

# JavaScript event callback guide



## Disclaimer

Copyright © 2012 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:

3900 North Capital of Texas Highway

Suite 300, Austin, Texas 78746

Toll-Free: (866) 522-9227 | Phone: (512) 551-6000 | Fax: (512) 551-6001

[www.bazaarvoice.com](http://www.bazaarvoice.com)

# Contents

<b>JavaScript event callback.....</b>	<b>5</b>
An overview of how the event callback works.....	5
Using the event callback.....	6
<b>Reporting dimensions and values.....</b>	<b>8</b>
Parameters that are always passed.....	8
group.....	9
bvProduct.....	9
eventSource.....	9
eType.....	10
eventTarget.....	10
Parameters that are sometimes passed.....	11
subjectType.....	12
eName.....	12
socialMedium.....	15
pageType.....	15
formAction.....	16
fieldGroup.....	16
fieldName.....	17
submissionBodyLengthSummary.....	17
submissionBodyLength.....	17
pageStatus.....	18
initialProductDisplay.....	18
bvExtension.BrandVoice.....	19
Parameters that are associated with product attributes.....	20
attributes.....	20
urlPageName.....	22
productId.....	22
leafCategoryId.....	22
rootCategoryId.....	23
linkDestinationLocation.....	23
<b>JavaScript event callback examples.....</b>	<b>24</b>
Product page displayed with ratings.....	24
Click to read all reviews.....	25
Successful submission of a review.....	26
Click to share a review on Facebook using social network buttons.....	27
Click to ask a question on a product or category page.....	28
Successful submission of an answer for a specific product.....	28
Campaigns event: Selecting text in content.....	29
<b>Sending data to your web analytics account.....</b>	<b>31</b>
Examples of tracking user interaction on product pages.....	32
Least granular: Track if a visitor interacted with Bazaarvoice.....	32
More granular: Track the Bazaarvoice product and type of interaction that occurred.....	33
Most granular: Track detailed information about visitor interactions.....	33
Bazaarvoice recommendation.....	34
Examples of tracking the steps of the content-submission process.....	35
Track granular submission analytics.....	36

Submission elements.....	36
Recommendations.....	37
Review length.....	37
Examples of tracking the Bazaarvoice attributes of a product.....	37
Ratings and reviews.....	39
Question and answer.....	39
Campaigns.....	40
<b>Reports.....</b>	<b>41</b>
Visitor engagement.....	41
Submission pages.....	42

# JavaScript event callback

The phrase *web analytics* describes the principal mechanism by which you can measure the online return on investment (ROI) that Bazaarvoice provides. Not only is this mechanism critically important to many businesses, but it often requires a complex technical implementation. Consequently, Bazaarvoice has designed a JavaScript-based solution that reduces the amount of technical effort you must expend through your analytics platform to measure visitor interaction with Bazaarvoice user-generated content (UGC) on your website. This solution is known as the Bazaarvoice *JavaScript event callback*, and it helps measure the impact of user interaction with UGC on the key conversion metrics that you are already measuring in your web analytics platform.

The JavaScript event callback sends detailed information about visitor interactions with Bazaarvoice UGC to a single JavaScript function that you implement on your site. As a result, you can relay granular information about visitor interaction with UGC to your web analytics provider simply by choosing the information that you want to capture from the event callback. You do not need to tag individual Bazaarvoice links manually to capture this information.

After you implement the event callback, it works with your web analytics platform to measure the impact of visitor engagement with Bazaarvoice UGC against your key performance metrics. The tracking of visitor engagement is the most basic, yet most important way to measure the impact of UGC on your site. Through visitor tracking, you can generate reports that help you make assertions like *Visitors who engage with Bazaarvoice show an X% lift in conversion over visitors who do not*.

Other common uses for the event callback include the following examples:

- Recording visitor interaction with specific pieces of UGC
- Determining the average rating and review count of a product while a visitor views a page
- Tracking the progress and rates of abandonment of a UGC-submission form

This document describes the steps that you must perform to use the JavaScript event callback to track visitor interactions with Bazaarvoice UGC on your site.

## An overview of how the event callback works

The event callback passes data about how UGC is consumed on your site to a JavaScript function when the following occurs:

- Bazaarvoice content appears on a page
- A visitor performs an action on Bazaarvoice content

When either event takes place, Bazaarvoice calls a JavaScript function that is implemented on your site and passes a single JavaScript Object Notation (JSON) object to it. Information about visitor interactions is passed to this object, which provides details about the event that occurred, including the following examples:

- The Bazaarvoice feature that was used
- Whether a page with Bazaarvoice content loaded or a visitor interacted with Bazaarvoice content
- The specific type of visitor interaction that occurred
- Product information, including the product ID, average rating, and number of reviews

This information can be relayed to your analytics system, typically through a JavaScript call to your analytics provider, while Bazaarvoice passes it to the event callback. Further, you can customize the information that you record with your web analytics provider as follows:

- Identify the information that you want to pass from the event callback
- Specify when you want to pass that information

## Using the event callback

The JavaScript event callback is not enabled by default. Use [Spark](#) to have the Bazaarvoice Support team enable the event callback before starting any development work.

Depending on your particular Bazaarvoice integration, the event callback refers to one of the following methods:

- If your Bazaarvoice integration uses `<iframe>` elements, define a new function called `BVAnalytics` on the appropriate pages. When implemented on a page, the `BVAnalytics` method uses the following format.

```
<script type="text/javascript">
function BVAnalytics(json) {
  if (json.eventSource == "Action") {
    alert("Bazaarvoice interaction happened");
  }
};
</script>
```

- If your integration uses the JavaScript integration method, your pages already feature a `$BV.ui` function that contains Bazaarvoice content. Consequently, you must specify an `onEvent` option when making `$BV` API calls. The following example shows the appropriate format for a ratings and reviews (R&R) implementation.

```
<script type="text/javascript">
$BV.ui("rr", "show_reviews", {
  onEvent: function(json) {
    if ( json.eventSource == "Action" ) {
      alert("Bazaarvoice interaction happened");
    }
  }
});
</script>
```

The same information is made available within the event callback regardless of the method that you use.

---

### Note

The example code in this document pertains to the `$BV.ui` method. However, you can achieve the appropriate results for the `BVAnalytics` method by substituting `BVAnalytics` for `$BV.ui` where necessary.

---

When you implement these examples on a page with Bazaarvoice content, each example returns an alert that indicates a visitor clicked a link or performed another action involving Bazaarvoice content. Both methods determine that an interaction occurred by examining the value that Bazaarvoice passed to the `eventSource` parameter of the JSON object when a visitor clicked a Bazaarvoice link. Because the `eventSource` parameter in these examples was passed a value of `Action`, we know that an interaction with Bazaarvoice occurred.

---

In a similar fashion, the event callback passes data to a variety of different parameters to describe the following information:

- Types of interactions that visitors take with Bazaarvoice content
- Attributes of the products with which visitors interact

Values that are passed to these parameters can be relayed to your web analytics system to record the type of interactions that take place on your site. The remaining sections in this document describe the parameters to which Bazaarvoice passes data and provide guidance for sending this data to your analytics provider.

# Reporting dimensions and values

The values that are passed into the parameters of the JavaScript event callback can also be relayed to your web analytics system. The granularity of the tracking in your web analytics provider depends on the following conditions:

- The parameters that you choose to relay to your web analytics provider
- The circumstances under which you choose to pass the values

The combination of values that are passed into the different parameters specifies the type of interaction that occurred.

The level of granularity that you can capture is highly flexible. For example, you can pass a value to your web analytics provider to indicate only that a visitor has read a review. Additionally, you have the option of passing more granular information for the same interaction, including the product ID, average rating, number of reviews, and interaction that triggered the read action.

The following types of parameters are associated with the event callback:

- Parameters that are always passed
- Parameters that are sometimes passed, depending on the event that occurred
- Parameters that are associated with product attributes

The following sections identify the parameters that belong to these types.

[Parameters that are always passed](#) on page 8

These parameters are always passed to the JSON object of the JavaScript event callback.

[Parameters that are sometimes passed](#) on page 11

These parameters that are passed to the JSON object of the JavaScript event callback only when specific types of events occur with Bazaarvoice.

[Parameters that are associated with product attributes](#) on page 20

These parameters describe the product- and page-related attributes that are captured when an event with Bazaarvoice occurs.

## Related Links

[JavaScript event callback examples](#) on page 24

[Examples of tracking user interaction on product pages](#) on page 32

## Parameters that are always passed

These parameters are always passed to the JSON object of the JavaScript event callback.

[group](#) on page 9

The `group` parameter answers the question *“What is the grouping for this event?”* and always contains the value Bazaarvoice.



[bvProduct](#) on page 9

The `bvProduct` parameter answers the question “Where is the user?” or “Which Bazaarvoice product is being used?”

[eventSource](#) on page 9

The `eventSource` parameter answers the question “Was the Bazaarvoice event triggered by a page loading or a user interaction?”

[eType](#) on page 10

The `eType` parameter answers the question “What general type of interaction occurred?”

[eventTarget](#) on page 10

The `eventTarget` parameter answers the question “With what type of Bazaarvoice content is the visitor interacting?”

## group

The `group` parameter answers the question “What is the grouping for this event?” and always contains the value `Bazaarvoice`.

Specifically, it indicates that one of the following event types has occurred:

- A page with Bazaarvoice UGC was loaded
- A visitor interacted with UGC

Use `group` to determine that the visitor viewed or interacted with Bazaarvoice UGC.

## bvProduct

The `bvProduct` parameter answers the question “Where is the user?” or “Which Bazaarvoice product is being used?”

It contains one of the following values:

- `RatingsAndReviews` – Indicates that the visitor viewed or interacted with the R&R product
- `AskAndAnswer` – Indicates that the visitor viewed or interacted with the Question and Answer (Q&A) product
- `Stories` – Indicates that the visitor viewed or interacted with the Campaigns product
- `Profiles` – Indicates that the visitor viewed or interacted with the Profiles product

Use `bvProduct` if you have implemented multiple Bazaarvoice products on your site and need to identify the one with which site visitors are interacting. This value provides clarification when parameters return similar values for visitors who read or write content that is associated with different Bazaarvoice products.

If you are interested in capturing the total number of visitor interactions with a specific Bazaarvoice product, pass this parameter in conjunction with `eventSource=Action`.

## eventSource

The `eventSource` parameter answers the question “Was the Bazaarvoice event triggered by a page loading or a user interaction?”

It contains one of the following values:

- `Display` – Indicates an initial page load due to either the initial display of Bazaarvoice content or a dynamic page update by AJAX
- `Action` – Indicates that a visitor interacted explicitly with Bazaarvoice content by clicking the content or by hovering the mouse pointer over certain pieces of it

Use `eventSource` to determine whether a visitor interacted with Bazaarvoice or a page that loaded Bazaarvoice content. To capture information about the average rating of a product when its page loads, for example, you need to capture it when `eventSource=Display` because that value indicates that a page with Bazaarvoice content was loaded. Use `eventSource=Action` only to capture interactions that visitors take with Bazaarvoice content.

## eType

The `eType` parameter answers the question “*What general type of interaction occurred?*”

It contains one of the following values:

- `Read` – Indicates the consumption of data without the contribution of UGC, such as reading, paginating, sorting, tag filtering, or expanding
- `Write` – Indicates engagement with any step of the content-submission process, such as the contribution of a review, question, answer, campaign item, profile, comment, or recommended product
- `Support` – Indicates the contribution of supporting data, such as the marking of UGC as helpful or inappropriate, the sharing of content by way of the social network buttons or the use of Facebook Like

Use `eType` to determine the overall type of event that occurred with Bazaarvoice content. This parameter groups specific visitor interactions to provide an overview of visitors who are reading, contributing, or supporting content. Use this parameter in conjunction with `eName` to provide a more granular view of the specific `read`, `write`, or `support` action that occurred.

## eventTarget

The `eventTarget` parameter answers the question “*With what type of Bazaarvoice content is the visitor interacting?*”

It contains one of the values from the following table.

This value...	Indicates that a visitor...	Under these conditions...
Review	Wrote, commented on, or provided inappropriate feedback or a helpfulness vote for a review	<code>bvProduct=RatingsAndReviews AND (eType=Write OR eType=Support)</code>
Question	Asked, commented on, or provided inappropriate feedback or a helpfulness vote for a question	<code>bvProduct=AskAndAnswer AND (eType=Write OR eType=Support)</code>

This value...	Indicates that a visitor...	Under these conditions...
Answer	Wrote, commented on, or provided inappropriate feedback or a helpfulness vote for an answer	bvProduct=AskAndAnswer AND (eType=Write OR eType=Support)
Comment	Contributed a comment or provided feedback on a comment	All conditions
Campaign	Submitted, rated, commented on, or provided feedback for campaign content	bvProduct=Stories AND (eType=Write OR eType=Support)
Profile	Edited, commented on, or provided feedback for a user nickname	All conditions
Product	Clicked Facebook Like functionality at the product level	All conditions
Gallery	Clicked Bazaarvoice-hosted media located in the Media Gallery	All conditions

Use `eventTarget` to identify the type of Bazaarvoice content with which visitors interact. For example, if `eType` possesses a value of `Write` with question and answer (Q&A), `eventTarget` identifies whether the visitor was in the process of writing a question or an answer. Similarly, if `eType` has a value of `Write` with R&R, `eventTarget` identifies whether the visitor was in the process of writing a review or a comment.

## Parameters that are sometimes passed

These parameters that are passed to the JSON object of the JavaScript event callback only when specific types of events occur with Bazaarvoice.

[subjectType](#) on page 12

The `subjectType` parameter answers the question *“Is the customer on a product- or a category-specific page?”*

[eName](#) on page 12

The `eName` parameter answers the question *“What specific type of event occurred?”*

[socialMedium](#) on page 15

The `socialMedium` parameter answers the question *“With which social network did the customer choose to share content?”* and is required on visitor interactions with the social network buttons.

[pageType](#) on page 15

The `pageType` parameter is always passed on submission pages and answers the question *“On which page of the submission process is the visitor?”*

[formAction](#) on page 16

The `formAction` parameter answers the question *“What button on a submission or feedback form was clicked?”* and is required for buttons on submission forms.

[fieldGroup](#) on page 16

The `fieldGroup` parameter answers the question *"Which group of fields on the review submission form are being interacted with?"* and is optional for fields on the submission form.

[fieldName](#) on page 17

The `fieldName` parameter answers the question *"Which group of fields on the review submission form are being interacted with?"* and is optional for fields on the submission form.

[submissionBodyLengthSummary](#) on page 17

The `submissionBodyLengthSummary` parameter answers the question *"Was the review that a user previewed a short, medium or long review?"* and is a required parameter when the review preview page is displayed.

[submissionBodyLength](#) on page 17

The `submissionBodyLength` parameter answers the question *"How many characters long is the review that's being previewed?"* and is a required parameter when the review preview page is displayed.

[pageStatus](#) on page 18

The `pageStatus` parameter describes error messages that are displayed to visitors and answers the question *"What type of error message did the visitor receive when attempting to submit content or feedback?"*

[initialProductDisplay](#) on page 18

The `initialProductDisplay` parameter answers the question *"Was this time the first time that Bazaarvoice content was displayed for the current page?"*

[bvExtension.BrandVoice](#) on page 19

The `bvExtension.BrandVoice` parameter answers the question *"Does the product being viewed feature syndicated content?"* and contains a value only when a product is viewed in a brand that features at least one active syndication match.

## subjectType

The `subjectType` parameter answers the question *"Is the customer on a product- or a category-specific page?"*

It contains one of the following values when `eventSource=Display`:

- `Product` – Identifies the page as product-specific
- `Category` – Identifies the page as category-specific

Use `subjectType` to limit interactions with Bazaarvoice to product pages or category pages.

## eName

The `eName` parameter answers the question *"What specific type of event occurred?"*

It contains one of the values that the following table identifies.

This value...	Indicates that a visitor clicked...	Under these conditions...
Paginate	To another page of Bazaarvoice content for a particular content type; loaded content results from paginating.	<code>eType=Read</code>

This value...	Indicates that a visitor clicked...	Under these conditions...
Sort	To sort Bazaarvoice content for a particular content type; loaded content results from sorting.	eType=Read
AttributeFilter	To use attribute filtering or Pros and Cons tags; loaded content results from attribute filtering.	eType=Read
WhiteSpace	Something other than a link – such as whitespace, text, or an image – in a Bazaarvoice-hosted container.	eType=Read OR eType=Write
ProductLink	A product-recommendation link provided by another customer to view the product.	eType=Read
	To add a product recommendation to a piece of content.	eType=Write
Write	A Write a(n) X link to start the submission process.	eType=Write
SocialConnect	To share content during the submission process associated with Facebook Connect or Twitter Connect; corresponds to a click on the submission form and not to any activity within resulting popup pages.	eType=Write
SocialAlerts	To receive a SocialAlerts transactional alerts email.	eType=Write
Associate	To submit a positive or negative helpfulness vote or the “Like (a product)” functionality of Bazaarvoice or Facebook.	eType=Support
ShoutIt	A link to share UGC with other visitors by way of a social networking site like Facebook or Twitter.	eType=Support
Inappropriate	To submit an inappropriate vote on a piece of UGC	eType=Support
Rate	To rate campaign content.	bvProduct=Stories AND eType=Support

This value...	Indicates that a visitor clicked...	Under these conditions...
ReadAll	To read all of the content related to the Read All X link.	eType=Read
ProductFollow	A Follow This Product or Follow New Reviews link.	eType=Read
Media	To view a hosted photo (attributes={mediaType=Photo}) or video (attributes={mediaType=Photo}) within the Bazaarvoice platform; the photo or video can reside within the Media Gallery (eventTarget=Gallery) or can appear inline, alongside Bazaarvoice content (eventTarget=Review Question Answer Story).	eType=Read
Search	To type a search query and click Search within a review, campaign item, or question.	eType=Read
ProfileLink	Clicked to view a user profile	eType=Read
ProductLink	Clicked on a user recommended product	eType=Read OR eType=Write
FacebookAuthLogin	Clicked the Facebook Login box.	eType= Write
FacebookAuthAllow	User confirmed Facebook credential authorization, allowing Bazaarvoice to programmatically access their profile data.	eType= Write
QuestionSubmit	Clicked the submit button for a question on the Social Homepage.	eType= Write
ResultsSelect	Clicked on a suggested question from the question submission form.	eType=Read
SubmitActivity	Clicked on an element within the submission form.	eType= Write

Use `eName` to identify the specific type of interaction that occurred with Bazaarvoice. This parameter is most helpful when used in conjunction with the `eType` parameter to determine what type of `read`, `write`, or `support` action took place. For example, if a visitor sorts reviews, this action is captured as a general `read` action with `eType`. However, the `eName` parameter indicates that the `read` action occurred as the result of the visitor sorting content.

Specific `eName` values are not currently associated with every interaction with Bazaarvoice.

## socialMedium

The `socialMedium` parameter answers the question “*With which social network did the customer choose to share content?*” and is required on visitor interactions with the social network buttons.

`socialMedium` must contain one of the following values when `eName=ShoutIt` OR `eName=SocialConnect`:

- `Facebook` – Indicates that a visitor clicked to share UGC with friends through Facebook
- `Twitter` – Indicates that a visitor clicked to share UGC with friends through Twitter
- `Digg` – Indicates that a visitor clicked to share UGC with friends through Digg
- `Delicious` – Indicates that a visitor clicked to share UGC with friends through Delicious
- `MySpace` – Indicates that a visitor clicked to share UGC with friends through MySpace
- `RenRen` – Indicates that a visitor clicked to share UGC with friends through RenRen
- `KaiXinWang` – Indicates that a visitor clicked to share UGC with friends through KaiXinWang
- `MSConnect` – Indicates that a visitor clicked to share UGC with friends through MSConnect

Use `socialMedium` to identify the social network through which a visitor chose to share UGC. The `socialMedium` parameter captures activity only on the initial click to share content. It does not track any subsequent activity on popup forms that visitors might need to complete to share UGC successfully.

## pageType

The `pageType` parameter is always passed on submission pages and answers the question “*On which page of the submission process is the visitor?*”

If present, `pageType` must contain one of the following values when `eType=Write` OR `eType=Support`:

- `Input` – Indicates that a visitor is on the input page of the submission process
- `Preview` – Indicates that a visitor is on the preview page of the submission process and has successfully entered the required criteria on the input page
- `Confirm` – Indicates that a visitor has submitted the form and advanced to the final page of the submission process, which is typically a **Thank You** page or other confirmation page

Use `pageType` to determine abandonment rates between steps of the submission process or to determine the frequency with which a visitor successfully submits content. When used in conjunction with `eventSource=Display`, the values of the `pageType` parameter can be interpreted as the loading of the various steps of the submission process. For example, you can determine that a visitor has reached the final step of a submission process and successfully submitted content or feedback when `pageType=Confirm` and `eventSource=Display` are both present.

Use the following criteria to measure the submissions of a certain type of content, such as reviews:

- `bvProduct=RatingsAndReviews`
- `eventSource=Display`
- `eventTarget=Reviews`

- `pageType=Confirm`

## formAction

The `formAction` parameter answers the question “*What button on a submission or feedback form was clicked?*” and is required for buttons on submission forms.

If present, `formAction` must contain one of the following values when `eventSource=Action` AND (`eType=Write` OR `eType=Support`):

- `Preview` – Indicates that a visitor clicked **Preview** on a submission form to preview a submission
- `Edit` – Indicates that a visitor has clicked **Edit** on a submission form to edit a submission
- `Submit` – Indicates that a visitor clicked **Submit** on a submission form to complete a submission
- `Cancel` – Indicates that a visitor clicked **Cancel** on a submission form to abandon a submission

Use `formAction` to measure button clicks on submission pages and to identify the point to which visitors progressed through the content submission process. For example, visitors who clicked **Preview** on a submission form but never proceeded to the preview page probably experienced an error or were missing information on the submission form.

## fieldGroup

The `fieldGroup` parameter answers the question “*Which group of fields on the review submission form are being interacted with?*” and is optional for fields on the submission form.

If present, `fieldGroup` must contain one of the following values:

- `Core` – Indicates that a visitor interacted with one of the following:
  - Clicked on a rating star
  - Typed more than two characters in the review title
  - Typed more than two characters in the review body
  - Selected Yes or No to the “Would you recommend this to a friend?” question
- `RatingDimensions` – Indicates that a visitor clicked on any one of the rating dimensions on the submission page
- `Tags` – Indicates that a visitor clicked on a tag checkbox or typed a character in the tag field on the submission page
- `AdditionalFields` – Indicates that a visitor clicked on a custom field on the submission page
- `ContextData` – Indicates that a visitor clicked on a context-data value such as the age, gender, or experience drop-down on the submission page

Use `fieldGroup` to measure clicks on submission pages with the groups of fields that users interact with. For example, capturing clicks on the `fieldGroup` `RatingDimensions` will capture all clicks on any custom rating dimensions being captured on the submission form.



## fieldName

The `fieldName` parameter answers the question *"Which group of fields on the review submission form are being interacted with?"* and is optional for fields on the submission form.

If present, `fieldGroup` must contain one of the following values:

- `Rating` – Indicates that a visitor clicked on one of the ratings stars on a submission page
- `Recommend` – Indicates that a visitor clicked on the Yes/No buttons to the "Would you recommend this to a friend?" question on a submission page
- `Title` – Indicates that a visitor typed more than two characters in the Review Title box on a submission page
- `Reviewtext` – Indicates that a visitor typed more than two characters in the Review Body box on a submission page
- `Pro` – Indicates that a visitor clicked on a pro checkbox or typed a character in the pro field on the submission page
- `Con` – Indicates that a visitor clicked on a con checkbox or typed a character in the con field on the submission page
- `EmailAddress` – Indicates that a visitor typed more than two characters in the email address field on the submission page
- `MoreThoughts` – Indicates that a visitor typed more than two characters in the More Thoughts field on the submission page
- `[Custom Value]` – Based on name of field on the submission form

Use `fieldName` to measure clicks on submission pages with the specific fields that users interact with. For example, you can use `fieldName` to capture clicks on a custom rating dimension such as value or quality.

## submissionBodyLengthSummary

The `submissionBodyLengthSummary` parameter answers the question *"Was the review that a user previewed a short, medium or long review?"* and is a required parameter when the review preview page is displayed.

The `submissionBodyLengthSummary` parameter must contain one of the following values:

- 0 to 200 characters --> `submissionBodyLengthSummary=PreviewShortReview`
- 201 to 400 characters --> `submissionBodyLengthSummary=PreviewMediumReview`
- More than 400 characters --> `submissionBodyLengthSummary=PreviewLongReview`

## submissionBodyLength

The `submissionBodyLength` parameter answers the question *"How many characters long is the review that's being previewed?"* and is a required parameter when the review preview page is displayed. The `submissionBodyLength` parameter will contain an integer with the number of characters present in the review text.

## pageStatus

The `pageStatus` parameter describes error messages that are displayed to visitors and answers the question “*What type of error message did the visitor receive when attempting to submit content or feedback?*”

If present, `pageStatus` must contain one of the values that the following table identifies.

This value...	Indicates this kind of message...	Under these conditions...
Authentication-Failure	Error message that appears when a visitor reaches a submission-container page without the appropriate encoding of the user authentication string (UAS)	eType=Write OR eType=Support
Cancelled	Confirmation message that appears after a visitor cancels the submission process	eType=Write OR eType=Support
Expired	Error message that appears when a visitor clicks Back after cancelling, effectively trying to resubmit the cancellation command for an submission that is already cancelled	eType=Write OR eType=Support
AlreadySubmitted	Error message that appears when a visitor attempts a duplicate submission soon after submitting the original content	eType=Write OR eType=Support
Own	Error message that appears when a visitor tries to submit helpfulness or inappropriate feedback for his or her own content	eType=Support
Duplicate	Error message that appears when a visitor tries to resubmit helpfulness or inappropriate feedback for content after the feedback has already been submitted	eType=Support
ValidationError	Message that functions as a catch-all for errors that are not otherwise identified	eType=Write OR eType=Support

Use `pageStatus` to measure the number of visitors who received an error when attempting to submit Bazaarvoice content. This metric helps to measure abandonment through the content-submission process by determining whether visitors encounter difficulties when writing content. Use this parameter in conjunction with `bvProduct` and `eventTarget` to identify the type of content that a visitor was attempting to submit when an error was received.

## initialProductDisplay

The `initialProductDisplay` parameter answers the question “*Was this time the first time that Bazaarvoice content was displayed for the current page?*”

It is required when the following conditions are satisfied:

- `eventSource=Display`

- The event represents the first instance of the JavaScript event callback function being called for the given URL of a page

Bazaarvoice display pages exist as entities that are separate from your pages. As a result, a Bazaarvoice page can load multiple times for a single loading of your page, such as when a visitor changes the sort order on a page or navigates to a second page of reviews. In such instances, the Bazaarvoice page loads, but your page does not reload.

The `initialProductDisplay` parameter fires as `True` when the Bazaarvoice display page and your page load at the same time. Both pages can load simultaneously during one or more of the following events:

- On product page load
- On submission container load
- On product page load after submission

The `initialProductDisplay` parameter is set separately for each `bvProduct` on a given page. For example, if R&R and Q&A appear on the same product page, the event callback is called twice. Further, the `initialProductDisplay` value for each JSON object – one for R&R and one for Q&A - is set to `True` when the page originally loads, identifying the loading as the first time that Bazaarvoice content was loaded for each Bazaarvoice product on the page.

## bvExtension.BrandVoice

The `bvExtension.BrandVoice` parameter answers the question “Does the product being viewed feature syndicated content?” and contains a value only when a product is viewed in a brand that features at least one active syndication match.

This parameter is populated with one of the following values only if you are a brand that has implemented content syndication:

- `Syndicating` – Identifies the product as one for which syndicated content is present
- `NotSyndicating` – Identifies the product as belonging to a brand that is syndicating content but for which no syndicated content is present

Use `bvExtension.BrandVoice` to determine whether a product includes syndicated content. For example, if a product that belongs to Brand X displays syndicated reviews from a manufacturer alongside reviews from a retailer, the `bvExtension.BrandVoice` parameter indicates whether that content appears on the product page.

Identify the products that feature active syndication when they are first displayed by capturing the value of `bvExtension.BrandVoice` in your web analytics system when `initialProductDisplay=True`. Use the data that you collect to compare the conversion rates of products with syndicated content to other products in the same brand that do not have syndicated content.

# Parameters that are associated with product attributes

These parameters describe the product- and page-related attributes that are captured when an event with Bazaarvoice occurs.

[attributes](#) on page 20

The `attributes` parameter answers the question *“What are the Bazaarvoice-related attributes of the product that the visitor is viewing?”*

[urlPageName](#) on page 22

The `urlPageName` parameter answers the question *“With which page did the event occur?”* and identifies the URL of the page that the visitor was viewing when the event with Bazaarvoice occurred.

[productId](#) on page 22

The `productId` parameter answers the question *“What is the product ID or SKU being acted upon?”* and specifies the product ID of the page if it resides within the product catalog.

[leafCategoryId](#) on page 22

The `leafCategoryId` parameter answers the question *“What is the category ID of the product being acted upon?”* and specifies the category ID of the page if it resides within the category hierarchy of your Bazaarvoice product feed.

[rootCategoryId](#) on page 23

The `rootCategoryId` parameter answers the question *“What is the top-level parent category being acted upon?”* and specifies the category ID of the top-level parent category of the page if it resides within the category hierarchy of your Bazaarvoice product feed.

[linkDestinationLocation](#) on page 23

The `linkDestinationLocation` parameter answers the question *“Where does this Bazaarvoice link point?”* and contains the value `External` if the Bazaarvoice link points to a domain that does not reside on your site.

## attributes

The `attributes` parameter answers the question *“What are the Bazaarvoice-related attributes of the product that the visitor is viewing?”*

It contains one of the following values.

Bazaarvoice feature	Variable	Format	Description
R&R	<code>numReviews</code>	Integer	Total number of approved reviews that are submitted for the product.

Bazaarvoice feature	Variable	Format	Description
	numRatingsOnlyReviews	Integer	Number of ratings-only reviews that are submitted for the product. By default, a ratings-only review contains fewer than 50 characters of review text and features no attached photos, videos, visitor-entered tags, or additional visitor-entered fields.
	percentRecommend	Integer (0-100)	Percentage of visitors who selected Yes as an answer to the question "Would you recommend this product to a friend?"
	avgRating	Float	Average rating for the product, expressed in a format that displays up to four decimal places.
Q&A	numQuestions	Integer	Total number of approved questions that are submitted about the subject.
	numAnswers	Integer	Total number of approved answers that are submitted about the subject.
Campaigns	numStories	Integer	Total number of approved content items that are submitted about the subject.
All	mediaType	String	Indicates that a visitor clicked hosted media within a piece of Bazaarvoice content. Possible values are <code>photo</code> and <code>video</code> .
	filterDimension	String	Describes the type of dimension by which a user filtered content. Examples values include: <ul style="list-style-type: none"> <li>• <code>star rating</code></li> <li>• <code>Age</code></li> <li>• <code>Gender</code></li> </ul>
	filterName	String	Describes the filter value by which a user filtered content. Example values include: <ul style="list-style-type: none"> <li>• <code>overallRating2Stars</code></li> <li>• <code>18to24</code></li> <li>• <code>Male</code></li> </ul>
	sortType	String	Indicates the type of sort functionality a visitor used to re-sort content. Example values include: <ul style="list-style-type: none"> <li>• <code>rating</code></li> <li>• <code>submissionTime</code></li> <li>• <code>helpfulness</code></li> </ul>

Bazaarvoice feature	Variable	Format	Description
	<code>sortDirection</code>	String	Indicates the direction of the sort functionality a visitor used to re-sort content. Possible values are <code>asc</code> and <code>desc</code> .

Use `attributes` to measure items like the following examples:

- Average rating of a product
- Number of reviews present when a product was viewed

When this information is passed into your web analytics system, you can report on the conversion rate of products with different ratings or varying numbers of reviews.

## urlPageName

The `urlPageName` parameter answers the question *“With which page did the event occur?”* and identifies the URL of the page that the visitor was viewing when the event with Bazaarvoice occurred.

Because `urlPageName` consists only of a URL path and file name, it is the equivalent of a product page URL without the protocol, hostname, and parameters.

The following URL provides an example product page URL:

```
http://www.client.com/category/product/sampleproduct.html
```

Given this URL, the following text represents the corresponding value for `urlPageName`:

```
category/product/sampleproduct.html
```

## productId

The `productId` parameter answers the question *“What is the product ID or SKU being acted upon?”* and specifies the product ID of the page if it resides within the product catalog.

The product ID that is passed to this parameter must match one of the product IDs that is included in your Bazaarvoice product feed.

`productId` is required when `eventSource=Display` but optional when `eventSource=Action`. Exclude this parameter if no or multiple values exist.

## leafCategoryId

The `leafCategoryId` parameter answers the question *“What is the category ID of the product being acted upon?”* and specifies the category ID of the page if it resides within the category hierarchy of your Bazaarvoice product feed.

---

`leafCategoryId` is required when `eventSource=Display` but optional when `eventSource=Action`. Exclude this parameter if no or multiple values exist.

## rootCategoryId

The `rootCategoryId` parameter answers the question “*What is the top-level parent category being acted upon?*” and specifies the category ID of the top-level parent category of the page if it resides within the category hierarchy of your Bazaarvoice product feed.

`rootCategoryId` is required when `eventSource=Display` but optional when `eventSource=Action`. Exclude this parameter if no or multiple values exist.

## linkDestinationLocation

The `linkDestinationLocation` parameter answers the question “*Where does this Bazaarvoice link point?*” and contains the value `External` if the Bazaarvoice link points to a domain that does not reside on your site.

`linkDestinationLocation` applies to social network links and other links that point to pages outside your domain. Such links must pass through the Bazaarvoice URL-shortening service.

The `linkDestinationLocation` parameter excludes clicks on Facebook Like links.

# JavaScript event callback examples

This section provides examples of populated JavaScript event callback parameters as well as the values that are passed to those parameters for multiple types of visitor interactions with Bazaarvoice content. These examples do not represent every possible interaction. Instead, they demonstrate the types of data that can be passed during a variety of different situations.

## [Product page displayed with ratings](#) on page 24

This topic shows examples of the information that is passed when ratings and reviews content is displayed for the first time during a product page load.

## [Click to read all reviews](#) on page 25

This topic shows examples of the information that is passed when a visitor clicks **Read All Reviews** in the product summary located at the top of a product page.

## [Successful submission of a review](#) on page 26

This topic shows examples of the information that is passed when a visitor writes a review and successfully advances to the confirmation page of the submission process.

## [Click to share a review on Facebook using social network buttons](#) on page 27

This topic shows examples of the information that is passed when a visitor clicks to use the social network button to share a review on Facebook.

## [Click to ask a question on a product or category page](#) on page 28

This topic shows examples of the information that is passed when a visitor clicks **Ask a Question** on a product or category page.

## [Successful submission of an answer for a specific product](#) on page 28

This topic shows examples of the information that is passed when a visitor successfully reaches the confirmation page of the answer-submission process.

## [Campaigns event: Selecting text in content](#) on page 29

When a visitor performs an action with Bazaarvoice but does not explicitly click a link, the action still triggers the JavaScript event callback. Such non-clicking interactions with content in Bazaarvoice containers are recorded as interactions with whitespace. This topic shows examples of the information that is passed when a visitor selects the text in the body of a campaign content item.

## Product page displayed with ratings

This topic shows examples of the information that is passed when ratings and reviews content is displayed for the first time during a product page load.

Parameters and values that are always passed:

- Group – Bazaarvoice
- bvProduct – RatingsAndReviews
- eventSource – Display
- eType – Read



---

Parameters and values that are sometimes passed, depending on the Bazaarvoice event that occurred:

- `subjectType` – Product
- `initialProductDisplay` – True

The `initialProductDisplay` parameter is set to `True` because the corresponding event identifies the first time that Bazaarvoice content was displayed on the product page. Subsequent visitor interactions with Bazaarvoice on the same page, such as the sorting of reviews, do not contain values for `initialProductDisplay`.

Parameters and values that are associated with product attributes:

- `productID` – SKU12345678
- `leafCategoryID` – BVsubcat12345
- `rootCategoryID` – BVcat12345
- `attributes.numReviews` – 98
- `attributes.avgRating` – 4.2143
- `attributes.percentRecommend` – 75
- `attributes.numRatingsOnlyReviews` – 2
- `urlPageName` – `category/product/samplepage.htm`

Use these values to capture such information as the average rating of a product when a visitor first viewed the relevant product page. These values can also be used to capture information about a page when a visitor interacts with it, helping you to answer the question *“Is visitor interaction with a product’s UGC proportional the product’s ratings?”*

## Click to read all reviews

This topic shows examples of the information that is passed when a visitor clicks **Read All Reviews** in the product summary located at the top of a product page.

Parameters and values that are always passed:

- `Group` – Bazaarvoice
- `bvProduct` – RatingsAndReviews
- `eventSource` – Action
- `eType` – Read

Parameters and values that are sometimes passed, depending on the Bazaarvoice event that occurred:

- `subjectType` – Product

Parameters and values that are associated with product attributes:

- `productID` – SKU12345678
- `leafCategoryID` – BVsubcat12345
- `rootCategoryID` – BVcat12345
- `attributes.numReviews` – 98

- `attributes.avgRating` – 4.2143
- `attributes.percentRecommend` – 75
- `attributes.numRatingsOnlyReviews` – 2
- `urlPageName` – `category/product/samplepage.htm`

The `eName` parameter is not passed when a visitor clicks **Read All Reviews**, even though most other types of read actions – such as sorting, paginating, and highlighting review text – feature defined `eName` parameters.

## Successful submission of a review

This topic shows examples of the information that is passed when a visitor writes a review and successfully advances to the confirmation page of the submission process.

Parameters and values that are always passed:

- `Group` – Bazaarvoice
- `bvProduct` – `RatingsAndReviews`
- `eventSource` – `Display`
- `eType` – `Write`
- `eventTarget` – `Review`

Parameters and values that are sometimes passed, depending on the Bazaarvoice event that occurred:

- `subjectType` – `Product`
- `pageType` – `Confirm`

Parameters and values that are associated with product attributes:

- `productID` – `SKU12345678`
- `leafCategoryID` – `BVsubcat12345`
- `rootCategoryID` – `BVcat12345`
- `attributes.numReviews` – 98
- `attributes.avgRating` – 4.2143
- `attributes.percentRecommend` – 75
- `attributes.numRatingsOnlyReviews` – 2
- `urlPageName` – `category/product/samplepage.htm`

The combination of `eventSource=Display` and `pageType=Confirm` indicates that a submission-confirmation page was displayed and that content was successfully submitted. The presence of `bvProduct=RatingsAndReviews` and `eventTarget=Review` indicates that the content was a review that was successfully submitted.

To measure the successful submissions of other types of content, search for `eventSource=Display` and `pageType=Confirm` along with different combinations of `bvProduct` and `eventTarget`.

# Click to share a review on Facebook using social network buttons

This topic shows examples of the information that is passed when a visitor clicks to use the social network button to share a review on Facebook.

Parameters and values that are always passed:

- Group – Bazaarvoice
- bvProduct – RatingsAndReviews
- eventSource – Action
- eType – Support
- eventTarget – Review

Parameters and values that are sometimes passed, depending on the Bazaarvoice event that occurred:

- subjectType – Product
- eName – ShoutIt
- socialMedium – Facebook

Due to the combination of `eType=Support` and `eName=ShoutIt`, the `socialMedium` parameter is populated with the name of the social network over which the visitor shares content.

Parameters and values that are associated with product attributes:

- productID – SKU12345678
- leafCategoryID – BVsubcat12345
- rootCategoryID – BVcat12345
- attributes.numReviews – 98
- attributes.avgRating – 4.2143
- attributes.percentRecommend – 75
- attributes.numRatingsOnlyReviews – 2
- urlPageName – category/product/samplepage.htm
- linkDestinationLocation – External

The `linkDestinationLocation` parameter is populated with a value of `External` because visitors who click the social network buttons are taken to a URL outside your domain.

## Click to ask a question on a product or category page

This topic shows examples of the information that is passed when a visitor clicks **Ask a Question** on a product or category page.

Parameters and values that are required:

- Group – Bazaarvoice
- bvProduct – AskAndAnswer
- eventSource – Action
- eType – Write
- eventTarget – Question

Parameters and values that are optional, depending on the Bazaarvoice event that occurred:

- subjectType – Category

The parameter `subjectType` is set to `Category` because visitors can ask and answer questions on category pages as well as on product pages.

- eName – Write

Parameters and values that are associated with product attributes:

- productID – SKU12345678
- leafCategoryID – BVsubcat12345
- rootCategoryID – BVcat12345
- attributes.numQuestions – 22
- attributes.numAnswers – 35
- urlPageName – category/samplepage.htm

The parameters `attributes.numQuestions` and `attributes.numAnswers` are present and populated with values because Q&A is present.

## Successful submission of an answer for a specific product

This topic shows examples of the information that is passed when a visitor successfully reaches the confirmation page of the answer-submission process.

---

Parameters and values that are always passed:

- Group – Bazaarvoice
- bvProduct – AskAndAnswer
- eventSource – Display
- eType – Write
- eventTarget – Answer

Parameters and values that are sometimes passed, depending on the Bazaarvoice event that occurred:

- subjectType – Product
- pageType – Confirm

Parameters and values that are associated with product attributes:

- leafCategoryID – BVsubcat12345
- rootCategoryID – BVcat12345
- attributes.numQuestions – 4
- attributes.numAnswers – 8
- urlPageName – client/submission\_container.htm

The combination of `eventSource=Display` and `pageType=Confirm` indicates that a submission-confirmation page was displayed and that content was successfully submitted. The presence of `bvProduct=AskAndAnswer` and `eventTarget=Answer` indicates that the content was an answer that was successfully submitted.

## Campaigns event: Selecting text in content

When a visitor performs an action with Bazaarvoice but does not explicitly click a link, the action still triggers the JavaScript event callback. Such non-clicking interactions with content in Bazaarvoice containers are recorded as interactions with whitespace. This topic shows examples of the information that is passed when a visitor selects the text in the body of a campaign content item.

Parameters that are always passed:

- Group – Bazaarvoice
- bvProduct – Stories
- eventSource – Action
- eType – Read

Parameters that are sometimes passed, depending on the Bazaarvoice event that occurred:

- eName – WhiteSpace
- subjectType – Product or Category

Parameters and values that are associated with product attributes:

- productID – SKU12345678
- The parameter `productID` does not exist on category pages.

- leafCategoryId – BVsubcat12345
- rootCategoryId – BVcat12345
- attributes.numStories – 98
- urlPageName – category/product/samplepage.htm

The same type of action is passed for all Bazaarvoice features. Because this example represents a campaigns page, however, the `attributes.numStories` parameter is populated with the number of content items that appear on the page. The `attributes` parameters for R&R and Q&A are not populated because they are not present on this page.

# Sending data to your web analytics account

Perform the following steps to capture details about visitor interaction with UGC:

1. Determine the method that you want to use for passing data to your web analytics provider.

Select a method of passing data that allows you to correlate interaction with Bazaarvoice to the key performance metrics that you are measuring on your site, such as conversion rate, revenue per visitor, and other success metrics. If possible, pass data into a report that allows you to segment it. Such reports are helpful for analyzing information in various ways, including rolling up detailed information about visitor interaction to determine the manner in which visitors who interact with UGC are performing as a whole.

2. Decide on the level of detail that you want to capture about visitor interaction with Bazaarvoice UGC.

For example, are you interested in learning how visitors who use R&R perform on your site as a whole, or are you interested in learning more about the content with which visitors most frequently interact?

3. Identify the parameters from the event callback that you need to pass to your web analytics platform to achieve the appropriate level of detail.

This section provides examples to help get you started.

4. Determine the manner in which you want to present the data in your web analytics reporting.

Form your web analytics tag to pass the data appropriately. Does your tagging method require the passing of data in a particular format? Can someone who views your reports interpret the end results easily?

5. Add the call to your web analytics provider into the JavaScript event callback function.

Each time an event occurs with Bazaarvoice and the event callback is populated with data, a call is made to your web analytics provider with information about that event.

[Examples of tracking user interaction on product pages](#) on page 32

The following are examples for sending data to your web analytics account when a visitor interacts with Bazaarvoice UGC on one of your product pages.

[Examples of tracking the steps of the content-submission process](#) on page 35

The following are examples of the code that you need to add to your Bazaarvoice submission pages to track each step of the content-submission process.

[Examples of tracking the Bazaarvoice attributes of a product](#) on page 37

This following are examples for passing data about the Bazaarvoice attributes of a product into your web analytics platform. Such attributes include the number and average rating of reviews, questions, answers, or campaign items.

# Examples of tracking user interaction on product pages

The following are examples for sending data to your web analytics account when a visitor interacts with Bazaarvoice UGC on one of your product pages. The examples differ only by the level of granularity about the visitor interaction that occurred. Place code that is similar to these examples on all your product pages to track visitor interaction with Bazaarvoice.

To relay data to your web analytics provider, replace the example `myExampleAnalyticsTrackEvent` function with an appropriate method for your web analytics provider. The syntax of the method that you choose might differ from the examples in this section. You can also alter the type of information that is passed to your provider by adding or removing event callback parameters.

## Least granular: Track if a visitor interacted with Bazaarvoice

Use this method to track the overall performance of visitors who use Bazaarvoice. Upon each visitor interaction with Bazaarvoice, this method sends a tag to your web analytics system with the same information, indicating that a visitor interaction with Bazaarvoice occurred.

If you are interested in the overall value of the visitors who use Bazaarvoice on your site but you possess limited segmentation capabilities in your web analytics provider, this level of detail provides a useful option for you.

The following example code shows a `show_reviews` call that has been updated with `onEvent` information in a R&R implementation.

```
<script type="text/javascript">
$BV.ui("rr", "show_reviews", {
  productId: "XXXXX",
  onEvent: function(json) {
    if (json.eventSource == "Action") {
      myExampleAnalyticsTrackEvent("Bazaarvoice Interaction Occurred");
    }
  }
});
</script>
```

In this example, `xxxxx` represents the ID of the product that is displayed on the page.

Information is passed to your web analytics provider for the following types of visitor interactions:

- Sorting content in R&R (*Bazaarvoice Interaction Occurred*)
- Sharing an answer with friends on Facebook through the social network buttons (*Bazaarvoice Interaction Occurred*)
- Clicking to post a comment on campaign content (*Bazaarvoice Interaction Occurred*)



## More granular: Track the Bazaarvoice product and type of interaction that occurred

The following example code presents a slightly more detailed example of data that can be passed into your web analytics provider.

```
<script type="text/javascript">
$BV.ui("rr", "show_reviews", {
productId: "XXXXX",
onEvent: function(json) {
if (json.eventSource == "Action") {
myExampleAnalyticsTrackEvent(json.group + json.bvProduct + json.eType);
}
}
});
</script>
```

In this example, `xxxxx` represents the ID of the product that is displayed on the page. This example provides information about the Bazaarvoice feature with which a visitor interacted and the general type of interaction that occurred. This information is easy to interpret and provides an overview of the ways in which visitors who read, write, or support UGC on each Bazaarvoice product perform on your site.

Information is passed to your web analytics provider for the following types of visitor interactions:

- Sorting content in ratings and reviews (*Bazaarvoice RatingsAndReviews Read*)
- Sharing an answer with friends on Facebook through the social network buttons (*Bazaarvoice AskAndAnswer Support*)
- Clicking to post a comment on campaign content (*Bazaavoice Stories Write*)

## Most granular: Track detailed information about visitor interactions

This example provides the most granular level of detail about the types of visitor interactions that occur. However, the related reports are typically more difficult to interpret because of the various combinations of values that are passed to your web analytics provider.

For example, some of the parameters in this example are not populated with values for each type of visitor interaction. As a result, some of the parameter values that are passed to your web analytics system are blank and appear in reporting as *undefined*. When you pass such JavaScript event callback parameters to your web analytics provider, Bazaarvoice recommends that you use a delimiter between parameters to help keep track of blank parameter values when reports are viewed.

The following example code shows a `show_reviews` call that has been updated with `onEvent` information in an R&R implementation.

```
<script type="text/javascript">
$BV.ui("rr", "show_reviews", {
productId: "XXXXX",
onEvent: function(json) {
if (json.eventSource == "Action") {
myExampleAnalyticsTrackEvent(json.group + "^" + json.bvProduct + "^"
+ json.eventSource + "^" + json.eType + "^" + json.eventTarget + "^"
+ json.eName + "^"
+(json.socialMedium || ""));
}
}
});
</script>
```

In this example, `xxxxx` represents the ID of the product that is displayed on the page. Information is passed to your web analytics provider for the following types of visitor interactions:

- Sorting content in R&R (Bazaarvoice^RatingsAndReviews^Action^Read^Review^Sort^)
- Sharing an answer with friends on Facebook through the social network buttons (Bazaarvoice^AskAndAnswer^Action^Support^Answer^ShoutIt^Facebook)
- Clicking to post a comment on campaigns content (Bazaarvoice^Stories^Action^Write^Comment^Write^)

## Bazaarvoice recommendation

To balance the level of granularity with the clarity of reporting, Bazaarvoice recommends passing the following combination of parameters to your web analytics platform.

```
<script type="text/javascript">
$BV.ui("rr", "show_reviews", {
productId: "XXXXX",
onEvent: function(json) {
if (json.eventSource == "Action") {
myExampleAnalyticsTrackEvent(json.group + "^" + json.bvProduct + "^" + json.eventSource + "^" +
json.eType + "^" + json.eventTarget + "^" + (json.eName || ""));
}
}
});
</script>
```

In this example, `xxxxx` represents the ID of the product that is displayed on the page. Information is passed to your web analytics provider for the following types of visitor interactions:

- Sorting content in R&R (Bazaarvoice^RatingsAndReviews^Action^Read^Review^Sort)
- Sharing an answer with others on Facebook through the social network buttons (Bazaarvoice^AskAndAnswer^Action^Support^Answer^ShoutIt)
- Clicking to post a comment on campaigns content (Bazaarvoice^Stories^Action^Write^Comment^Write)

# Examples of tracking the steps of the content-submission process

The following are examples of the code that you need to add to your Bazaarvoice submission pages to track each step of the content-submission process.

Adding this code to your submission pages allows you to measure the following values:

- Number of visitors who advance to the first page of the submission process
- Number of visitors who successfully submit content
- Drop-off between these two numbers
- List of events a visitor clicked on within the submission form

The following code runs when the input page, preview page, or confirmation page of any Bazaarvoice submission process is displayed to a visitor.

```
<script type="text/javascript">
$BV.ui("submission_container", {
  onEvent: function(json) {
    if ( json.eventSource == "Display" &&
        (json.eType == "Write" || json.eType == "Support") ) {
      myExampleAnalyticsTrackEvent("Bazaarvoice Submission Step: " + json.bvProduct + json.eventTarget
        + json.pageType);
    }
  }
});
</script>
```

The combination of the `bvProduct` and `eventTarget` parameters identifies the type of UGC that the visitor was in the process of submitting. The `pageType` parameter identifies the page of the submission process that was viewed.

# Track granular submission analytics

## Submission elements

The following is an example of the code that you need to place on your product pages to track activity on the Bazaarvoice submission page.

```
<script type="text/javascript">
$BV.ui("submission_container", {
  onEvent: function(json) {
    if ( json.eventSource == "Action" &&
        (json.eType == "Write" && json.eventTarget == "Review" && json.pageType == "Input") ) {
      myExampleAnalyticsTrackEvent("Bazaarvoice Submission Activity: " + json.bvProduct
+ "^" + json.eventTarget + "^" + (json.fieldName || json.eName || "Unspecified"));
    }
  }
});
</script>
```

JSON Field Name	Description
Rating	New event is fired when a user selects (not hovers) one of the rating stars on the submission form.
RatingDimension   <labelname>	New event is fired after a user selects a rating value on one of the available dimensions. The value of each rating-dimension is captured.
Recommend	New event fired after a user selects Yes/No to the question "Would you recommend this to a friend?"
Title	New event fired after a user types more than two characters in the Review Summary/Title box.
Reviewtext	New event fired after a user types more than two characters into the review body box. Another event is fired when the user reaches the preview page. This event checks the length of the review and fires a value based on the length of the review :0 to 200 characters--> submissionBodyLengthSummary=PreviewShortReview201 to 400 characters --> submissionBodyLengthSummary=PreviewMediumReview More than 400 characters --> submissionBodyLengthSummary=PreviewLongReview
Pro	New event fired after a user selects a pro tag check box or types a character into a user defined pro field.
Con	New event fired after a user selects a con tag check box or types a character into a user defined con field.
Userlocation	New event fired after a user inputs their location.

## Recommendations

The following is an example of the code you need to place on your product pages to when a visitor recommends a product in the Bazaarvoice submission page.

```
<script type="text/javascript">
$BV.ui("submission_container", {
  onEvent: function(json) {
    if ( json.eventSource == "Action" &&
        (json.eType == "Write" && json.eventTarget == "Review" && json.pageType == "Input") ) {
      myExampleAnalyticsTrackEvent("Bazaarvoice Submission Action: " + json.bvProduct +
        "^" + json.eventTarget + "^" + json.eName);
    }
  }
});
</script>
```

## Review length

The following is an example of the code that you need to place on your product pages to track the length of the review on the Bazaarvoice submission page.

```
<script type="text/javascript">
$BV.ui("submission_container", {
  onEvent: function(json) {
    if ( json.eventSource == "Display" && json.pageType == "Preview" && json.eventTarget == "Re-
view" ) {
      myExampleAnalyticsTrackEvent("Bazaarvoice Review Length: " + json.submission-
BodyLengthSummary + "^" + json.submissionBodyLength);
    }
  }
});
</script>
```

# Examples of tracking the Bazaarvoice attributes of a product

The following are examples for passing data about the Bazaarvoice attributes of a product into your web analytics platform. Such attributes include the number and average rating of reviews, questions, answers, or campaign items.

By capturing the attributes of a product at the time a visitor viewed it, you can analyze the effects on the conversion rate of metrics like the average rating or number of reviews. This analysis helps answer the following questions:

- Do products with higher ratings convert better than lower-rated products?
- If so, by how much?

To examine the overall influence of product attributes on conversion, Bazaarvoice recommends passing product-attribute information to your web analytics provider only when a product page is displayed to a visitor, not upon each visitor interaction with UGC. To minimize the number of server calls made to your web analytics provider, Bazaarvoice recommends passing this data to your web analytics provider when the page first loads, if possible.

Do not place your web analytics tags solely within the JavaScript event callback. Although Bazaarvoice strives for consistent uptime through our redundant and globally load-balanced infrastructure, it is recommended that you place another web analytics tag in the code that executes during the extremely unlikely event that Bazaarvoice is

down. This code is executed only if Bazaarvoice does not load. For information about the JavaScript blocks that execute for each product in this case, refer to your copy of the Bazaarvoice *QuickStart Implementation Guide*.

To analyze the effects of product attributes on your key performance metrics, ensure that the web analytics report to which you pass product-attribute information can report on the conversion rates of individual products.

Because product-attribute information is passed as numeric values, Bazaarvoice recommends combining this information into more meaningful groups to aid in analysis. You can use JavaScript to complete this task when the information is passed into your web analytics provider, or you can use segmentation, classification, or another method to complete this task after the data is in the provider. Ways in which you can group this information include the following examples:

- Number of reviews, questions, answers, or campaign items:
  - 0
  - 1-5
  - 6-10
  - 11-25
  - 26-50
  - 51-100
  - 101+
- Average rating:
  - -1.5
  - +1.5 – 2.0
  - +2.0 – 2.5
  - +2.5 – 3.0
  - +3.0 – 3.5
  - +3.5 – 4.0
  - +4.0 – 4.5
  - +4.5 – 5.0
- Recommend percentage:
  - 100
  - 91-99
  - 76-90
  - 51-75
  - 26-50
  - 1-25
  - 0

## Ratings and reviews

The following provides an example of the code that you need to place on your product pages to track information about the Bazaarvoice review attributes for a particular product, including the average rating, number of reviews, and percent of customers who recommend the product.

```
<script type="text/javascript">
$BV.ui("rr", "show_reviews", {
productId: "XXXXX",
onEvent: function(json) {
if ( json.eventSource == "Display" &&
json.eType == "Read" && json.initialProductDisplay ==
"True" ) {
myExampleAnalyticsTrackEvent( json.attributes.avgRating);
}
}
});
</script>
```

In this example, `xxxxx` represents the ID of the product that is displayed on the page. This code runs only when a page that contains R&R content loads.

## Question and answer

The following is an example of the code that you need to place on your product pages to track information about the Bazaarvoice Q&A attributes for a particular product, including the number of approved questions or approved answers for the product.

```
<script type="text/javascript">
$BV.ui("qa", "show_questions", {
productId: "XXXXX",
subjectType: "product",
onEvent: function(json) {
if ( json.eventSource == "Display" &&
json.eType == "Read" ) {
myExampleAnalyticsTrackEvent( json.attributes.numAnswers);
}
}
});
</script>
```

In this example, `xxxxx` represents the ID of the product that is displayed on the page. This code runs only when a page that contains Q&A content loads.

## Campaigns

The following is an example of the code that you need to place on your campaigns pages to track the number of content items for that category.

```
<script type="text/javascript">
$BV.ui("sy", "show_stories", {
productId: "XXXXX",
subjectType: "product",
onEvent: function(json) {
if ( json.eventSource == "Display" &&
json.eType == "Read" ) {
myExampleAnalyticsTrackEvent( json.attributes.numAnswers );
}
});
</script>
```

In this example, `XXXXX` represents the ID of the product that is displayed on the page. This code runs only when a page that contains campaigns content loads.



---

# Reports

This following identifies best practices for reporting on visitor interactions with Bazaarvoice UGC and the effects of product attributes against key performance metrics.

## Visitor engagement

After you finish tagging your site and start capturing information about the interaction of visitors with the UGC on your site, set up reporting for the visitors who did and did not use Bazaarvoice.

It is recommended that you segment all visitors who interacted with Bazaarvoice content as visitors who used Bazaarvoice during their visit. For comparison purposes, build another segment of visitors who viewed pages that displayed Bazaarvoice content but who did not interact with it. This segment provides a performance baseline against which to measure visitors who interacted with Bazaarvoice. Analyze both segments against your key performance indicators (KPIs) to calculate the increase that visitors who engage with Bazaarvoice bring to your site. This increase represents the value that visitors who engage with UGC contribute to your site.

Build the following segments:

- Used Bazaarvoice – Identifies visitors who performed the following actions:
  - Viewed a page that displayed Bazaarvoice content, such as a product page
  - Clicked a **Read all Reviews** link, a tab over Bazaarvoice content, or any other Bazaarvoice link that results in the firing of a tag
- Did Not Use Bazaarvoice – Identifies the following types of visitors:
  - Visitors who viewed a page that displayed Bazaarvoice content, such as a product page
  - Visitors who did not click a **Read all Reviews** link, a tab over Bazaarvoice content, or any other Bazaarvoice link that results in a firing of a tag

This segment excludes visitors who did not actively interact with Bazaarvoice. Visitors who simply view content by scrolling to it without clicking it are not excluded, making this segment a conservative one.

When setting up these segments, verify the accuracy of the filter for pages that display Bazaarvoice content. For instance, filtering on a product view can sometimes skew reports. If you perform filtering on the URL, ensure that the regular expression is accurate. If you are unable to limit your segments to visitors who viewed a product view, you can still compare visitors who used Bazaarvoice against all site visitors. However, metrics are likely be skewed in favor of Bazaarvoice in such an instance because not all visitors viewed pages that contain Bazaarvoice content. Visitors who did not visit a product page are probably not as far down the consideration path as visitors who did visit a product page.

The following image shows an example report that displays the performance comparison of visitors who engaged with Bazaarvoice content versus those who did not.



### Client Name Social Commerce Insights Results:

January 11, 2012 to February 10, 2012

	Ratio/% Lift	Used Bazaarvoice	Did Not Use Bazaarvoice
<b>VISITOR BEHAVIOR</b>			
>> Visits	8.7%	7,978	91,988
>> Page Views / Visit	105.4%	7.9	3.8
>> Time on Site (min)	162.2%	5.8	2.2
<b>E-COMMERCE</b>			
>> Orders	15.8%	212	1,339
>> Revenue	15.8%	\$17,236	\$108,962
>> Conversion Rate	82.2%	2.66%	1.46%
>> AOV	-0.1%	\$81.30	\$81.38
>> Revenue / Visit	82.4%	\$2.16	\$1.18

\* Used Bazaarvoice = visits that used Bazaarvoice UGC during their visit

\* Did Not Use Bazaarvoice = visits that reached a product page but did not engage with any Bazaarvoice UGC content during their visit

## Submission pages

If you enable tracking on your Bazaarvoice submission pages, you can use the information that you capture to calculate drop-off rates for specific types of UGC. These numbers provide an indication of whether visitors encounter obstacles while trying to submit content. Specifically, you can use this information to determine the number of visitors who reached each of the following steps of the submission process:

1. Input page
2. Preview page
3. Confirmation page

By examining the number of visitors who reach each page of the submission process, you can determine the abandonment rates between steps. When determining these rates, be sure to examine each type of Bazaarvoice content separately – such as reviews, comments, questions, and answers – as each type features a different submission process.

Due to the amount of content that visitors must enter successfully to reach the preview page, the abandonment rate between the input and preview pages is typically higher than the abandonment rate between the preview and confirmation pages. However, many clients require their customers to log on and authenticate before they can submit content. Consequently, the point of greatest drop-off tends to lie between a visitor's initial click to write content and the visitor reaching the input page. To measure the drop-off between visitors who click to write content and visitors who reach the input page, tag visitor interactions on your submission pages as well as on your product pages.

---

When examining this behavior, keep in mind that the difference between visitors who click to write content and visitors who reach the input page might be skewed if you have implemented a post-interaction email campaign to solicit additional reviews. Such visitors land directly on an input page and do not need to click a link to start the submission process.

Use your web analytics system to measure the number of visitors who submit content by observing the number who reach the confirmation page of the submission process. For any particular content type, this number typically equals the volume of content that visitors have submitted during a specific time period. However, differences exist between the number of visitors who arrived at the confirmation page and the actual amount of content that was submitted to your site for the following reasons:

- The method by which your web analytics system counts visits to the confirmation page likely differs from the method by which the volume of Bazaarvoice content is submitted.
- Bazaarvoice moderators might not have approved the content that a visitor submitted, resulting in a visit to the confirmation page without the content appearing on your site.



3900 North Capital of Texas Highway

Suite 300, Austin, Texas 78746

Toll-Free: (866) 522-9227 | Phone: (512) 551-6000 | Fax: (512) 551-6001

[www.bazaarvoice.com](http://www.bazaarvoice.com)