# Share Bar

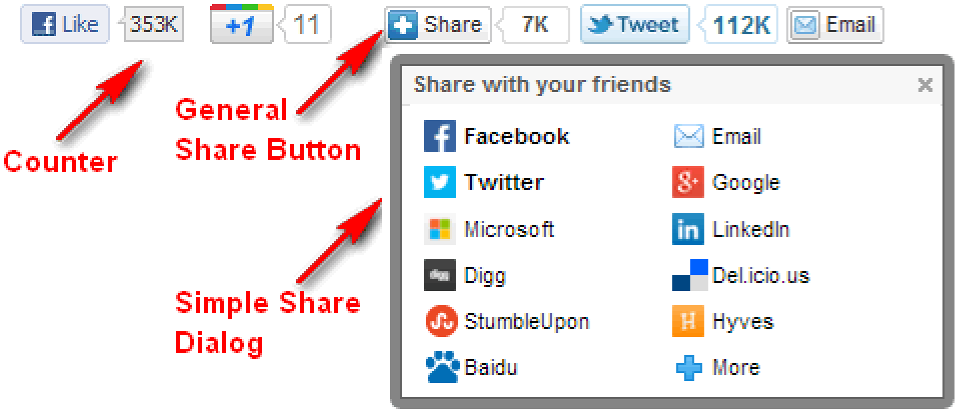
## Introduction

The Share Bar add-on allows you to easily place in your site a bar with different buttons for sharing content to social destinations. The Share Bar add-on is built out of one or more buttons, arranged horizontally. Each button represents a direct bookmark destination, a general share button, or an email button.

Once a user clicks a direct bookmark button, the window of the provider will be opened for direct bookmarking. Once a user clicks the general share button, the Simple Share mode of the Share add-on will popup, enabling your user to share and bookmark content and site activities to selected social destinations.

The Share Bar buttons may be displayed with or without counters, representing the number of shares.
For more information regarding the counters, refer to the socialize.getProviderShareCounts API method.

You can use the **Share Bar Wizard** to customize the visual aspects of the Share Bar add-on design, view, and grab the customized code.



### What's in this Guide

This document is a step-by-step guide for integrating the Share Bar add-on and also provides a reference and guide for the various customization options. For convenience, we have divided this document into levels of complexity:

- Quick Start Implementation - learn how to quickly integrate the Share Bar add-on, and share to the various social networks.
- Changing to Multiple Selection Share - learn how to change the default share display.
- Share Bar add-on Customization - learn how to customize the default design of the buttons in the share bar.

## Quick Start Implementation

**Before you begin:** Make sure that you have already set up social network application settings before you continue. For more information on how to set up an external application for use with your Gigya account, see the Setting Up External Applications in Social Networks documentation and select a specific network.

### Implementation Overview

Adding the Share Bar add-on is simple, and involves only adding a few lines of code:

- Step 1: Define the action fields for the share item. The newsfeed item that is published to the social networks is constructed from the fields of this object.

  This can be done either by constructing a UserAction object, or by adding Open graph meta-tags to your HTML, though it is recommended to implement **both**.
- Step 2: Define the Share Bar add-on's params object, including a comma-separated list of providers, representing the buttons to display in the share bar.
- Step 3: Call the socialize.showShareBarUI method to activate the Share Bar add-on.

## Code Example

Add the following code snip to the of your page to activate the Share Bar add-on that is presented in the screenshot above:

```
<DIV id=divButtons></DIV> <!-- Share Bar add-on DIV Container -->

<script language="javascript">
    // Step 1: Construct a UserAction object and fill it with data
 var ua = new gigya.socialize.UserAction();
 ua.setLinkBack("http://www.youtube.com/watch?v=jqxENMKaeCU");
 ua.setTitle("HOME");

    // Step 2: Define the Share Bar add-on's params object
 var params ={
  userAction:ua,
  shareButtons:'facebook-like,google-plusone,share,twitter-tweet,email', //
list of providers
  containerID: 'divButtons',
        cid:''
 };

    // Step 3: Load the Share Bar add-on:
    gigya.socialize.showShareBarUI(params);
</script>
```

## Code Explanation

### Step 1: Construct a UserAction object and fill it with data

The UserAction object represents a User Action. We use the UserAction object constructor for this purpose:

```
var ua = new gigya.socialize.UserAction();
```

After you have constructed the object, you can use the various UserAction member methods to set the object's properties. For example:

```
ua.setTitle("This is my title");
ua.setDescription("This is the textual content of my post");
...
```

The UserAction's object properties represent the various items of which a User Action may consist: Title, Description etc.

**Note**: In addition to 'User object', you must define Open Graph Tags on pages that use the Share add-on. Defining the share data using only 'User object' is not recommended, as some social networks use OG-Tags for page scraping.
Set up the tags to get a better structured news feed post on Facebook ('like'), Twitter and Google (+1), and invoke the 'share' dialog by adding *facebookDialogType: 'share'* to the params object, as the default 'Feed dialog' was deprecated by Facebook. In addition to OG tags, some sites support extended tagging schemas. Read more on how to define OG and extended tags on the 'Content Sharing' page.

### Step 2: Define the Share Bar add-on's parameters

```
var params = {
 userAction:ua,
 shareButtons:'facebook-like,google-plusone,share,twitter-tweet,email', //
list of providers
 containerID: 'divButtons',
    cid:''
};
```

The Share Bar add-on has three required parameters:

- *containerID* - The identifier of the DIV element, where the Share Bar add-on should be embedded.
- *shareButtons*- The list of providers.
- *userAction* - The userAction object that we have created in Step 1.
- *facebookDialogType* - Options are: 'share' (default) or 'feed'. The 'share' dialog will scrape the post data from the open graph tags defined on the page, while the 'feed' dialog will use data from the 'user action object'.

In this example the ShareButtons parameter is comprised of a list of providers. Instead, you may define an array of ShareButton objects:

```
var params ={
 userAction:ua,
 shareButtons:
 [
  { // General Share Button
   provider:'share',
   tooltip:'General Share Button',
   userMessage:'default user message'
  },
  { // Google +1 button
   provider:'google-plusone',
   tooltip:'Recommend this on Google',
   userMessage:'default user message'
  },
  { // Facebook Like button
   provider:'facebook-like',
   tooltip:'Recommend this on Facebook',
   action:'recommend',
   font:'arial'
  },
  { // Twitter Tweet button
   provider:'twitter-tweet',
   tooltip:'Share on Twitter',
   defaultText:'Twitter message'
  },
  { // Email button
   provider:'email',
   tooltip:'Email this',
   iconImgUp:'http://mySite.com/images/myEmailIcon.gif',
            iconImgOver:'http://mySite.com/images/myEmailIconOver.gif'
         }
 ],
 containerID: 'divButtons',
    cid:''
};
```

The add-on has additional optional parameters you can use. Please refer to the method's parameters table for the entire list of optional parameters.

### Step 3: Invoke the Share Bar add-on

Call the socialize.showShareBarUI API method:

```
gigya.socialize.showShareBarUI(params);
```

Like every Gigya API method, the socialize.showShareBarUI method receives one parameter - the parameters object, which is the object we have defined in Step 2. (Learn more about Gigya's Basics here).

## A Working Example

In the Share Bar add-on demo you may find a complete working example of a share bar that is used to publish user actions. You may view the code, run it and view the outcome.

## How the URL to be Shared is Determined

The share buttons are used to share content, and more specifically share a URL. The URL to be shared is determined according to a certain logic. The following list shows the order in which Gigya checks whether a URL exists, and if it does, this is the URL used for sharing:

1. The ShareButton object's 'url' field
2. The linkback field of the ShareButton object's userAction
3. The linkback field of the UserAction
4. The Open Graph URL tag (og:url tag)
5. The canonical meta tag* (if defined)
6. The URL of the current page

> When sharing or shortening URLs via any Gigya Add-ons, APIs, or methods, the URL being used must be a publicly accessible URI. If the URI is behind a firewall, an HTTP Auth, or does not respond within 5 seconds of a request to retrieve the URL of the page, even if the URL is within a whitelisted domain, Gigya will respond with errorCode **400120** - **Invalid Site Domain**, and the request will fail.

*Defining a canonical meta tag is done by adding a rel="canonical" link to the section of the non-canonical version of each HTML page. To specify a canonical link to the page http://www.example.com/product.php?item=swedish-fish, create a element as follows:

```
<link rel="canonical"
href="http://www.example.com/product.php?item=swedish-fish"/>
```

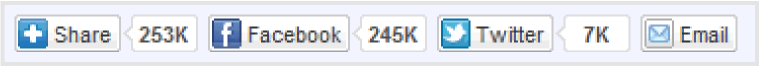
## Changing to Multiple Selection Share

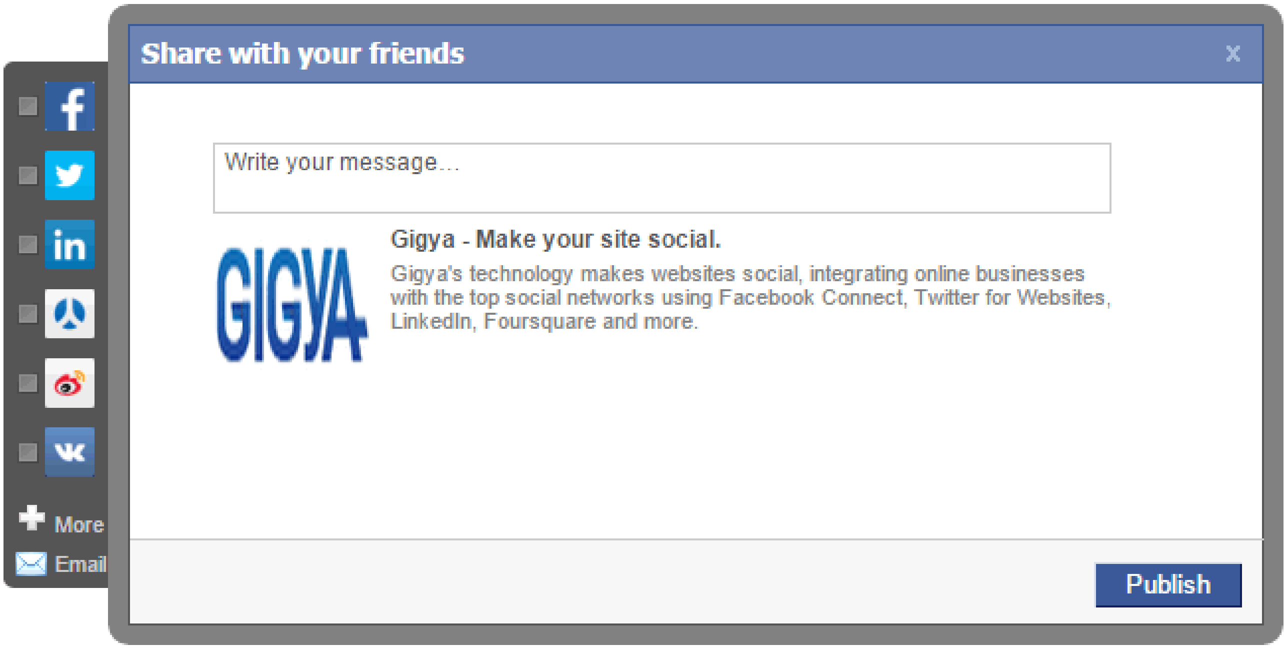
When using the general share button in the Share Bar add-on, the default Share dialog that is displayed is the Simple Share dialog. You can change this to the multiple selection share, which is the more elaborate Share add-on.

Change the simple share to multiple selection share by adding the optional *operationMode* parameter to the Share Bar parameters, with the value 'multiSelect':

```
var params = {
 userAction:ua,
 shareButtons:'share,facebook,twitter,email', // list of providers
 containerID: 'divButtons',
    operationMode:'multiSelect',
};
```

The general share button will now trigger the full Share add-on:





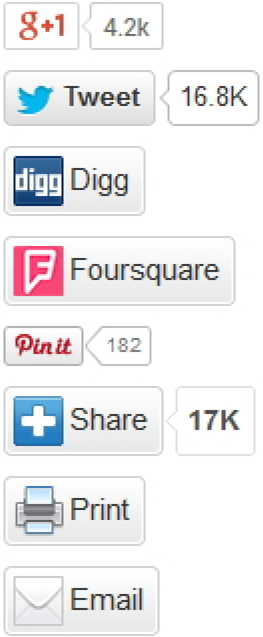
Gigya - Make your site social.

Gigya's technology makes websites social, integrating online businesses with the top social networks using Facebook Connect, Twitter for Websites, LinkedIn, Foursquare and more.

You can now use the Share Bar add-on with the full Share add-on capabilities. Please refer to the Share add-on Guide for further reading.

## Share Bar UI in Mobile View

When the socialize.showShareBarUI **deviceType** parameter is set to **mobile**, the share bar UI is displayed in the mobile mode, with the share bar either in vertical presentation, or horizontal presentation:



**Note:** When the Share Bar is set to deviceType mobile or to auto (on mobile), clicking the email share will open the mail client rather than the Gigya share UI.

## Special Buttons

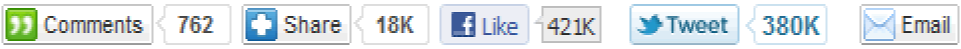
### Adding a Comments Counter

You can add a **Comments** button to your Share Bar, with a counter indicating the number of comments made in that specific comments stream. The comments stream is identified by its unique categoryID and streamID.

In order to display the **Comments** button with a counter, define a ShareButton object with '*comments*' as the provider, and the categoryID and streamID that identify the comments stream:

```
shareButtons:
 [
  { // Comments Button
   provider:'comments',
   categoryID: 20358285,
   streamID: '1'
  },
  { // General Share Button
   provider:'share',
   tooltip:'General Share Button',
   userMessage:'default user message'
  },
  { // Facebook Like button
   provider:'facebook-like',
   tooltip:'Recommend this on Facebook',
   action:'recommend'
  },
  { // Twitter Tweet button
   provider:'twitter-tweet',
   tooltip:'Share on Twitter',
   defaultText:'Twitter message'
  },
  { // Email button
   provider:'email',
   tooltip:'Email this',
   iconImgUp:'http://mySite.com/images/myEmailIcon.gif',
            iconImgOver:'http://mySite.com/images/myEmailIconOver.gif'
        }
 ]
```

The Share Bar that is displayed for this code is:



## Localization for Facebook Like Button

In order to display the **Facebook Like** button, define a ShareButton object or string with *'facebook-like*' as the provider. If you wish to display the button in a different language, the language needs to be set in the socialize JS URL before the add-on is loaded. Learn about this in Localization Support for 3rd Party add-ons.

## Twitter Tweet Button

### Special Twitter Button Parameters

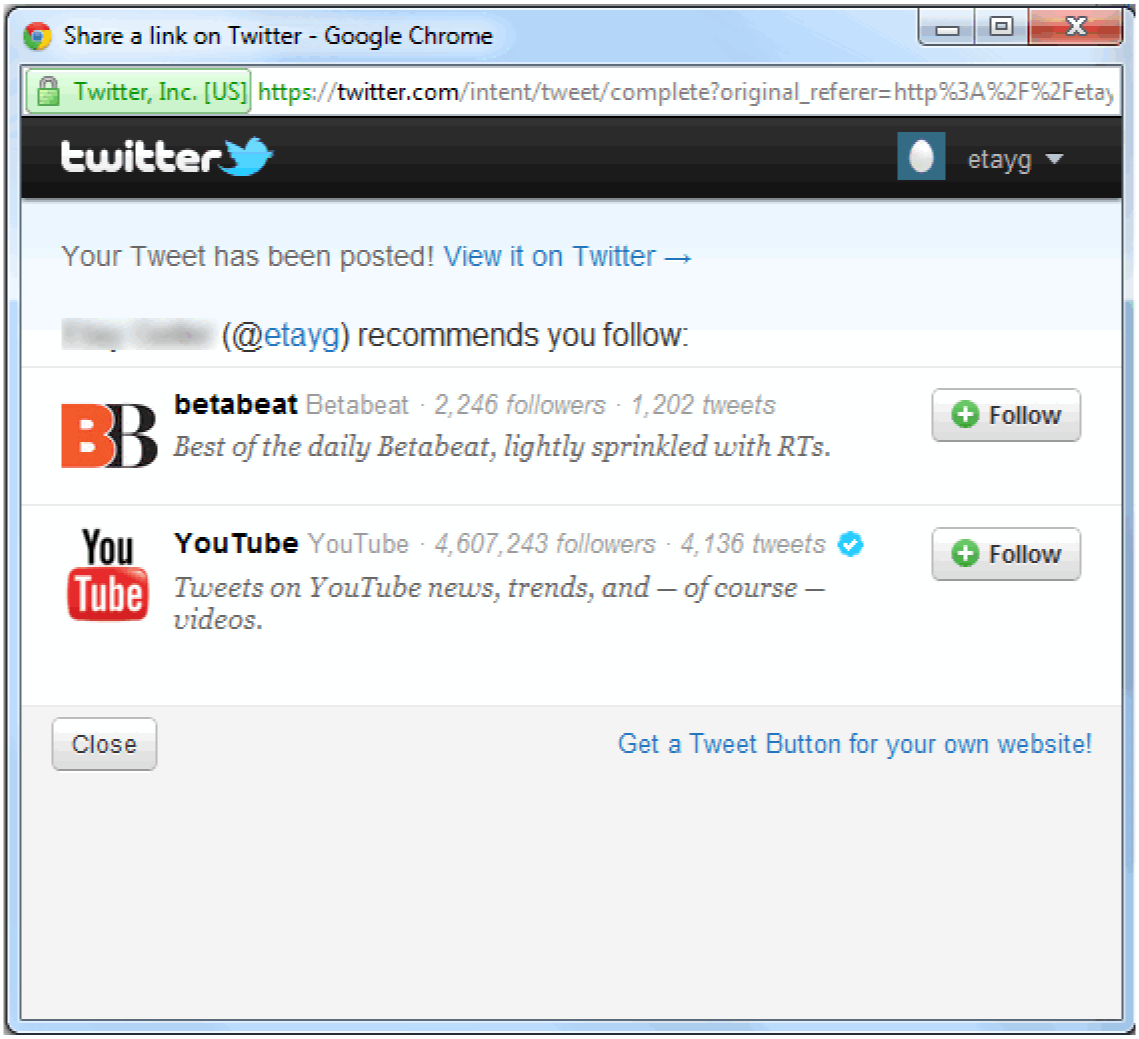
You can configure the Twitter Tweet button so that:

- After tweeting, you can suggest to the user to follow related accounts, by using the *'related'* parameter.
- You can assign the screen name of the user to whom to attribute the Tweet, by using the *'via'* parameter.

Define a ShareButton object with 'provider', 'related', and 'via' fields:

```
shareButtons:
[
    { // Twitter Tweet button
        provider:'twitter-tweet',
        related: 'betabeat,youtube',
        via:'etayg'
    }
]
```

The following Twitter window includes the screen name of the user to whom to attribute the Tweet, i.e. @etayg, and two recommended related accounts:


Share a link on Twitter - Google Chrome
Twitter, Inc. [US]
https://twitter.com/intent/tweet/complete?original_referer=http%3A%2F%2Fetay
twitter
etayg

Your Tweet has been posted! View it on Twitter →

[redacted] (@etayg) recommends you follow:

**betabeat** Betabeat · 2,246 followers · 1,202 tweets
*Best of the daily Betabeat, lightly sprinkled with RTs.*

**YouTube** YouTube · 4,607,243 followers · 4,136 tweets
*Tweets on YouTube news, trends, and — of course — videos.*

Follow
Follow
Close
Get a Tweet Button for your own website!

The Twitter Tweet button has additional optional parameters you can use. Please refer to the ShareButton object reference socialize.showShareBarUI for the entire list of optional parameters.

## Google +1 Supported Browsers

In order to display the **Google +1** button, define a ShareButton object or string with '*google-plusone*' as the provider. There are few browsers in which the **Google +1** button is not supported. Please refer to the Supported browsers for the updated list by Google.

## Google+ Interactive Post

Google+ provides an interactive post feature. Using this feature users can post shares to G+ with a special call-to-action button, and have a deep link to their mobile apps.

Generally, the interactive button allows a post to appear in Google+ with a special "call to action" link, which is displayed in addition to the standard "content" link. For example, "goodsongs" can set an interactive Google+ button, so that whenever users share songs from goodsongs.com, in Google+, each shared song will have a "Listen" button next to it ("call to action") and not just a standard link to the song ("content" link).

In addition, the feature allows configuring "deep links" to be used in mobile apps, so the shared song will not just open the "goodsongs" app (when accessed through a mobile platform), but also directs it to the right place where this song can be played.

Add the Google+ Interactive Post by defining a ShareButton object with provider "googleplus-interactive" and the button's fields:

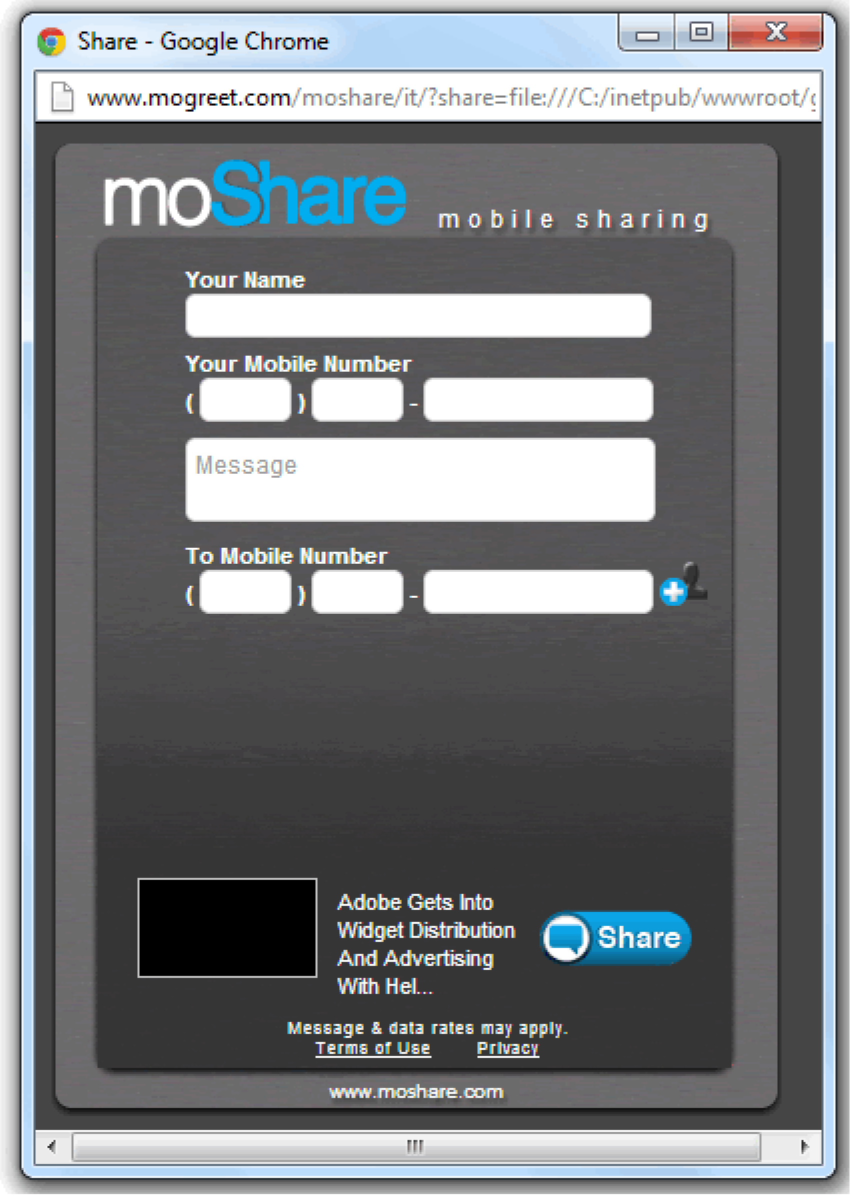
```
shareButtons:
[
    { // Google+ Interacvive Post button:
        provider:'googleplus-interactive',
        callToActionLabel: 'CREATE',
        callToActionURL: 'http://plus.google.com/pages/create',
        callToActionDeepLinkID: '/pages/create',
        clientID: 'xxxx.apps.googleusercontent.com',
        contentURL: 'https://plus.google.com/pages/',
        contentDeepLinkID: '/pages',
        cookiePolicy: 'single_host_origin',
        preFillText: 'Create your Google+ Page too!'
    }
]
```

We also allow CSS customization of the button. For more information, please refer to: https://developers.google.com/+/web/share/interactive#button_design
For more information about the Google+ Interactive Post button in general, and about each field, please refer to https://developers.google.com/+/web/share/interactive.

## Sharing via Text Message

In order to share via text message, define a ShareButton object or string with *'moshare-sms*' as the provider. moShare cannot be used over https.

When you click on the moShare icon, the following window will open:



www.mogreet.com/moshare/it/?share=file:///C:/inetpub/wwwroot/

Message & data rates may apply.

## Send to Kindle

In order to send to Kindle using the Share Bar, define a ShareButton object or string with *'kindle'* as the provider.

Marking the **title**, **author**, and **publication date** on your pages can be done using the following methods: Custom CSS Selectors, Facebook's Open Graph, schema.org, hNews, and Blogger / Blogspot. In the same manner, you can mark the **main body of content** on your pages using Custom CSS Selectors, schema.org, hNews, and Blogger / Blogspot

- **Custom Css Selectors** - you can mark the title, author, and publication date on your pages using Custom CSS Selectors. Read about the possible CSS Selectors in the socialize.showShareBarUI method.
- **Facebook's Open Graph** meta tags allow you to define the title, author, page name and publication date. An example of markup metadata with the Open Graph specification:

```
<!doctype html>
<html>
<head>
  <title>Kindles are Fantastic</title>
  <meta property="og:title" content="Kindles are The Best" />
  <meta property="og:article:author" content="Johannes Gutenberg" />
  <meta property="og:article:published_time" content="2007-11-19" />
</head>
<body>
  <h1>Kindles are The Best</h1>
  <h2>By Johannes Gutenberg</h2>
  <h2>2007-11-19</h2>
  <div>
    <p>I am of the humble opinion the Kindle is the best e-reader
available.</p>
  </div>
</body>
</html>
```

- **schema.org** is a standard to provide information about a website for search engines, it allows them to identify the relevant metadata of the page. An example of markup metadata with the schema.org specification:

```
<!doctype html>
<html>
<head>
  <title>Kindles are Fantastic</title>
</head>
<body>
  <h1 itemprop="name">Kindles are Fantastic</h1>
  <h2>By Johannes Gutenberg</h2>
  <h2>2007-11-19</h2>
  <div itemprop="articleBody">
    <p>I am of the humble opinion the Kindle is the best e-reader
available.</p>
  </div>
</body>
</html>
```

- **hNews** is an open standard for news content, it allows you to describe journalistic work. AN example of markup metadata with the hNews specification:

```
<!doctype html>
<html>
<head>
  <title>Kindles are Fantastic</title>
</head>
<body>
  <h1 class="entry-title">Kindles are The Best</h1>
  <h2 class="byline"<ac:structured-macro
ac:name="anchor"><ac:parameter ac:name="">By Johannes
Gutenberg</ac:parameter></ac:structured-macro></h2>
  <h2>2007-11-19</h2>
  <div class="hentry">
    <p>I am of the humble opinion the Kindle is the best e-reader
available.</p>
  </div>
</body>
</html>
```

- **Blogger / Blogspot** - Using the send to Kindle button with Blogger / Blogspot - select this option if you are using a "Classic" Blogger template.

## Provider mixi

Since mixi is a provider that requires an API key in order to work, you must add a parameter called ***mixiKey*** with the mixi API key to the showShareBarUI parameters:

```
var params = {
 userAction:ua,
 shareButtons:'share,mixi,facebook,twitter,email', // list of providers
    containerID: 'divButtons',
    mixiKey:'<Insert the mixi API Key here>'
};
```

## Provider Pinterest

When you add Pinterest to your share bar, you must also define a UserAction object, provide a media item and add the media item to your UserAction object:

```
// Define an image media item:
var image = {
  type: 'image',
  src:
'http://www.infozoom.ru/wp-content/uploads/2009/08/d06_19748631.jpg',
  href: 'http://www.infozoom.ru'
  };

// Define a UserAction onject
var ua = new gigya.socialize.UserAction();
ua.addMediaItem(image);

// Define Share Bar add-on's Parameters
var shareBarParams = {
  userAction:ua,
  shareButtons:
"share,stumbleupon,pinterest,email,twitter-tweet,google-plusone,facebook-l
ike",
  containerID: 'divButtons' // location of the Share Bar add-on
};

// Load Share Bar add-on
gigya.socialize.showShareBarUI(shareBarParams);
```

Read more about adding media items to the UserAction object.

**Note:** If you want to add the Pin It button to mobile apps, please refer to Pinterest's documentation:

- For iOS: http://developers.pinterest.com/ios/
- For Android: http://developers.pinterest.com/android/

# Share Bar add-on Customization

The Share Bar design can be customized using:

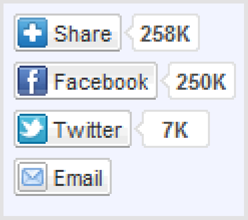
- The socialize.showShareBarUI method parameters
- HTML Elements Style (CSS) Classes
- Templates

## Method Parameters

You can customize the Share Bar add-on using the following socialize.showShareBarUI method parameters:

- layout - using this parameter you can arrange the buttons in a vertical bar instead of a horizontal bar
- showCounts - using this parameter you can display the counters above the corresponding buttons instead of on their right, or choose to display the bar with no counters
- noButtonBorders - using this parameter you can display buttons without borders

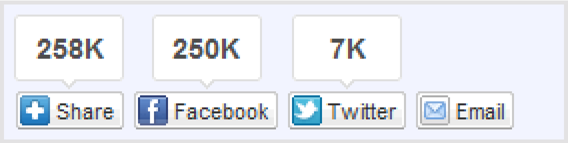
### Vertical Layout



To arrange the buttons in a vertical bar instead of a horizontal bar, add the *layout* parameter to the Share Bar add-on's parameters, with the value 'vertical':

```
var params = {
 userAction:ua,
 shareButtons:'share,facebook,twitter,email', // list of providers
 containerID: 'divButtons',
    layout:'vertical',
};
```

## Counters on Top

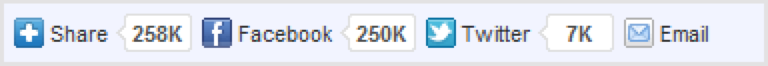


To display the counters above the corresponding buttons instead of on their right, add the *showCounts* parameter to the Share Baradd-on's parameters, with the value 'top':

```
var params = {
 userAction:ua,
 shareButtons:'share,facebook,twitter,email', // list of providers
 containerID: 'divButtons',
    showCounts:'top',
};
```

In the same manner you can display the Share Bar without counters, by adding the *showCounts* parameter to the Share Bar add-on's parameters, with the value 'none'.

## No Button Borders



To arrange the buttons without button borders, add the *noButtonBorders* parameter to the Share Bar add-on's parameters, with the value 'true':

```
var params = {
 userAction:ua,
 shareButtons:'share,facebook,twitter,email', // list of providers
 containerID: 'divButtons',
    noButtonBorders:true,
};
```

## Applying a New Button Design

It is possible to customize any non-native share button in the Share Bar. Applying a new design can be done categorically for all non-native buttons by applying image URLs to the *buttonImages* parameter of the socialize.showShareBarUI method, or specifically to a single button by supplying an image URL to the *iconImgUp* Paramter within the shareButton object.

**Note**: Button designs can only be applied to buttons generated by Gigya. socialize.showShareBarUI also allows you to use the social network's native buttons; however, styling applied to these will have no effect.

Native social network buttons which cannot be customized are listed in the description of the '*shareButtons*' parameter of the socialize.showShareBarUI method parameters, they include: *facebook-subscribe, facebook-send, facebook-share, facebook-like, google-plusone, googleplus-share, googleplus-interactive, getglue, moshare-sms, twitter-tweet, twitter-hashtag, twitter-mention, comments, foursquare, foursquare-save, pinterest-pinit, kindle, hatena-bookmark and lineit.*

The following are the categorical customization fields:

- *buttonLeftImgUp* - An image for the left part of the button
- *buttonCenterBGImgUp* - An image for the center part of the button
- *buttonRightImgUp* - An image for the right part of the button

To replace the default button images with your alternative images, set these fields with URLs to the corresponding images.

Add the *buttonImages* parameter to the Share Bar parameters with your own button images:

```
var params = {
 userAction:ua,
 shareButtons:'share,facebook,twitter,email', // list of providers
 containerID: 'divButtons',

buttonImages:{buttonLeftImgUp:'Button_Left.png',buttonCenterBGImgUp:'Button_Middle.png',buttonRightImgUp:'Button_Right.png'}
    };
```

You may also define images to the 'Disabled' 'Mouse over' and 'Down' states of the button, using the corresponding fields of the *buttonImages* parameter:

- *buttonLeftImgDisabled, buttonCenterBGImgDisabled, buttonRightImgDisabled* -  Images for Disabled button (left, center and right).
- *buttonLeftImgOver, buttonCenterBGImgOver, buttonRightImgOver* - Images for Mouse Over state (left, center and right).
- *buttonLeftImgDown, buttonCenterBGImgDown, buttonRightImgDown* - Images for Down state (left, center and right).

For example:

```
buttonImages:
 {
 buttonLeftImgUp:'Button_Left.png',buttonCenterBGImgUp:'Button_Middle.png'
,buttonRightImgUp:'Button_Right.png',

buttonLeftImgDown:'Button_Left_Down.png',buttonCenterBGImgDown:'Button_Mid
dle_Down.png',

buttonRightImgDown:'Button_Right_Down.png',buttonLeftImgDisabled:'Button_L
eft_Disabled.png',

buttonCenterBGImgDisabled:'Button_Middle_Disabled.png',buttonRightImgDisab
led:'Button_Right_Disabled.png',

buttonLeftImgOver:'Button_Left_Over.png',buttonCenterBGImgOver:'Button_Mid
dle_Over.png',
    buttonRightImgOver:'Button_Right_Over.png'
 }
```

## Customizing a Specific Button

Share buttons are defined via the ***shareButtons*** parameter of the socialize.showShareBarUI method. While button images are obtained by default according to the provider of each button, they can also be customized.
To specify an image for a single, specific button (e.g altering the Facebook share button image), set the ***iconImgUp*** parameter, within the **button object** you want to change, to the image URL you would like to set as the share button.

For example:

```
var shareBarParams = {
        shareButtons:
        [
            { // General Share Button
                provider:'share',
                tooltip:'General Share Button',
            },
            { // Google+
                provider:'google',
                tooltip:'Recommend this on Google',
                iconImgUp:'www.exampleImages.com/googleShare.ico'
            },
            { // Facebook
                provider:'facebook',
                tooltip:'Recommend this on Facebook',
                iconImgUp:'www.exampleImages.com/facebookShare.ico'
            },
        ],
        userAction:ua,
        containerID: 'shareBar'
    };
    // Load Share Bar add-on
    gigya.socialize.showShareBarUI(shareBarParams);
```

## HTML Elements Style (CSS) Classes

Any elements on the add-on that has an ID can be overridden with supported style attributes. You may use Firebug to identify an HTML element on the add-on and assign a new style to it. Please make sure to use the add-on's container ID when assigning style, so it will not affect other elements on the page.

For example:

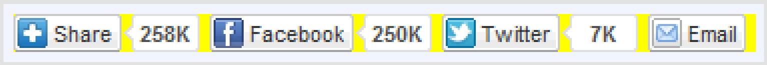
```
<style> #divButtons .gig-share-button-container {background-color:yellow;}
</style>
```

There are five classes that you can use to customize your buttons, overriding the default design:

- gig-share-button-container - overrides the default design of a button and counter
- gig-share-button - overrides the default design of a single button
- gig-share-button-text - overrides the default design of a button's label
- gig-share-counter - overrides the default design of a single counter
- gig-share-counter-text - overrides the default design of a counter's value

The following examples demonstrate how to change the default background color of each class specified here. You may change the font size, font color, border, etc. In order to more

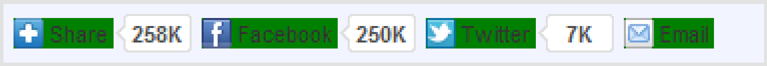
### Button and Counter



To design a button and counter with a yellow background, add the following to the section of your page:

```
<style> #divButtons .gig-share-button-container {background-color:yellow;}
</style>
```

## Button

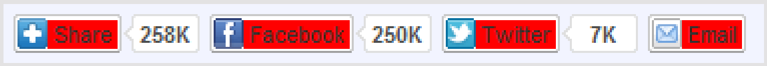


To design a button with a green background, add the following to the section of your page:

```
<style> #divButtons .gig-share-button {background-color:green;}</style>
```

In this example, the button is without borders to allow the green background to be more visible.

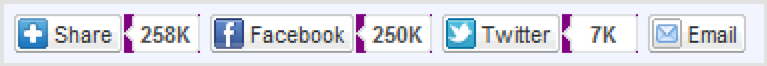
## Button Label



To design a button label with a red background, add the following to the section of your page:

```
<style>#divButtons .gig-share-button-text {background-color:red;} </style>
```

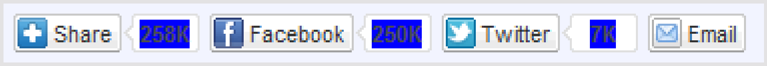
## Counter



To design a button counter with a purple background, add the following to the section of your page:

```
<style> #divButtons .gig-share-counter {background-color:purple;} </style>
```

## Counter Value



To design a counter value with a blue background, add the following to the section of your page:

```
<style> #divButtons .gig-share-counter-text {background-color:blue;}
</style>
```

### Additional Classes

In addition, you can further customize your share bar buttons using the following classes:

- gig-button-up
- gig-button-down
- gig-button-over
- gig-button-disabled

For example, add the following to the section of your page to change the button text to yellow when the button is disabled:

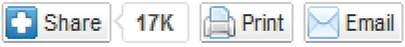
```
<style>#divButtons .gig-button-disabled .gig-share-button-text
{color:yellow} </style>
```

## Using Button Templates

The **buttonTemplate** and **buttonWithCountTemplate** parameters of the socialize.showShareBarUI method enables you to override the default design of buttons in the Share Bar by defining an HTML template representing the appearance of each button. Within the template, you may use CSS attributes to style the button.

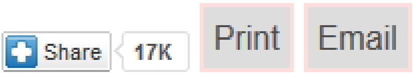
- The **buttonTemplate** parameter affects the design of buttons that do not include a counter, such as "Print" and "Email". The HTML template supports the placeholders *$iconImg, $text, $onClick*.
- The **buttonWithCountTemplate** parameter affects the design of buttons with counters, such as the simple "Share" button. The HTML template supports the placeholders *$iconImg, $text, $onClick*, *$count*.

For example, here is a simple bar with no buttonTemplate parameter defined:



The following picture shows the same bar with the following line added to the parameters of the method:

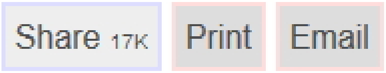
```
buttonTemplate: '<div style="padding: 5px; background-color: #ddd; border:
2px #fdd solid; font-size: medium" onclick="$onClick">$text</div>'
```



Note that the Share button is not affected, because it is a button with a counter. To apply a template to the Share button, add a buttonWithCountTemplate parameter as well, e.g.:

```
buttonWithCountTemplate: '<div style="padding: 5px; background-color: #eee;
border: 2px #ddf solid; font-size: medium" onclick="$onClick">$text
$count</div>'
```

The result:



**Notes:**

- When using **buttonTemplate** or **buttonWithCountTemplate**, you *must* specify the *$onClick* placeholder (e.g. "onclick="$onClick" in the example above).
- Custom design can only be applied to *generic* and *non-native* buttons. Styling applied to native social network buttons, such as the Facebook "Like" button, has no effect. To change the appearance of native social network buttons, use the specific parameters available for each button, as detailed in the description of the ShareButton object.