Bazaarvoice for SAP Hybris Extension Integration Guide

Version: 2.8

Disclaimer

Copyright © 2018 Bazaarvoice.
All rights reserved.

The information in this document:

- Is confidential and intended for Bazaarvoice clients. No part of this document may be shared with anyone outside your company or your company-partners without prior approval from Bazaarvoice.
- Is provided “as is” without warranty of any kind either expressed or implied, including, but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.
- Is periodically updated to be in sync with the improvements and/or changes in our product offers.
- May be changed without notice. Some examples depicted herein are provided for illustration only and are not guarantees of a specific result. You bear the risk of using this document.

Bazaarvoice is not liable for any technical or typographical errors that might be inadvertently present in this document.

Bazaarvoice might have patents and/or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents, or legal rights to any other intellectual property in any Bazaarvoice service or product.

All Bazaarvoice brand and product names are trademarks or registered trademarks of Bazaarvoice in the United States and may be protected as trademarks or registered trademarks in other countries. All other product, service, or company names mentioned here are claimed as trademarks and trade names by their respective companies.

Contact Us:

Bazaarvoice Inc.
10901 Stonelake Blvd.
Austin, TX 78759
Tel (toll-free): (866) 522-9227 | Tel (toll): (512) 551-6000
www.bazaarvoice.com | info@bazaarvoice.com

Table of Contents

Introduction

Mapping a Hybris Environment to the Bazaarvoice Platform
- Multi-site Integrations

Requesting Bazaarvoice to Provision Hosted Services

Plugin update

Installation and Configuration
- Requirements
- Before You Begin
- Installing the Extension
- Configuring Your Installed Bazaarvoice Extension
  - Creating a New Bazaarvoice Configuration
  - Assigning the Bazaarvoice Configuration to your Hybris Website/Storefront
  - Configuring Localization
  - Creating the Product Feed Export CronJob
  - Passing Additional Product IDs to Bazaarvoice
    - Configurable UPC Data Source

Website/Storefront Modifications
- Product Details Page Integration
- Product Listing / Search Results Page Integration
- Implementing Bazaarvoice BV Pixel
- Implementing Bazaarvoice Submission container

Website/Storefront Modifications description
- Master / Page Custom Tag Changes
  - Master.tag File
  - Page.tag File
- Product Listing / Search Results Page Integration
  - 2) productListerItem.tag:
  - Removing Sort by “Top Rated” Feature
- Product Details Page Integration
    - Update productDetailsPanel.tag File
    - Update productReviewsTab.jsp File
- Implementing Bazaarvoice BV Pixel

Appendix A: Explanation of Configuration Settings
- Backoffice Control Panel Values

Local.Properties Configuration Settings
Locating your Locales Enabled within Your Bazaarvoice Instance
Locating your Deployment Zone Name
Locating your Cloud SEO Key

Version History

Introduction

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

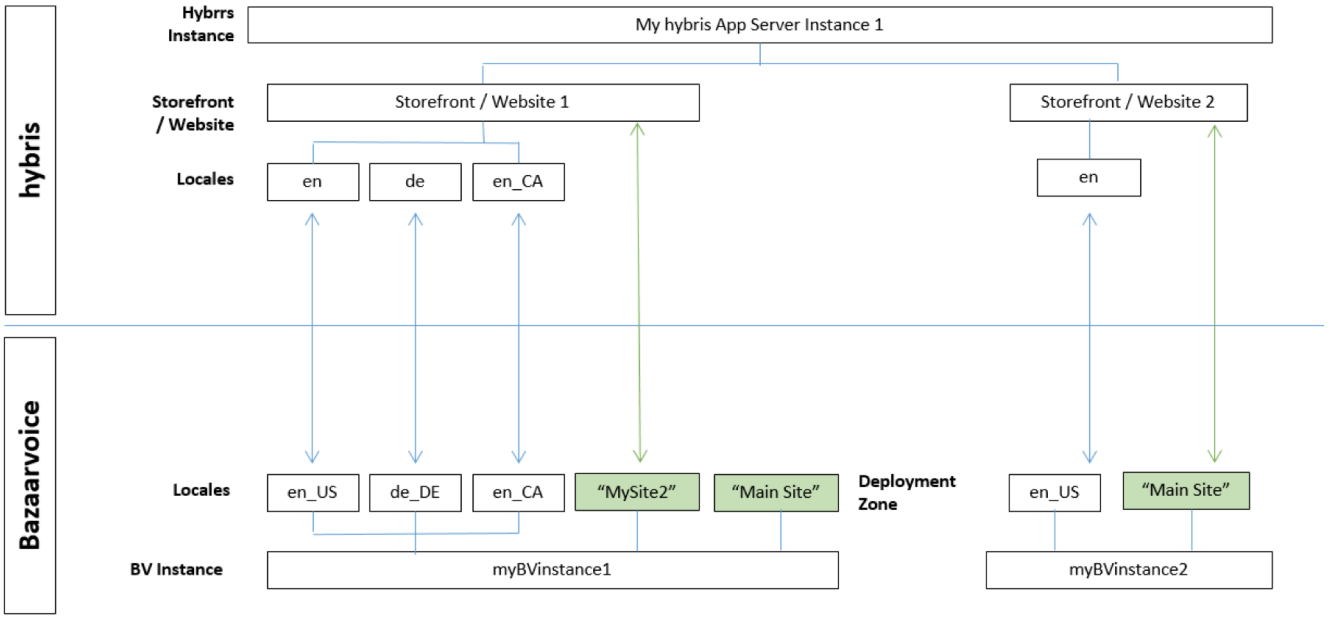
The extension is supplied “as-is” and is intended to function as a starter kit for clients to implement Bazaarvoice. It includes support for commonly required integration points, but does not include pre-built functionality for the exhaustive list of Bazaarvoice features. This extension may need to be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

Mapping a Hybris Environment to the Bazaarvoice Platform

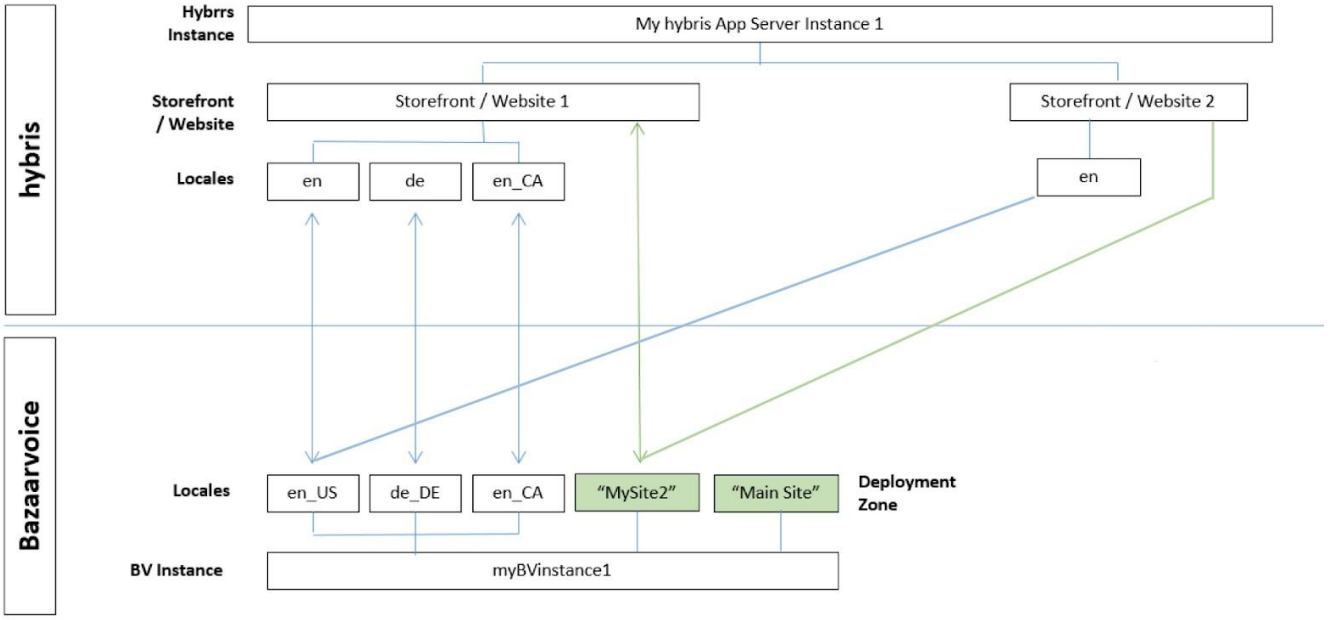
The Bazaarvoice for Hybris extension is architected in the following way:

- **Option 1** Each storefront can have a dedicated BV instance (see diagram)
- **Option 2** All storefronts can be mapped to 1 BV instance (see diagram)
- **Option 3** A combination of the two
- Each Bazaarvoice Instance can support multiple Locales and Deployment Zones
- Locales within a given Hybris Storefront/Website can be mapped to a Locale within a particular Bazaarvoice instance
- Bazaarvoice supports only [language]_[country] locale values (e.g. en_US, fr_FR, etc.). Hybris supports both [language] (e.g. de) as well as [language]_[country] locale values (e.g. en_US, fr_FR, etc.). You can map a language-only Hybris Locale to a language-country combination for Bazaarvoice (e.g. de maps to de_DE).
- Each Hybris Storefront/Website can be mapped to a Deployment Zone within a particular Bazaarvoice instance (Note: a Deployment Zone is a mechanism to create and manage alternate displays within a given Bazaarvoice instance independent of locale configurations).
  The following diagram illustrates this concept:

Each Storefront maps to a unique BV instance.



All storefronts map to 1 BV instance



Multi-site Integrations

If a Hybris client wants to integrate Bazaarvoice into multiple Hybris Sites within a single Hybris app server environment, the following considerations should be made:

- The Backoffice control panel must be configured separately for each Hybris Storefront/Website.
- The Bazaarvoice configuration must be configured to the desired Hybris Storefronts/Websites.
- The product feed CronJob (or DCC if enabled) must be configured for each desired Hybris Storefront/Website.
- Code modifications must be made to each desired Hybris Storefront/Website

Requesting Bazaarvoice to Provision Hosted Services

Use of the Bazaarvoice for Hybris Extension by itself does not grant nor enable use of the Bazaarvoice offerings (e.g. Ratings & Reviews, etc.). Bazaarvoice must separately enable the hosted Bazaarvoice SAAS services for the offerings to available on the Hybris platform. This may require a separate agreement with Bazaarvoice.

Complete the following steps:

1. Contact a Bazaarvoice representative with the following information (for each Hybris Storefront/Website) to provision Bazaarvoice hosted services:
    a. Number of Bazaarvoice instances required
    b. Suggested Client/Brand Name (for each instance)
    c. Domain (for each instance)
    d. Locale(s) required (for each instance)
2. Request login credentials (Client Name, SFTP username and SFTP password) for each BV instance.

Purchasing Access to the Bazaarvoice Platform

You must have a contract with Bazaarvoice for access to the BV platform. Contact partner-support@bazaarvoice.com to establish a contract with Bazaarvoice.

Plugin update

In case you have installed Bazaarvoice extension 1.x version

Before updating the Hybris version and before deleting the old Bazaarvoice extension version from your custom folder and substituting it with a new version, make note of any Bazaarvoice customizations made by your team. Find them, copy them to a safe place and apply again after updating.

Updating the plugin will not remove your previous hmc configuration settings, also all previous hmc configurations will appear in the Backoffice, so you do not need to reconfigure those settings (you can skip the following paragraph: “Configuring Your Installed Bazaarvoice Extension”).

In case you need to pull bug fixes

You need to substitute existing Bazaarvoice extension with new one which contains fixes, before that you need to take care about your own Bazaarvoice customizations if you have such. Find them, copy to safe place and apply again after substitution.

Installation and Configuration

This document highlights how you must install and configure the Bazaarvoice extension for Hybris.

Requirements

The following section details the requirements for the extension.

- Hybris System Requirements
    - Hybris version since 1905 (including Hybris 2005 version)
    - Java 11
    - Storefront accelerator - the extension has been tested on the Hybris B2C accelerator.
        - *It is possible to use the extension on a custom storefront*
- Bazaarvoice Requirements
    - Bazaarvoice Conversations

Installing the Extension

The Bazaarvoice extension requires a working installation of Hybris. If you do not have a working installation of Hybris, please install Hybris per the instructions provided on the Hybris wiki.

1. Download and unpack the Bazaarvoice extension package.
2. Copy the custom folder content within the Bazaarvoice extension download into the custom folder of your ${HYBRIS_BIN} directory. If you do not have a custom folder in your **${HYBRIS_BIN}** directory, create one, or leverage your preferred naming convention for Hybris extensions.
3. Update the **${HYBRIS_CONFIG}\localextensions.xml** for your installed Hybris instance, and add the Bazaarvoice extension as an extension following the standard:

```
<extension name="bazaarvoice" />
```

4. If you do not have any customized backoffice extension in your project, add the bvbackoffice extension to **${HYBRIS_CONFIG}\localextensions.xml** as an extension following the standard:

```
<extension name="bvbackoffice" />
```

5. If you already have your own customized backoffice extension, follow next instructions:

- Copy configuration from file:

/custom/bvbackoffice/resources/bvbackoffice-backoffice-config.xml and add it to your backoffice configuration file:
/custom/{your-backoffice-extension}/resources/{your-backoffice-extension-name}-backoffice-config.xml

- Copy all properties from:
  /custom/bvbackoffice/resources/bvbackoffice-backoffice-labels/labels.properties to the same file in your backoffice extension:
  /custom/{your-backoffice-extension}/resources/{your-backoffice-extension-name}-backoffice-labels/labels.properties

6. Modify your **extensioninfo.xml** file within your storefront extension to include the Bazaarvoice add on as shown in the code snippet below. If you are using the accelerator as a template, the storefront extension is located at **${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront**

```
<requires-extension name="bazaarvoice" />
```

7. Modify your **local.properties** file to define additional configuration settings used for the Product Catalog Feed CronJob. See Appendix A for an explanation of each configuration option.

```
# Bazaarvoice settings
ftpServerDirectory=/import-inbox
ftpServerPort=
xmlBaseFileName=productfeed

#add the following two properties if you want to save product feed files
in project directory
localXmlOutputDirectory=/xmlFolder
localXmlArchiveDirectory=/archiveFolder

#add the following two properties if you want to save product feed files
to some specific directory
specificXmlOutputDirectory=/path/to/dir/xmlFolder
specificXmlArchiveDirectory=/path/to/dir/archiveFolder
```

8. Build your Hybris system using the **ant all** command.

9. Start up your Hybris server, and then execute an **update** selecting **bazaarvoice** and **bvbackoffice** extension from the list of projects as well as **Update running system**, **Create essential data**, **Localize types**

Configuring Your Installed Bazaarvoice Extension

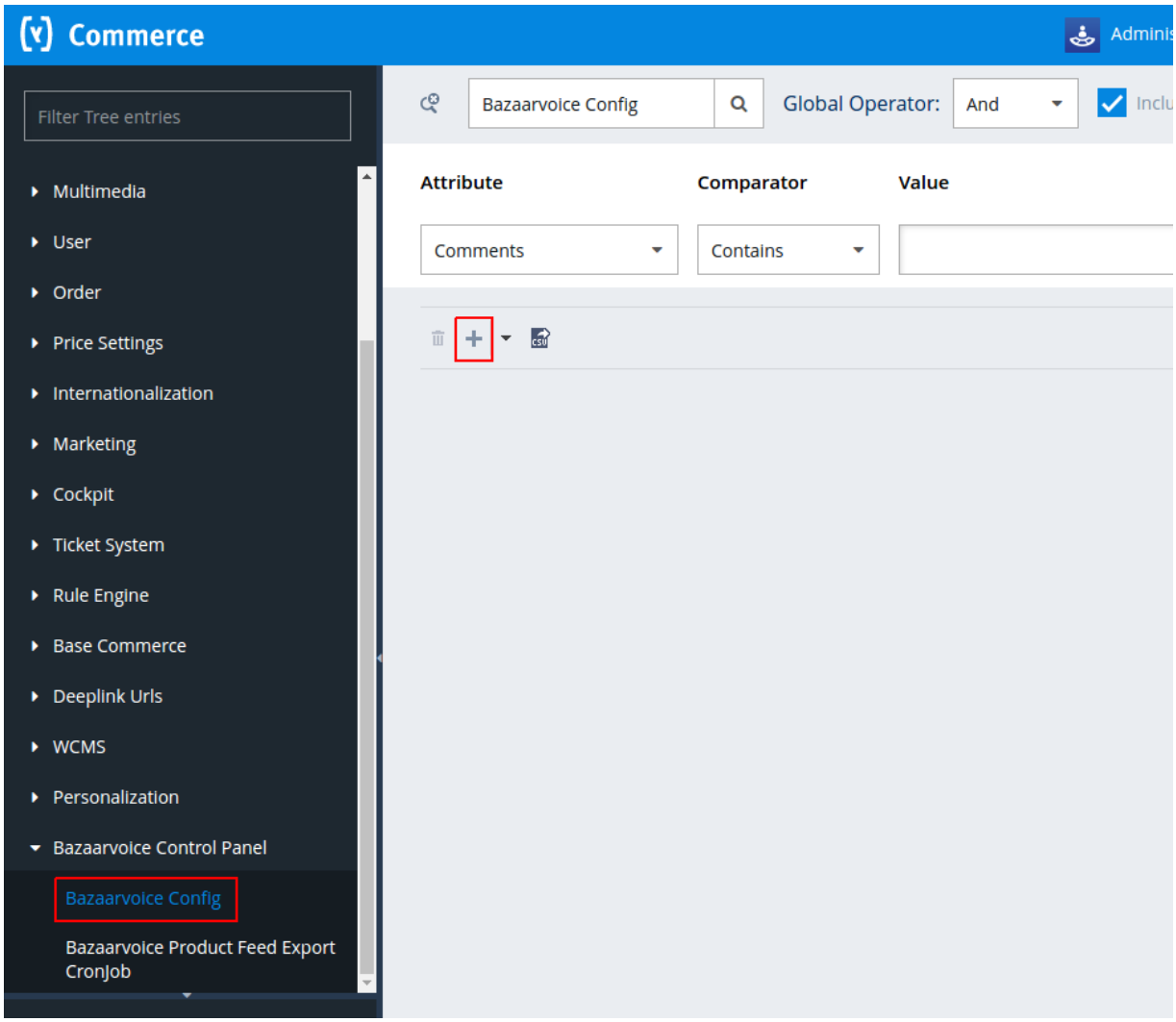
NOTE:
Below section is describing how to create bazaarvoice configuration via Backoffice which is the most descriptive way. However you have an ability to do this via impex scripts as well. Please take a look into
/bazaarvoice/resources/impex/essentialdata_bazaarvoice.impex file to see how to build the configurations via impex script

Creating a New Bazaarvoice Configuration

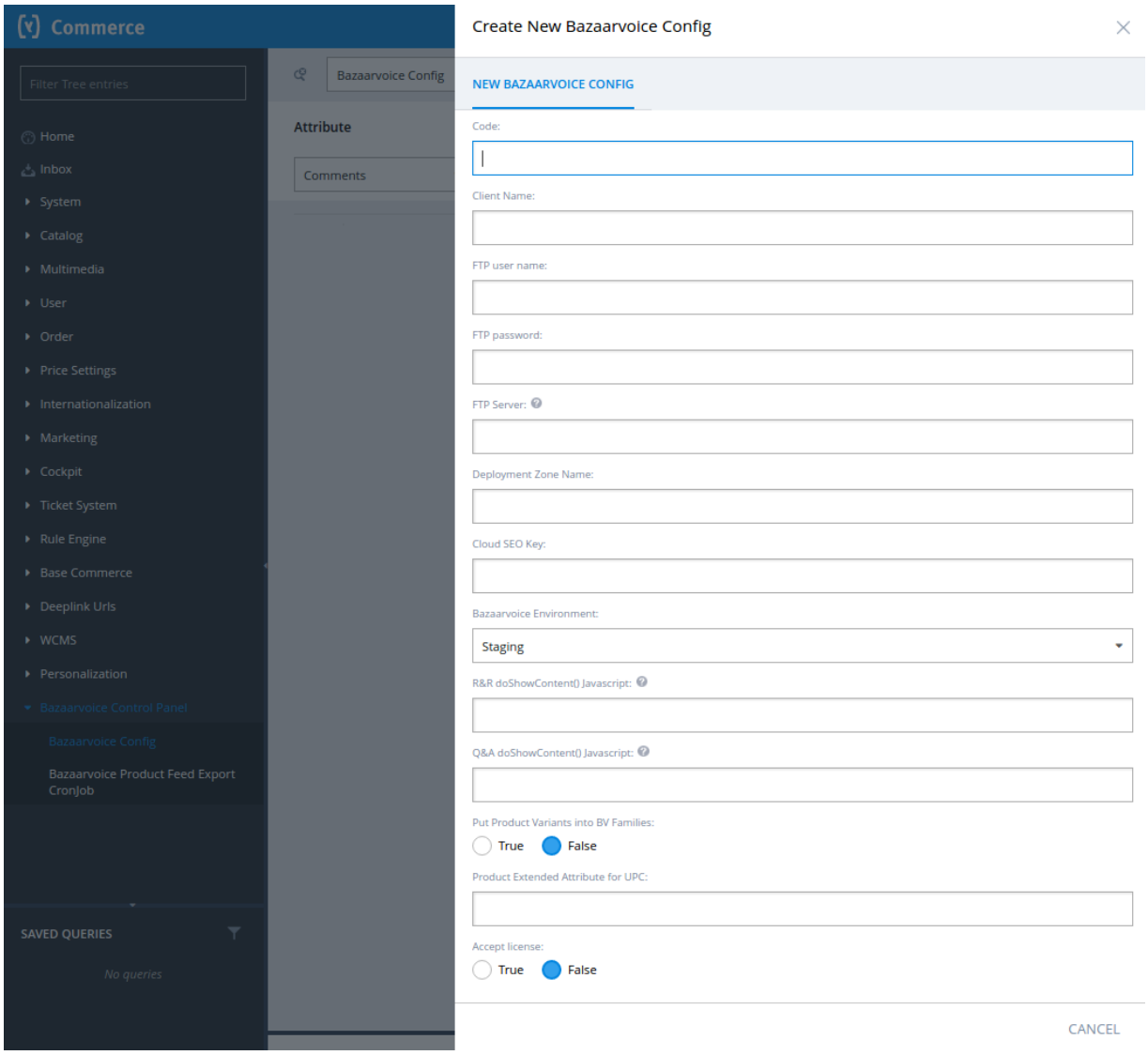
Configuration is required to configure export of products from within a Hybris catalog to the Bazaarvoice SFTP servers. The configurations within the Backoffice holds the data required to connect to the Bazaarvoice SFTP server like the username, password, server location, etc.

1. Login to the **Backoffice** of the Hybris instance with the Bazaarvoice extension installed.

2. Find **Bazaarvoice Control Panel** > **Bazaarvoice Config** > Press "**+**" button

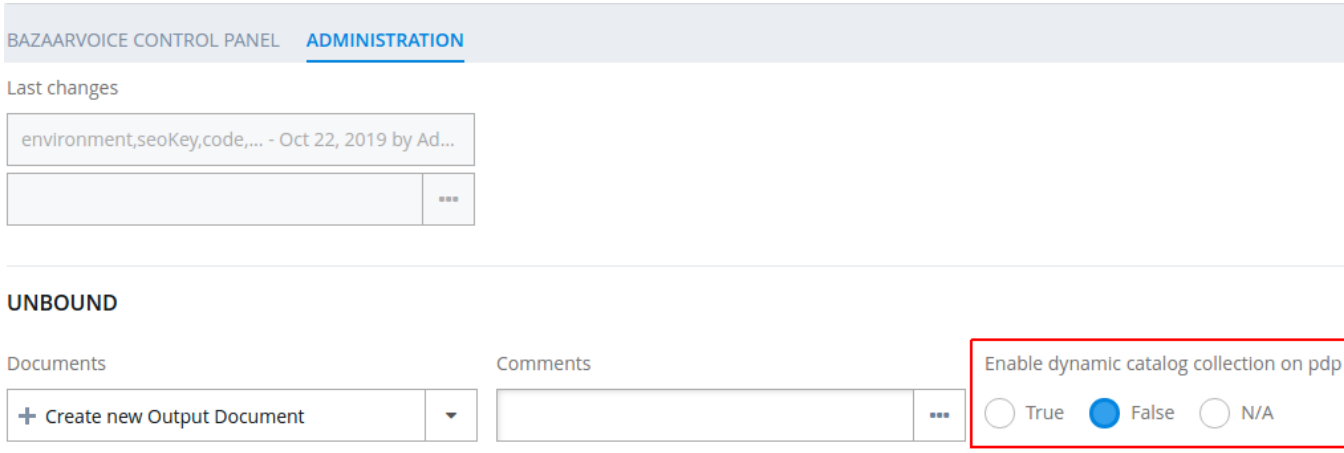


3. Complete the form with the information provided by Bazaarvoice for your specification BV configuration (see **Appendix A** for specific instructions).



4. Check the **Accept License** option.
5. Upon completing the form, please click **DONE** to save the new configuration into the Hybris system. "**DONE"** button appears only after completing form with all required fields. Required field list: **Code, Client Name, FTP user name, FTP password, FTP Server, Deployment Zone Name, Cloud SEO Key, Bazaarvoice Environment**
6. If dynamic catalog collection needs to be disabled, open just created bazaarvoice config, go to the Administration tab and click **"False"** under **"Enable dynamic catalog collection on pdp"**

like on the screenshot below.



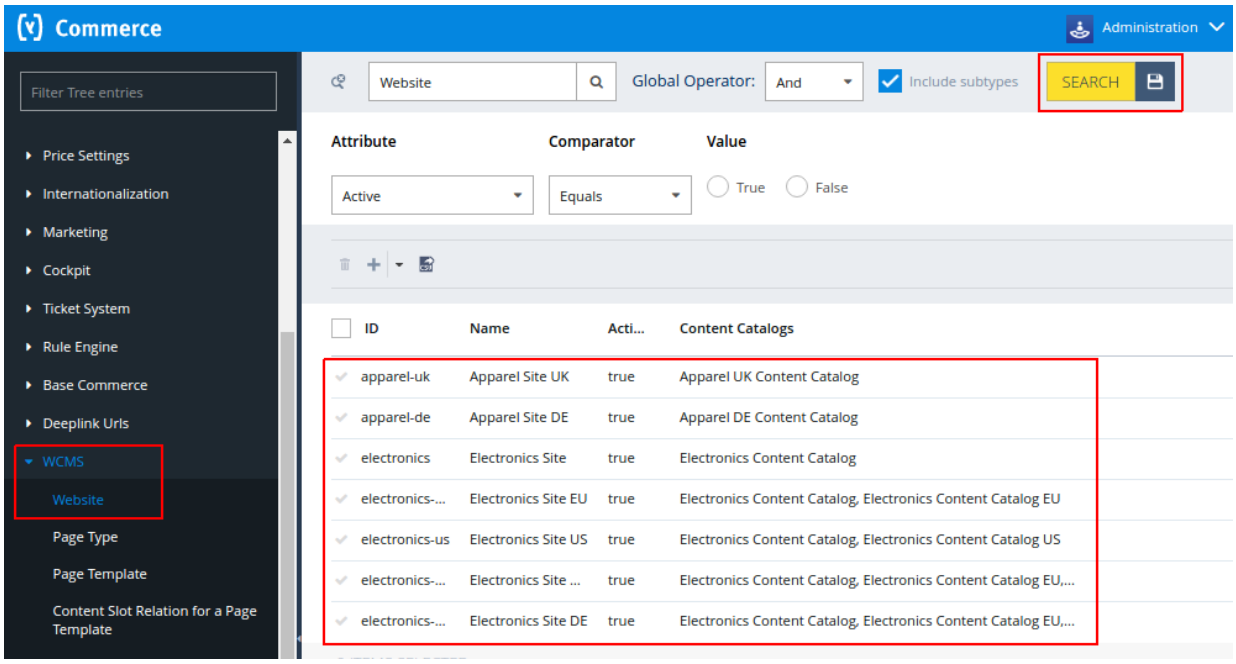
7. **Enable Product Prefix** - *This setting is only required when merging multiple sites into one BV instance.* This setting adds a siteid prefix to the id used to collect reviews on the site. The id can be product id, variant, or variation group. For example: siteid-productid.
8. **Enable Store Level Categories** - *This setting is only required when merging multiple sites into one BV instance.* This adds "Site Name" as a top level category, allowing each site's category structure to be managed separately. This also adds the siteid as a prefix to the category ids, allowing each category to be configured independently.
9. **Aggregate Attributes for Syndication** - This setting affects how reviews will display on syndicated sites. Bazaarvoice syndicates reviews based on the UPC or EAN field. This setting aggregates all of the UPCs or EANs defined across product variants so that each variant id sent to BV includes all the variant UPCs or EANs. For example: A product has three variants. Each variant has 1 UPC defined. This setting will send three UPCs for each variant id.

Assigning the Bazaarvoice Configuration to your Hybris Website/Storefront

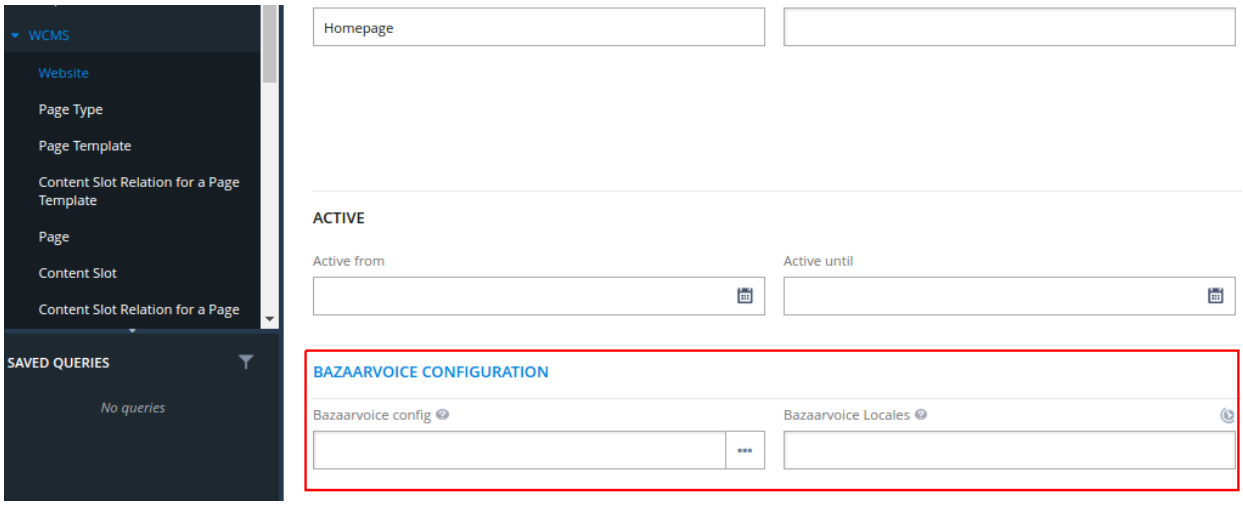
Complete the following steps to assign the Bazaarvoice configuration to your Hybris website/storefront.

1. Stay logged into **Backoffice**.
2. Click **WCMS** > **Website** (in the left navigation menu).

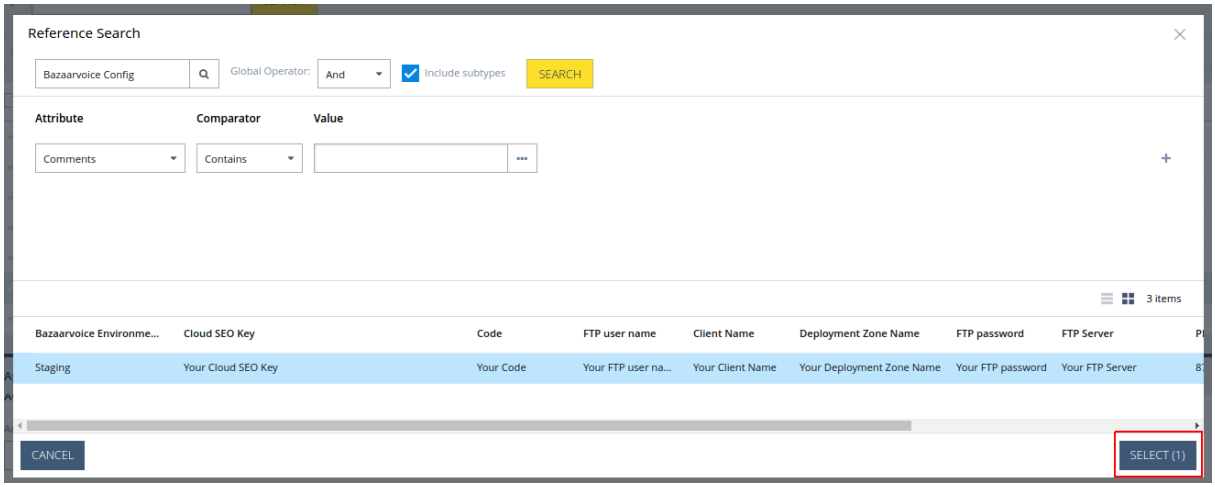
3. From the **Website** section, search for the name of the website that you want to assign to Bazaarvoice.



4. Select the desired Website to open up the Website configuration screen.
5. Click the **Properties** tab for your selected website, and navigate down to the bottom of the page to find the section entitled **Bazaarvoice Configuration**.



6. Click on the Bazaarvoice Config "**...**" button to open up the Backoffice's search interface to find the Bazaarvoice Config object you just created.

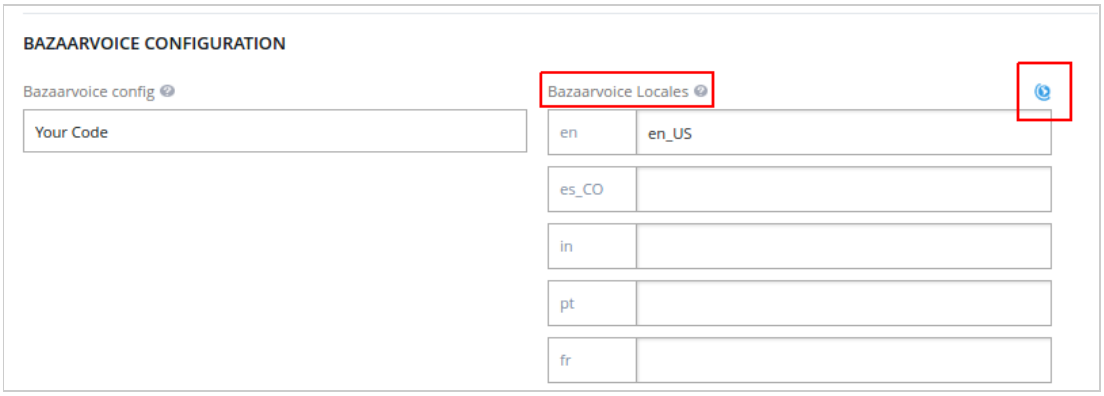


7. Select the object, and the Backoffice will attach it to the web site. After selecting a Bazaarvoice configuration object, click **SELECT**.

Configuring Localization

The Bazaarvoice for Hybris extension is built to support multi-locale implementations. Within Hybris, each Hybris Website/Storefront can be configured to support multiple locales. You must have this list of locales explicitly enabled within the corresponding Bazaarvoice instance.

1. Log into the **Backoffice**.
2. Click **WCMS** > **Website**
3. Select the desired **Website**, and click the **Properties** tab.
4. (Scroll to the bottom) Under the **Bazaarvoice Configuration** section, for each Hybris language/locale enabled, enter the desired Bazaarvoice locale. These should be [language]_[country] combination locales (e.g. en_US, fr_CA, de_DE, etc.).



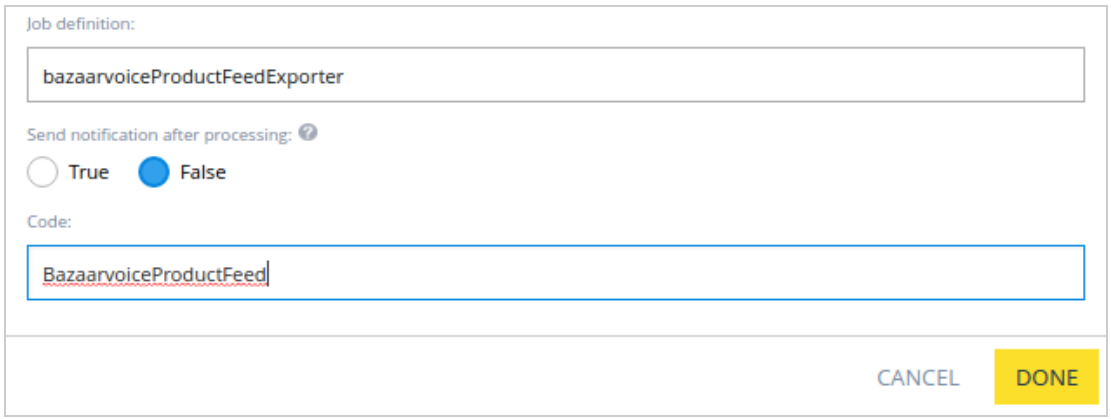
IMPORTANT: Provide this list of locales to your Bazaarvoice representative so they can be enabled in

the Bazaarvoice platform. See Appendix A for how to determine the locales currently enabled in the Bazaarvoice platform.

Creating the Product Feed Export CronJob

The Bazaarvoice for Hybris extension creates a new section within the Backoffice specifically for the product catalog feed required by Bazaarvoice.

1. Login to the **Backoffice**.
2. Click **Bazaarvoice Control Panel** > **Bazaarvoice Product Feed Export CronJob**.
3. Click "**+**" button (at top).
4. Fill fields according to the following:
   Code: (call it anything)
   Job definition: select **bazaarvoiceProductFeedExporter** from pull-down list



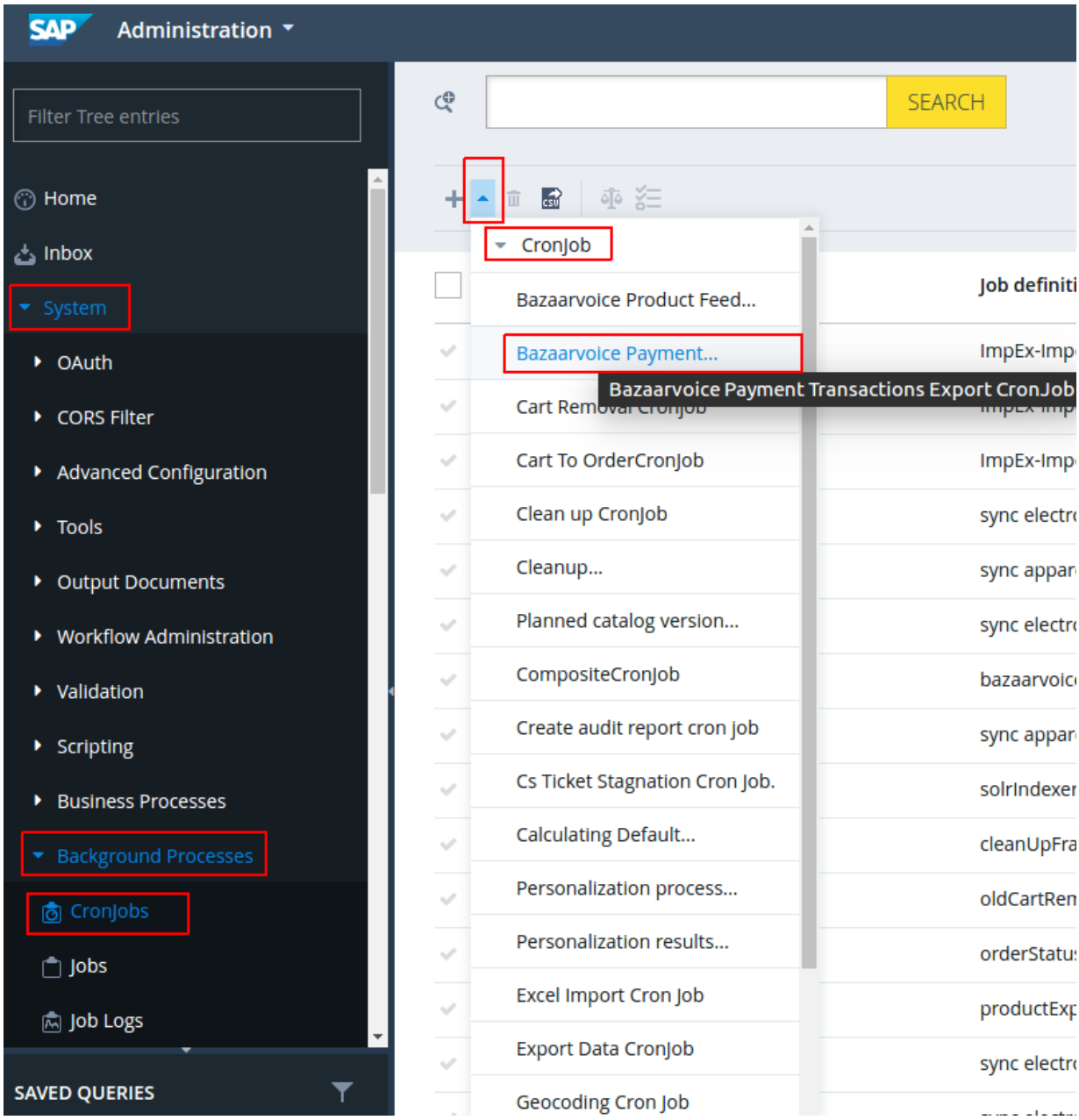
   Set the **hybris Website/Storefront** variable to be the desired Hybris Website/Storefront as the data source of the product feed.

   Leave all other fields with their default values.
5. Click **DONE**.

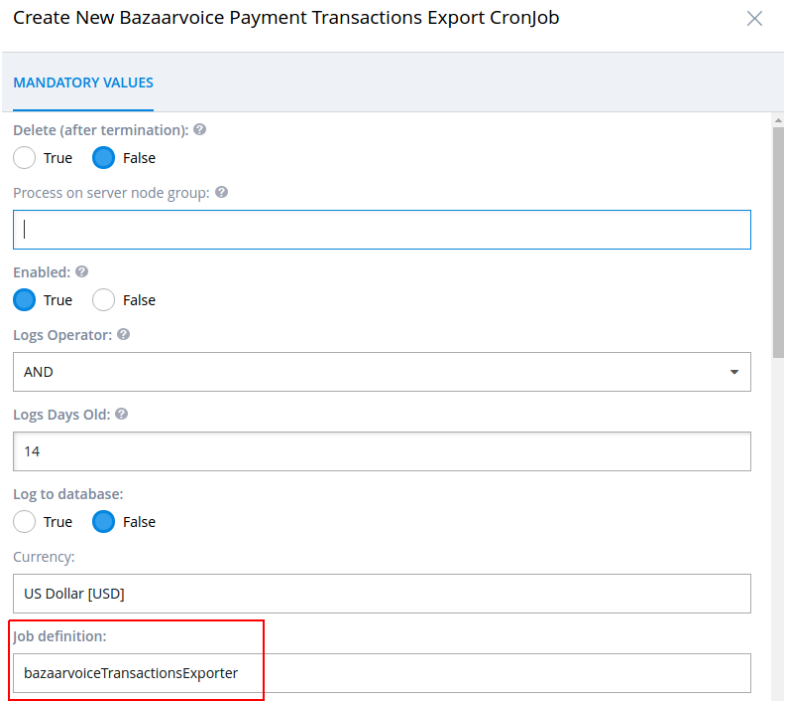
	Recommended Time for Product Feed Exports While you can schedule your CronJob to run at any time you desire, it is recommended to have the Product Feed export run and be completed before Midnight (12:00AM) in your local region. This is when Bazaarvoice begins to process the feeds uploaded to them during that day. Contact your Bazaarvoice representative for the recommended time based on your location.

Creating the Payment Transactions Export CronJob

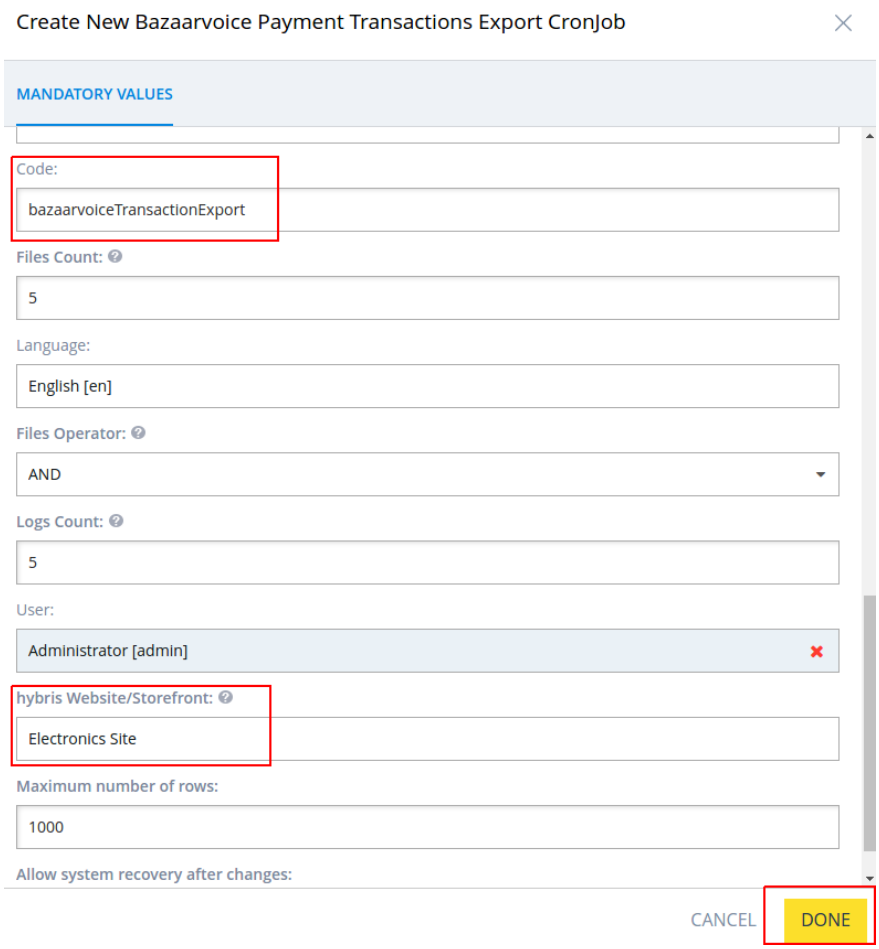
Go to Backoffice -> System -> Background processes -> Cronjobs.
Click create cronjob like described on screenshot below.



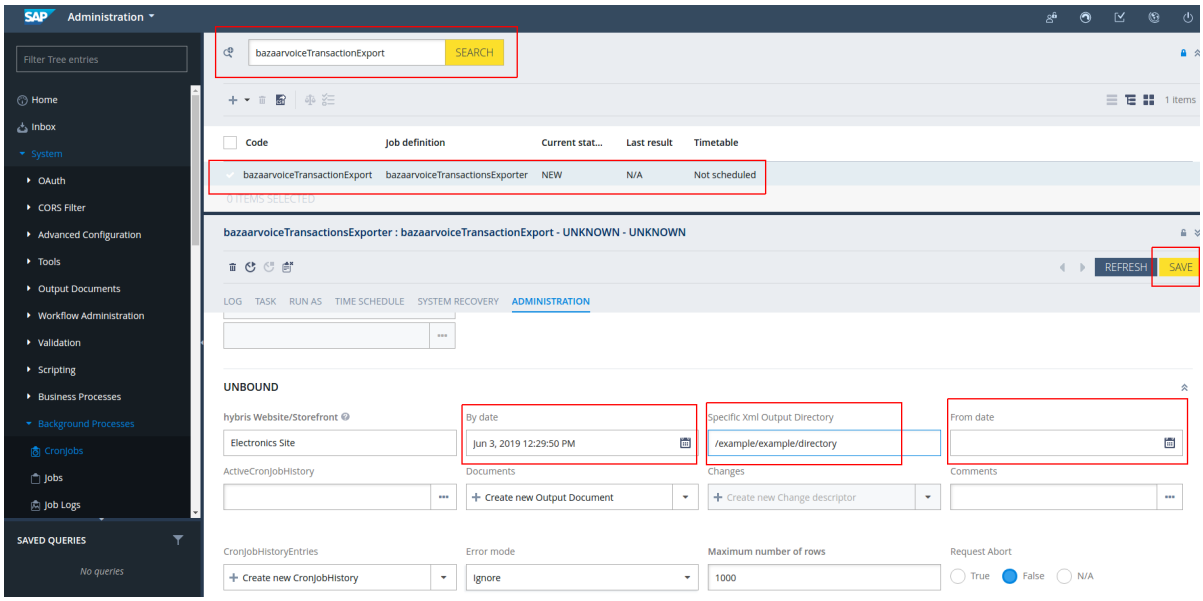
The following window will be displayed.
In the **Job definition** field type: **bazaarvoiceTransactionsExporter.**



Scroll down and fill **Code** field with any value (for example: bazaarvoiceTransactionExport).
Choose desired storefront in **hybris Website/Storefront** field and click **DONE** button.



At this point, cronjob is created and can be found by search.
Find cronjob, and go to **Administration** tab like on the screenshot below.



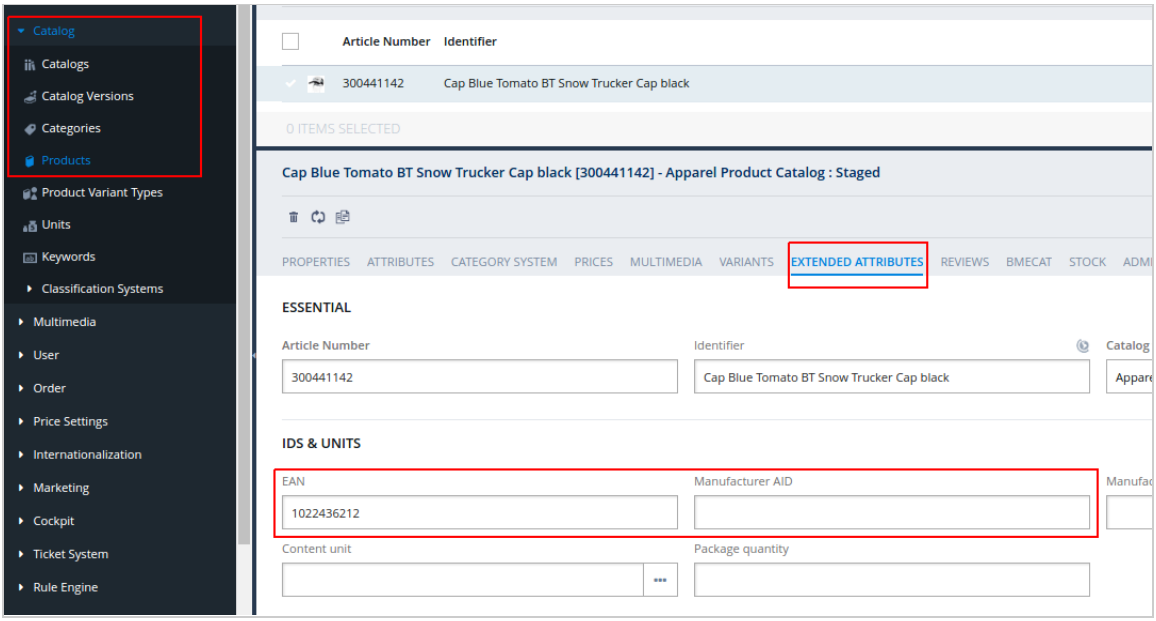
Desired period can be adjusted in “**From Date**” and “**By Date**” fields of cronjob. If you left “**From Date**” and “**By Date**” empty, cronjob will export all of transactions.
Also, those fields function separately from each other, and they can be specified separately. For example, if you enter only “**From Date**” and leave “**By Date**” empty, all transactions from that date until now will be exported.
By default transactions is being exported by transaction creation date. By setting “**Filter by shipping date**” field to true, transactions will be exported by shipping date, desired period can be adjusted in “**From Date**” and “**By Date**” fields. ”
Note: “**Filter by shipping date**” should only be used if using the native Hybris shipping model "consignment". In this case, transaction will be exported if it’s order has DeliveryStatus = Shipped.

“**Export yesterday’s transactions**” - by setting this to true, only transactions that have been created yesterday will be exported. Note, if this field is true the following field’s values become unnecessary: “**From Date**”, “**By Date**” and “**Filter by shipping date**”, hence they will be ignored by cronjob.
“**Ftp server directory**” - directory on Bazaarvoice’s ftp server where transactions will be exported to. Default vaue is “/ppe/inbox”

Passing Additional Product IDs to Bazaarvoice

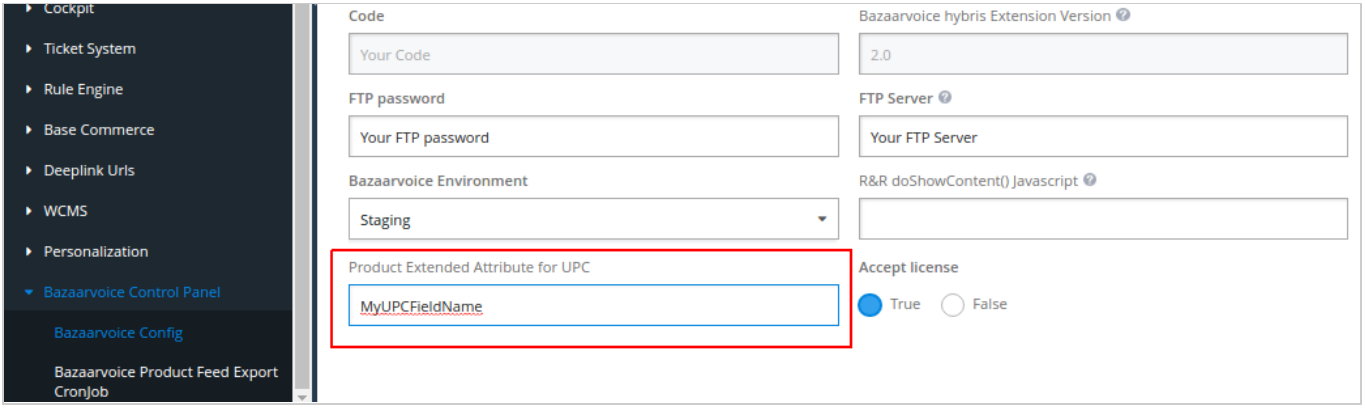
The Bazaarvoice for Hybris extension supports providing additional product IDs to Bazaarvoice (e.g EAN, Manufacturer Part Number, UPC, etc.), which can assist with enabling participation in the Bazaarvoice Network.

Hybris Source Data Field	Bazaarvoice Product Feed XML Export
EAN	EAN
Manufacturer AID	Manufacturer Part Number
(configurable)	UPC



Configurable UPC Data Source

The Hybris platform does not have a default standard field for UPC data values. A Hybris customer must insert UPC data into a custom field. The Bazaarvoice for Hybris extension can be configured to source UPC data from this custom field and insert that data into the Bazaarvoice product feed XML export as UPC. Enter the custom field name in the Bazaarvoice Config Backoffice panel “Product Extended Attribute for UPC” setting. This setting is not case sensitive.



Website/Storefront Modifications

Information below describes bazaarvoice tags and modifications which should be applied to your storefront from Bazaarvoice add-on.

Bazaarvoice tags description

Bazaarvoice tags enable bv functionality within storefront. Table below describes each of Bazaarvoice tag

Tag Name	Description
productPageReviewDetailsTab.tag	Includes the Bazaarvoice Ratings & Reviews detail tab onto a page
productReviewSummary.tag	Inserts the Bazaarvoice Summary content
productPageQuestionAndAnswerTab.tag	Includes the Bazaarvoice Question & Answer detail tab onto a page
cloudSeoInclude.tag	Enables BV Cloud SEO functionality onto PDP*
inlineRatings.tag	This tag allows you to include the BV Inline Ratings onto a page
trackTransactionPageView.tag	Enables BV Pixel functionality on a page
dynamicCatalogCollection.tag	Enables BV DCC on PDP page

If you have another pages which need bv functionality, you can include the Bazaarvoice code in each of the views by just placing the custom tag where you would like to see Bazaarvoice content
If you want to use bazaarvoice with your own storefront you can copy this tags to your storefront and plug them to your pages (see information below how to do that)

All this tags located in the following package:
bazaarvoice/acceleratoraddon/web/webroot/WEB-INF/tags/responsive/bazaarvoice

Information below describe what modifications should be applied to accelerator-based storefront. You can update custom storefront as well by take this instructions as an example.

Master / Page Custom Tag Changes

The Bazaarvoice add-on works best when changes are made to the master.tag and page.tag custom tags provided by the Hybris accelerator. Making changes to these pages allows the Bazaarvoice JavaScript code to reside within the <head> … </head> section of the compiled page.

Updated files

- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\tags\responsive\template\master.tag
- ${HYBRIS_BIN}/ext-template/yacceleratorstorefront/web/webroot/WEB-INF/tags/responsive/template/page.tag

Master.tag File

The master.tag file should be updated to include the Bazaarvoice JavaScript needed by the various elements throughout the pages. The changes for this file are in the listing below:

```
<%@ tag body-content="scriptless" trimDirectiveWhitespaces="true" %>
<%@ attribute name="pageTitle" required="false" rtexprvalue="true" %>
<%@ attribute name="metaDescription" required="false" %>
<%@ attribute name="metaKeywords" required="false" %>
<%@ attribute name="pageCss" required="false" fragment="true" %>
<%@ attribute name="pageScripts" required="false" fragment="true" %>

<%@ taglib prefix="template" tagdir="/WEB-INF/tags/responsive/template" %>
<%@ taglib prefix="analytics" tagdir="/WEB-INF/tags/shared/analytics" %>
<%@ taglib prefix="addonScripts" tagdir="/WEB-INF/tags/responsive/common/header" %>
<%@ taglib prefix="generatedVariables" tagdir="/WEB-INF/tags/shared/variables" %>
<%@ taglib prefix="debug" tagdir="/WEB-INF/tags/shared/debug" %>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="htmlmeta" uri="http://hybris.com/tld/htmlmeta"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>

<%--BAZAARVOICE_MODIFICATION_START--%>
<%@ attribute name="showBV" required="false" %>
<%@ taglib prefix="bvJavascript"
tagdir="/WEB-INF/tags/addons/bazaarvoice/responsive/template"%>
<%--BAZAARVOICE_MODIFICATION_END--%>

<spring:htmlEscape defaultHtmlEscape="true" />

<!DOCTYPE html>
```

```
<html lang="${fn:escapeXml(currentLanguage.isocode)}">
<head>
  <title>
     ${not empty pageTitle ? pageTitle : not empty cmsPage.title ?
fn:escapeXml(cmsPage.title) : 'Accelerator Title'}
  </title>
  <%--BAZAARVOICE_MODIFICATION_START--%>
  <c:if test="${showBV == 'true'}">
     <bvJavascript:bazaarvoiceHeaderJavascript />
  </c:if>
  <%--BAZAARVOICE_MODIFICATION_END--%>
```

Note: *Only the lines in color and comments have added. The lines in black are native. This rule is fair for the all next files as well.*

Page.tag File

The page.tag file should be updated to include the Bazaarvoice JavaScript needed by the various elements throughout the pages. The changes for this file are in the listing below:

```
<%@ tag body-content="scriptless" trimDirectiveWhitespaces="true"%>
<%@ attribute name="pageTitle" required="false" rtexprvalue="true"%>
<%@ attribute name="pageCss" required="false" fragment="true"%>
<%@ attribute name="pageScripts" required="false" fragment="true"%>
<%@ attribute name="hideHeaderLinks" required="false"%>

<%@ taglib prefix="template" tagdir="/WEB-INF/tags/responsive/template"%>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="header" tagdir="/WEB-INF/tags/responsive/common/header"%>
<%@ taglib prefix="footer" tagdir="/WEB-INF/tags/responsive/common/footer"%>
<%@ taglib prefix="common" tagdir="/WEB-INF/tags/responsive/common"%>
<%@ taglib prefix="cart" tagdir="/WEB-INF/tags/responsive/cart" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>

<spring:htmlEscape defaultHtmlEscape="true" />

<%--BAZAARVOICE_MODIFICATION_START--%>
<%@ attribute name="showBV" required="false" %>
<%--<template:master pageTitle="${pageTitle}">--%>
<%--BAZAARVOICE_MODIFICATION_END after the next line--%>
<template:master pageTitle="${pageTitle}" showBV="${showBV}">
```

These pages create an attribute that can now be added into any template called “showBV”. This

attribute allows the developer to choose what pages include the Bazaarvoice JavaScript by adding in the attribute showBV and setting it to true within the template:page tag within the accelerator. The code listing below shows an example:

```
<template:page pageTitle="${pageTitle}" showBV="true">
```

Product Listing / Search Results Page Integration

To include Bazaarvoice capabilities on product listing, and search results page, first page templates have modified to state they are going to leverage Bazaarvoice. This is done by adding in the showBV attribute to the page code as detailed in the previous section. The following pages should be updated to include Bazaarvoice:

- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\views\responsive\pages\category\productListPage.jsp
- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\views\responsive\pages\category\productGridPage.jsp

The code below shows this update for both files:

```
<%@ page trimDirectiveWhitespaces="true" %>
<%@ taglib prefix="template" tagdir="/WEB-INF/tags/responsive/template" %>
<%@ taglib prefix="cms" uri="http://hybris.com/tld/cmstags" %>

<%--BAZAARVOICE_MODIFICATION_START--%>
<%--<template:page pageTitle="${pageTitle}">--%>
<template:page pageTitle="${pageTitle}" showBV="true">
  <%--BAZAARVOICE_MODIFICATION_END--%>
```

To place Bazaarvoice on Product Listing Pages, the following files should be updated to include the Bazaarvoice components.

Updated files:

- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\tags\responsive\product\productListerGridItem.tag
- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\tags\responsive\product\productListerItem.tag

The following code block shows how the hybris provided inline ratings replaced with the Bazaarvoice inline ratings within each of the tag files:

1) productListerGridItem.tag:

```
<%@ tag body-content="empty" trimDirectiveWhitespaces="true" %>
<%@ attribute name="product" required="true"
type="de.hybris.platform.commercefacades.product.data.ProductData" %>
<%@ taglib prefix="product" tagdir="/WEB-INF/tags/responsive/product" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="action" tagdir="/WEB-INF/tags/responsive/action" %>
<%@ taglib prefix="ycommerce" uri="http://hybris.com/tld/ycommercetags" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>

<spring:htmlEscape defaultHtmlEscape="true" />

<%--BAZAARVOICE_MODIFICATION_START--%>
<%@ taglib prefix="bazaarvoice"
tagdir="/WEB-INF/tags/addons/bazaarvoice/responsive/bazaarvoice"%>
<%--BAZAARVOICE_MODIFICATION_END--%>

<spring:theme code="text.addToCart" var="addToCartText"/>
<c:url value="${product.url}" var="productUrl"/>
<c:set value="${not empty product.potentialPromotions}" var="hasPromotion"/>

<c:set value="product-item" var="productTagClasses"/>
<%--BAZAARVOICE_MODIFICATION_START--%>
<bazaarvoice:inlineRatings product="${product}"/>
<%--BAZAARVOICE_MODIFICATION_END--%>
```

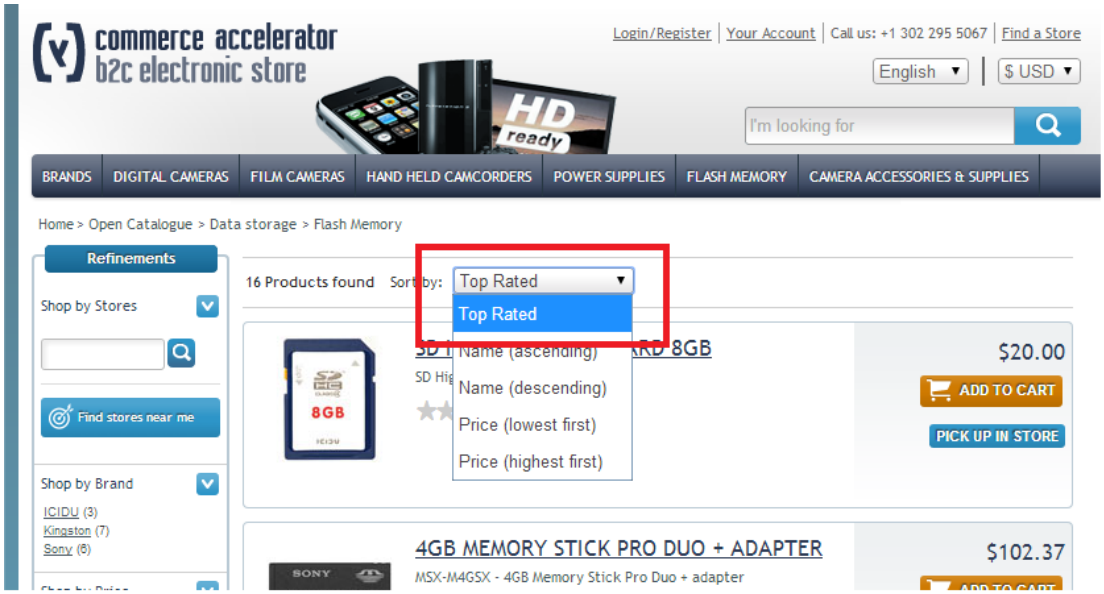
2) productListerItem.tag:

```
<%@ tag body-content="empty" trimDirectiveWhitespaces="true" %>
<%@ attribute name="product" required="true"
type="de.hybris.platform.commercefacades.product.data.ProductData" %>
<%@ taglib prefix="product" tagdir="/WEB-INF/tags/responsive/product" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="action" tagdir="/WEB-INF/tags/responsive/action" %>
<%@ taglib prefix="ycommerce" uri="http://hybris.com/tld/ycommercetags" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
```

```
<%--BAZAARVOICE_MODIFICATION_START--%>
<%@ taglib prefix="bazaarvoice"
tagdir="/WEB-INF/tags/addons/bazaarvoice/responsive/bazaarvoice"%>
<%--BAZAARVOICE_MODIFICATION_END--%>

<spring:htmlEscape defaultHtmlEscape="true" />

<spring:theme code="text.addToCart" var="addToCartText"/>
<c:url value="${product.url}" var="productUrl"/>

<c:set value="${not empty product.potentialPromotions}" var="hasPromotion"/>

<c:set value="product__list--item" var="productTagClasses"/>

<%--BAZAARVOICE_MODIFICATION_START--%>
<bazaarvoice:inlineRatings product="${product}"/>
<%--BAZAARVOICE_MODIFICATION_END--%>
```

Removing Sort by “Top Rated” Feature

The default B2C accelerator includes an option to sort products by “Top Rated” in the Category Listing Page.



Because Bazaarvoice ratings information is not imported and indexed as part of the integration, this sort option is not possible with the default Bazaarvoice extension.

The following is the IMPEX snippet that the customer will need to run from the hybris Administration Console (hAC) to remove the sort by option. You should replace the **electronicsProductType** with the product type for your Website/Storefront.

```
$solrIndexedType=electronicsProductType
REMOVE SolrSort;&sortRefID;indexedType(identifier)[unique=true];code[unique=true]
;sortRef2;$solrIndexedType;topRated
```

Product Details Page Integration

To include Bazaarvoice capabilities on product details pages, page templates should be modified to state they are going to leverage Bazaarvoice. This should be done by adding in the showBV attribute to the page code as detailed earlier in this document. The following pages should be updated to include Bazaarvoice:

- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\views\responsive\pages\product\productLayout1Page.jsp
- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\views\responsive\pages\product\productLayout2Page.jsp

The code below shows this update for both files:

```
<%@ page trimDirectiveWhitespaces="true"%>
<%@ taglib prefix="template" tagdir="/WEB-INF/tags/responsive/template"%>
<%@ taglib prefix="cms" uri="http://hybris.com/tld/cmstags"%>
<%@ taglib prefix="product" tagdir="/WEB-INF/tags/responsive/product"%>

<%--BAZAARVOICE_MODIFICATION_START--%>
<%--<template:page pageTitle="${pageTitle}">--%>
<template:page pageTitle="${pageTitle}" showBV="true">
  <%--BAZAARVOICE_MODIFICATION_END--%>
```

Update productDetailsPanel.tag File

- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\tags\responsive\product\productDetailsPanel.tag

The following code block shows the changes within the **productDetailsPanel.tag**.

```
<%@ tag body-content="empty" trimDirectiveWhitespaces="true"%>
<%@ taglib prefix="product" tagdir="/WEB-INF/tags/responsive/product"%>
<%@ taglib prefix="cms" uri="http://hybris.com/tld/cmstags"%>
<%@ taglib prefix="ycommerce" uri="http://hybris.com/tld/ycommercetags"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>

<%--BAZAARVOICE_MODIFICATION_START--%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="bazaarvoice"
      tagdir="/WEB-INF/tags/addons/bazaarvoice/responsive/bazaarvoice"%>
<bazaarvoice:cloudSeoInclude product="${product}"/>

<c:if test="${cmsSite.bvConfig.dccEnabled}">
<bazaarvoice:dynamicCatalogCollection bazaarvoiceDCCProductData="${bazaarvoiceDCCProductData}"/>
</c:if>
<%--BAZAARVOICE_MODIFICATION_END--%>

<div class="product-details page-title">
  <ycommerce:testId code="productDetails_productNamePrice_label_${product.code}">
    <div class="name">${fn:escapeXml(product.name)}<span class="sku">ID</span><span
class="code">${fn:escapeXml(product.code)}</span></div>
  </ycommerce:testId>
  <%--BAZAARVOICE_MODIFICATION_START--%>
  <%--<product:productReviewSummary product="${product}" showLinks="true"/>--%>
  <bazaarvoice:productReviewSummary product="${product}" />
  <%--BAZAARVOICE_MODIFICATION_END--%>
```

Update productReviewsTab.jsp File

- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\views\responsive\pages\product\productReviewsTab.jsp

```
<%@ page trimDirectiveWhitespaces="true"%>
<%@ taglib prefix="product" tagdir="/WEB-INF/tags/responsive/product"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
```

```
<%--BAZAARVOICE_MODIFICATION_START--%>
<%@ taglib prefix="bazaarvoice"
        tagdir="/WEB-INF/tags/addons/bazaarvoice/responsive/bazaarvoice" %>
<%--BAZAARVOICE_MODIFICATION_END--%>

<div id="tabreview" class="tabhead">
  <a href="">${fn:escapeXml(title)}</a> <span class="glyphicon"></span>
</div>
<div class="tabbody">
  <div class="container-lg">
     <div class="row">
        <div class="col-md-6 col-lg-4">
           <div class="tab-container">
              <%--BAZAARVOICE_MODIFICATION_START--%>
              <%--<product:productPageReviewsTab product="${product}" />--%>
              <bazaarvoice:productPageReviewDetailsTab product="${product}" />
              <%--BAZAARVOICE_MODIFICATION_END--%>
```

Add bazaarvoice Questions & Answers tab:

For adding bazaarvoice Q&A tab the following impex should be executed in hac. Then the same for online catalog version

```
$contentCatalog=electronicsContentCatalog
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCa
talog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged
]
$lang=en

#Create new Q&A component and add to TabPanelContainer
INSERT_UPDATE
JspIncludeComponent;$contentCV[unique=true];uid[unique=true];name;title[lang=$
lang];page;actions(uid,$contentCV);
;;BazaarVoiceProductQATabComponent;BazaarVoice Product QA Tab;Questions And
Answers;/WEB-INF/views/addons/bazaarvoice/responsive/pages/product/bazaarvoice
QATab.jsp;;
INSERT_UPDATE
CMSTabParagraphContainer;$contentCV[unique=true];uid[unique=true];name;visible
;simpleCMSComponents(uid, $contentCV);&componentRef
;;TabPanelContainer;Tab
container;true;ProductDetailsTabComponent,ProductSpecsTabComponent,deliveryTab
,ProductReviewsTabComponent,BazaarVoiceProductQATabComponent;TabPanelContainer
```

Implementing Bazaarvoice BV Pixel

Update file:

- ${HYBRIS_BIN}\modules\base-accelerator\yacceleratorstorefront\web\webroot\WEB-INF\views\responsive\pages\checkout\checkoutConfirmationLayoutPage.jsp

The code below shows this update:

```
<%@ page trimDirectiveWhitespaces="true"%>
<%@ taglib prefix="template" tagdir="/WEB-INF/tags/responsive/template"%>
<%@ taglib prefix="cms" uri="http://hybris.com/tld/cmstags"%>

<%--BAZAARVOICE_MODIFICATION_START--%>
<%@ taglib prefix="bazaarvoice"
tagdir="/WEB-INF/tags/addons/bazaarvoice/responsive/bazaarvoice"%>
<%--BAZAARVOICE_MODIFICATION_END--%>

<%--BAZAARVOICE_MODIFICATION_START--%>
<template:page pageTitle="${pageTitle}" showBV="true">
   <%--BAZAARVOICE_MODIFICATION_END--%>
   <cms:pageSlot position="SideContent" var="feature"
class="accountPageSideContent" element="div">
       <cms:component component="${feature}" element="div"
class="accountPageSideContent-component"/>
   </cms:pageSlot>
   <cms:pageSlot position="TopContent" var="feature" element="div"
class="accountPageTopContent">
       <cms:component component="${feature}" element="div"
class="accountPageTopContent-component"/>
   </cms:pageSlot>
   <div class="account-section">
       <cms:pageSlot position="BodyContent" var="feature" element="div"
class="account-section-content checkout__confirmation__content">
           <cms:component component="${feature}" element="div"
class="checkout__confirmation__content--component"/>
       </cms:pageSlot>
   </div>
   <cms:pageSlot position="BottomContent" var="feature" element="div"
class="accountPageBottomContent">
       <cms:component component="${feature}" element="div"
class="accountPageBottomContent-component"/>
```

```
    </cms:pageSlot>

    <%--BAZAARVOICE_MODIFICATION_START--%>
    <bazaarvoice:trackTransactionPageView order="${orderData}"/>
    <%--BAZAARVOICE_MODIFICATION_END--%>
```

Add submission container page:

Execute the following impex in hac (for stage and online catalog version):

```
#SUBMISSION CONTAINER PAGE
$contentCatalog=electronicsContentCatalog
$contentCV=catalogVersion(CatalogVersion.catalog(Catalog.id[default=$contentCatalog]),CatalogVersion.version[default=Staged])[default=$contentCatalog:Staged]

#submissionContainer. Create new page template and page
INSERT_UPDATE PageTemplate;$contentCV[unique=true];uid[unique=true];frontendTemplateName;restrictedPageTypes(code);active[default=true]
;;SubmissionContainerPageTemplate;addon:/bazaarvoice/pages/bvcontainer/bvSubmissionContainer;ContentPage
INSERT_UPDATE ContentPage;$contentCV[unique=true];uid[unique=true];name;masterTemplate(uid,$contentCV);defaultPage;approvalStatus(code)[default='approved'];homepage[default='false']
;;submissionContainer;Submission Container;SubmissionContainerPageTemplate;true;

#Since we created new page template, we need to attach slots to it according to hybris requirenments. It does not matter which slot you will create for this page
INSERT_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='SubmissionContainerPageTemplate'];contentSlot(uid,$contentCV)[unique=true];allowOverwrite
;;SiteLogo;SiteLogo;;SiteLogoSlot;true
```

Submission container page should be accessible by the following url:
[YourBaseSitePath]/bv/submissioncontainer?bvaction=rr_submit_review&bvproductid=product1

ProductPageController.java

Put `BazaarvoiceDefaultService as class variable as shown in the snippet below`
And modify **`productDetail()`** `method in` **ProductPageController.java**

```
/*BAZAARVOICE_MODIFICATION_START*/
@Resource(name = "bazaarvoiceDefaultService")
private BazaarvoiceDefaultService bazaarvoiceDefaultService;
/*BAZAARVOICE_MODIFICATION_END*/


@RequestMapping(value = PRODUCT_CODE_PATH_VARIABLE_PATTERN, method =
RequestMethod.GET)
public String productDetail(@PathVariable("productCode") final String
encodedProductCode, final Model model,
     final HttpServletRequest request, final HttpServletResponse response)
     throws CMSItemNotFoundException, UnsupportedEncodingException
{
  final String productCode = decodeWithScheme(encodedProductCode, UTF_8);
  final List<ProductOption> extraOptions =
Arrays.asList(ProductOption.VARIANT_MATRIX_BASE,
ProductOption.VARIANT_MATRIX_URL,
        ProductOption.VARIANT_MATRIX_MEDIA);

  final ProductData productData =
productFacade.getProductForCodeAndOptions(productCode, extraOptions);

  final String redirection = checkRequestUrl(request, response,
productDataUrlResolver.resolve(productData));
  if (StringUtils.isNotEmpty(redirection))
  {
     return redirection;
  }

  updatePageTitle(productCode, model);


  populateProductDetailForDisplay(productCode, model, request, extraOptions);

  model.addAttribute(new ReviewForm());
  model.addAttribute("pageType", PageType.PRODUCT.name());
  model.addAttribute("futureStockEnabled",
Boolean.valueOf(Config.getBoolean(FUTURE_STOCK_ENABLED, false)));

  /*BAZAARVOICE_MODIFICATION_START*/
     String storefrontContextRoot =
getSiteConfigService().getProperty("storefrontContextRoot");
     String siteFullUrl = getSiteConfigService().getProperty("website." +
getCmsSiteService().getCurrentSite().getUid() + ".https");
     String websiteUrl = siteFullUrl.replace(storefrontContextRoot, "");
     model.addAttribute("websiteUrl", websiteUrl);

     final ProductModel productModel =
productService.getProductForCode(productCode);
```

```
     final List<ProductOption> options = new
ArrayList<>(Arrays.asList(ProductOption.VARIANT_FIRST_VARIANT,
ProductOption.BASIC,
             ProductOption.URL, ProductOption.PRICE, ProductOption.SUMMARY,
ProductOption.DESCRIPTION, ProductOption.GALLERY,
             ProductOption.CATEGORIES, ProductOption.REVIEW,
ProductOption.PROMOTIONS, ProductOption.CLASSIFICATION,
             ProductOption.VARIANT_FULL, ProductOption.STOCK,
ProductOption.VOLUME_PRICES, ProductOption.PRICE_RANGE,
             ProductOption.DELIVERY_MODE_AVAILABILITY,
ProductOption.VARIANT_MATRIX_MEDIA, ProductOption.VARIANT_MATRIX_BASE,
             ProductOption.VARIANT_MATRIX_URL));

     Set<ProductData> productDataForBazaarvoice = new HashSet<>();
     final ProductData currentProductData =
productFacade.getProductForCodeAndOptions(productData.getCode(), options);

/*gather base product and it`s variants to the set*/
     if (!(productModel instanceof VariantProductModel)) {
         productDataForBazaarvoice.add(currentProductData);
         if (currentProductData.getVariantOptions() != null) {
             for (VariantOptionData variantOptionData :
currentProductData.getVariantOptions()) {
                 final ProductData variantProductData =
productFacade.getProductForCodeAndOptions(variantOptionData.getCode(), options);
                 productDataForBazaarvoice.add(variantProductData);
             }
         }
     } else {
         /*gather variants from base product to the set*/
         if (currentProductData.getBaseOptions() != null) {
             for (BaseOptionData baseOptionData :
currentProductData.getBaseOptions()) {
                 if (baseOptionData.getOptions() != null) {
                     for (VariantOptionData variantOptionData :
baseOptionData.getOptions()) {
                         final ProductData variantProductData =
productFacade.getProductForCodeAndOptions(variantOptionData.getCode(), options);
                         productDataForBazaarvoice.add(variantProductData);
                     }
                 }
             }
         }
     }
     /*gather all required product data for bazaarvoice DCC functionality*/

bazaarvoiceDefaultService.populateModelWithProductInfo(productDataForBazaarvoice,
model);
     /*BAZAARVOICE_MODIFICATION_END*/


  final String metaKeywords =
```

```
MetaSanitizerUtil.sanitizeKeywords(productData.getKeywords());
  final String metaDescription =
MetaSanitizerUtil.sanitizeDescription(productData.getDescription());
  setUpMetaData(model, metaKeywords, metaDescription);
  return getViewForPage(model);
}
```

Perform “and clean all” and restart hybris. Storefront modifications complete!

Schema structure:

Schema structure can be included on product details page as shown in bold here: https://knowledge.bazaarvoice.com/wp-content/conversations/en_US/Display/display_integration.html#full-code-example

```
<div itemscope itemtype="https://schema.org/Product">
     <span itemprop="name">ProductName</span>
     <div data-bv-show="rating_summary" data-bv-productId="ProductID123”></div>
     <div data-bv-show="reviews" data-bv-productId="productID123"></div>
     <div data-bv-show="questions" data-bv-productId="productID123"></div>

</div>(Closing the div)
```

Appendix A: Explanation of Configuration Settings

Backoffice Control Panel Values

The following table describes the Backoffice control panel settings required by the Bazaarvoice for Hybris extension.

Backoffice Configuration Setting	Where to locate?	Example Value	Comment
Code	Determined by client	BV_config_electronics	Only used as unique Id for the BV configuration object w/in hybris; not used by Bazaarvoice
Client Name	Contact your Bazaarvoice representative	MyCompanyName	This value is case sensitive.
SFTP user name	Contact your Bazaarvoice representative	mycompanyname	Typically the lowercase transform of the Client Name value (i.e. if Client Name is “Acme”, the FTP username is “acme”)
SFTP password	Contact your Bazaarvoice representative	mTy@378FmH99	
SFTP Server	Standard BV SFTP server host names	Staging: sftp-stg.bazaarvoice.com Production: sftp.bazaarvoice.com	Contact Bazaarvoice representative to determine if non-standard FTP server is used for your deployment (e.g. “C7” cluster)
Deployment Zone Name	Bazaarvoice Configuration Hub (see steps below)	Main Site	Default value for all BV clients is “Main Site”. This should **not** be a Deployment Zone ID such as “main_site”.
Cloud SEO Key	Bazaarvoice Configuration Hub (see steps below)	mycompanyname-8ff9cdaafe1738bf4d77d49e09e080da	

Bazaarvoice Environment	Determined by client	Staging/Production	Changing this value does *not* change the FTP server from Staging to Production
R&R doShowContent() Javascript	Determined by client	$("#prod_tabs").tabs ({selected:2});	
Q&A doShowContent() Javascript	Determined by client	$("#prod_tabs").tabs ({selected:3});	
Put Product Variants into BV Families	Determined by client's implementation	True/False	Defines BV Product Families in product feed. Should set this value when Hybris product variants are used in client catalog. See standard BV product feed documentation for details.
Enable dynamic catalog collection on pdp		True	Enable/disable dynamic catalog collection on pdp.

Clarification on "**Put Product Variants into BV Families"** checkbox

1) This checkbox does not enable variants, variant products are enabled by default, in other words they included into product feed by default

2) If this checkbox is set to TRUE value, the following info will be included into Product variant information in the feed:

```
<Attributes>
    <Attribute id="BV_FE_FAMILY">
        <Value>1978440_md</Value>   //<------ base product`s code of variant
    </Attribute>
    <Attribute id="BV_FE_EXPAND">
        <Value>BV_FE_FAMILY:1978440_md</Value>   //<------ base product`s code of variant
    </Attribute>
</Attributes>
```

3) Checkbox just include base product`s code into variant in the way displayed above.

Local.Properties Configuration Settings

The following table describes local.properties settings required by the Bazaarvoice for Hybris extension.

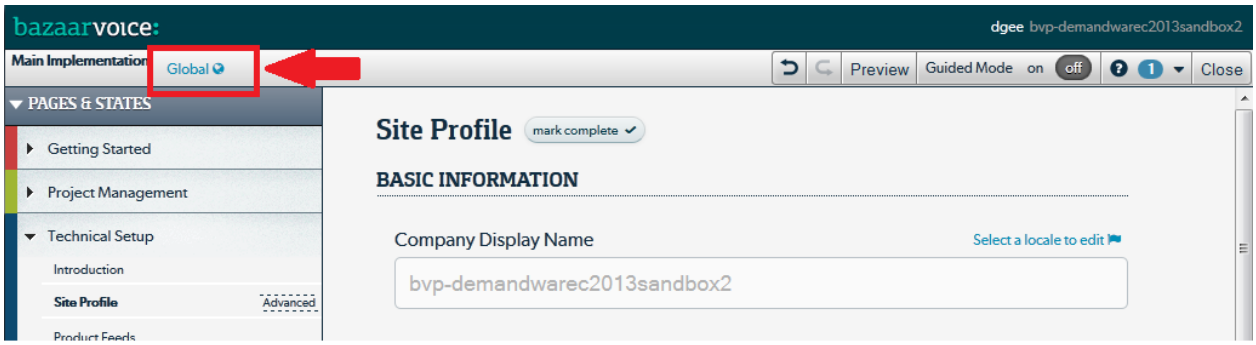
Local.Properties Configuration Setting	Example Value	Comment
ftpServerDirectory=/import-inbox	/import-inbox	This value should always be "/import-inbox" unless instructed otherwise by Bazaarvoice.
xmlBaseFileName=productfeed	productfeed	This value is concatenated with the current date to construct the product feed filename.
localXmlOutputDirectory=/xmlFolder	/xmlFolder	Need write permission to this directory; product feed is dropped to project directory after it has been created, but before it is FTP'd to BV
localXmlArchiveDirectory=/archiveFolder	/archiveFolder	Need write permission to this directory; product feed is moved to project directory after it has been FTP'd to BV
specificXmlOutputDirectory=/path/to/dir/xmlFolder	/home/username/xmlFolder	Add this property if you need to save product feed to some specific directory before it is FTP'd to BV
specificXmlArchiveDirectory=/path/to/dir/archiveFolder	/home/username/archiveFolder	Add this property if you need to save product feed to some specific directory after it has been FTP'd to BV

Locating your Locales Enabled within Your Bazaarvoice Instance

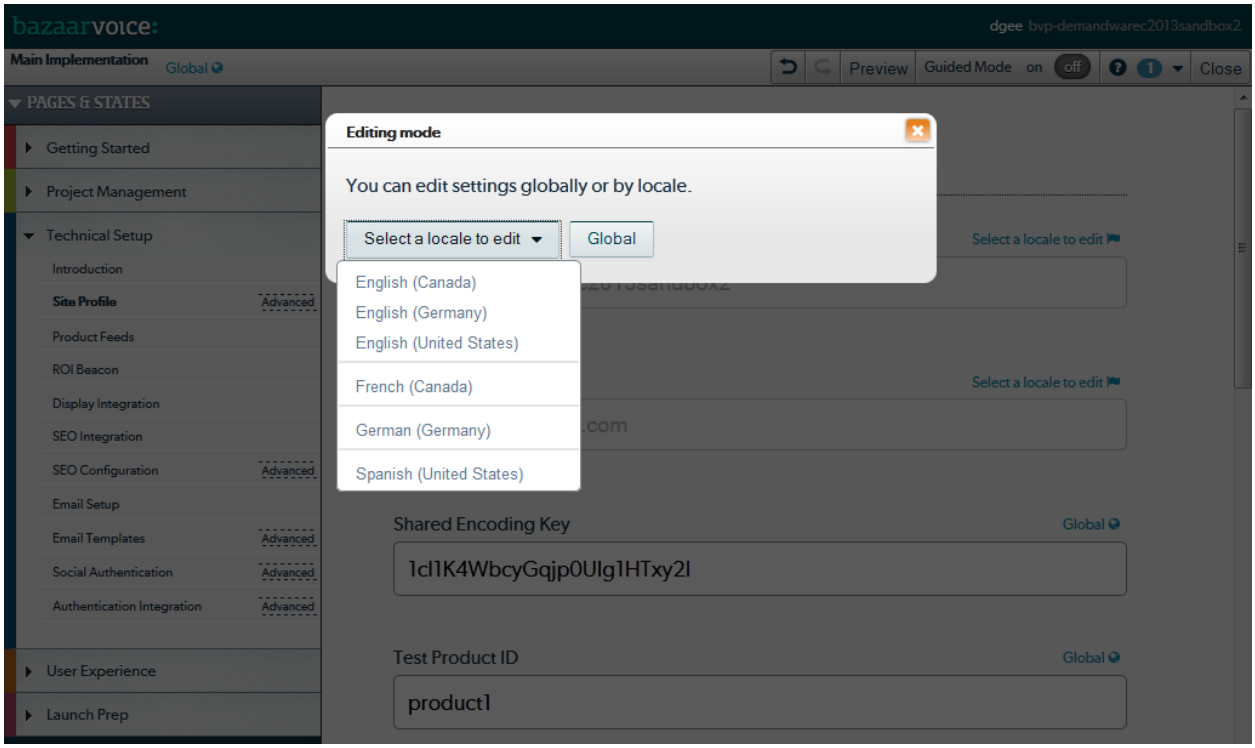
Complete the following steps to locate your Locales enabled within your Bazaarvoice instance.

1. Login to http://login.bazaarvoice.com.
2. Click **Settings** > **Manage Applications**.
3. You can have more than one **Deployment Zone**.
4. Select your desired **Implementation** object.

5. Select **Global** in the upper menu bar.



6. Click Select a locale to edit. The pull down list will display the enabled locales.

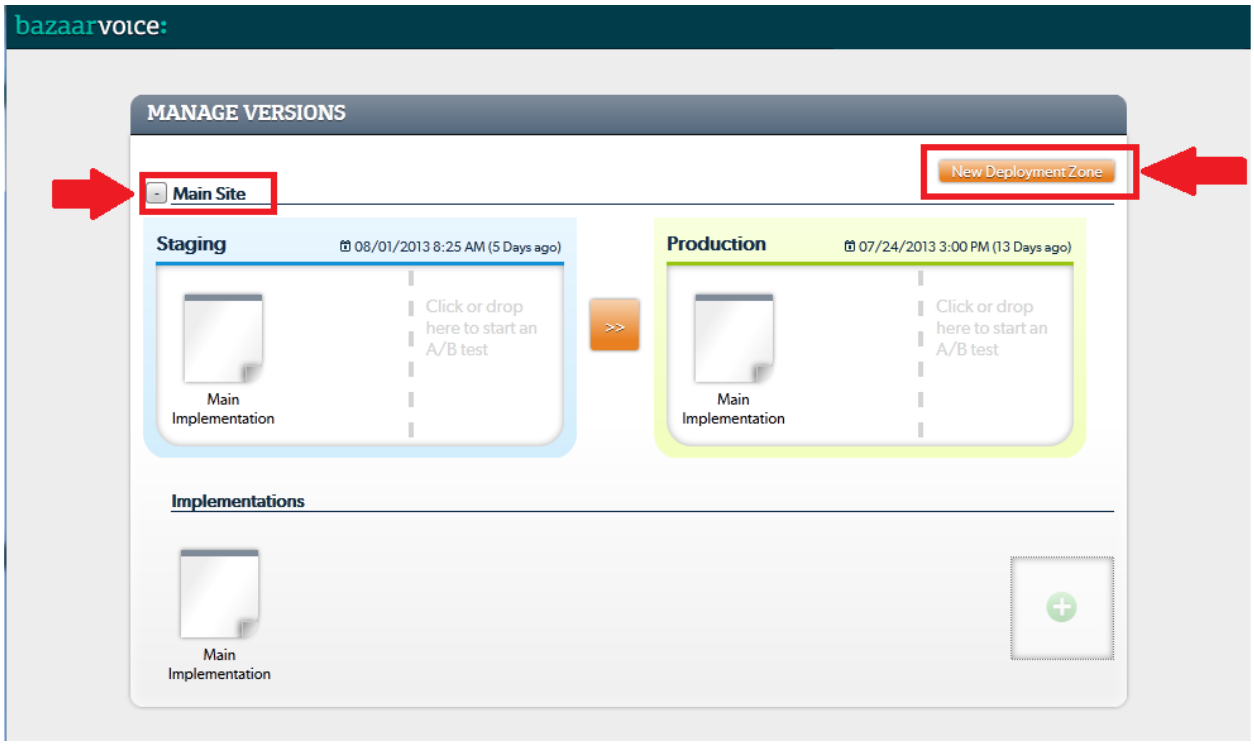


Locating your Deployment Zone Name

Complete the following steps to locate your Deployment Zone Name value.

1. Login to http://login.bazaarvoice.com.
2. Click Settings > Manage Applications.
3. Note your Deployment Zone name. The default Bazaarvoice Deployment Zone is named “Main Site”. Alternatively, you can create a new Deployment Zone by clicking New Deployment Zone.

You can have more than one Deployment Zone.

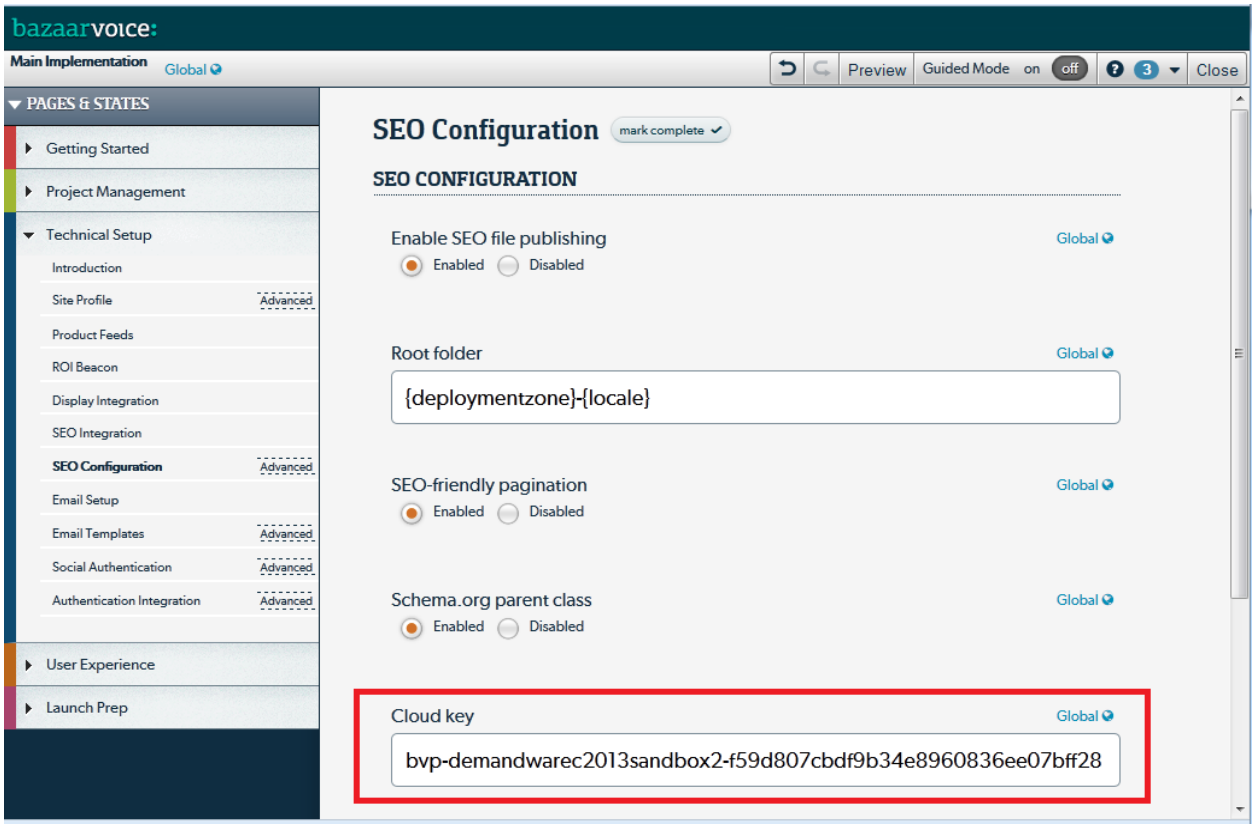


Locating your Cloud SEO Key

A Cloud SEO Key is required to implement the Bazaarvoice SEO solution. Complete the following to locate the key:

1. Login to http://login.bazaarvoice.com.
2. Click **Settings** > **Manage Applications**.
3. Select your desired **implementation**.
4. Select **Global** in the upper left menu bar.
5. Click **SEO Configuration** in the left navigation panel.
6. Ensure that **Root Folder** is set to "**{deploymentzon}-{locale}**".

7. Note the **Cloud Key** value.



Appendix B: How to Update products in bulk

The following impex script shows how you can update some field for some amount of products in bulk for STAGED catalog version:

```
$productCatalog=yourProductCatalogName
$catalogVersion=catalogVersion(catalog(id[default=$productCatalog]),version[default='Staged'])[unique=true,default=$productCatalog:Staged]

UPDATE Product;$catalogVersion;code[unique=true];fieldName
;;productCode1;"value1"
;;productCode2;"value2"
```

Parts which are in bold should be replaced by your actual values. This impex can be executed via Hybris HAC. Then the same should be executed for ONLINE catalog version as well.

More information you can find in official hybris documentation:
https://help.hybris.com/1811/hcd/8bdf595b86691014b902e0974f71491a.html

Version History

Version	Release Date	Changes
2.8.1	10 MAR 2021	• Fixed an issue in product feed when product ean field is empty, an issue was appeared in plugin 2.8.0 version. Jira ticket - BAZ-83
2.8.0	14 AUG 2020	• Added the following features: Enable Product Prefix, Enable Store Level Categories, Aggregate Attributes for Syndication
2.7.10	21 APR 2020	• Add hided html tags with plugin version to all bazaarvoice functionality
2.7.9	06 FEB 2020	• Add ability to send transactions to bazaarvoice server

2.7.8	28 NOV 2019	● Add ability to export yesterday's transactions ● Add ability to export transactions by shipping date
2.7.7	30 OCT 2019	● Add ability to enable/disable dcc via backoffice, ● Minor dcc fixes
2.7.6	18 OCT 2019	● Fix product feed ● Update jsch library with latest version 0.1.55 ○ This updates cipher used in file transfers
2.7.5	11 OCT 2019	● Implement Bazaarvoice DCC functionality
2.7.4	27 SEP 2019	● Make plugin compatible with hybris 1905
2.7.3	22 SEP 2019	● Update bv pixel fields
2.7.2	04 JUN 2019	● Implement export transactions cronjob
2.7.1	15 MAY 2019	● Allow special utf-8 symbols in product feed
2.7.0	11 MAR 2019	● In the product feed remove ParentExternalId from Category tag if category does not have supercategories
2.6.0	22 FEB 2019	● Remove close relation to native hybris accelerator ● Since that version plugin can be used with all hybris versions since 6.6
2.5.0	07 FEB 2019	● Fix compatibility with hybris v1811
2.3.0	01 OCT 2018	● Removed redundant file
2.2.0	15 MAY 2018	● Minor bug fixes
2.1.0	20 APR 2018	● Support Backoffice ● Support for Cloud Hosted Hybris ● Updates to container page canonical url ● Updates to product feed (product family update) ● Updates to documentation (Schema.org/backoffice)
2.0.0	15 FEB 2018	● Support Hybris 6.5 ● Support BV.JS on Category, PDP, and Checkout pages ● Updated SEO SDK version (v4) ● Update product feed schema version to 14.7 ● Added support for submission container page
1.1.1	10 APR 2015	● Added support for removing non-UTF8 characters from BV product feed export

		● Added support for SFTP ● Fix issue non UTF-8 characters ● Fix issue for sftp port numbers always using port 21 ● Stripped out non-ascii characters from product feed export ● Remove tags that do not contain any data
1.1.0	16 DEC 2014	● Updated SEO SDK version (v2.1.0.1) ● Added support for UPC, EAN, and Manufacturer part numbers in the product feed export ● Updated documentation steps to address SEO instruction gap ● Fixed issue with hmc setting to accept license
1.0	01 MAY 2014	Initial release of extension

Bazaarvoice Inc.
10901 Stonelake Blvd.
Austin, TX 78759
Tel (toll-free): (866) 522-9227 | Tel (toll): (512) 551-6000
www.bazaarvoice.com | info@bazaarvoice.com