

Bazaarvoice

Bazaarvoice for Magento Extension Implementation Guide v6.0 (Beta)

Version 6.0 Beta

Introduction

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

This is a beta.

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
- Redis NoSQL (optional for CE 1.8 and later, EE 1.13 and later)
 - redis-server version 2.6.9 or later
 - phpredis version 2.2.2 or later
- 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 Connect 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 Functionality

If the Magento native reviews module is integrated with your storefront, disable it 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.

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 Magento installation:
`app/design/frontend/[your_interface]/[your_theme]/template/catalog/product/view.html`
3. Insert the following code where you want the Ratings Summary to appear:

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

```
@@ -49,11 +49,11 @@
    <div class="product-main-info">
        <div class="product-name">
            <h1><?php echo $_helper->productAttribute($_product, $_product->getName(), 'name') ?></h1>
        </div>
        <p class="product-ids"><?php echo $this->__('SKU# %s', $this->escapeHtml($_product->getSku())); ?></p>
-        <?php echo $this->getReviewsSummaryHtml($_product, false, true)?>
+        <?php echo $this->getChildHtml('bv reviewssummary') ?>
        <?php echo $this->getChildHtml('alert_urls') ?>
        <?php echo $this->getChildHtml('product_type_data') ?>
    </div>
    <?php echo $this->getChildHtml('tierprices') ?>
    <?php echo $this->getChildHtml('extrahint') ?>
```

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

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

```
@@ -97,19 +97,17 @@
    <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 $this->getChildHtml('bv reviews') ?>
+        </div>
+    </dd>
    </dl>
    <script type="text/javascript">
        //
        var collateralTabs = new Enterprise.Tabs('collateral-tabs');
        Event.observe(window, 'load', function() {
            collateralTabs.select();
        });
        //]]&gt;
    &lt;/script&gt;
    &lt;?php endif; ?&gt;
    &lt;?php if ($this-&gt;canEmailToFriend()): ?&gt;</pre>
</div>
<div data-bbox="142 503 853 557" data-label="Text">
<p>If you have put the ratings and reviews content behind a tab, see the Advanced Configuration appendix section <i>Putting Bazaarvoice Content Behind a Tab</i> for details on implementing the <code>doShowContent()</code> callback function.</p>
</div>
<div data-bbox="142 588 743 606" data-label="Section-Header">
<h2>Adding Inline Ratings to the Category (and/or Search Results) Pages</h2>
</div>
<div data-bbox="142 606 856 643" data-label="Text">
<p>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:</p>
</div>
<div data-bbox="171 643 856 867" data-label="List-Group">
<ol>
<li>1. Ensure that <b>Enable Hosted Inline Ratings</b> is checked in the Magento control panel.</li>
<li>2. Open the following file in your Magento installation:<br/>app/design/frontend/enterprise/default/template/catalog/product/list.phtml</li>
<li>3. Locate the line that resembles the following example:<br/>$_helper = $this-&gt;helper('catalog/output');</li>
<li>4. Add the following code immediately after this line:<br/>$_bvHelper = $this-&gt;helper('bazaarvoice');</li>
<li>5. In the same file, locate the two occurrences of the following text:<br/>&lt;h2 class="product-name"&gt;</li>
<li>6. Add the following code immediately after the &lt;/h2&gt; element that closes each occurrence:</li>
</ol>
</div>
<div data-bbox="489 916 507 934" data-label="Page-Footer">
<p>5</p>
</div>
```

```
<div id="BVRRIInlineRating-<?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() ?>
```

```
@@ -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'): ?>
@@ -52,13 +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 $productNameStripped ?>">
                <?php echo $product->getName() ?></a></h2>
            <?php if($product->getRatingSummary()): ?>
            <?php echo $this->getReviewsSummaryHtml($product) ?>
            <?php endif; ?>
+            <div id="BVRRIInlineRating-<?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 $productNameStripped ?>" class="link-l
('Learn More') ?></a>
            </div>
@@ -98,13 +98,11 @@
    <ul class="products-grid">
        <?php endif ?>
        <li class="item"><?php if(($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($product, 'small_image', 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="BVRRIInlineRating-<?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><?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->getLayout()->getBlock('questions')->toHtml(); ?>
```

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

```
@@ -97,19 +97,23 @@
    <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 $this->getChildHtml('bv_questions') ?>
+       </div>
+   </dd>
    </dl>
    <script type="text/javascript">
    //
        var collateralTabs = new Enterprise.Tabs('collateral-tabs');
        Event.observe(window, 'load', function() {
            collateralTabs.select();
        });
    //]]&gt;
    &lt;/script&gt;
    &lt;?php endif; ?&gt;
    &lt;?php if ($this-&gt;canEmailToFriend()): ?&gt;</pre></div><div data-bbox="142 583 853 637" data-label="Text"><p>If you have put the questions and answers content behind a tab, see the Advanced Configuration appendix section <i>Putting Bazaarvoice Content Behind a Tab</i> for details on implementing the <code>doShowContent()</code> callback function.</p></div><div data-bbox="142 686 327 706" data-label="Section-Header"><h2>Bazaarvoice SEO</h2></div><div data-bbox="142 708 715 726" data-label="Text"><p>The Magento Extension utilizes the Bazaarvoice Cloud SEO solution.</p></div><div data-bbox="142 727 640 745" data-label="Text"><p>Complete the following steps in the Magento control panel.</p></div><div data-bbox="171 745 800 824" data-label="List-Group"><ol><li>1. Set Enable Cloud SEO to Yes.</li><li>2. Locate your Cloud SEO Key value in the Bazaarvoice Configuration Hub (<a href="http://login.bazaarvoice.com">http://login.bazaarvoice.com</a>).</li><li>3. Enter your key value into the Cloud SEO Key control panel setting.</li></ol></div><div data-bbox="142 838 828 875" data-label="Text"><p>Additionally, the product page templates must be updated to include Bazaarvoice content. See related section of this document.</p></div><div data-bbox="489 916 506 934" data-label="Page-Footer"><p>7</p></div>
```


ROI Beacon

The Bazaarvoice for Magento 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.

```
<script type="text/javascript" src="https://display-stg.uqc.bazaarvoice.com/static/bvp-magentoc2013sandbox2/Main+Site/en_US/bvapi.js"></script>
<script type="text/javascript">
$BV.$I.trackTransactionPageView({"orderId":"24","tax":"0.00","shipping":"0.00","total":"129.99","city":"Baltimore","state":"MD","country":"US","currency":"USD","items":
[{"sku":"1112","name":"Chair_English","price":"129.99","quantity":"1","imageUrl":"https://bv2013.storefrontconsulting.com/media/catalog/product/cache/1/image/265x/9df78eab33525d08de5fb8d27136e95/c/h/cha
ir.jpg"}],"userId":null,"email":"chris@storefrontconsulting.com","nickname":"chris@storefrontconsulting.com","locale":"en_US","partnerSource":"Magento Extension r6.0.0"});
</script>
```

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

Product Feed

The Bazaarvoice for Magento 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.

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:

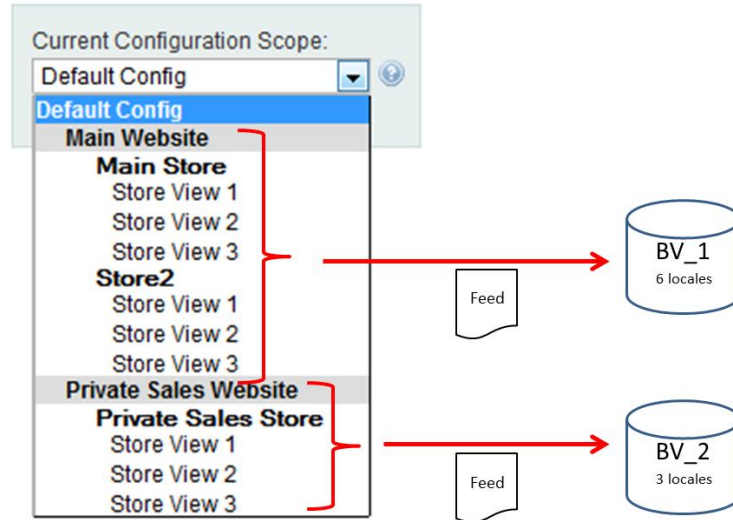
- Magento Instance
 - Magento Website
 - Magento Store (Group)
 - Magento Store View

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**.

Feed Generation Scope Magento Website [GLOBAL]

* This determines whether separate feed files will be generated and sent to BV for each website, store or store view.

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

Current Configuration Scope: Main Website

Configuration

- GENERAL
- General
- Web
- Design
- Currency Setup
- Store Email Addresses
- Contacts
- Bazaarvoice
- Content Management

Bazaarvoice Save Config

General

Enable Bazaarvoice Extension Yes Use Default [STORE VIEW]

Environment Staging Use Default [STORE VIEW]

Client Name my-main-website Use Default [STORE VIEW]

FTP Password password123 Use Default [STORE VIEW]

Set at Website 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.

Bazaavoice

Save Config

General

Enable Bazaavoice Extension

Yes

▲ Enable all Bazaavoice Magento extension features at this Magento Scope

Environment

Staging

▲ BV environment that this Magento scope is integrating with

Client Name

my-main-website

▲ Client Name as defined in your BV platform account

FTP Password

password123

▲ Password used to access the BV FTP server for feed upload/download processes.

Deployment Zone

Main Site

▲ Deployment Zone Name as defined in your BV platform account

☒ Use Website [STORE VIEW]

☒ Use Website [STORE VIEW]

☒ Use Website [STORE VIEW]

☒ Use Website [STORE VIEW]

☒ Use Website [STORE VIEW]

Locale

en_US

▲ The BV locale for this Magento Configuration Scope.
For example: "en_US". The locale must be enabled within your BV platform account.

☐ Use Website [STORE VIEW]

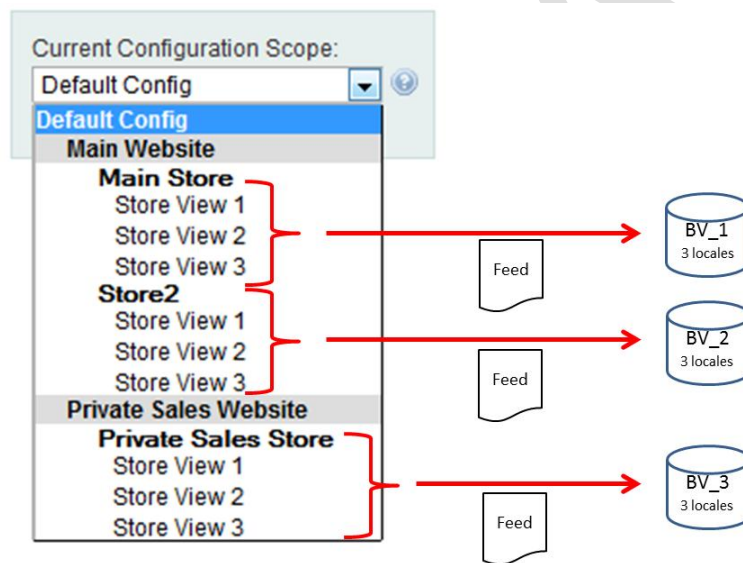
Set at
Store View
scope

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

Feed Generation Scope **Magento Store / Store Group** [GLOBAL]

This determines whether separate feed files will be generated and sent to BV for each website, store or store view.

- 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.

The image displays two screenshots of the Bazaarvoice configuration interface, illustrating the correct configuration for the 'Locale' and 'Deployment Zone' settings across different Store Views.

Top Screenshot (Store View 1): The 'Current Configuration Scope' is set to 'Store View 1'. The 'Locale' is configured as 'en_US'. The 'Deployment Zone' is set to 'Main Site'. A red box highlights the 'Locale' field, and a red arrow points to the text: "Set Locale at Store View scope (unique locale to Store Group)".

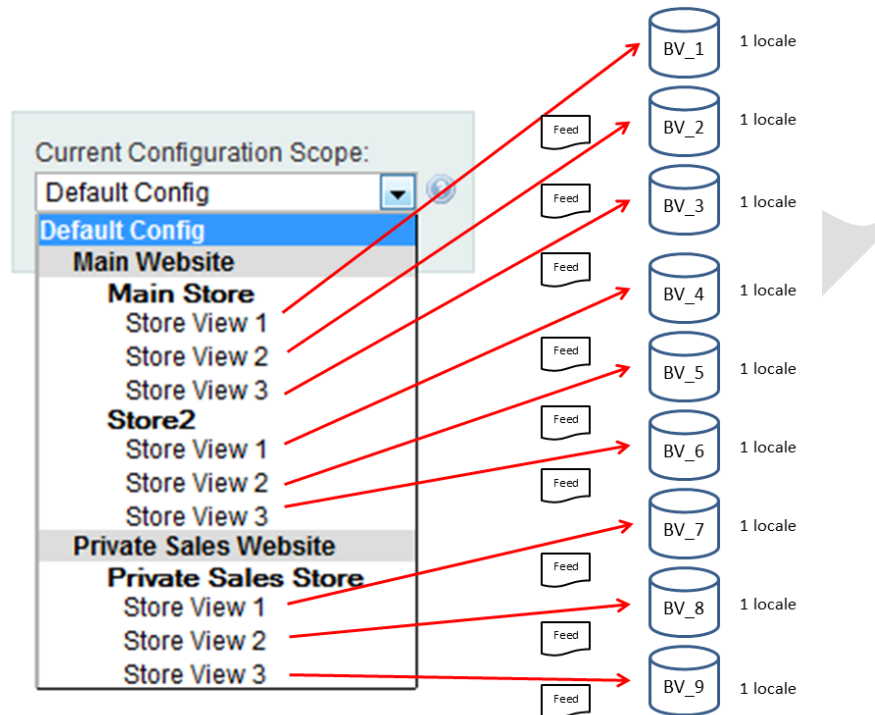
Bottom Screenshot (Store View 2): The 'Current Configuration Scope' is set to 'Store View 2'. The 'Locale' is configured as 'fr_CA'. The 'Deployment Zone' is set to 'Main Site'. A red box highlights the 'Locale' field, and a red arrow points to the text: "Set at Store View scope (must match across all Store Views in Group)".

Both screenshots show the 'General' configuration tab with fields for 'Enable Bazaarvoice Extension', 'Environment', 'Client Name', 'FTP Password', 'Deployment Zone', and 'Locale'. The 'Locale' field is highlighted in red in both screenshots, indicating its importance in the configuration.

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 Views.

Current Configuration Scope: **Store View 1**

Bazaarvoice Save Config

General

Enable Bazaarvoice Extension ☒ Yes Use Website [STORE VIEW]
Enable all Bazaarvoice Magento extension features at this Magento Scope

Environment Use Website [STORE VIEW]
BV environment that this Magento scope is integrating with

Client Name Use Website [STORE VIEW]
Client Name as defined in your BV platform account

FTP Password Use Website [STORE VIEW]
Password used to access the BV FTP server for feed upload/download processes

Deployment Zone Use Website [STORE VIEW]
Deployment Zone Name as defined in your BV platform account

Locale Use Website [STORE VIEW]
The BV locale for this Magento Configuration Scope. For example: "en_US". The locale must be enabled within your BV platform account.

Set at Store View scope
(unique BV client instance values)

Current Configuration Scope: **Store View 2**

Bazaarvoice Save Config

General

Enable Bazaarvoice Extension ☒ Yes Use Website [STORE VIEW]
Enable all Bazaarvoice Magento extension features at this Magento Scope

Environment Use Website [STORE VIEW]
BV environment that this Magento scope is integrating with

Client Name Use Website [STORE VIEW]
Client Name as defined in your BV platform account

FTP Password Use Website [STORE VIEW]
Password used to access the BV FTP server for feed upload/download processes

Deployment Zone Use Website [STORE VIEW]
Deployment Zone Name as defined in your BV platform account

Locale Use Website [STORE VIEW]
The BV locale for this Magento Configuration Scope. For example: "en_US". The locale must be enabled within your BV platform account.

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	**Manufacturer	Brand
This section contains a list of the localized category names that are found at the StoreView configuration scope.	Names	*This is the category ID/Url path of the category that the product is in.	CategoryExternalID
This section contains the category url with the Store View parameter added	CategoryPageUrls	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	Names

		as found in the Store View configuration scope	
		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 the Magento Manufacturer attribute is mapped to the Bazaarvoice Brand attribute. This is configurable to use the value of an 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.

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 created 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 happen 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 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

Appendix D: Advanced Configuration

Putting Bazaarvoice Content Behind a Tab

When Bazaarvoice content is put behind a tab, Bazaarvoice provides the JavaScript callback function `doShowContent()` 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	file_name.xml	You can specify the name of your product feed
Product Feed 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	manufacturer	By default this configuration is set to manufacturer, but can be changed to a field of your choosing if you are not using the built in Magento manufacturer attribute
Category Id Use URL Path	true or false	By default this is set to “true”. When set to true the 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

		<p>shirts is the category.</p> <p>When set to false the actual Category Id will be used.</p> <p>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.</p>
--	--	--

Example code

```
<default>
<bazaarvoice>
<bv_config>
```

```
<!-- Hard code export path and filename in class constants -->
<product_feed_export_filename>zz_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

The following attribute code will be used to locate the brand of products which is sent in the product feed:

```
-->
<product_feed_brand_attribute_code>manufacturer</product_feed_brand_attribute_code>
```

```
<!-- 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>
```

```
</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>
    </bazaarvoice_product_feed>
    <bazaarvoice_purchase_feed>
      <schedule>
        <cron_expr>30 2 * * 6 </cron_expr><!-- Every Saturday at 2:30 AM -->
      </schedule>
    </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.

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**.

Available tasks Generate Schedule Cron Configuration

Select All Unselect All Select Visible Unselect Visible 1 items selected				Actions	Run now	Submit
Code	Cron Expression	Model	Status			
<input type="checkbox"/> aggregate_sales_report_best sellers_data	0 0 ***	sales/observer::aggregateSalesReportBestsellersData	ENABLED			
<input type="checkbox"/> aggregate_sales_report_coupons_data	0 0 ***	salesrule/observer::aggregateSalesReportCouponsData	ENABLED			
<input type="checkbox"/> aggregate_sales_report_invoiced_data	0 0 ***	sales/observer::aggregateSalesReportInvoicedData	ENABLED			
<input type="checkbox"/> aggregate_sales_report_order_data	0 0 ***	sales/observer::aggregateSalesReportOrderData	ENABLED			
<input type="checkbox"/> aggregate_sales_report_refunded_data	0 0 ***	sales/observer::aggregateSalesReportRefundedData	ENABLED			
<input type="checkbox"/> aggregate_sales_report_shipment_data	0 0 ***	sales/observer::aggregateSalesReportShipmentData	ENABLED			
<input type="checkbox"/> aggregate_sales_report_tax_data	0 0 ***	tax/observer::aggregateSalesReportTaxData	ENABLED			
<input type="checkbox"/> aoescheduler_heartbeat	* / 5 *	aoe_scheduler/heartbeatTask::run	ENABLED			
<input type="checkbox"/> bazaarvoice_download_inlineratings	30 1 ***	bazaarvoice/retrieveInLineRatingsFeed::retrieveInLineRatingsFeed	ENABLED			
<input checked="" type="checkbox"/> bazaarvoice_product_feed	0 1 ***	bazaarvoice/exportProductFeed::exportDailyProductFeed	ENABLED			
<input type="checkbox"/> bazaarvoice_purchase_feed	30 2 * * 6	bazaarvoice/exportPurchaseFeed::exportPurchaseFeed	ENABLED			

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`