>
        },
        containerPrefix : 'BVRRInlineRating'
      });
      </script>
    <?php endif; ?>
    <!-- End BV Inine Ratings -->
```

```
@@ -32,14 +32,16 @@
 */
?>
<?php
    $_productCollection=$this->getLoadedProductCollection();
    $_helper = $this->helper('catalog/output');
+   $_bvHelper = $this->helper('bazaarvoice');
?>
<?php if(!$_productCollection->count()): ?>
<p class="note-msg"><?php echo $this->__('There are no products matching the selection.') ?></p>
<?php else: ?>
+<?php echo $this->getChildHtml('inline_ratings') ?>
<?php echo $this->getToolbarHtml() ?>
<?php echo $this->getAdditionalHtml() ?>
<div class="category-products">
    <?php // list mode ?>
    <?php if($this->getMode()!='grid'): ?>
```

```
@@ -52,11 +54,11 @@
            <?php // Product description ?>
            <div class="product-shop">
                <div class="f-fix">
                    <?php $_productNameStripped = $this->stripTags($_product->getName(), null, true); ?>
                    <h2 class="product-name"><a href="<?php echo $_product->getProductUrl() ?>" title="<?php echo $_productNameStrip
er->productAttribute($_product, $_product->getName() , 'name'); ?></a></h2>
-                   <?php if($_product->getRatingSummary()): ?>
-                   <?php echo $this->getReviewsSummaryHtml($_product) ?>
-                   <?php endif; ?>
+                   <div id="BVRRInlineRating-<?php echo $_bvHelper->getProductId($_product) ?>"></div>
                    <?php echo $this->getPriceHtml($_product, true) ?>
                    <div class="desc std">
                        <?php echo $_helper->productAttribute($_product, $_product->getShortDescription(), 'short_description') ?>
                        <a href="<?php echo $_product->getProductUrl() ?>" title="<?php echo $_productNameStripped ?>" class="link-
('Learn More') ?></p>
                    </div>
```

```
@@ -98,13 +98,11 @@
    <ul class="products-grid">
    <?php endif ?>
        <li class="item<?php if(($i-1)%$_columnCount==0): ?> first<?php elseif($i%$_columnCount==0): ?> last<?php endif; ?>">
            <a href="<?php echo $_product->getProductUrl() ?>" title="<?php echo $this->stripTags($this->getImageLabel($_pred
?>" class="product-image"><img src="<?php echo $this->helper('catalog/image')->init($_product, 'small_image')->resize(135); ?>
?php echo $this->stripTags($this->getImageLabel($_product, 'small_image'), null, true) ?>" /></a>
            <h2 class="product-name"><a href="<?php echo $_product->getProductUrl() ?>" title="<?php echo $this->stripTags($_
>"><?php echo $_helper->productAttribute($_product, $_product->getName(), 'name') ?></a></h2>
-           <?php if($_product->getRatingSummary()): ?>
-           <?php echo $this->getReviewsSummaryHtml($_product, 'short') ?>
-           <?php endif; ?>
+           <div id="BVRRInlineRating-<?php echo $_bvHelper->getProductId($_product) ?>"></div>
            <?php echo $this->getPriceHtml($_product, true) ?>
            <div class="actions">
                <?php if($_product->isSaleable()): ?>
                    <button type="button" title="<?php echo $this->__('Add to Cart') ?>" class="button btn-cart" onclick="set
dToCartUrl($_product) ?>')"><span><span><?php echo $this->__('Add to Cart') ?></span></span></button>
                <?php else: ?>
```

Questions and Answers

Within the directory structure of your Magento installation, open the Product page template file `view.phtml`, as appropriate. This file typically resides in the following locations:

```
/app/design/frontend/enterprise/default/template/catalog/product/
view.phtml
```

Insert the following code where you want the Questions & Answers block to appear:

```
<?php echo $this->getChildHtml('bv_questions') ?>
```

Ensure that **Enable Questions & Answers** is checked in the Magento control panel.

```
@@ -97,19 +97,25 @@
                <dt class="tab"><span><?php echo $this->escapeHtml($this->getChildData($alias, 'title')) ?></span></dt>
                <dd class="tab-container">
                    <div class="tab-content"><?php echo $html ?></div>
                </dd>
            <?php endforeach;?>
+               <dt id="tab-reviews" class="tab"><span><?php echo $this->__('Reviews')?></span></dt>
+               <dd class="tab-container">
+                   <div class="tab-content">
+                       <?php echo $this->getChildHtml('bv_reviews') ?>
+                   </div>
+               </dd>
+               <dt id="tab-questions" class="tab"><span><?php echo $this->__('Q&A')?></span></dt>
+               <dd class="tab-container">
+                   <div class="tab-content">
+                       <?php echo $this->getChildHtml('bv_questions') ?>
+                   </div>
+               </dd>
            </dl>
-           <script type="text/javascript">
-           //<![CDATA[
-               var collateralTabs = new Enterprise.Tabs('collateral-tabs');
-               Event.observe(window, 'load', function() {
-                   collateralTabs.select();
-               });
-           //]]>
-           </script>
        <?php endif; ?>
        <?php if ($this->canEmailToFriend()): ?>
```

If you have put the questions and answers content behind a tab, see the Advanced Configuration appendix section *Putting Bazaarvoice Content Behind a Tab* for details on implementing the doShowContent() callback function.

Bazaarvoice SEO

This version of the extension utilizes bvseosdk.php as of the following commit:
https://github.com/bazaarvoice/HostedUIResources/commit/69168eae518238b1b0223d74af900d2fcaac3d78

The Magneto Extension utilizes the Bazaarvoice Cloud SEO solution.
Complete the following steps in the Magento control panel.

1. Set Enable Cloud SEO to Yes.

2. Locate your Cloud SEO Key value in the Bazaarvoice Configuration Hub (http://login.bazaarvoice.com).

3. Enter your key value into the Cloud SEO Key control panel setting.

Additionally, the product page templates must be updated to include Bazaarvoice content. See related section of this document.

ROI Beacon

The Bazaarvoice for Magneto Extension includes support for the Bazaarvoice ROI beacon. This feature is an analytics tag on the checkout page. This tag provides a small set of data elements to BV when an online user completes the checkout process.

Note: Around lines 58-61 when viewing source of the Success page you will see the beacon if enabled in the Magento Admin. You need to make sure you are viewing the current source of the success page. By default the view source functionality in many browsers reloads the page in order to view it. In this case the information will not be visible.

1. Ensure that Enable ROI beacon is checked in the Magento control panel.

Product Feed

The Bazaarvoice for Magneto Extension includes support for generating a product catalog feed to Bazaarvoice and upload the feed via FTP to Bazaarvoice.

By default, the Bazaarvoice Product Feed is scheduled to run every day at 1 AM (Note: the time zone is set based on the locale set in Magento at the default configuration scope).

Complete the following to enable the Product Feed:

- Ensure that the Magento cron job is enabled
- Ensure that Enable Product Feed is checked in the Magento control panel.

See the Advanced Configuration section for instructions on how to verify that the Magento cron job is running, as well as how to change the default time of day for the cron jobs.

Bazaarvoice Families

The Bazaarvoice for Magento Extension includes support for creating Bazaarvoice Product Families where the Configurable product is the Parent, and Simple product is the Child. The sku or External ID is used as the Family Name. When product family support is enabled both the configurable, and simple associated products will be added to the family, and EXPAND will be set to true on both. We have added the ability for you to set EXPAND on simple products to false by adding a short xml snippet to the app/etc/local.xml file. See Appendix D Advanced Configuration.

With EXPAND set to true both configurable, and simple products will display the totals for the entire family when displayed in the catalog. When EXPAND set to false only the configurable product will display the totals for the family, and the simples will display only their content.

Below is how each piece of Bazaarvoice is affected when Bazaarvoice Product Families are enabled:

Product Feed

- Configurable sku will be family name
- Configurable will be a member of it's family
- All associated simple products will be member of family
- Simples are included in feed whether they are visible or not.
- When simple is visible
    - We will send it's category
    - We will send it's url
    - We will send it's image

- When simple is Not Visible Individually
    - We will send parent category
    - We will send parent url
    - We will send it's base image if one exists otherwise send parent

Inline Ratings

- No Change here what is displayed depends on the EXPAND setting. When EXPAND true simples will display the totals of all family content. When set to false simples will display only their rating and review numbers.

Product Detail

- Similar to above
- On products with single configurable option the ratings and reviews displayed will reflect the results for the resulting external id as selected. This means that when EXPAND set to false you would be able to see only the Ratings and Reviews for the simple product as selected based on the configurable option. An example of this would be if you have flavors as a configurable option. When the cherry flavor is selected you would see content relating to the cherry flavor.

Cloud SEO

- Functions normally

ROI Beacon

- Simple sku is sent
- Simple name is sent
- Simple image is sent if base image available otherwise config image url is sent.

Purchase Feed

- Same as ROI Beacon

Submission

- All submissions will be attributed to the simple product as selected.

Appendix A: Mapping your Magento Environment to your Bazaarvoice Environment

Magento is a very flexible platform. As such, the following scenarios describe how the Magento extension can handle different ways to map the Magento environment to the Bazaarvoice environment.

Background

Magento provides the following hierarchy of scope:

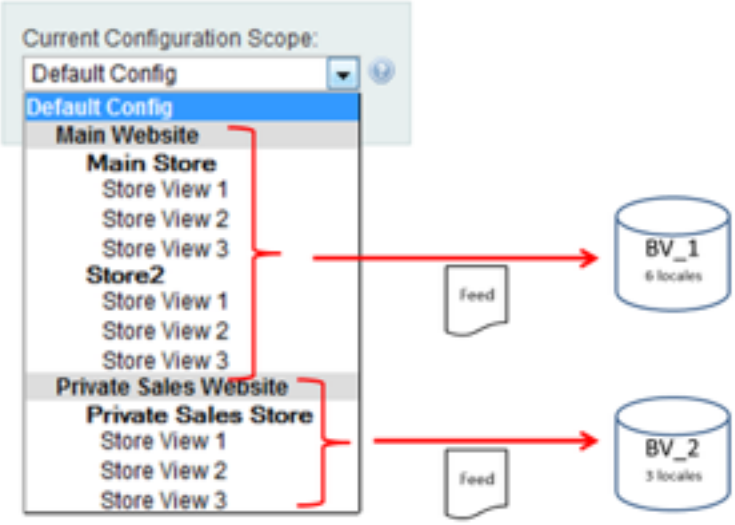
- Magento Instance
    - Magento Website
        - Magento Store (Group)
            - Magento Store View

A Bazaarvoice client may exist as one or more instances of the Bazaarvoice platform, depending on client needs. Each instance can support multiple locales. Each instance can consume only one product catalog feed (w/ localized product and category attributes for each configured locale).

Scenario 1: Each Magento Website maps to a Separate Bazaarvoice Instance

Description of Scenario

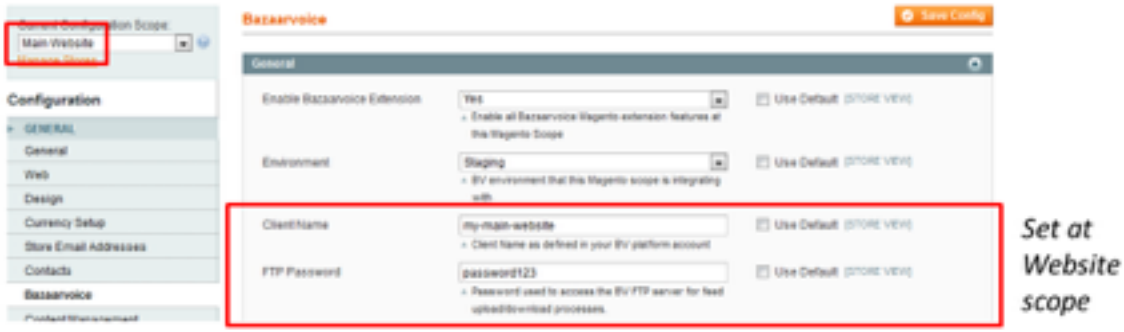
- BV provisions an instance for each Magento Website
- Locales are mapped at the Store View level
- A single product catalog feed is generated for the entire Magento Website (the feed will contain localized attributes (e.g. Product Page Urls, Product Names, etc.) for each Store View in the Website)
- **Limitations**:
    - Each Store View must map to a unique BV Locale within a given BV instance
    - All stores within a single website must share the same category tree



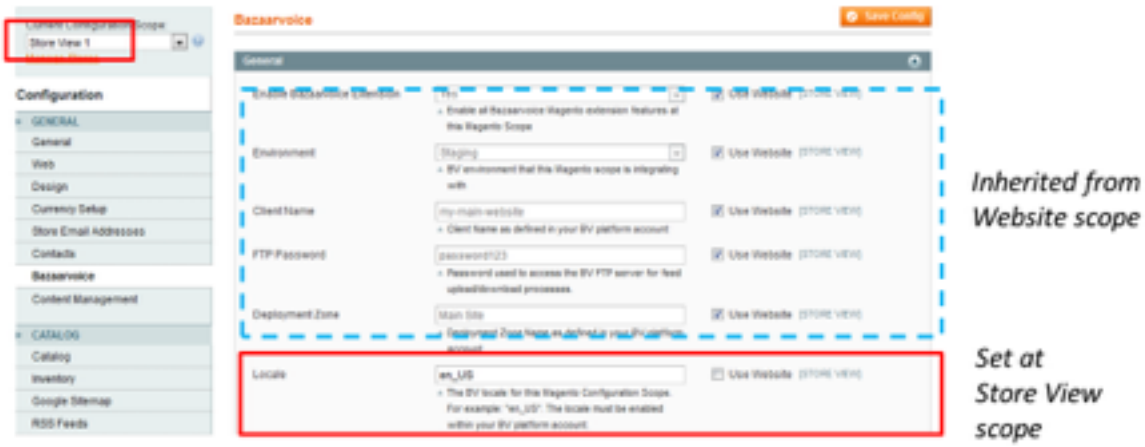
How to Setup the Scenario

- Contact BV to provision a separate BV instance for each desired Magento Website.
- In control panel, set **Feed Generation Scope** to **Magento Website**.

- Set BV client-level settings (Client Name, FTP Password, Cloud SEO Key) at the Magento Website configuration scope.



- Set Locale at each Store View configuration scope
- Other BV client-level settings (Client Name, FTP Password, SEO Cloud Key) should inherit from the Website scope.

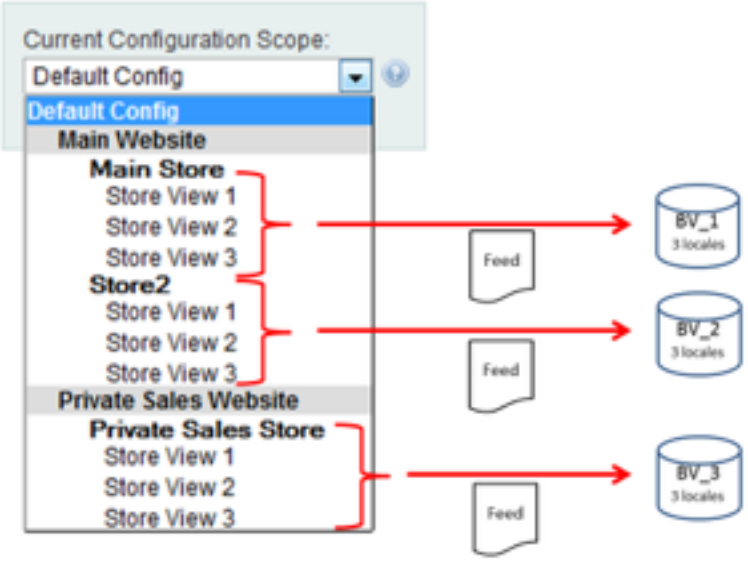


- (optional) Deployment Zone can be configured at the Store View scope as well.

Scenario 2: Each Magento Store (Group) maps to a Separate Bazaarvoice Instance

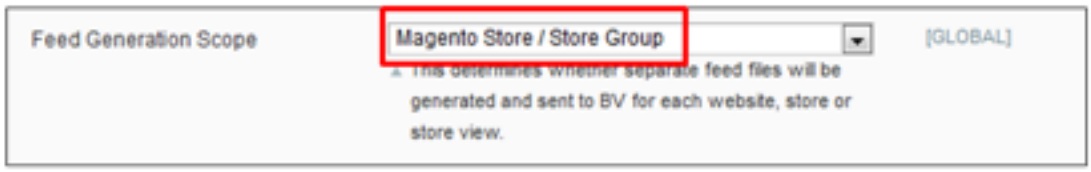
Description of Scenario

- BV provisions an instance for each Magento Store (Group)
- Locales are mapped at the Store View level
- A single product catalog feed is generated for Magento Store (Group) (the feed will contain localized attributes (e.g. Product Page Urls, Product Names, etc.) for each Store View in each Store (Group))
- Limitation: each Store View must map to a unique BV Locale within a given BV instance



How to Setup the Scenario

- Contact BV to provision a separate BV instance for each desired Magento Store (Group).
- In control panel, set **Feed Generation Scope** to **Magento Store / Store Group**.

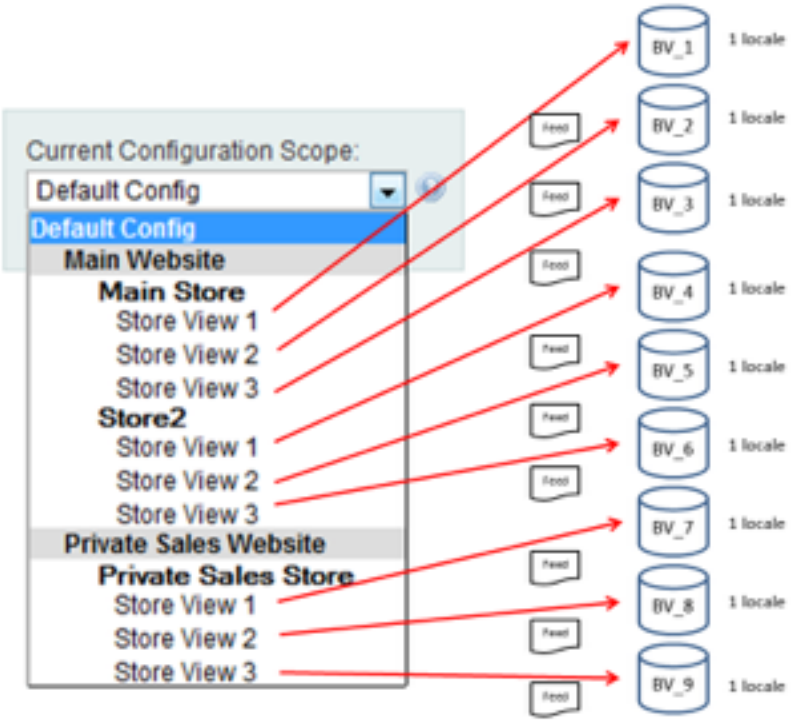


- Because settings cannot be configured at the Magento Store (Group) level, you must set the BV client-level settings (Client Name, FTP Password, Cloud SEO Key) at the Magento Store View configuration scope ***and ensure that these settings match for all Store Views within the given Store (Group)***.
- Locale should be set at the Store View configuration scope.
- (Optionally) Deployment Zone can be set at the Store View configuration scope.

Scenario 3: Each Magento Store View maps to a Separate Bazaarvoice Instance

Description of Scenario

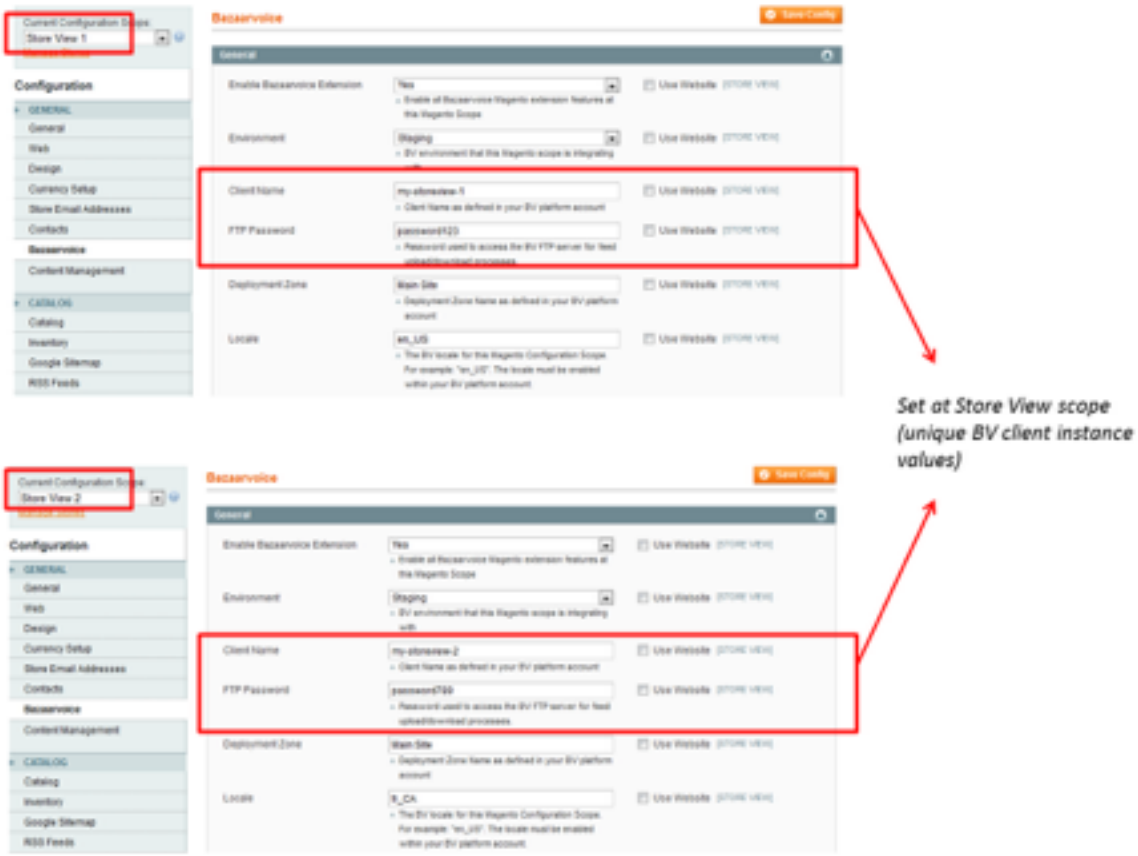
- BV provisions an instance for each Magento Store View
- Locales are mapped at the Store View level
- A single product catalog feed is generated for Magento Store View (the feed will contain a single set of localized attributes (e.g. Product Page Urls, Product Names, etc.) for the Store View locale)



How to Setup the Scenario

- Contact BV to provision a separate BV instance for each desired Magento Store View.
- In control panel, set **Feed Generation Scope** to **Magento Store View**.

- All control panel settings should be configured at the Magento Store View configuration scope.
- Because each Store View will map to a separate BV instance, the same locale can be used across the Store View.



Appendix B: Upgrading from Earlier Version of Magento Extension

The Bazaarvoice for Magento Extension version 6.x represents a significant upgrade to previous versions of the extension. The 6.x version is explicitly built to integrate with the new Bazaarvoice Conversations 2013 platform.

- Hosted Inline Ratings (no longer relies on feeds)
- Cloud-base SEO solution (no longer relies on feeds)

Upgrading to the new Bazaarvoice Conversations 2013 platform requires considerable coordination with Bazaarvoice, and involves additional effort beyond upgrading the Bazaarvoice for Magento extension.

Contact your Bazaarvoice representative if you are looking to upgrade to the new Bazaarvoice platform.

Appendix C: Code Logic and Data Mapping

Changes Made to the Magento Environment as part of the Installation Process

New Data Table Fields

During Installation of the extension a field is created on the sales_flat_order table called “sent_in_bv_postpurchase_feed” The value will be NULL by default, and set to 1 as orders are sent to Bazaarvoice in the order feed.

Product Feed Field Mapping

This section details the contents of what is included in the product feed. In general the product feed contains categories, and products.

CATEGORIES		**PRODUCTS**	
Magento Attribute	**Bazaarvoice Attribute**	**Magento Attribute**	**Bazaarvoice Attribute**
*ID/Url Path of the category	ExternalID	sku	ExternalId
*ID/url path of the Parent Category	ParentExternalId	Name	Name
The category name	Name	Short Description	Description
The category page url	CategoryPageUrl	**NULL	Brand
This section contains a list of the localized category names that are found at the StoreView configuration scope.	Names	***NULL	UPC
This section contains the category url with the Store View parameter added	CategoryPageUrls	*This is the category ID/Url path of the category that the product is in.	CategoryExternalID

		This is the url of the product page.	ProductPageUrl
		This is the url of the product image. This will call the cached version of the file.	ImageUrl
		This is the localized version of the product name as found in the Store View configuration scope	Names
		This is the localized version of the Short Description as found in the Store View configuration scope	Descriptions
		This is the localized version of the product page	ProductPageUrls
		This is the localized version of the product image	ImageUrls

* The value for the ID is dependent on how you have the configuration set. By default it is set to use the category url path. This can be overridden in the app/etc/local.xml file. The details of doing this can be found in the Advanced Configuration section.
** By default there are no Magento attributes mapped to the Bazaarvoice Brand attribute. This is configurable to use the value of a “Select” attribute of your choosing by overriding the value in the app/etc/local.xml. See the Advanced Configuration setting of this document for further details.
*** By default there are no Magento attributes mapped to the Bazaarvoice UPC attribute. This is configurable to use the value of a “Text” attribute of your choosing by overriding the value in app/etc/local.xml. See the Advanced Configuration setting of this document for further details.

Include Categories Algorithm

Root Category Based on the Feed configuration setting
Include categories at level greater than 1
Include only active categories
Find all categories that have the root path included in their path
If the count is greater than 0 write the categories to the file

Include Products Algorithm

Retrieve the products from the website
Only include Enabled products

Only include products that are visible in the catalog aka Visibility does not equal Not Visible Individually
Write to the file if there are more than 0 matching products.
It is ensured that the CategoryExternalId is one that is included in the category section of the feed.

Purchase Feed Algorithm

```
There are 2 config settings for purchase feed process:
  - Enabled / Disabled
  - Triggering Event  (Can be Shipping or Purchase)
The purchase feed code adds one extra field / attribute to the Magento
order record:
  - "ALREADY SENT IN FEED" (Field defaults to false when new order cre-
ated by Magento)
Purchase Feed process follows this algorithm:
   > Init new local feed file
   > Find all store groups in Magento Instance (Also know as stores,
this is the level above store view)
   > For each store group
     > Check if purchase feed is enabled in the default store view
within the group
     > If yes,
       > Find all Magento order meeting these criteria:
         (order is in a store within current store group)
         (order status = 'complete' or 'closed')
         (order created_date >= 30 days ago)
         (order "ALREADY SENT IN FEED" field = false)
       > For each order found:
         > Does order meet all these criteria:
           (order "ALREADY SENT IN FEED" field = false)
           (order status is not 'cancelled')
           (order->store field is empty) (really this should never hap-
pen with Magento)
           (order has at least 1 item)
           (order has a customer email address)
         > And, does order meet one or the other of these criteria:
           ("TRIGGERING EVT" = Shipping  AND  order has completely
shipped  AND  more than 1 day has passed since the last shipment)
or
           ("TRIGGERING EVT" = Purchase  AND  more than 1 day has
passed since purchase happened)
         > If yes,
           > Add order to local feed file
           > Set "ALREADY SENT IN FEED" field = true for order
         > If no, do nothing
     > If no, do nothing   > Save local feed, send local feed file to
BV via FTP
```

ROI Beacon Data Mapping

Below is a data mapping that shows what is being sent to Bazaarvoice via the ROI beacon.

Order Details	
Magento	**Bazaarvoice**
Order Increment id	orderId
TaxAmount	tax
ShippingAmount	shipping
GrandTotal	total
City	city
RegionId Code	state
CountryId	country
OrderCurrencyCode	currency
*CustomerId	userid
CustomerEmail	email
CustomerEmail	nickname
StoreLocale	locale
Magento Extension Plugin	partnerSource
Order Items	
Sku	sku
Name	name
Price	price
QtyOrdered	quantity
ImageUrl	imageUrl

* When Guest Checkout is used there is not UserId for the customer. In this case we are sending the MD5 hash of the customer email address.

String Replacement for Ids with Special Characters

Because the Bazaarvoice platform cannot accept product, category, or brand ID values that contain special characters, the Magento extension automatically transforms any offending ID values based on the following table:

Special Character	Replaced With
&	bv38_amp_bv59
%	bv37
(space)	bv32
/	bv47
$	bv36
@	bv64
#	bv35
]	bv93
[	bv91
“	bv125
`	bv96
~	bv126
<	bv38_lt_bv59
>	bv38_gt_bv59
\	bv92
#	bv35
!	bv33
^	bv94
(	bv40
>	bv41
|	bv124
=	bv61
?	bv63

Appendix D: Advanced Configuration

Putting Bazaarvoice Content Behind a Tab

When Bazaarvoice content is put behind a tab, Bazaarvoice provides the JavaScript callback function doShowConent() to allow Bazaarvoice to trigger the tab opening when a “read all reviews” link is clicked on the product page.
See the sections “Questions and Answers integration code” and/or “Ratings and Reviews integration code” sections of the standard Bazaarvoice integration documentation for details on this function: https://config.bazaarvoice.com/resources/v1/definitions/helpDocs/CodeIntegrationGuide_C2013.pdf

The Bazaarvoice for Magento extension provides the following easy-to-use control panel setting to insert the JavaScript required to utilize this callback function:

- For R&R: **JS Callback function to display R&R content (doShowContent)**
- For Q&A: **JS Callback function to display Q&A content (doShowContent)**

Configuration Settings Outside the Magento Control Panel

There are several configuration parameters that can be hard coded into your site.

The following parameters can be hard coded in your app/etc/local.xml file.

Product Feed File Name	<product_feed_export_filename>	file_name.xml	You can specify the name of your product feed
Product Feed Export Path	<product_feed_export_export_path>	/import-inbox	This setting allows you to control the path the file will be uploaded to on the BV ftp server
Product Feed Brand Attribute Code	<product_feed_brand_attribute_code>	NULL	By default this configuration is set to NULL, and can be set to a field of your choosing. Note: it must be a “Select” Type attribute in Magento.

Product Feed UPC Attribute Code	<product_feed_upc_attribute_code>	NULL	By default this configuration is set to NULL, and can be set to a field of your choosing. Note: it must be set to a “Text” Type attribute in Magento.
Product Feed EAN Attribute Code	<product_feed_ean_attribute_code>	NULL	By default this configuration is set to NULL, and can be set to a field of your choosing. Note: it must be set to a “Text” Type attribute in Magento.
Product Feed Set Simple Products to Expand	<product_feed_export_bvfamilies_expand>	true	By default this configuration is set to true. Which means that simple products will display totals based on the families. When set to false only content for the simple product will be displayed.
Category Id Use URL Path	<category_id_use_url_path>	true or false	By default this is set to “true”. When set to true he CategoryId used in feeds will be the url path of the category tree separated by hyphens. Ex: apparel-shirts where apparel is the top level category, and shirts is the category. When set to false the actual Category Id will be used. Ex: This is the id that can be found when navigating to Catalog>Manage Categories, and selecting a category. The id will be displayed at the top of the page to the right of the category name.

FTP Host Name	<ftp_host_name>	hostname	Used to override the default BV FTP server hostnames. If this is not specified, the standard ftp-stg.bazaarvoice.com and ftp.bazaarvoice.com will be used. <ftp_host_name></ftp_host_name> Example setting: sftp7-stg.bazaarvoice.com

Example code

```
<default>
   <bazaarvoice>
      <bv_config>
         <!-- Hard code export path and filename in class constants -->
         <product_feed_export_filename>productfeed.xml</product_feed_export_filename>

         <!-- Hard code export path and filename in class constants -->
         <product_feed_export_export_path>/import-inbox</product_feed_export_export_path>

         <!-- Magento Brand Attribute Code
         <product_feed_brand_attribute_code>manufacturer</product_feed_brand_attribute_-
code>

         <!-- Magento UPC Attribute Code -->
         <product_feed_upc_attribute_code>upc</product_feed_upc_attribute_code>

         <!—Magento EAN Attribute Code —>
         <product_feed_ean_attribute_code>ean</product_feed_ean_attribute_code>

         <!-- Disable Expand on Simple products -->
         <product_feed_export_bvfamilies_expand>false</product_feed_export_bvfamilies_ex-
pand>

         <!-- If true, use Magento category URL path as category id, otherwise use Mage category
entity primary key -->
         <category_id_use_url_path>true</category_id_use_url_path>

         <!-- FTP Connection Host Name.  If this is not specified, the standard ftp-stg.bazaar-
voice.com and ftp.bazaarvoice.com will be used -->
         <ftp_host_name></ftp_host_name>

         <!-- Display code - We still need this for some clients -->
```

```
        <display_code></display_code>
      </bv_config>
    </bazaarvoice>
</default>
```

Purchase Feed

Note: for most Bazaarvoice clients, ROI beacon data can be used as a replacement for the purchase feed feature. It is not necessary to enable both the ROI beacon as well as the purchase feed feature simultaneously.

For clients who insist on enabling the purchase feed, the following should be completed:

- Ensure that **Enable Product Feed** is checked in the Magento control panel.

By default, the feed is configured to run every day at 1:30 AM (Note: the time zone is set based on the locale set in Magento at the default configuration scope).

Changing Default Time of Day of Cron Schedules for Bazaarvoice Feeds

The section below shows the cron tab configuration section of the config.xml file, which can be edited to the time that you prefer.

The time zone is determined by the locale set in Magento at the default configuration scope.

The format used is a standard cron expression. See this article for details: http://en.wikipedia.org/wiki/Cron

```
<crontab>
  <jobs>
    <bazaarvoice_product_feed>
      <schedule>
        <cron_expr>0 1 * * *</cron_expr><!-- Every day at 1:00 AM -->
      </schedule>
      <run>
        <model>bazaarvoice/exportProductFeed::exportDailyProductFeed</model>
      </run>
    </bazaarvoice_product_feed>
    <bazaarvoice_purchase_feed>
      <schedule>
        <cron_expr>30 2 * * 6</cron_expr><!-- Every Saturday at 2:30 AM -->
      </schedule>
      <run>
        <model>bazaarvoice/exportPurchaseFeed::exportPurchaseFeed</model>
      </run>
    </bazaarvoice_purchase_feed>
  </jobs>
</crontab>
```

Appendix E: Troubleshooting

Contact Bazaarvoice for Support

Contact Bazaarvoice Support (email support@bazaarvoice.com) if you have issues installing or configuring the extension. It is helpful to provide Magento Admin panel credentials and SSH access for Bazaarvoice to efficiently provide support.

Bazaarvoice Logging

As of version 6.1.2 All logging form the Bazaarvoice Connector can be found in the standard Magento log directory as bazaarvoice.log

How to Verify that the Magento Cron Job is Running

To ensure the Magento cron job is running you can run crontab -l (the –l is to list the cron jobs that are running.) from the command line. Typically the job is scheduled as the root user, but could vary depending on your hosting environment. If you are unsure contact your system administrator or hosting company.

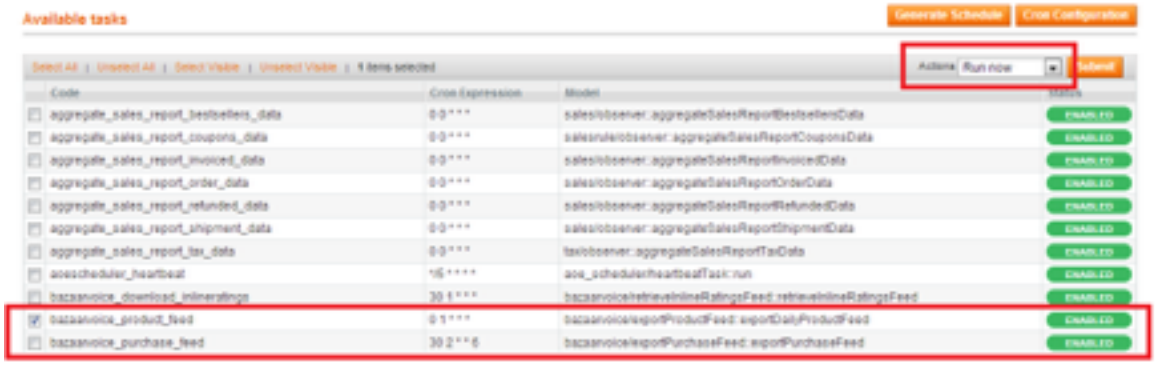
```
crontab -l
        */5 * * * * /bin/bash /path/to/magento/cron.sh
```

How to Manually Run a Bazaarvoice Feed

Run Feeds Using AOE Scheduler Plug-in

You can also run the feeds manually by installing the following extension.
http://www.magentocommerce.com/magento-connect/aoe-scheduler.html
This is the extension key that you can use in versions 1.5+ and 1.10+ from Magento Connect
http://connect20.magentocommerce.com/community/Aoe_Scheduler

1. Install plug-in
2. Navigate to **System** > **Scheduler** > **Schedule Configuration**
3. Select the desired BV feed (code = “bazaarvoice_product_feed” or “bazaarvoice_purchase_feed”)
4. In the **Actions** pull-down control, select **Run now**.
5. Click **Submit**.



Run Feeds Using Command Line

Finally, you can manually run the feeds from the command line. We have included scripts with the extension included in the /path/to/magento/shell directory.

Product Feed: bv_export_product_feed.php
Purchase Feed: bv_export_order_feed.php

Example:
php –f ./shell/bv_export_product_feed.php

NOTE: These must be run from the Magento root directory so as to properly call the app/Mage.php

Appendix F: Migrating to Production

Steps for Testing Integration in Staging

1. Log in to Magento Admin.
2. Make sure the Bazaarvoice control panel setting Environment is set to the value “Staging”.
3. Follow this Implementation Guide to complete the integration with Bazaarvoice.
4. Confirm with Bazaarvoice team that integration is functioning properly (e.g. product feed is received by BV, product and category page properly displays BV content, etc.)

Before Launch on Production.

1. Staging Magento instance is updated to be a replica of production.
2. Cron jobs are disabled
3. Staging Magento instance is setup to connect to Production BV environment.
4. A product feed is triggered which will deliver the feed to the BV production environment, and processed.
5. Once complete and verified on the staging version of the Magento site connected to the BV production environment
6. Procedure below is carried out.

Steps to Move from Staging to Production

1. Disable and clear caches
2. Release code (ensure any changes in app/etc/local.xml are migrated to prod)
3. Log out of Magento Admin
4. Log into Magento Admin
5. Duplicate configuration in System > Configuration > GENERAL > Bazaarvoice
6. Set Environment to Production

Following the procedure above there is the issue of the urls that will be provided to Bazaarvoice This could be handled by setting the base urls in the Magento staging instance to the production urls, and using local host entries to point at the staging magento instance. This would allow you to view the site, and generate content with the production values.