Documentation / Documentation Home

Google Integration

Created by Unknown User (malynow), last modified by Anton Lashuk on янв 05, 2019

Extension page: http://ecommerce.aheadworks.com/magento-extensions/google-integration.html
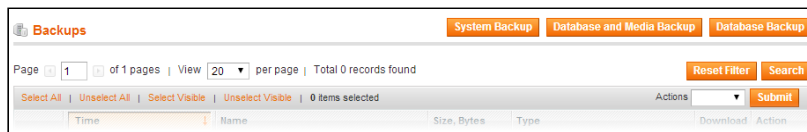
Installation

Installing Google Integration extension

1. Backup your web directory and store database.
   Click to view details
   You can make backup copies with any tool you find appropriate

   If you are going to use the native Magento backup function, navigate to **System -> Tools -> Backups** and perform *System* and *Database* backups

 | Backups | System Backup | Database and Media Backup | Database Backup |
   |---|---|---|---|
   | Page 1 of 1 pages | View 20 per page | Total 0 records found | Reset Filter Search |
   | Select All | Unselect All | Select Visible | Unselect Visible | 0 items selected | Actions | Submit |
   | Time | Name | Size, Bytes | Type | Download Action |

2. Log in to the Magento backend

3. Disable compilation
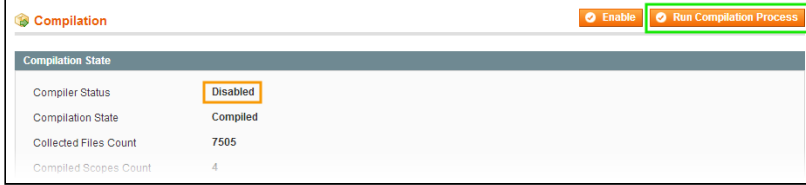   Click to view details
   Navigate to **System -> Tools -> Compilation**.

   If Compiler status is Disabled, you can skip to the next step

   If Compiler is enabled, disable it.
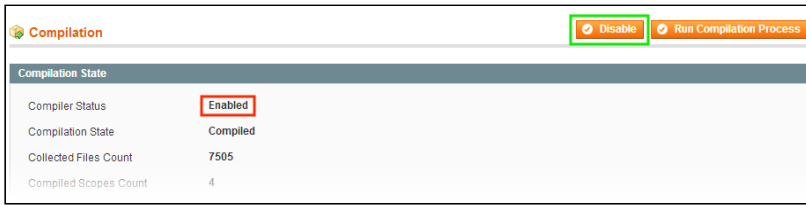
IMPORTANT: after the extension is installed, you can enable the compilation again; <u>IT IS CRUCIAL</u> that you use "**Run Compilation Process**" function, not just "Enable button"

Installing an extension with the Compilation enabled will result in store downtime.
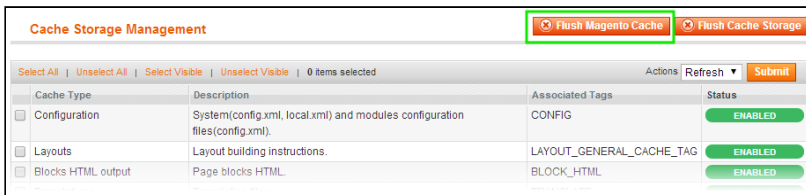
---

4. Flush store cache
Click to view details
You can flush the store cache in 2 ways:

- **via the backend**:

Navigate to **System -> Cache Management** menu, and click **Flush Magento Cache** button

- **via filesystem**:

On your server, navigate to Magento root folder, then proceed to **/var/cache/;** delete all the content there.

---

5. Download the extension package from your account and extract the downloaded archive

---

6. Copy the content of **/Step_1/** folder to your store's root directory

If you are using a custom theme,
read the instructions here.
In case you are using <u>a custom theme</u>, it is recommended to copy the design files to your current theme's folders. In case there are <u>several themes</u> in use at the same store, the design files must be copied to each of them.

- Find this folder in the package: */step_1/app/design/frontend/base/default/* ; copy its content to */app/design/frontend/[your_package]/[your_theme]/*
- Find this folder in the package: */step_1/skin/frontend/base/default/* ; copy its content to */skin/frontend/[your_package]/[your_theme]/*

7. Copy the content of **/Step_2/** folder to your store's root directory

8. Flush store cache again; log out from the backend and log in again.

Registering project on Google

To set up the extension navigate to the **Google Developers Console** page. Here it will be required to create a new Google Project for later use with the Google Integration extension.

1. Add new Project;
   Click to view details
   Navigate to the **Google Developers Console** page and click
   Create Project button.
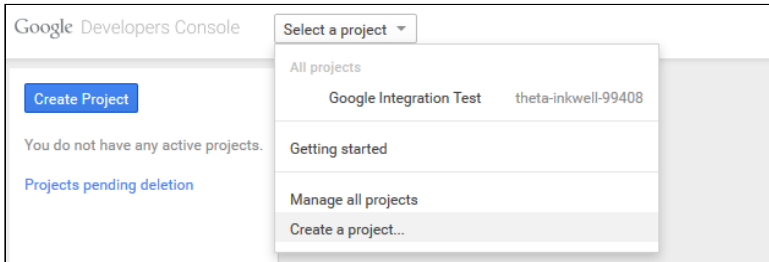
   Name the new project and click Create button.

2. Enable Google+ API;
   Click to view details
   1) Navigate to APIs & auth > API section;

   2) Click Google+ API;

   3) Click Enable API button.
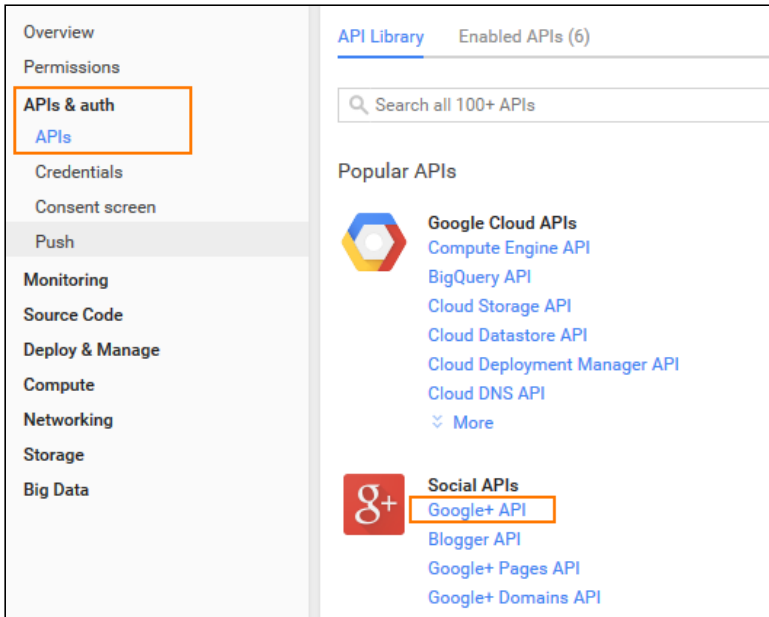
3. Configure Consent screen;
   Click to view details

Navigate to **APIs & auth > Consent screen** and specify the required information including: Homepage URL, Google+ page ID.

Consent screen

The consent screen will be shown to users whenever you request access to their private data using your client ID

Note: This screen will be shown for all of your applications registered in this project

Email address ⓘ

[▼]

Product name

[   ]

Homepage URL (Optional)

[   ]

Product logo (Optional) ⓘ

[   ]

🚫   This is how your logo will look to end users
Max size: 120x120 px

Privacy policy URL (Optional)

[   ]

Terms of service URL (Optional)

[   ]

Google+ page ID (Optional) ⓘ

[plus.google.com/ page ID]

[Save]   [Reset]

4. Create new client ID;
   Click to view details
   Navigate to **APIs & auth > Credentials** and create new client ID

Create Client ID

Application type

◉ Web application
Accessed by web browsers over a network.

○ Service account
Calls Google APIs on behalf of your application instead of an end-user. Learn more

○ Installed application
Runs on a desktop computer or handheld device (like Android or iPhone).

Authorized JavaScript origins
Cannot contain a wildcard (http://*.example.com) or a path (http://example.com/subdir).

[https://www.example.com]

Authorized redirect URIs
One URI per line. Needs to have a protocol, no URL fragments, and no relative paths. Can't be a public IP Address.

[https://www.example.com/googleconnect/login/callback/
https://www.example.com/index.php/googleconnect/login/callback/]

[Create Client ID]   [Cancel]

1) Select **Web application** Application type;

2) Specify your store URL in **Authorize JavaScript origins** field;

3) Paste:

- yourstore_URL/googleconnect/login/callback/
- yourstore_URL/index.php/googleconnect/login/callback/

> **Authorized redirect URIs MUST** end with **"/"** (slash symbol).

In **Authorize redirect URIs** field.

4. Get Client ID and Client secret.
   Click to view details
   Once Client ID created **Client ID for web application** block will appear.

Client ID for web application	
Client ID	919891364445-ts9e2m9l05...
Email address	919891364445-ts9e2m9l05...
Client secret	kqbOmxG6mbsDAmuyVy4nn...
Redirect URIs	https://www.example.com/googleconnect/login/callback/
	https://www.example.com/index.php/googleconnect/login/callb...
JavaScript origins	https://www.example.com

Edit settings   Reset secret   Download JSON   Delete

Copy Client ID and Client secret to the appropriate fields of the Google Integration settings at your store backend.

Configuration

All the extension configuration options can be found under **System > Configuration > AheadWorks extensions**.

As the extension introduces several aspects of the Google integration functionality all settings are splint into three (3) sections:

- Google Connect;
- Google "+1" Button;
- Google Share Button.

Google Connect

Google Connect settings section allows managing basic extension configuration.

General Options

In the **Google Connect** settings section one is suggested to specify the Google Client ID and Client Secret.

- **Enable Extension** - allows enabling/disabling extension output;
- **Client ID** - Google Client ID;
- **Client secret** - Google Client secret;
- **Title** - defines the title of the Google **Sign In** button.
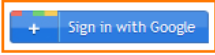
> How does Google sign in button look like?

General Options

To get "API key" register your application on Google APIs Console

Enable Extension	Yes	[STORE VIEW]
Client ID		[STORE VIEW]
Client Secret		[STORE VIEW]
Title	Sign in with Google	[STORE VIEW]
	▲ The title of the button.	

Content Options

- **Display on Login Page** - defines if the Google **Sign In** button should be displayed at the Login page;
- **Display on Checkout Page** - defines if the Google **Sign In** button should be displayed at the Checkout page;
- **Display Custom Buttons** - allows disabling/enabling all custom buttons placed via **CMS > Widget**;
- **Display Widget Buttons** - allows disabling/enabling **ALL** Google Integration related buttons.

Google "+1" Button

Google "+1" Button settings allow managing **"+1"** button related settings.

General Options

- **Button Code Type** - defines the code that should be used for the **+1** button functionality (leave the default value if you are not sure if you want to change it);
- **URL of a page** - allows specifying particular **+1** URL of the store;
- **Display Count Box** - defines if the number of people who "liked" the page should be displayed along with **+1** button;

- **Size** - defines the size of **+1** button:
    - Small;
    - Medium;
    - Standard;
    - Tall.

Content Options

- **Display on CMS pages** - defines if the Google **+1** button should

be displayed at the CMS pages;

- **Display on Home Page** - defines if the Google **+1** button should be displayed at the Home page;
- **Display on categories page** - defines if the Google **+1** button should be displayed at the category pages;
- **Display on products page** - defines if the Google **+1** button should be displayed at the product pages;
- **Display Custom Buttons** - allows disabling/enabling all custom buttons placed via **CMS > Widget**;
- **Display Widget Buttons** - allows disabling/enabling **ALL** Google Integration related buttons.
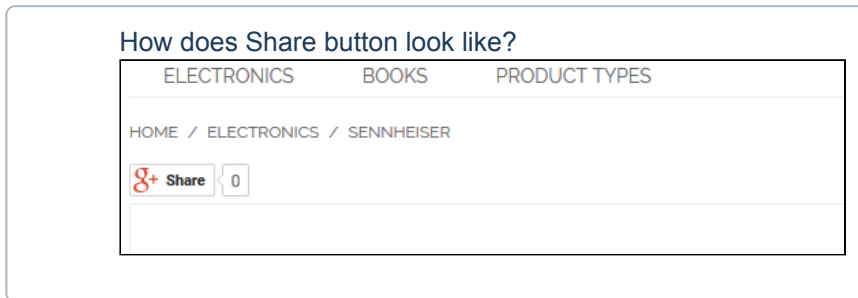
Google Share Button

Google Share Button settings allows managing **Share** button related settings.

General Options

- **Annotation** - defines the look of the annotation to display next to the **Share** button:
  - Bubble(horizontal);
  - Inline;
  - Bubble(vertical);
  - None;
- **Size** - defines the size the **Share** button:
  - Small;
  - Medium;
  - Large;

How does Share button look like?

- **Width** - **Share** button block width in pixels or %;
- **HTML5 valid syntax** - defines if the extension should use HTML5 valid syntax.

Content Options

- **Display on CMS pages** - defines if the Google **+1** button should be displayed at the CMS pages;
- **Display on Home Page** - defines if the Google **+1** button should be displayed at the Home page;
- **Display on categories page** - defines if the Google **+1** button should be displayed at the category pages;
- **Display on products page** - defines if the Google **+1** button should be displayed at the product pages;
- **Display Custom Buttons** - allows disabling/enabling all custom buttons placed via **CMS > Widget**;
- **Display Widget Buttons** - allows disabling/enabling **ALL** Google Integration related buttons.

Custom positions

Google Integration extension allows placing it's buttons and blocks using widget **Layout Updates** or **CMS pages** widgets.

The extension comes with the following widgets available:

- Google "+1" Button;
- Google Share Button;
- Google Connect Button.

Each widget represents related functional aspect of the extension.

Layout Update

Custom Google button widgets can be introduced to different store pages from **CMS > Widget** section.

1. Click **Add New Widget Instance**;
2. Specify required Google widget in the **Type** dropdown list;
3. Specify required **Theme** you want the widget to be displayed at;
4. Click **Continue** button;
5. Set required widget parameters in the Frontend Properties section;
6. In the Layout Updates section by clicking **Add Layout Update** button specify the store pages you want the widget to be displayed at.
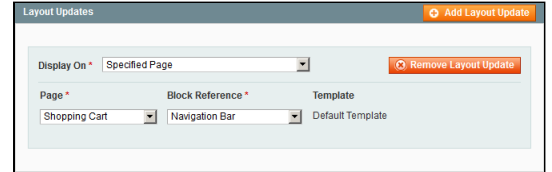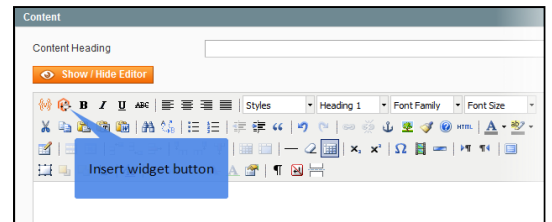
Once the widget is saved required Google widget will be displayed at the set store pages.

CMS Pages

Google Integration widgets can be inserted in the CMS pages from **CMS > Pages > Manage Content** section.

1. Select existing or create new CMS page;
2. In the Content block click **Insert Widget** button of the WYSIWYG editor;
3. Select required Google Integration widget.

The widget will be automatically inserted in the CMS page.

Uninstallation

1. Disable compilation, in case it is enabled.
2. Login to your FTP, navigate to **app/etc/modules/**
3. Open the files: AW_Google_Connect.xml, AW_Google_Plusone.xml, AW_Google_Share.xml and change the following line:

```
<active>true</active>
```

to

```
<active>false</active>
```

Now your Magento is unaware of the existence of the extension

4. Clear the cache under var/cache
5. Make sure that the site is working properly, otherwise roll back the changes and apply to our technical support.
6. If everything works fine, it is safe to delete the files of the extension.
7. In case you need to clean the database, **make a backup** and then run the following query in MySQL:

```
DROP TABLE 'googleconnect_synch';
```

Note, if you are using the table with prefixes, you must specify them in all table names.

If you are not sure how to do that or expect any troubles with it, please contact your server administrator regarding the matter.

If you remove the tables as it is described above, you will need to configure the extension again after it is reinstalled.

Troubleshooting

After the extension installation the store gives an error, or blank page, or suggests to start Magento installation procedure.

Change the owner of the extracted extension files to the web server user and set 775 permissions on them. Clear the store cache and try again.

There is no *aheadWorks extensions* under my configuration section, or having the extension tab clicked I get a blank page, or *Access Denied* error.

Clear the store cache, browser cookies, logout and login again.

I've set up everything correctly, inserted the HTML code but there is nothing on that page.

Clear the store cache, clear your browser cache and domain cookies and refresh the page.

My configuration changes do not appear on the store.

Clear the store cache, clear your browser cache and domain cookies and refresh the page.

You can always find the latest version of the software, full documentation, demos, screenshots and reviews
at http://ecommerce.aheadworks.com
License agreement: http://ecommerce.aheadworks.com/LICENSE-M1.txt
Contact us: http://ecommerce.aheadworks.com/contacts/
Copyright © 2019 aheadWorks Co. http://www.aheadworks.com

Нравится   Станьте первыми кому понравится это   Ни одной