**MexBS Facebook Shop Integration v1.0.0 User Guide**

Thank you for purchasing the Facebook Shop Integration extension and welcome to the MexBS family! In this document, we will guide you through the installation steps, and will show you how to use and get the maximum from your extension.

**Installation**

In this tutorial, we will show you how to easily add a shop tab to your Facebook page, and to sync your Magento store with it.

**Step 1 – Extension Installation**

In this step we will install the extension on your server.

Before we begin, please ensure that you have a backup of your data before proceeding with the installation of the extension. This is because although we will make our best to support you, we cannot take responsibility of any data loss that might occur during the installation/adjustments of this extension.

[ 1 ] Go to the Magento installation directory on your server, and make sure to create the following path –
app/code/Mexbs/Fbshop

[ 2 ] Unpack the mexbs_fbshop-1.0.0.zip file. Copy all the extracted files from the mexbs_fbshop-1.0.0 directory to the app/code/Mexbs/Fbshop path that you’ve just created.

[ 3 ] Connect via SSH and go to your Magento root directory

[ 4 ] Determine the mode that your Magento is running on, by running the following command:

```
php bin/magento deploy:mode:show
```

[ 5 ] If you are in **developer** or **default** mode, run the following commands:

```
php bin/magento module:enable Mexbs_Fbshop --clear-static-content
php bin/magento setup:upgrade
```

If your Magento version is 2.0.x or 2.1.x run:

```
php bin/magento setup:static-content:deploy
```

If your Magentno version is 2.2.x run:

```
php bin/magento setup:static-content:deploy -f
```
[6] If you are in **production** mode, run the following commands:

```bash
php bin/magento module:enable Mexbs_Fbshop --clear-static-content
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
```


**Step 2 – Configure the extension**

1) Go to the admin panel, “Stores → Configurations → General → Facebook Shop Integration”.

2) Select “Yes” in “**Feed Generation Schedule Enabled**”, and fill in the frequency of the schedule in “**Feed Generation Schedule**”. This sets up the cron that regenerates the feed file that Facebook fetches to update your shop. The default value of “**Feed Generation Schedule**” is “0 0 * * *” – which means - every night 12am.

3) Fill in a unique identifier for your feed in “**Feed Unique Identifier**”. It doesn't have to be the ID that Facebook gives to your feed. It can be for example a date of today in combination with the shop name. The extension needs a unique identifier for the feed, to be able to generate unique IDs for your product.

The reason for this is that if two feeds will have a product with the same ID, only the first feed that created the products will be able to edit them. Therefore, the product IDs in the feed must be unique –
Step 3 – Create mappings for the Facebook fields

The extension allows you to set mappings between the products attributes and the Facebook fields. (See the full list of all Facebook fields here - https://developers.facebook.com/docs/marketing-api/dynamic-product-ads/product-catalog#required-fields). Upon the installation, the extension creates default mappings between the product attributes and the Facebook fields that have the same name. For example: color, description and more. It also creates the mapping between product attribute “name” and the Facebook field “title”.

Some of the Facebook fields are filled in dynamically by the extension. That is, you don’t need to create mappings for those fields. Those fields are - id, availability, image_link, link, price, sale_price, sale_price_effective_date

Some of the Facebook fields are required. Therefore, if the extension doesn’t have a mapping for the required fields, it displays an error message in the feed generation panel.

1) Go to Marketing → Feed Actions. In our example, the missing mappings are “condition” and one of - “gtin”, “mpn”, “brand” (you can see that the error message says that). That is, we have to create a mapping for the Facebook field “condition”, and we have to create at least one mapping for either “gtin”, “mpn” or “brand” –

![Image of Feed Actions with error message]

The error message says that the mapping for “condition” is missing. It also says, that one of the mappings is missing - for either “gtin”, “mpn” or “brand”.

2) To add the missing mappings, go to Marketing → Facebook Shop Integration → Product Attributes Mapping. Click on “Add New Mapping” -
3) In the Facebook API Field drop down, select “condition”, and in the Product Attribute Code drop down, select the attribute that will hold the value for the condition of the product (in our example it’s “facebook_condition”) -

If you don’t have such attribute, you can create it. (If you don’t know how to create a product attribute and to mass update it’s value over the products, read “Appendix 1 – creating a product attribute and mass updating it’s value over the products”) –

4) Click “Save".
Now, since one of the fields “gtin”, “mpn” or “brand” are required by Facebook, we will create a mapping for “brand” (we could create a mapping for “gtin” or “mpn” instead). For this, follow the points 2-4 for “brand”.

Now, when all the required fields are mapped, we are ready to proceed to the next step.

**Step 4 – Mark the products that we want to be in the Facebook shop**

In this step we will mark the products that we want to show up in our Facebook shop.

1) Go to Products → Catalog. Check the products that you want to be in the Facebook feed, click Actions → Change Product Facebook Feed Status → In Feed –

You can see which products are in the Facebook feed, by looking at the “Is Product in Facebook Feed” column in the products grid, or by filtering by it.
**Step 5 – Initial feed file generation**

The Facebook shop tab that you will configure in the next step will use a feed file to sync itself with your Magento shop. This feed file will be generated automatically using the cron schedule that you’ve set up in step 2.

However, when you will set up your Facebook shop in the next step, Facebook will require a feed file. Since we don’t want to wait for the Magento cron to create the feed file for the first time, we will do it the first time manually –

1) Go to Marketing → Facebook Shop Integration → Feed Actions. Select the store view that you want to generate the feed file for, and click “Schedule Feed Generation Now” –

The message “The feed generation is scheduled. It should start running soon. You can leave this page and come to check later ...” should show up in the “Feed Generation Log” window. It means that the feed generation will start on the next cron run, that is, if the cron in your server is set up to run every minute, the generation should start within a minute (it can take more time if there is currently another job running – that is, the feed generation will start after that the other current jobs are finished running).

Once the feed generation will start, you will see it’s progress in the “Feed Generation Log” window –
Once the feed generation is done, you can proceed to the next step.

**Step 6 – Add the Shop tab to your Facebook page**

1) Go to your Facebook page that you would like to add the tab in (you need to be the administrator of the page).
2) Click on “Settings” at the top right. Click on “Templates and Tabs”, and scroll to the bottom –
3) Click on “Add a Tab” -
4) Click on the “Add Tab” button next to the “Shop” line –

The “Shop” line should disappear.

5) Click on “Close” to close the popup.
6) Now, click on “Page” on the top left, to go back to the page view –
7) Now you should see the “Shop” tab in the tabs on the left –

The “Shop” tab is now showing up.
8) Click on the “Shop” tab. You should get a popup. Check the terms checkbox, and click “Continue” –
9) Now you should see a popup, asking you to select the checkout method. Select “Check Out on Another Website” and click “Continue” –
Another popup will show up, asking about the shop currency. Choose the currency of your Magento store, and click “Save” –

Now you’ve successfully set up the “Shop” tab on your page, and you are ready to proceed to the next step – setting up the catalog feed.

Step 7 – Set up the catalog feed on Facebook

1) Go to the Facebook bussiness manager at https://business.facebook.com/. If you don’t have a bussiness manager account, simply follow the sign up process.

2) Click on “Bussiness Manager” link on the top left –
3) Click on “Catalogs” –

You should see a catalog with the name of your page. (In our example it is “Demo Shirts 2 French”). Click on it –
4) Click on “Product Data Sources” -
5) Click on “Add Data Source” –

6) Go to your Magento admin panel, Marketing → Facebook Shop Integration → Feed Actions. Switch to the desired store view, and copy the URL of the link “Download the feed file” by right clicking it and selecting “Copy link address” –

7) Go back to the Facebook business manager. In “How do you want to upload your data feed?”, select “Set a Schedule”. In “Add a Data Feed URL” paste the URL that you’ve just copied. In “Schedule Your Automatic Uploads”, set the schedule by which Facebook will fetch your feed file.
We recommend setting up your schedule with a delay from the cron schedule that you’ve set in Step 2. For example, if your cron schedule is every night at 12am, we recommend setting the Facebook schedule an hour after that, that is every night at 1am. However, if your cron schedule is more frequent than daily (ie: once in two hours, once an hour etc), we recommend setting the Facebook frequency to “Hourly”.

In “Enter Data Feed Name”, you can enter any name of your choice. We suggest adding a “Magento” to the name, to indicate that this feed is generated by Magento.

In “Select Default Currency”, select the currency of your Magento store.

Click “Start Upload” –
Add Products With Data Feeds

Upload a file or add a URL

Need to set up a data feed?

Download CSV Template  Follow Step-by-Step Guide

You can also set up a data feed by using the Facebook Feed plugin on Shopify, WooCommerce, BigCommerce or Magento

How do you want to upload your data feed?

Upload Once
Recommended if your inventory rarely changes. You can upload a data feed file to update your inventory.

Set a Schedule
Recommended if your inventory regularly changes. You can set a schedule to automatically update your inventory from a feed URL

1) Select “Set a Schedule”

Add a Data Feed URL
Enter your data feed URL with one of our supported URL types: http, https, or ftp (for example: "https://www.mynetwork.com/feeds/feed12 wastes.csv"). Note that this is different than the URL for your website, Facebook News Feed and Facebook Page Shop.

2) Paste the feed link that you’ve just copied from Magento

Add your data feed URL login details (optional)
If your data feed URL is password protected, please enter your
Username
Password

Schedule Your Automatic Uploads
Select when you want to automatically upload products from this data feed URL.

3) Set the schedule by which Facebook will fetch the feed file

Hourly  Daily  Weekly

At 1:00 AM

Enter Data Feed Name
Choose a name that will help you find this data feed in the future.

4) Set the feed name. We suggest adding “Magento” to it

Magento French Store

Select Default Currency
Select the default currency in which you want to display prices in your data feed (e.g., USD for Americans). You can update the default currency at any time in the feed settings.

5) Select the Magento store currency

USD - US Dollars

Start Upload

6) Click
You should see a loading screen –

8) Once done, click “Done” –

9) Now go to your Facebook page → Shop tab. You should see some products starting to show up, most likely missing the images (the images will show up after that Facebook will finish the processing of your products) –
Note: you should be logged in as an admin of the page in order to see the processed products (the non admin users will only see the products when their processing is done).

10) Click on one of the products. You will most likely see the message “Your product is being processed”, as said it means that Facebook is still processing your product. On a first upload it usually takes between 10 minutes and an hour –
Hooray! You've just finished setting up the integration between your Magento store and your Facebook shop tab! This is how it will look when the processing is done –
**FAQ 1: Setting some products to be in the feed**

Go to Products → Catalog. Check the products that you want to be in the Facebook feed, click Actions → Change Product Facebook Feed Status → In Feed –

This will cause the products to be in feed on the next feed generation. That is, those products will show up in your Facebook shop on the next Facebook fetch.

**FAQ 2: Setting some products to NOT to be in the feed**

Go to Products → Catalog. Check the products that you don't want to be in the Facebook feed, click Actions → Change Product Facebook Feed Status → Not in Feed –
This will cause the products to not to be in the feed on the next feed generation. That is, those products will disappear from your Facebook shop on the next Facebook fetch.

**FAQ 3: Generating the feed file manually**

Sometimes you’d want to generate the feed file manually instead of waiting for the cron to do it –

1) Go to Marketing → Facebook Shop Integration → Feed Actions. Select the store view that you want to generate the feed file for, and click “Schedule Feed Generation Now” -
The message “The feed generation is scheduled. It should start running soon. You can leave this page and come to check later ...” should show up in the “Feed Generation Log” window.

It means that the feed generation will start on the next cron run. That is, if the cron in your server is set up to run every minute, the generation should start within a minute (it can take more time if there is currently another job running – that is, the feed generation will start after that the current jobs will finish running).

Once the feed generation will start, you will see its progress in the “Feed Generation Log” window –
FAQ 4: Checking the feed generation logs

If there is a current feed generation running, or it finished running and there is no new feed generation scheduled yet, you can see its log by going to Marketing → Facebook Shop Integration → Feed Actions, and looking into the Feed Generation Log –

Otherwise, if you want to look at the previous generation logs, go to Marketing → Facebook Shop Integration → Feed Generation Logs –
The Product IDs column shows only part of the IDs of the products that were added to the feed. To see all the product IDs, click on the row.

**FAQ 5: In the Feed Generation Log I get the message “Important: the rows include 147 configurable products, they WON'T be added to the feed, but only their children.”. Why?**

It is a totally normal behavior. The reason for this message is that Facebook doesn't need the information of the parent product, but only of its children. That is, if for example you have a configurable product named “t-shirt A”, and two children “t-shirt A red” and “t-shirt A green”, Facebook only needs the “t-shirt A red” and “t-shirt A green” to be in the feed, and it will display the product “t-shirt A” automatically based on the information of its children.

**FAQ 6: What attributes does Facebook support for the variations of configurable options?**

Facebook only supports color, size, material and pattern for product variations. That is if you have a configurable product children that has those attributes, it will be grouped by Facebook to one product with options dropdowns, like the following –
However, if you have a configurable product that has other attributes, Facebook won’t group its children into one product. Instead, it will display all of its children products.
FAQ 7: When user clicks on “Check out on website” – where does the link go?

The extension behaves similarly to the Shopify’s native Facebook shop integration –

For simple, configurable, downloadable and virtual products – it automatically adds the product to the cart (with the selected options) and brings the user to the checkout page –
For bundled and grouped products – it brings the user to the product page.

**Appendix 1 – creating a product attribute and mass updating its value over the products**

In this example we will create an attribute “facebook_condition”.

1) To create a product attribute, go to Stores → Attributes → Product. Click “Add New Attribute” -

2) In the “Properties” tab fill in the following values –
New Product Attribute

Attribute Properties

- Default Label: Facebook Condition
- Catalog Input Type for Store Owner: Dropdown
- Values Required: No

Manage Options (Values of Your Attribute)

<table>
<thead>
<tr>
<th>Is Default</th>
<th>Admin*</th>
<th>Default Store View</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td></td>
<td></td>
</tr>
<tr>
<td>used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>refurbished</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Advanced Attribute Properties

- Attribute Code: facebook_condition
- Scope: Store View
- Unique Value: No
- Input Validation for Store Owner: None
- Add to Column Options: No
- Use in Filter Options: No
3) In the “Manage Labels” tab you can leave the fields untouched –

4) In the “Storefront Properties” tab you can leave the fields untouched –
5) Click “Save Attribute”.

Now we will mass update the attribute value over the products. In order to do this, we first need to assign the attribute to the attribute sets of the products that we want to update –

6) Go to Stores → Attribute Sets –

7) Click on one of the Attribute Sets that the products you want to update are in. In our case, some of the products that we want to update are bags, therefore, we click on “Bags” –
8) Drag and drop the attribute facebook_condition from the right column to anywhere on the left column.
9) Click “Save”.
10) Repeat the steps 6-9 for any other attributes sets that the products that you want to update are in.

Now we are going to mass update the attribute “facebook_condition” over the products in our store.

11) Go to Catalog → Products
12) Check the products that you want to update, click on Actions → Update Attributes –
In the "Attributes" tab, scroll down to the "Facebook Condition" attribute, click on "Change", select the desired value (in this example we chose "new"), and click "Save" –

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Check the products that you want to update</td>
<td>2) Click &quot;Update Attributes&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Repeat the points 11-13 for any other products for which you want to update the attribute “Facebook Condition”.

14) Repeat the points 11-13 for any other products for which you want to update the attribute “Facebook Condition”.
Get Support!

We are here and we will do our best to help. Contact us on support@mexbs.com and we will get back to you ASAP with help!

Free software updates

From time to time we are going to release some cool updates for your extension. And you will get those updates for free! Currently we are working on upgrading our website, in order to make the software updates appear directly in your customer account. But for now, you will get the software updates by email on demand (because we don’t want to spam you), or every time there is new release (if it’s ok for you to be spammed by good software ;)).

Don’t forget to send your feedback!

Every feedback is good for us! Please send us your feedback to sales@mexbs.com or write it directly in the reviews section on the product page. It will be a pleasure for us to hear your opinion!

Enjoy your extension!
MexBS Team