

Advanced Product Feeds Manual

Welcome to the Advanced Product Feeds User Manual

Whether you are new to Advanced Product Feeds or an advanced user, you can find some useful information here.

In this guide, you'll learn how to:

- [Install extension](#)
- [Create and import feed templates](#)
- [Make and generate your first feed](#)
- [Adjust your feeds](#)

How to install the extension

1. Back up your store's database and web directory.
2. Log in to the SSH console of your server and navigate to the root directory of the Magento 2 store.
3. Copy installation instructions from the page [My Downloadable Products](#) to the SSH console and press ENTER.
4. Run the command `php -f bin/magento module:enable Mirasvit_Core Mirasvit_Feed Mirasvit_Report` to enable the extension.
5. Run the command `php -f bin/magento setup:upgrade` to install the extension.
6. Run the command `php -f bin/magento cache:clean` to clean the cache.
7. Deploy static view files

```
rm -rf pub/static/frontend/*; rm -rf pub/static/backend/*; rm -rf var/view_preprocessed/*; php -f bin/magento setup:static-content:deploy
```

Note

If you install the module manually to the "app/code/" directory (Magento 2.3 only), you must additionally install the required `zfl/zend-reflection` library through composer:

```
composer require "zfl/zend-reflection"
```

Manage templates

To manage templates, go to **Catalog ? Advanced Product Feeds ? Templates**.

The extension includes more than 45 ready-to-use templates for all the most popular price comparison engines (Google Shopping, Amazon, eBay, Shopzilla, etc.).

Note

Before you can create a new template, you need to check the product feed specification for the specified marketplace (template format type, fields delimiter, required fields, etc.).

You can create templates for any comparison shopping engine.

To create a new template, follow these steps:

1. Go to **Catalog ? Advanced Product Feeds ? Templates**, then click the **Add Template** button in the upper right-hand corner. You'll see following fields:



☰ New Template ▾

TEMPLATE INFORMATION

Template Information

Content Settings

General Information

Name

File Type



Copyright © 2022 Magento Commerce Inc. All rights reserved.

1. Fill in the following fields:

- **Name** - the name of the new template.
- **File Type** - the feed output format. There are 3 types that are available for the data feed:
 - **CSV** - comma-separated values with each item placed on a new line. File extension is *.csv*.
 - **TXT** - same as a CSV file, but with a *.txt* extension.
 - **XML** - uses tags to define blocks of content. Information about your items is enclosed within these tags, which are indicated by angle brackets. The file extension is *.xml*.
- At the **Content Settings** tab, you need to configure the template depending on particular requirements and the file type. Currently supported formats are:
 - **CSV, TXT**
 - **XML**

2. Click the **Save** button.

List of the pre-installed templates

Template name	Format
AdForm	CSV
AllYouNeed	CSV
AmazonAds	TXT
AmazonSellerCentral (example)	TXT
Awin	TXT
Become Europe	CSV
Belboon	CSV
Beslist	XML
Billiger.de	CSV
Bing Shopping	TXT
BizRate	TXT
Choozen	CSV
Ciao	CSV
Commission Junction	CSV
CJ	XML
DooYoo	CSV
Domodi	XML
eBay(Commerce Network)	CSV
eBay.com (Store)	CSV
Facebook Dynamic Ads	XML
Facebook (storefront)	CSV
Fishpond	CSV
FashionChick	CSV
GetPrice Categories	XML
GetPrice Products	XML
Google Shopping	XML
Google Shopping (configurable products)	XML
Google Shopping Review	XML
Google Shopping Update	XML

Template name	Format
idealo.it	CSV
it.bestshopping.com	CSV
Kelkoo	XML
Kieskeurig	XML
LeGuide.com	TXT
Marktplaats	XML
Newegg	XML
Newegg(inventory)	XML
Nextag	TXT
pagineprezzi	TXT
Partner-Ads	XML
PriceGrabber	TXT
PriceMe	XML
PriceRunner	XML
PriceSpy	TXT
Rakuten (Apparel)	TXT
Sears.com Inventory	XML
Sears.com Item	XML
Sears.com Price	XML
ShareASale	CSV
ShopMania	XML
Shopping.com	XML
ShopPrice	XML
Shopzilla	TXT
SingleFeed	CSV
The Find	CSV
TradeDoubler	CSV
TradeTracker	CSV
Twenga	CSV
Webgains	CSV
Yandex Market	XML

How to create a new data feed

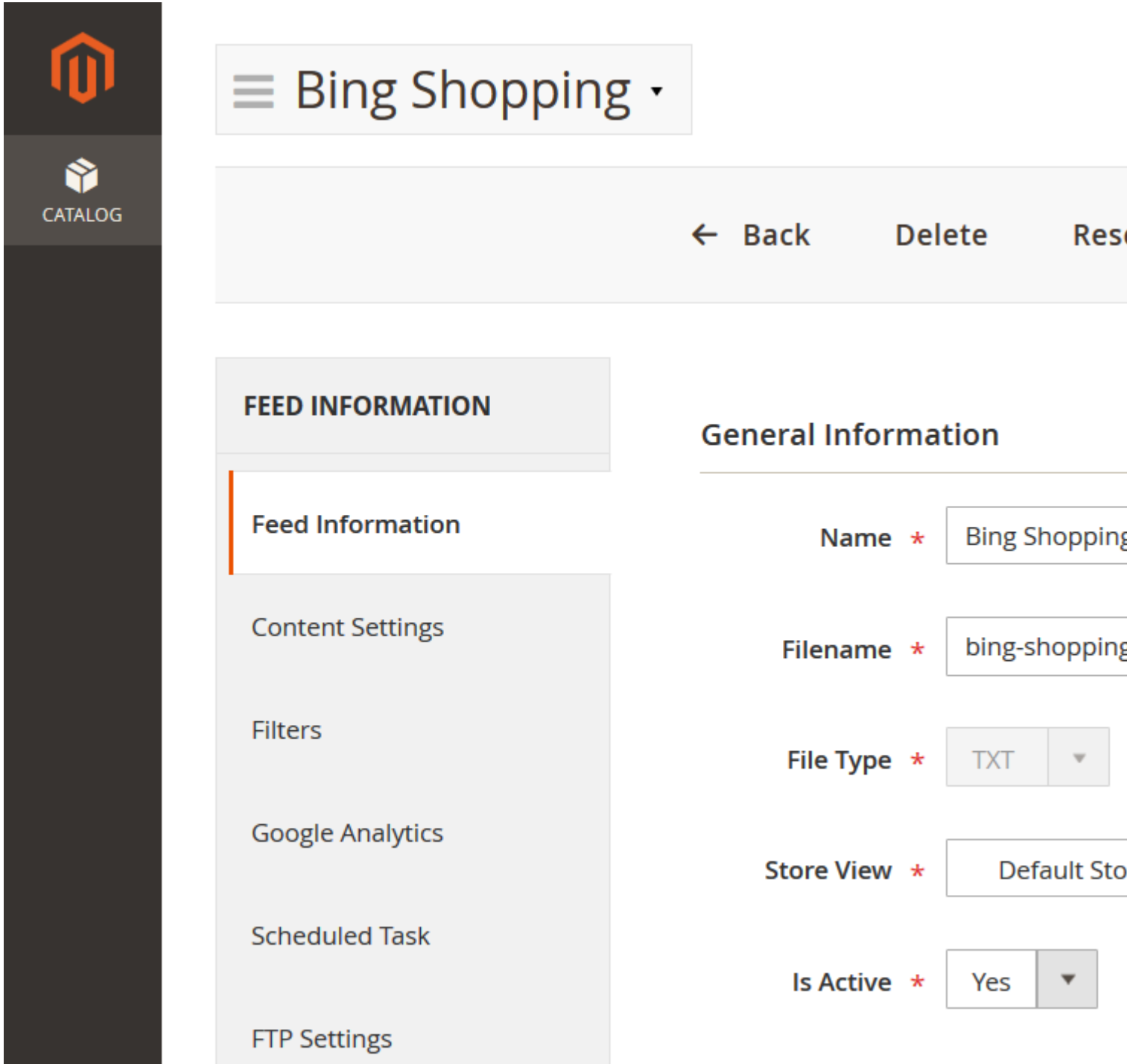
Note

Pre-installed templates may require additional attributes and settings due to specific information related to the items being sold or country location. Make a point of checking a given product's attribute requirements using marketplace specifications.

To create a new data feed, take the following steps:

1. Go to **Catalog ? Advanced Product Feeds ? Feeds**. Press the button **Add Feed**.

2. Select one of the existing templates to create a feed. To create an empty feed, select **Empty Template**.
3. Press button **Continue**.
4. Fill in the following requirement fields:
 - **Name** - name of the data feed.
 - **Filename** - name of the data feed file. The file will be located at [magento_path]/pub/media/feed/filename.
 - **Store View** - the store view for which a data feed will be generated.
 - **Is Active**



The screenshot displays the Magento Admin interface for configuring a Bing Shopping feed. On the left, a dark sidebar contains the Magento logo and a 'CATALOG' menu item. The main content area features a header with a hamburger menu icon and the text 'Bing Shopping'. Below the header, there are three buttons: 'Back', 'Delete', and 'Res...'. A 'FEED INFORMATION' section is highlighted, with a sidebar menu listing 'Feed Information', 'Content Settings', 'Filters', 'Google Analytics', 'Scheduled Task', and 'FTP Settings'. The 'General Information' form includes the following fields:

- Name ***: Bing Shopping
- Filename ***: bing-shopping
- File Type ***: TXT (dropdown menu)
- Store View ***: Default Sto...
- Is Active ***: Yes (dropdown menu)

Additionally, if you selected **Empty Template**, you will need to fill in these

- * **File Type** - there are three file types available for the data feed.
 - * **CSV** (/feed/csv) - comma-separated values where each item is placed on a new line.
 - * **TXT** (/feed/csv) - same as for a CSV file, but with a *.txt* extension.
 - * **XML** (/feed/xml) - uses tags to define blocks of content. Information on the XML format is available in the [XML format](#) section of the [Data Feeds](#) guide.

1. Press the button **Save and Continue Edit**.
2. To generate a data feed, press the button **Generate** at the top right corner of the page.

How to configure an XML Feed

If you select the **XML** file type at the tab **Content Settings**, you can create/edit XML schema for your feed.

By default, we provide templates for XML feeds, so you can easily copy and change it according to your requirements.

Usually, Comparison Shopping Engines provide a template of the XML file. Based on this template, you can create your own XML schema.

Typical xml schema:

```
<?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0" xmlns:g="http://base.google.com/ns/1.0">
  {% for product in context.products %}
  <item>
    <attribute_1><![CDATA[{{ product.attribute_1 }}]]></attribute_1>
    <attribute_2><![CDATA[{{ product.attribute_2 }}]]></attribute_2>
    .....
  </item>
  {% endfor %}
</rss>
```

Product cycle block:

```
{% for product in context.products %}
  .....
{% endfor %}
```

Inside this block, you can use any product attribute.

Category cycle block:

```
{% for category in context.categories %}
  .....
{% endfor %}
```

Inside this block, you can use any category attribute.

Review cycle block:

```
{% for review in context.reviews %}
  .....
{% endfor %}
```

Inside this block, you can use any review attribute.

Attribute (pattern) block:

```
<attribute_1><![CDATA[{{ product.attribute_1 }} ]]></attribute_1>
<attribute_1><![CDATA[{{ category.name }} ]]></attribute_1>
<attribute_1><![CDATA[{{ review.nickname }} ]]></attribute_1>
```

The attribute code must be enclosed in double curly brackets: `{{ product.attribute_code }}`. You can use all attribute codes available at **Store > Attributes > Product** and all static attributes (ex. `entity_id`, `created_at`, etc.).

Additionally, you can place any available pattern in curly brackets. [Full pattern list of patterns]

Note

Characters like `<` and `&` are illegal in XML elements.

- `<` will generate an error because the parser interprets it as the start of a new element.
- `&` will generate an error because the parser interprets it as the start of a character entity.

We suggest that you enclose all patterns in a **CDATA** block `<attribute><![CDATA[{{ pattern }}]]></attribute>`. In this case, the xml data feed will be valid.

How to configure CSV, TXT Feed

If you select a **CSV** or **TXT** file type at the tab **Content Settings**, you can create/edit attribute schema for your feed.

Note

When you use pre-installed templates, you need to verify that the attributes from the template response are set for the same values as your store attributes.

If this attribute does not exist in your store, set the appropriate product attributes or patterns for the same line.

Content Settings

Before creating the attribute scheme, you need to fill in the required file settings:



☰ New Feed ▾



DASHBOARD



SALES



CATALOG



CUSTOMERS



MARKETING



CONTENT



REPORTS



STORES



SYSTEM



FIND PARTNERS
& EXTENSIONS

FEED INFORMATION

Feed Information

Content Settings

Filters

Google Analytics

Scheduled Task

FTP Settings

Additional

History

Content Settings

Fields Delimited

Fields enclosure

Include Column
Header

Extra Header

Column

+ Add column

- **Fields Delimiter** - delimiter which allows you to split a text into columns in your feed file. Supported delimiters are:
 - Comma ","
 - Tab "\t"
 - Colon ":"
 - Space " "
 - Vertical pipe "|"
 - Semi-colon ";"
- **Fields enclosure** - allows you to enclose data in your feed file.
- **Include Columns Header** - set to "Yes" to include a header row (attribute names) in the first line of your feed file.
- **Extra header** - set to "Yes" to include an additional header row in the first line of your feed file. It will always be above the first attributes row or columns header.

Field Mapping

In the field mapping table, you can add/remove rows, change rows ordering, set the output type, and the symbols limit. Each row in a mapping table is a column in the data feed file.

To add a new column to your CSV feed, you need to create a new row and fill it with a few parameters:

- **Column Name** - the header column name.
- **Value Type** - the following types are available:
 - **Pattern** - allows you to enter a static value or use patterns like those from the XML template.
 - **Attribute** - allows you to select any store attribute from the drop-down list.
 - **Parent Product** - allows you to export configurable products. In this case, simple associated products will have an attribute. If you have configurable or bundle products, we suggest you use this Type **Parent Product** with the fields: "Product URL", "Grouped ID"
- **Value** - allows you to select the attribute or put in place the pattern for the column output.

Note

To modify the **Value** output, click on the cogwheel in the required column to use the **Add Modifier** button. There will appear a list of available modifiers for usage.

To check information about modifiers, go to the page: [Output Filters](#)

FTP settings

The extension can automatically deliver data feed files via FTP to the Shopping Engine Service.

Note
Check the marketplace merchant account for FTP details, or inquire about FTP credentials at the marketplace Support Center

To configure FTP delivery, take the following steps:

1. Open the tab **FTP Settings** at the feed edit page.
2. At the tab, you need to enable FTP delivery and fill in the fields below:
 - **Protocol** - you can select a FTP/SFTP or SFTP connection.

- **Host Name** - the server on which you would like to send your feed.
 - **User Name** - the username of the FTP server.
 - **Password** - the password of the FTP server.
 - **Path** - an optional field, enter a path to your merchant folder provided by the Shopping Engine Service.
 - **Passive mode** - most FTP servers work in Passive mode, even when you are using a firewall.
3. Press the button **Save And Continue Edit**.

After enabling FTP delivery, you can manually run the delivery of the feed by pressing the button **Delivery Feed** at the right top corner of the page.

Additionally, the extension can deliver the feed by a set schedule after each feed generation.

Schedule Task Settings

You can configure your feed to generate the data feed file according to a set schedule.



If the FTP settings are enabled, the feed will be automatically delivered to the marketplace after generation by the schedule.

Note

To generate a feed by schedule, Magento Cron must be configured.

To configure the schedule, take the following steps:

1. Open the tab **Scheduled Task** at the feed edit page.
2. Set the **Status - Enabled** to enable the feed generated by schedule
3. Set up the following fields:
 - **Days of the week** - days of the feed generation.
 - **Time of the day** - time of the feed generation.
4. Press the button **Save And Continue Edit**.



CATALOG


☰ Bing Shopping ▾

← Back Delete Res


FEED INFORMATION

Feed Information

Content Settings

Filters 

Google Analytics

Scheduled Task 

FTP Settings

Additional

Scheduled Task

Enabled

 Yes ▾

If enabled, the ext
feed by schedule,

Days of the
week

- Sunday
- Monday**
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday

For example, if the selected **days** are *Monday, Wednesday* and the **time** *03:00AM, and 05:00AM*, the feed will be generated 4 times during a given week.

Note

If you already have a few feeds with scheduled tasks, in order to prevent cron job errors, you need to set a different scheduler **time** for each feed.

Email Notifications

Open the feed's **Additional** tab on the feed's edit page.

The screenshot shows a web interface for editing a feed. On the left is a dark sidebar with an orange logo and a 'CATALOG' button. The main area has a top navigation bar with 'Back', 'Delete', and 'Res' buttons. Below this is a 'FEED INFORMATION' section with a list of options: 'Feed Information', 'Content Settings', 'Filters' (with a pencil icon), 'Google Analytics', 'Scheduled Task' (with a pencil icon), 'FTP Settings', 'Additional' (with a pencil icon and a red vertical bar to its left), and 'History'. To the right, the 'Email Notifications' section is visible, featuring an 'Email' field containing 'storeadmin@' and a 'Notification Events' section with a blue highlight on 'Successful D'.

The extension can automatically send an email to the notifications for the following events:

- Successful Export
- Unsuccessful Export
- Successful Delivery
- Unsuccessful Delivery

Fill in the following lines:

- **Email** - an email address for email notifications (use a comma separator to include more than one email address).

- **Notification Events** - sets events for further email notification.

Press the button **Save And Continue Edit**.

Google Analytics Campaign

Note

Google Analytics should be configured and activated in order to use this feature.

The extension can automatically append Google Analytics campaign parameters to your product's URLs.

To configure a **Google Analytics Campaign**, take the following steps:


1. Open the tab **Google Analytics** at the feed's edit page.
2. Fill in the 3 required fields:
 - **Campaign Source** - Identifies a search engine, newsletter name, or other source (i.e. google, citysearch, newsletter)
 - **Campaign Medium** - Identifies a medium such as email or cost-per-click (i.e. cpc, banner, email)
 - **Campaign Name** - Identifies a specific product promotion or strategic campaign (i.e product, promo code, or slogan)
Also, as an option, you can fill in other fields:
 - **Campaign Term** - Identifies paid keywords.
 - **Campaign Content** - Differentiates ads or links that point to the same URL.
3. Press the button **Save And Continue Edit**.



FEED INFORMATION

Feed Information

Content Settings

Filters 

Google Analytics

Scheduled Task

FTP Settings

Google Analytics

Campaign Source

Required. Referrer

Campaign Medium

Required. Marketing

Campaign Name

Required. Product

Campaign Term

Identify the paid k

After adding Google Analytics parameters, you need to generate your feed. In the feed's file, all product URLs will be

```
http://example.com/product.html?fep=...&fee=...&utm_source=...&utm_medium=...&utm_name=...
```

After feed generating, you do not need to make any additional configuration adjustments.

Additionally, in the campaign fields, you are free to use any pattern.

To track a **Google Analytics Campaign**, log in to your account, go to **Traffic Sources > Campaigns**, and select the campaign source from the list.

Additional Settings

The extension allows you to set up additional settings for a feed export. Go to the feed tab **Additional**.

Email Notifications

Check the **Email Notifications** tab.

Report the Configuration

For each feed, you can enable/disable tracking clicks and orders by changing the setting **Enable Reports**.

If the feature is enabled, the extension appends two special arguments to the product's URL (fee=, fep=) to track clicks and orders,

where **fee** is the ID of the feed, and **fep** is the ID of the product. For example

<http://example.com/product.html?fee=1&fep=3>

Product Filters

Filtering is one of the most important parts of the feed creation process.

By using filters, you will not export products that have zero price or are out of stock, products without images, disabled products, etc.

To create a new product filter, take the following steps:

1. Go to **Catalog ? Advanced Product Feeds ? Manage Filters**. Press the button **Add Filter**.
2. Set the filter **Name**.
3. Select the feeds for which the filter will be applied. Also, you can select applied filters on the feed edit page.
4. At the **Rules** tab, you can specify all conditions you require.
5. Select an attribute from the list and use the filter conditions. The extension allows you to set the **conditions combination**, using **if ALL** and **if ANY, TRUE, or FALSE** rules.

Note

Do not set in the filter the **Visibility** conditions: **Catalog** or **Catalog, Search** if you have configurable products.

Otherwise, simple **Associated Products**, which contain the status **Not Visible Individually**, will not be included in the feed.

Product Filter Examples

- **Filter example for Simple products**

For example, you seek to export only **Simple** products from the **Furniture** category with a product **Price** that is greater than 50, and products that are available in the **Stock**:

- Set **Status is Enabled** (not necessarily)
- Set **Product Type is Simple Product**
- Add a new **Conditions Combination**. Set the **if ANY** rule. Add 2 conditions for the attribute **Visibility**:
 - **Visibility is Catalog**
 - **Visibility is Catalog, Search**

This condition allows you to include products with the **Visibility** status: **Catalog** or **Catalog, Search**. If you don't set this condition combination, the feed will also be generated with products that have status: **Not Visible Individually**.

Thus, Configurable **Associated Products** will not be included in the feed.

Continue setting up the main conditions:

- Set **Price greater than 50**
- Set **Stock availability is In Stock**
- Select **Category** attribute and set the condition to **is one of**. Click the **Chooser** icon, and you will see the store category tree. To include products from certain categories, tick the **Furniture** category and their subcategories from the list. **Category ID's** will be added automatically to the text line.

- **Filter example for Simple and Configurable products**

For example, you want to export **Simple** and **Configurable** products, which are **not** from the **Cell Phones** and **Cameras** categories.

The product **Name** contains the word **ecco** and the product **quantity** is equal to or greater than 10:

- Set **Status is Enabled** (not necessarily)
- Set **Name contains ecco**
- Set **Stock availability is In Stock**
- Set **Quantity equals or greater than 10**
- Select **Category** attribute and set the condition to **is not one of**. Click the **Chooser** icon, and you will see the store category tree. To exclude products from certain categories, tick the **Cell Phones** and **Cameras** categories and subcategories from the list. **Category ID's** will be added to the text line automatically.

- **How to exclude products without Base images**

Add the following condition to your filter:

Base Image is not ...

- **How to exclude children products if their configurable parent product is not active or is out of stock**

Add the following condition to your filter:

Is Salable is Yes

List of Patterns

All patterns must be enclosed in curly brackets. In patterns, you can use codes of attributes, filters, links to parent products, base php functions, and calculations.

The base pattern schema `{{ entity.attribute | filter | filter }}`

Attribute Patterns

1.Primary Attributes

- `{{ product.attribute_set }}` - the name of the assigned attribute set to the product
- `{{ product.description }}` - the description of the product
- `{{ product.status }}` - the status of the product

Possible values:

- Enabled
- Disabled
- `{{ product.name }}` - the name of the product
- `{{ product.meta_description }}` - the meta description of the product from the SEO section
- `{{ product.meta_keyword }}` - the meta keyword of the product from the SEO section
- `{{ product.meta_title }}` - the meta title of the product from the SEO section
- `{{ product.entity_id }}` - the ID of the product
- `{{ product.name }}` - the name of the product
- `{{ product.type_id }}` - the type of the product (simple, configurable, etc.)
- `{{ product.url }}` - the direct URL to the product
- `{{ product.url_with_options }}` - the direct URL to the product with the custom options (size, color, etc.)

This type of URL allows you to automatically select child options on the product page after the customer opens the link. At the end of the URL, there will be additional options for the parameters, f.e.:
`http://store.com/some-product-key.html#142=5596&93=5477`

- `{{ product.sku }}` - an identifier of the product
- `{{ product.short_description }}` - the short description of the product
- `{{ product.url_key }}` - the URL key of the product
- `{{ product.visibility }}` - the visibility of the product

Possible values:

- Not Visible Individually
- Catalog
- Search
- Catalog, Search

2.Prices & Taxes

- `{{ product.cost }}` - the price at which products are or have been bought by a merchant or retailer
- `{{ product.final_price }}` - the final price (saleable) of the product

The price of a product, after applying special price and catalog price rules.

- `{{ product.final_price_tax }}` - the final price (saleable) of the product with tax included
- `{{ product.msrp }}` - the manufacturer's suggested retail price, a price recommended for the sale of an item in all retail outlets.
- `{{ product.minimal_price }}` - the lowest price among all active child products
- `{{ product.price }}` - the price of the product (without discounts, catalog rules, etc.)
- `{{ product.regular_price }}` - the regular price of the product
- `{{ product.special_price }}` - a special price of the product

The special price of the product.

The special price ignores the values of *Special Price From Date* and *Special Price To Date*

- `{{ product.special_from_date }}` - the date of starting sales of the product
- `{{ product.special_to_date }}` - the date of ending sales of the product
- `{{ product.tax_class_id }}` - the tax class of the product
- `{{ product.tier_price }}` - the tier price of the product
- `{{ product.tax_rate }}` - the tax rate for the product

3.Category

- `{{ product.category_ids }}` - the IDs of the assigned categories to the product

Note

If the product is assigned to multiple categories, the extension automatically selects **Category** using this logic:

The most nested category is always selected. For example, if a product is assigned to a few categories at different levels, the attribute `{category}` returns the **name** of the category, which is the most nested in the category tree. If the product is assigned to several categories at the same level, the extension selects the category with the lowest position of the product. You can change the position of the product at **Catalog > Manage Categories**, in the tab **Category Products**

- `{{ product.category.entity_id }}` - the ID of the assigned category to the product

Note

If the product is assigned to several categories, the extension selects a **Category Id** using this logic:

The most nested category id is always selected. For example, if a product is assigned to multiple categories at different levels, the attribute `{category_id}` returns the **id** of the category, which is the most nested in the category tree.

If a product is assigned to several categories at the same level, the extension selects the category id with the lowest position of the product. To change the position of the product, go to **Catalog > Manage Categories**, in the tab **Category Products**

- `{{ product.category.path }}` - the path of the category names

E.g. Computers > Notebooks > Apple

Note

If the product is assigned to several categories, the extension selects a **Category Path** using this logic:

The most nested category path is always selected. For example, if a product is assigned to multiple categories at different levels, the attribute returns the **path** that is the most nested in the category tree. If a product is assigned to several categories at the same level, the extension selects the category with the lowest position of the product. To change the position of the product, go to **Catalog > Manage Categories**, in the tab **Category Products**

4.Images

- `{{ product.image }}` - the direct URL to the base image of the product
- `{{ product.images }}` - the direct URLs gallery images of the product, separated by a comma
- `{{ product.thumbnail }}` - the direct URL to the thumbnail image of the product
- `{{ product.small_image }}` - the direct URL to the small image of the product
- `{{ product.gallery[0] }}`, `{{ product.gallery[1] }}` ... - a direct url to gallery images of the product
- `{{ product.rating_summary }}` - the average product rating (from 0 to 5)

5.Stock Attributes

- `{{ product.is_in_stock }}` - the stock status of the product

Possible values:

- **0** - Out of Stock
- **1** - In Stock
- `{{ product.qty }}` -the quantity of the product
- `{{ product.qty_children }}` - the quantity of children in stock products
- `{{ product.stock_status }}` - the stock status of the product

Possible values:

- Out of Stock
- In Stock

6.Reviews

Note

To export review attributes, they should be set in the Reviews context.
For example, to export, the review title must be set in the following construction:

```
{% for review in context.reviews %}  
<title><![CDATA[{{ review.title }}]></title>  
{% endfor %}
```

- `{{ review.id }}` - the review ID
- `{{ review.nickname }}` - the nickname of the customer who left review
- `{{ review.created_at }}` - the date of the review creation
- `{{ review.title }}` - the title (summary) of the review
- `{{ review.detail }}` - the review text
- `{{ review.rating }}` - the review rating
- `{{ review.reviews_count }}` - the number of approved reviews for this product
- `{{ review.product.[product_attribute] }}` - allows for use of the product attributes at the Review context construction, where instead of `[product_attribute]`, there should be set a product attribute code.
For example, a product's SKU: `{{ review.product.sku }}`

7. Multi Source Inventory Sources

Note

Multi-Source Inventory (MSI) is available, starting from Magento 2.3

Allows you to export MSI Sources using the following pattern:

```
{{ product.inventory:source_code }} - where instead of source_code, there should be an  
Inventory Source Code set.
```

Parent product values

You can use the suffix **.parent** (`{{ product.parent.name }}`, `{{ product.parent.price }}`, `{{ product.parent.url }}` etc), if you need to return the value to the parent product.

Example

If the current product associated with a configurable/grouped/bundled product, the pattern `{{ product.parent.url }}`, will return the URL to the parent product. If the extension can't find the parent product, it will use the current product.

Note: The parent suffix is very useful when you export **simple** products with visibility **Not Visible Individually**. In this case, the product can't have a direct link, so you must use a link to the parent product.

Example

```
{% for product in context.products %}  
  
  {% for image in product.gallery %}  
  <picture>{{ image }}</picture>  
  {% endfor %}
```

```
<created>{{ product.created_at | dateFormat: 'd.m.Y H:i:s' }}</created>
{% endfor %}
```

How to hide a special price if it is empty

```
{% if product.special_price %}
    <g:sale_price><![CDATA[{{ product.special_price }} USD]]></g:sale_price>
{% endif %}
```

How to hide the shipping weight if it is empty

```
{% if product.weight %}
    <g:shipping_weight><![CDATA[{{ product.weight }} kg]]></g:shipping_weight>
{% endif %}
```

Filters

Filters are simple methods that modify the output of numbers, strings, variables, and arrays. They are placed within an output tag `{{ }}` and are separated with a pipe character `|`.

String/HTML Filters

- lowercase - converts a string into lowercase.

```
{{ product.name | lowercase }}
```

Original: Dash Digital Watch
Output : dash digital watch

- uppercase - converts a string into uppercase.

```
{{ product.name | lowercase }}
```

Original: Dash Digital Watch
Output : DASH DIGITAL WATCH

- replace - replaces all occurrences of a string with a substring.

```
{{ product.name | replace: 'Digital', 'Analog' }}
```

Original: Dash Digital Watch
Output : Dash Analog watch

- append - appends characters to a string.

```
{{ product.name | append: ' - best choice' }}
```

Original: Dash Digital Watch
Output : Dash Digital Watch - best choice

- prepend - prepends characters to a string.

```
{{ product.name | prepend: 'Best choice - ' }}
```

Original: Dash Digital Watch
Output : Best choice - Dash Digital Watch

- capitalize - capitalizes the first word in a string.

```
{{ product.color | capitalize }}
```

Original: dark red
Output : Dark red

- escape - escapes HTML tags in a string.

```
{{ product.description | escape }}
```

- nl2br - inserts a
 linebreak HTML tag in front of each line break in a string.

```
{{ product.short_description | nl2br }}
```

- remove - removes all occurrences of a substring from a string.

```
{{ product.name | remove: 'Digital' }}
```

Original: Dash Digital Watch
Output : Dash Watch

- stripHtml - strips all HTML tags from a string.

```
{{ product.description | stripHtml }}
```

- strip all newlines (\n, \r) from a string.

```
{{ product.description | strip_newlines }}
```

- Replace each newline (\n) with html break - replace \n to the
.

```
{{ product.description | newline_to_br }}
```

- truncate - truncates a string down to 'x' characters.

```
{{ product.name | truncate: '15' }}
```

Original: Dash Digital Watch
Output : ash Digital Wa

- truncate string down to x words - truncates a string down to 'x' words.

```
{{ product.name | truncatewords: '2' }}
```

Original: Dash Digital Watch

Output : Dash Digital

- plain - converts any text to plain format.

```
{{ product.description | plain }}
```

- ifEmpty - return argument, if the value is an empty string

```
{{ product.color | ifEmpty: 'Multi color' }}
```

Original:

Output : Multi color

- dateFormat - converts string to specified date-time format.

```
{{ product.created_at | dateFormat: 'd.m.Y H:i' }}
```

Original: 2016-02-18 10:11:12

Output : 18.02.2016 10:11

- json - converts an object or array to JSON format.

```
{{ product.gallery | json }}
```

Number Filters

- ceil - rounds an output up to the nearest integer.

```
{{ product.weight | ceil }}
```

Original: 1.423

Output : 2

- floor - rounds an output down to the nearest integer.

```
{{ product.weight | floor }}
```

Original: 1.423

Output : 1

- round - rounds the output to the nearest integer or specified number of decimals.

```
{{ product.weight | round: '2' }}
```

Original: 1.423
Output : 1.42

```
{{ product.weight | round }}
```

Original: 1.423
Output : 1

- numberFormat - formats the number to a specified format (php function).

```
{{ product.price | numberFormat: '2', '.', ',' }}
```

Price/Currency Filters

- price - formats a price to the default format.

```
{{ product.price | price }}
```

Original: 100.4200
Output : 100.42

- convert - converts a price from the base currency to the specified currency.

```
{{ product.price | convert: 'EUR' }}
```

Original: 100
Output : 92.28

- inclTax - include a tax to the product price.

```
{{ product.price | inclTax | price }}
```

- exclTax - exclude a tax from the product price.

```
{{ product.price | exclTax | price }}
```

- Addition - adds the specified value.

```
{{ product.price | plus: '2' }}
```

Original: 100
Output : 102

- Subtraction - subtracts the specified value.

```
{{ product.price | minus: '2' }}
```

Original: 100
Output : 98

- Multiplication - multiplied by the specified value.

```
{{ product.price | times: '2' }}
```

Original: 100

Output : 200

- Division - divided by the specified value.

```
{{ product.price | divided_by: '2' }}
```

Original: 100

Output : 50

- Modulo - the remainder after dividing by the specified value.

```
{{ product.price | modulo: '2' }}
```

Original: 100

Output : 0

Array Filters

- first - return the first element in the array.
- last - return the last element in the array.
- join - join an array to a string using glue.
- count - return the number of elements in the array.
- select - select values for a key from the array.

URL Filters

- secure - return a secure URL (with https://)
- unsecure - return an unsecure URL (with http://)
- mediaSecure - return a media URL using "Base URL for User Media Files"

```
{{ product.image | mediaSecure }}
```

Original: http://example.com/pub/media/catalog/product/m/h/mh03-black_main

Output : https://cdn-secure.com/catalog/product/m/h/mh03-black_main.jpg

- mediaUnsecure - return a media URL using "Secure Base URL for User Media Files "

```
{{ product.image | mediaUnsecure }}
```

Original: http://example.com/pub/media/catalog/product/m/h/mh03-black_main
Output : http://cdn.com/catalog/product/m/h/mh03-black_main.jpg

Image Filters

- resize - resize an image

```
{{ product.image | resize: 100, 100 }}
```

Original: http://example.com/pub/media/catalog/product/m/h/mh03-black_main
Output : http://example.com/pub/media/cache/200x200/catalog/product/m/h/mh03-black_main.jpg

Note

- ```
{{ product.name | truncate: '150' }}
```
- ```
{{ product.description | plain | truncate: '1000' }}
```
- ```
{{ product.weight | round: '2' }} kg
```
- ```
{{ product.manufacturer | ifEmpty: 'not set' }}
```

Dynamic attributes

The dynamic attribute is an attribute that can return values depending on particular conditions or the value of other attributes.

To create a new dynamic attribute, follow these steps:

1. Go to **Product ? Advanced Product Feeds ? Dynamic Attributes**. Press the button **Add Attribute**.
2. Fill in the following fields:
 - Name - name of the dynamic attribute
 - Code - code of the dynamic attribute. You will see this code when you select the dynamic attribute in the templates.
 - Set conditions:
 - Press "Add "OR" condition"
 - Press "Add "AND" condition"
 - Select attribute on the left corner
 - Select condition
 - Select or input condition value
 - Select Output Type
 - Select attribute or input pattern - when the condition is true, the dynamic attribute will return this value.
 - Press the button Save. *To set a default value, leave the Conditions column without conditions.*



CATALOG

☰ Color ▾

General Information

Name * Color

Attribute Code * color

Conditions

Conditions			
↕	Color [color] ▾	is ▾	not set
+ Add "AND" condition			
↕	Color [color] ▾	is not ▾	not set

Example

- Name: Additional Stock Status
- Code: stock_status

Conditions:

General Information

Name * Additional Stock Status

Attribute Code * stock_status

Conditions

Conditions				
↑	Qty	▼	equals or greater than	▼
↑	Qty	▼	less than	▼
	Qty	▼	equals or greater than	▼
↑	Qty	▼	is	▼

+ Add "OR" condition

Usage:

XML templates: {{ product.dynamic:stock_status }}

CSV templates: find your dynamic attribute in the dropdown list, dynamic attribute section:

Include Columns Header

Extra Header

Column Name		Value Type
<input type="button" value="↕"/> <input type="button" value="⚙"/>	<input type="text" value="title"/>	<input type="text" value="Attribute"/> <input type="button" value="▼"/>
<input type="button" value="↕"/> <input type="button" value="⚙"/>	<input type="text" value="stock_status"/>	<input type="text" value="Attribute"/> <input type="button" value="▼"/>

Output:

```
<?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0" xmlns:g="http://base.google.com/ns/

<item>
  <name><![CDATA[Joust Douffle Bag]]></name>
  <qty>9</qty>
  <stock status>Product is almost sold out</stock
</item>

<item>
  <name><![CDATA[Strive Shoulder Pack]]></name>
  <qty>100</qty>
  <stock status>Product in stock</stock status>
</item>











<item>
  <name><![CDATA[Ida Workout Parachute Pant]]></na
  <qty>0</qty>
  <stock status>Product is out of stock</stock sta
</item>
</rss>
```

Dynamic variables

The dynamic variable is a user-defined php code that must return a string value. The PHP code is stored on your server in your Magento installation folder. The extension fetches it from the file.

To create a new dynamic variable, follow these steps:

1. Go to **Catalog ? Advanced Product Feeds ? Dynamic Variables**. Press the button **Add Variable**.
2. Fill in the following fields:
 - Name - name of the dynamic variable
 - Variable Code - code of the dynamic variable. You will see this code when you select this dynamic variable in the templates.

-  DASHBOARD
-  SALES
-  CATALOG
-  CUSTOMERS
-  MARKETING
-  CONTENT
-  REPORTS
-  STORES
-  SYSTEM
-  FIND PARTNERS & EXTENSIONS

Variable for GTIN

General Information

Name * Variable for GTIN

Variable Code * GTIN_variable

Code

```

1 <?php
2 $gtin = $product->getGtin();
3 if (!$gtin) {
4     $gtin = $product->getId();
5     $gtin .= str_repeat('0', 12 - strlen($gtin));
6 }
7 echo $gtin;
8 ?>
9

```

\$product and \$objectManager available in context for use

Variable source code located in the file: app/code/Mirasvit/Feed/GTIN_variable.p

1. Create the PHP code for the Dynamic variable:

- Login to your server.
- Navigate to the **var/mst_feed/** or **app/code/Mirasvit/Feed/** folder
- Create a PHP file in one of these folders named exactly as the Variable Code you have entered previously.
For example, if the Variable Code is *GTIN_variable*, you need to create a file `var/mst_feed/GTIN_variable.php` or `app/code/Mirasvit/Feed/GTIN_variable.php`
- Paste the PHP code for your variable in the file and save it.

Note

Use the **Echo** function in the PHP code instead of **Return** to get the product data.

Note

For Magento Cloud, it is recommended to move variables from `var/mst_feed` to `app/code/Mirasvit/Feed` folder before deployment.

1. Return to your Magento admin and reload the page or click Save button. The section Code will show the PHP code that was fetched for this variable from your server.

To see an example of the dynamic variable output you can use the field right to the Code section. Enter product ID's separated by commas to see the data for several products.

Examples

Note

Ensure the code of your dynamic variable is enclosed in PHP opening `<?php` tag:

```
<?php
```

Your Dynamic Variable code

- **How to get the correct GTIN**

Name: GTIN

Code: gtin

PHP Code:

```
<?php
$gtin = $product->getGtin();
if (!$gtin) {
    $gtin = $product->getId();
    $gtin .= str_repeat('0', 12 - strlen($gtin));
}
echo $gtin;
```

Usage:

```
{{ product.variable:gtin }}
```

Output:

```
124200000000
```

- **How to add 20% to the final price**

Name: Add Percentage

Code: percentage

PHP Code:

```
<?php
$price = $product->getFinalPrice();
echo number_format($price*1.2, 2);
```

Usage:

```
{{ product.variable:percentage }}
```

- **How to get the qty of product from all stocks**

Name: Summary stocks qty

Code: sum_stocks_qty

PHP Code:

```
<?php
$sourcesData = $objectManager->get('\Magento\InventoryCatalogAdminUi\Model\GetS
->execute($product->getSku());
```

```
$sumQty = 0;
```

```
foreach ($sourcesData as $sourceData){
    $sumQty += $sourceData['quantity'];
}
```

```
echo $sumQty;
```

Usage: {{ product.variable:sum_stocks_qty }}

- **I would like to keep the Special Price field empty, unless the final price is lower then the regular price**

Name: Custom Special Price

Code: custom_spec_price

PHP Code:

```
<?php
$finalPrice = $product->getFinalPrice();
$regularPrice = $product->getPrice();
```

```
if ($finalPrice < $regularPrice) {
    echo number_format($finalPrice, 2);
} else {
    echo '';
}
```

Usage: {{ product.variable:custom_spec_price }}

Category Mapping

Note

What is category mapping?

The category names you are using in your Magento store aren't always the same as those used by Comparison Shopping Engines to reference your products.

This means you have to find out which Shopping Engine categories are most closely aligned with your store's categories.

To create a new category mapping, follow these steps:

1. Go to **Catalog ? Advanced Product Feeds ? Category Mapping**. Press the button **Add Category Mapping**.



Google Shopping Mapping ▾


CATALOG

General Information

Name *

Root Catalog (ID: 1)

Default Category (ID: 2)

What's New (ID: 38)

Women (ID: 20)

Men (ID: 11)

Gear (ID: 3)

Collections (ID: 7)

1. Fill in the mapping **Name**.
2. Fill in the new names related to your store categories.
If the child category doesn't have its own related name, it will use the related name of the parent category.

Note

When you start typing the category names, the extension will automatically show all the available categories from the Google Shopping taxonomy txt file.

You can add other txt files with taxonomies to use it for category mapping in the folder **magento_root/vendor/mirasvit/module-feed/src/Feed/Setup/data/mapping**

3. Press the button **Save**.

Now you can use the new category mapping in your feeds. You can select the newly-created **Category Mapping attributes** from the drop-down list of template attributes.

Note

- For CSV feeds, at the attribute selector, you need to select the attribute **Category Mapping: Name**
- For XML feeds, you should use the pattern, `{{product.mapping:mapping_id}}`

Example

For example: `{{ product.mapping:1 }}`

Where **1** is a category mapping id.

Reports

The extension allows you to track the clicks and orders after the customer redirects from the marketplaces.

Note

The extension can track the clicks and orders only via special arguments **ff=** and **fp=**, which are appended to the product URLs.

To enable this option, go to the feed tab **Additional > Enable Reports** and set **Yes**.

The extension allows you to show the Report graphics by **Feed, SKU, Day, Week, Month, Year**.

Main columns of the **Report**:

- **Number of Clicks** - numbers of times which the customer redirects from the marketplace.
- **Number of Orders** - numbers of the orders which were made after the customer has been redirected from the marketplace.
- **Revenue** - revenue of the order.
- **Revenue per Click** - average revenue amassed for each individual click.

Import/Export Data

Go to **Products > Import/Export Data**.

The extension allows you to import and export the following feed entities:

- **Templates**

- **Filters**
- **Dynamic Attributes**
- **Dynamic Categories**
- **Dynamic Attributes**

Select the required tab and import/export the required files.

Command Line Interface

Usage: `php -f bin/magento [options]`

- `mirasvit:feed:export` - generate all active feeds
- `mirasvit:feed:export --id=<id>` - generate feed with specified id
- `mirasvit:feed:delivery` - delivery of all active feeds
- `mirasvit:feed:delivery --id=<id>` - delivery feed with specified id

Note

`mirasvit:feed:export --step=100` - if your server has sufficient resources, you can use the additional parameter `--step`. Changing this parameter can decrease feed generation time or server resources that are required. By default, step equals 50.

How to upgrade the extension

To upgrade the extension, take the following steps:

1. Back up your store's database and web directory.
2. Log in to the SSH console of your server and navigate to the root directory of the Magento 2 store.
3. Run this command to update the current extension with all dependencies: `composer require mirasvit/module-feed:* --update-with-dependencies`.

Note

In some cases, the command above is not applicable, or you are unable to update just the current module, and need to upgrade all Mirasvit modules in a bundle. In this case, the command above will have no effect.

Instead, run the `composer update mirasvit/*` command, which will update all Mirasvit modules installed in your store.

4. Run the command `php -f bin/magento module:enable Mirasvit_Core Mirasvit_Feed Mirasvit_Report` to re-enable the extension.
5. Run the command `php -f bin/magento setup:upgrade` to install updates.
6. Run the command `php -f bin/magento cache:clean` to clean the cache.

7. Deploy static view files

```
rm -rf pub/static/frontend/*; rm -rf pub/static/backend/*; rm -rf  
var/view_preprocessed/*; php -f bin/magento setup:static-  
content:deploy
```

Disabling the Extension

Temporarily Disable

To temporarily disable the extension, please take the following steps:

1. Log in to the SSH console of your server and navigate to the root directory of the Magento 2 store.
2. Run the command `php -f bin/magento module:disable Mirasvit_Feed` to disabled the extension.
3. Log in to the Magento backend and refresh the store cache (if enabled).

Extension Removal

To uninstall the extension, please take the following steps:

1. Log in to the SSH console of your server and navigate to the root directory of the Magento 2 store.
2. Run the command `composer remove mirasvit/module-feed` to remove the extension.
3. Log in to the Magento backend and refresh the store cache (if enabled).

Troubleshooting

This section describes the most common problems that customers report, and how they can be resolved:

- [Blank page when I have navigate to the Catalog > Feeds](#)
 - [I have added category filter by fews categories, but my feed doesn't contain any product](#)
 - [Missing or incorrect required attribute \[price\]](#)
 - [Missing two out of three attributes \[GTIN, Brand, MPN\]](#)
-

There is a blank page when I have navigate to the Catalog > Feeds

You probably don't have the required library `zf1/zend-reflection`.

Solution: Please, try to install the library through composer:
`composer require "zf1/zend-reflection"`

I have added a category filter by fews categories,

but my feed doesn't contain any products

Such troubles usually occur when you use the "CATEGORY IS 1,2,3,4..." condition. This means that only products assigned to ALL of those categories simultaneously will be exported in the feed.

Solution: Try to change the condition to the "CATEGORY IS ONE OF..." instead, and regenerate the feed.

A required attribute is missing or incorrect [price]

Try adding a currency directly to the price attribute from the content settings of the feed, and regenerate it.

Before: <g:price>{{ product.final_price }}</g:price>

After: <g:price>{{ product.final_price }} EUR</g:price>

Also, you can define the currency directly in the **Google Merchant Center**:

Diagnosics

List

Feeds

Product suggestions

PROCESSING **SETTINGS** FEED RULES

Country of sale, language and destinations

Select the country and language for product data.

Country

^ United States

ADD

Settings apply to all countries and languages.

Feed settings	Encoding: Auto-detect; De
Fetch schedule	Daily
Default currency	No default currency
ID rule	Rule not set

Two out of three attributes are missing [GTIN, Brand, MPN]

It happens when gtin and brand attributes are not provided. Try creating a dynamic attribute with the following statements:

New Attribute

General Information

Name * Identifier Exists

Attribute Code * identifier_exists

Conditions

Conditions



Manufacturer [manufacturer]



is



UPC [upc]



is



Manufacturer [manufacturer]



is



UPC [upc]



is not



Manufacturer [manufacturer]



is not



UPC [upc]



is



+ Add "OR" condition

and use it in the `<g:identifier_exists>` tag.

Here is an example of how you can use it in the feed:

Content Settings

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <rss version="2.0" xmlns:g="http://base.google.com/ns/1.0">
3   <channel>
4     <title>Data feed Title</title>
5     <link>{{ context.store.base_url }}</link>
6     <description>Data feed description.</description>
7     {% for product in context.products %}
8     <item>
9       <g:id>{{ product.sku }}]&gt;&lt;/g:id&gt;
10      &lt;title&gt;<![CDATA[{{ product.name | truncate: '150' }}]&gt;&lt;/title&gt;
11      &lt;link&gt;<![CDATA[{{ product.url }}]&gt;&lt;/link&gt;
12      &lt;g:price&gt;{{ product.final_price }}&lt;/g:price&gt;
13      &lt;description&gt;<![CDATA[{{ product.description | stripHtml | plain | truncate: '5000' }}]&gt;
14      &lt;g:product_type&gt;<![CDATA[{{ product.category.path }}]&gt;&lt;/g:product_type&gt;
15      &lt;g:google_product_category&gt;<![CDATA[]]&gt;&lt;/g:google_product_category&gt;
16      &lt;g:image_link&gt;<![CDATA[{{ product.image }}]&gt;&lt;/g:image_link&gt;
17      {% for image in product.gallery %}
18      &lt;g:additional_image_link&gt;<![CDATA[{{ image }}]&gt;&lt;/g:additional_image_link&gt;
19      {% endfor %}
20      &lt;g:condition&gt;new&lt;/g:condition&gt;
21      &lt;g:availability&gt;{{ product.stock_status }}&lt;/g:availability&gt;
22      &lt;g:shipping_weight&gt;{{ product.weight }} kg&lt;/g:shipping_weight&gt;
23      &lt;g:brand&gt;<![CDATA[{{ product.manufacturer }}]&gt;&lt;/g:brand&gt;
24      &lt;g:mpn&gt;<![CDATA[{{ product.sku }}]&gt;&lt;/g:mpn&gt;
25      &lt;g:gtin&gt;<![CDATA[{{ product.upc }}]&gt;&lt;/g:gtin&gt;
26      &lt;g:identifier_exists&gt;{{ product.dynamic:identifier_exists }}&lt;/g:identifier_exists&gt;
27    &lt;/item&gt;
28    {% endfor %}
29  &lt;/channel&gt;
30 &lt;/rss&gt;</pre></div><div data-bbox="85 574 158 588" data-label="Text"><p>Attribute</p></div><div data-bbox="347 574 610 589" data-label="Text"><p>Dynamic Attribute: Identifier Exists</p></div><div data-bbox="74 610 452 627" data-label="Text"><pre>{{ product.dynamic:identifier_exists }}</pre></div><div data-bbox="47 684 262 714" data-label="Section-Header"><h2>Change Log</h2></div><div data-bbox="47 734 113 756" data-label="Section-Header"><h3>1.4.9</h3></div><div data-bbox="47 776 160 793" data-label="Text"><p>(2024-07-24)</p></div><div data-bbox="47 811 101 828" data-label="Section-Header"><h4>Fixed</h4></div><div data-bbox="78 848 858 882" data-label="List-Group"><ul><li>• Fixed import on AWS S3</li><li>• Fixed the issue with inclTax and exclTax modifiers (Unsupported operand types: string * int)</li></ul></div><div data-bbox="47 915 114 938" data-label="Section-Header"><h3>1.4.8</h3></div>
```

(2024-07-11)

Improvements

- Use feed filename instead of id for temporary file names
- Updated the logic for the "Special Price From Date" and "Special Price To Date" modifiers
- Use a relative path while uploading to the bucket (AWS S3 compatibility)
- CSP enhancements

Fixed

- Fixed the issue with the modifier arguments when the pipe sign is used
-

1.4.7

(2024-06-27)

Fixed

- Fixed inline script errors (CSP)
 - Fixed url_with_options modifier. Display attribute codes instead of attribute IDs.
 - Fixed product grid on filter page (Mageplaza_ProductGrid compatibility)
-

1.4.6

(2024-06-17)

Improvements

- Redirect to feed creation page if ID is not set on feed edit page
- Return the parent product if it is not a simple product

Fixed

- Fixed file locking issue on shared NFS disk volumes
 - Fixed the issue with product URLs (admin path is included in the URL)
-

1.4.5

(2024-05-29)

Improvements

- Return the lowest category of the product that belongs to the root category of the store for which the feed is created
- Return the parent product if it is not a simple product

Fixed

- Fixed the issue with CSV feeds (some rows are displayed as one line)
 - Fixed the issue with product URLs (admin path is included in the URL)
 - Removed spaces from the "Gallery Image Collection"
-

1.4.4

(2024-04-26)

Features

- Added "Strip style tag" modifier

Fixed

- Fixed preview in CSV feeds (conflict between jQuery and lib/web/legacy-build.min.js)
-

1.4.3

(2024-04-18)

Improvements

- Add a script for reports only if the reports feature is enabled in the configurations

Fixed

- Fixed the issue with incorrect feed URL
-

1.4.2

(2024-03-21)

Improvements

- Check that the feed is generated before delivery
- Added ability to get the "Manage Stock" attribute

Fixed

- Fixed the issue with custom static attributes
 - Fixed the issue with export generation on Magento PWA
-

1.4.1

(2024-03-04)

Fixed

- Fixed the issue with an empty first line in CSV/TXT feeds when the header is excluded
 - Fixed the issue with getting stock status for a bundle product
 - Fixed the issue with feed email notifications not being sent
 - Fixed compatibility with PHP 7.1
-

1.4.0

(2024-02-21)

Features

- Added ability to manage feeds in remote storage

Improvements

- Remove modifier arguments if no modifier is set (CSV, TXT feeds)
- Get categories in a feed only for the selected store
- Optimized the loading of the "Content Settings" tab when the number of product attributes exceeds 1000

Fixed

- Added a patch to fix CSV and TXT feeds where modifier arguments were stored incorrectly
 - Fixed the issue with getting the lowest category for a product
-

1.3.19

(2024-01-26)

Improvements

- Using plain JS for compatibility with the Hyva theme (Reports feature)

Fixed

- Fixed the issue with incorrect product URLs in review feeds
 - Fixed the issue where the product filter was not being applied to review feeds
 - Fixed display of "Filters" on feed edit page
 - Fixed the issue with the export cron (TypeError: strtotime(): Argument #1 (\$datetime) must be of type string, null given)
 - Fixed the issue with the append filter (TypeError: Mirasvit\Feed\Export\Filter\NumberFilter::append(): Return value must be of type string, null returned)
-

1.3.18

(2024-01-18)

Improvements

- Creating a secure SSL-FTP connection when using the "FTP / FTPS" protocol in the "FTP Settings" tab of the feed

Fixed

- Fixed the issue with the getParents function in Magento B2B
-

1.3.17

(2024-01-05)

Improvements

- Added a check that the parent product ID is different from the child product ID

Fixed

- Fixed the issue with the category.parent_category.id pattern
-

1.3.16

(2023-12-22)

Fixed

- Fixed the issue with the getFinalPrice function (returns bool in some cases)
 - Fixed the issue with the "mirasvit:feed:export" command when using the --step parameter
-

1.3.15

(2023-12-08)

Improvements

- Changed the format_serialized column type from text to longtext in the mst_feed_feed and mst_feed_template tables
-

1.3.14

(2023-11-23)

Features

- Added the ability to get all parent products in an XML feed

Fixed

- Fixed the issue with stock_status and is_in_stock attributes in feeds when MSI is enabled
 - Removed store codes from feed URLs
-

1.3.13

(2023-11-16)

Improvements

- Removed the duplicate code related to feed and feed templates

Fixed

- Fixed adding and removing rows, modifiers, and validators in the field mapping table
 - Fixed the error when running the mirasvit:feed:cron command
 - Fixed the issue with the feed preview loading from the "FTP Settings" tab
-

1.3.12

(2023-11-09)

Improvements

- Exclude processing of self-included dynamic attributes

Fixed

- Fixed the issue with the category_ids attribute in CSV and TXT feeds
-

1.3.11

(2023-10-13)

Fixed

- Fixed the error in Facebook feed preview
 - Fixed the issue with the return type of the "convert" modifier
-

1.3.10

(2023-10-06)

Fixed

- Fixed the issue with the return type value (feedStoreResolver plugin)
 - Fixed the issue with the "Format date" modifier
-

1.3.9

(2023-10-04)

Features

- Added the ability to filter feeds by stock availability in MSI

Improvements

- Improved request structure for validating and saving dynamic variables
-

1.3.8

(2023-09-25)

Features

- Added the ability to set the "excludes" filter for the "if" tag in feed content settings
- Added the ability to store dynamic variables in the var/mst_feed folder

Fixed

- Fixed the issue with the CustomGento_DefaultStoreCodeRemover module compatibility
 - Fixed the issue with feed filters when Magento_Inventory is disabled
 - Fixed the type error for null category attribute during feed generation
 - Fixed the issue with the "contains" filter for the "if" tag in feed content settings
-

1.3.7

(2023-09-12)

Fixed

- Fixed the return type of the getChildrenInStock function
-

1.3.6

(2023-09-08)

Fixed

- Fixed the issue with the truncate words filter
-

1.3.5

(2023-09-04)

Fixed

- Fixed the issue with the incorrect feed URL
 - Fixed the type of ff and fp parameters
-

1.3.4

(2023-08-23)

Fixed

- Fixed the issue with Number Filter (affects from 1.3.0)
-

1.3.3

(2023-08-22)

Fixed

- Fixed the issue with String Filter (affects from 1.3.0)
-

1.3.2

(2023-08-18)

Fixed

- Fixed bugs related to incorrect data types (affects from 1.3.0)
-

1.3.1

(2023-08-15)

Fixed

- Fixed the issue with the error while generating feeds (PHP8 compatibility, affects from 1.3.0)
-

1.3.0

(2023-08-14)

Improvements

- Code quality improved
 - Backend interface changed to UI components
-

1.2.11

(2023-04-25)

Fixed

- fixed an issue with not capturing modifier input values in XML feed templates
 - fixed an issue with missing liquid modifiers in php 8.1
 - fixed error on the Filter edit page - "The 'componentType' configuration parameter is required for the 'product_columns' component"
 - fixed an issue with missing block names in layouts
 - fixed an issue in the filter 'Preview products' button - "Column 'entity_id' in group statement is ambiguous"
 - fixed some styles for the buttons and progress bar
-

1.2.10

(2023-03-15)

Improvements

- Added support of Magento 2.4.6
- Search filter in templates and feeds grid improved

Fixed

- PHP 8.2
-

1.2.9

(2022-12-15)

Improvements

- Updated logic (location of code) for dynamic variables

Fixed

- Fixed the issue with filter preview (stock status)
-

1.2.8

(2022-09-20)

Fixed

- fixed an issue with not displaying all created feeds on the main feeds list
 - fixed an issue with the feed generations with email notifications
 - fixed an issue with js cookies on Magento 2.4.5 with caused problem with loading checkout page
-

1.2.7

(2022-09-14)

Improvements

- Feed statistics on edit page
 - Filter products preview
-

1.2.6

(2022-09-08)

Improvements

- Admin grids updated
- Added feeds stats to feeds grid

Fixed

- fixed an issue with ETA errors during feed generation
-

1.2.5

(2022-08-18)

Fixed

- Fixed the issue with products data for some products are not displayed in review feed
-

1.2.4

(2022-08-09)

Fixed

- fixed an issue with return value of console commands
-

1.2.3

(2022-07-05)

Fixed

- fixed issue with feed generation progress bar focus
-

1.2.2

(2022-06-20)

Improvements

- remove db_schema_whitelist.json
-

1.2.1

(2022-05-23)

Improvements

- updated module report version in the composer
-

1.2.0

(2022-05-13)

Improvements

- switch to declarative DB schema
-

1.1.38

(2022-04-05)

Fixed

- fixed issue with using liquid patterns in feed templates
-

1.1.37

(2022-03-23)

- Support of PHP8.1

1.1.36

(2021-12-23)

Fixed

- fixed the problem with the Category Mapping autocomplete searches
-

1.1.35

(2021-11-09)

Improvements

- Demo
-

1.1.34

(2021-09-28)

Improvements

- added eCommerce Network template
 - updated Google autocomplete categories in the Category Mappings
-

1.1.33

(2021-05-07)

Fixed

- Issue with IS NOT filter condition
-

1.1.32

(2021-05-07)

Improvements

- Implemented LockManager interface
- Added the ability to use additional images in the filter rule conditions

Fixed

- Fixed an issue with duplicated cron export feed generation
-

1.1.31

(2021-04-12)

Fixed

- Fixed an issue with the incorrect filtering 'Amount of Children in stock' via cron generation
 - Fixed an issue with the missing Category Mapping autocomplete
-

1.1.30

(2021-01-28)

Improvements

- Added ability to enable or disable fast filtering mode at the feed 'Additional' tab
 - Added Trovaprezzi template
-

1.1.29

(2020-12-30)

Fixed

- fixed an issue with incorrect output of the filter condition 'IS'
 - fixed an issue with the encoding characters at the feed preview
 - fixed an issue with the inability to set empty field option at the filter condition
-

1.1.28

(2020-12-03)

Improvements

- Added ability to disable FB Metadata (tab "Additional")
-

1.1.27

(2020-11-18)

Improvements

- speed up the time of the feed export with the Category Mappings
 - added Multi Source Inventory stocks filter conditions
-

1.1.26

(2020-11-13)

Fixed

- Incorrect class name for notification about installation of zf1/zend-reflection library
-

1.1.25

(2020-11-09)

Improvements

- module compatibility with the php 7.4
- added an error message if Zend Reflection library wasn't installed

Fixed

- fixed an issue with the incorrect filter Delete message
-

1.1.24

(2020-10-21)

Fixed

- invalid ID attribute

Improvements

- added new attribute 'Item group ID'
-

1.1.23

(2020-09-28)

Fixed

- fixed an issue with the incorrect filtering by categories
-

1.1.22

(2020-09-11)

Improvements

- added metadata to the Facebook feeds
-

1.1.21

(2020-09-10)

Fixed

- fixed an issue with the filtering incorrect amount of products
-

1.1.20

(2020-09-08)

Improvements

- Filter interface and applying logic
-

1.1.19

(2020-09-05)

Fixed

- Minor fixes
-

1.1.18

(2020-07-29)

Improvements

- Support of Magento 2.4
-

1.1.17

(2020-07-03)

Fixed

- fixed an issue with the filtering non-default attributes
-

1.1.16

(2020-06-12)

Improvements

- added ability to export salable quantities of Multi Source Inventory stocks

Fixed

- fixed an issue with the using filter conditions
-

1.1.15

(2020-05-15)

Fixed

- fixed an issue with the incorrect exporting of the "Product Type" attribute values at the csv and txt feeds
 - fixed an issue with the mirasvit sorting product collection during feed generation
-

1.1.14

(2020-05-12)

Fixed

- fixed errors during the filtration step
-

1.1.13

(2020-04-29)

Improvements

- Improved filtration time
-

1.1.12

(2020-04-14)

Fixed

- fixed an issue with the exporting inactive special prices
-

1.1.11

(2020-03-23)

Fixed

- fixed an issue with the incorrect select options of the rule condition 'manage stock'
-

1.1.10

(2020-03-04)

Fixed

- Issue when input for cron enabling/disabling don't show current cron status for feed
 - Price format
-

1.1.9

(2020-02-19)

Fixed

- fixed an issue with the error during new category mapping creation
-

1.1.8

(2020-02-04)

Features

- added cron info to the 'Scheduled Task' feed tab
-

1.1.7

(2019-12-27)

Features

- added new product attribute 'Product Url with custom options'
-

1.1.6

(2019-12-18)

Features

- added new multi source inventory filters 'manage stock' and 'qty'

Fixed

- icon class in the button text
 - incorrect selection field for the filter condition 'manage stock'
-

1.1.5

(2019-11-22)

Fixed

- Rename route "feed" to "mst_feed" (for prevent possible conflicts)
 - Undefined offset on Magento EE
 - chmod not permitted in Helper/To.php
-

1.1.3

(2019-10-29)

Fixed

- Installation issues on Magento 2.2.x
-

1.1.2

(2019-10-25)

Fixed

- Undefined offset 0 on Magento EE
-

1.1.1

(2019-10-25)

Fixed

- Magento Marketplace tests
-

1.1.0

(2019-10-11)

Improvements

- added new filter conditions 'Created At' and 'Updated At'
-

1.0.110

(2019-09-17)

Fixed

- fixed an issue with AM/PM format at the cron schedule list
 - fixed an issue with mass exporting templates at the grid
 - updated to eqp requirements
-

1.0.109

(2019-08-22)

Fixed

- Updated templates
 - Added notification about missing feed Filename before generation
-

1.0.108

(2019-08-12)

Improvements

- Added new 10 templates
 - Minor templates updates
-

1.0.107

(2019-08-08)

Improvements

- Coding Standart Refactoring

Fixed

- minor template issue
-

1.0.106

(2019-06-05)

Improvements

- added new template 'Amazon Seller Central (example)'
 - added an ability to duplicate Dynamic Attributes at the 'Action' column
-

1.0.105

(2019-05-28)

Improvements

- added an ability to export Multi Source Inventory sources (Magento > 2.3)
-

1.0.104

(2019-05-22)

Fixed

- fixed an issue with the tracking orders at the Feed Reports
 - fixed an issue with incorrect exporting 'Reviews Count' attribute
-

1.0.103

(2019-04-10)

Fixed

- fixed an issue with the incorrect filtering rule condition 'Status(Parent Product)'
 - fixed an error at the feed Preview page: 'Exception: for tag was never closed'
-

1.0.102

(2019-03-28)

Improvements

- added HMTL Entity Decode modifier
-

1.0.101

(2019-03-19)

Fixed

- Issue with filtration by Multi Select attribute
-

1.0.100

(2019-02-22)

Fixed

- fixed an issue with the incorrect store view export via cron schedule generation
-

1.0.99

(2019-02-07)

Fixed

- fixed an issue with the incorrect enclosure exporting in the csv and txt feeds

Improvements

- added new attribute 'Gallery image collection'
-

1.0.98

(2019-02-04)

Fixed

- Changed column length for dynamic category
-

1.0.97

(2019-01-10)

Fixed

- fixed an issue with the generation feeds by cron after new year
-

1.0.96

(2019-01-09)

Fixed

- fixed an issue with the incorrect output of the 'Is in Stock' attribute
-

1.0.95

(2018-12-28)

Improvements

- Ability to filter products without category
 - added new modifiers
-

1.0.94

(2018-12-20)

Fixed

- fixed an issue with incorrect output after applying "Append" modifier to the numbers
 - fixed an issue with the empty output instead of zero value at the Dynamic Attribute
-

1.0.93

(2018-12-14)

Improvements

- added 'Final Price with Tax' product attribute for export
- updated Google Shopping Review template

Fixed

- fixed an issue with the creating new feed using Empty Template
-

1.0.92

(2018-12-11)

Fixed

- fixed an issue with the incorrect work of Math Modifiers: Addition, Substraction, Multiplication, Division
 - added new text Modifiers
-

1.0.91

(2018-11-30)

Improvements

- M2.3 support
-

1.0.90

(2018-11-29)

Fixed

- fix of the problem with the data serialization for magento versions < 2.3
-

1.0.89

(2018-11-28)

Fixed

- support of magento 2.3
-

1.0.88

(2018-11-23)

Fixed

- export full path for small image pattern
-

1.0.87

(2018-11-20)

Improvements

- added "Product ID" filter condition
 - added "Qty of children in stock products" attribute for export
-

1.0.86

(2018-11-19)

Fixed

- getLevel() issue
-

1.0.85

(2018-11-15)

Improvements

- Added new templates: "Facebook Dynamic Ads", "Domodi", "Marktplaats"
 - Improved "Google Shopping Review" template according to the new XML Schema requirements
 - Updated existing templates
-

1.0.84

(2018-11-09)

Improvements

- added 'Stock Status' product attribute

Fixed

- fixed an issue with the incorrect categories export according to their position
-

1.0.83

(2018-10-25)

Fixed

- fixed an issue with the incorrect timezone export date at the 'time' pattern
 - fixed an issue with the exporting empty feed if any filter conditions are applied
-

1.0.82

(2018-10-02)

Fixed

- Possible issue with sorting
-

1.0.81

(2018-09-11)

Improvements

- Performance
-

1.0.80

(2018-09-06)

Improvements

- Added memory/iteration time to CLI command
-

1.0.79

(2018-09-05)

Fixed

- Feed emails are not working
- When using filters out of stock products excluded from feed
- Issue with truncate filter (multibyte strings)
- Fixed error "A technical problem with the server created an error. Try again to continue what you were doing. If the problem persists, try again later."

Improvements

- Created separate option group for Category Mappings
-

1.0.77

(2018-08-09)

Improvements

- improved the "Is Salable" filter condition to exclude children products according to their parent salable status

Fixed

- fixed an issue with the incorrect export of the category path by levels
-

1.0.76

(2018-07-24)

Fixed

- fixed an issue with the incorrect Review Summary Rating export
 - fixed an issue with the SQL errors during Magento installation
-

1.0.74

(2018-07-23)

Fixed

- fixed the Eval function Error via using the Dynamic Variables
-

1.0.73

(2018-07-17)

Fixed

- added compatibility of Notification Email classes with the Magento 2.1.x versions
-

1.0.72

(2018-07-16)

Fixed

- fixed an issue with the incorrect feeds delivery and export via CLI commands
 - fixed an issue with the sending notification emails
-

1.0.71

(2018-07-10)

Fixed

- Pie chart is not displayed: set default column for chart
-

1.0.70

(2018-06-28)

Improvements

- added 'Final Price' filter condition
- improved 'Is Salable' filter condition

Fixed

- fixed issues with processing image and image sizes filter conditions
 - fixed an issue with the incorrect and missing enclosures in the txt and csv feeds
-

1.0.69

(2018-06-04)

Fixed

- fixed an issue with the incorrect feed schedule time execution by cron
-

1.0.68

(2018-03-19)

Fixed

- fixed an issue with incorrect filtering of "Stock Availability" condition ()
-

1.0.67

(2018-03-12)

Fixed

- fixed an issue with incorrect filtering products by category ids

- Reports are not visible in 'developer' mode
-

1.0.66

(2018-02-26)

Fixed

- Report is not displayed (affects since 1.0.65)
-

1.0.65

(2018-02-23)

Improvements

- compatibility with latest version of Mirasvit module Reports

Fixed

- fixed an issue with the showing store categories on the Category Mapping page
-

1.0.64

(2018-02-15)

Improvements

- Added filter condition to export products by amount of in stock children
-

1.0.63

(2018-02-02)

Bugfixes

- fixed an issue with the files locking on Windows #23
-

1.0.62

(2017-12-15)

Fixed

- Issue with yaml parsing library

1.0.61

(2017-12-07)

Fixed

- Fixed issue with deleting dynamic variables from the mass action grid
-

1.0.60

(2017-11-22)

Feature

- Feed generation report

Improvements

- Display number of generated items in the feed
-

1.0.59

(2017-11-17)

Fixed

- Issue with price export by cron
-

1.0.58

(2017-10-12)

Fixed

- Fixed an issue with the exploding array instead of string
-

1.0.57

(2017-10-12)

Fixed

- Fixed an issue related to incorrect export of parent product values in Magento Enterprise edition
 - Fixed an issue with the exploding array instead of string
-

1.0.56

(2017-09-28)

Improvements

- Compatibility with Magento 2.2
-

1.0.55

(2017-06-27)

Fixed

- Dynamic attribute conditions do not work properly when admin user has custom permissions
- Solve 'Duplicate entry' error occurred on a filtration step

Improvements

- Modifier to remove all non-utf8 characters
 - Ability to limit gallery collection
-

1.0.54

(2017-06-13)

Fixed

- Issue with image resolver
-

1.0.53

(2017-05-05)

Fixed

- Product.price to product.finalPrice for google templates
- Fixed a feed generation error via CLI - "Area code is already set"
- Fixed feed generation error via command Command Line Interface - "Area code is already set"

Features

- Ability to use Product attributes in the Google Analytics tabs
-

1.0.52

(2017-03-30)

Improvements

- Changed clicks logging mechanism to more stable

Fixed

- CI for import/export
-

1.0.51

(2017-03-24)

Features

- Import/export Feed entities

Documentation

- updated documentation
-

1.0.50

(2017-03-21)

Fixed

- Possible issue with filtration
-

1.0.49

(2017-02-27)

Fixed

- Fixed an issue with patterns preview
-

1.0.48

(2017-02-27)

Improvements

- Changed report version to 1.2.*

Fixed

- Fixed an issue with dynamic attributes

1.0.47

(2017-02-20)

Improvements

- Ability to export swatches values
-

1.0.46

(2017-02-14)

Improvements

- Added dedicated cron group for feed generation process

Fixed

- Fixed an issue with Plain filter
-

1.0.45

(2017-02-01)

Fixed

- Fixed an issue with special char "|" in filters
-

1.0.44

(2017-01-20)

Improvements

- Changed logic of exporting configurable product attributes. If configurable product return empty value, module select values for child products
-

1.0.43

(2017-01-19)

Fixed

- Fixed an issue with thumbnail images
-

1.0.42

(2017-01-11)

Fixed

- Fixed an issue with json
-

1.0.41

(2017-01-10)

Improvements

- Ability to export all attributes (CSV header XALL)
 - Added json filter `{{ product.gallery | json }}`
-

1.0.40

(2017-01-09)

Fixed

- Fixed an issue with tax rate preview
-

1.0.39

(2017-01-06)

Improvements

- Implemented lock mechanism for prevent parallel feed generation (CLI)
-

1.0.38

(2017-01-06)

Fixed

- Fixed an issue with category mapping (edit page)
-

1.0.37

(2017-01-06)

Fixed

- Fixed an issue with mapping
-

1.0.36

(2017-01-03)

Fixed

- Fixed an issue with nested category mapping
-

1.0.35

(2017-01-03)

Fixed

- Added symfony/yaml to depends

Improvements

- Ability to place category taxonomy files to pub/media/feed/mapping
-

1.0.33

(2016-12-21)

Fixed

- Fixed an issue with filtration by category (simple products that not visible in catalog)
-

1.0.32

(2016-12-15)

Improvements

- Improved performance for csv feed edit page
-

1.0.31

(2016-12-14)

Fixed

- Fixed an issue with current date
-

1.0.30

(2016-12-09)

Improvements

- Compatibility with M2.2
-

1.0.29

(2016-12-06)

Fixed

- Fixed an issue with pattern output for dynamic attributes
-

1.0.28

(2016-12-05)

Fixed

- Fixed an issue with parent selector in dynamic attributes
-

1.0.27

(2016-11-11)

Fixed

- Fixed an issue with filtration by Yes/No attributes
-

1.0.26

(2016-11-04)

Fixed

- Fixed an issue with Status filter
-

1.0.25

(2016-11-02)

Fixed

- Changed crontab
 - Fixed an issue with feed delivery (SFTP)
-

1.0.24

(2016-10-21)

Fixed

- Fixed an issue with compilation
-

1.0.23

(2016-10-20)

Fixed

- Fixed an issue with deleting dynamic attribute
 - Fixed an issue with filter by stock quantity
-

1.0.21

(2016-10-12)

Improvements

- Use the same font-awesome.min.css for all extensions
-

1.0.20

(2016-10-12)

Fixed

- Fixed an issue with dynamic attributes
-

1.0.19

(2016-10-07)

Fixed

- Fixed an issue with images url
 - Fixed an issue with wrong product url (multi-store configuration)
 - Select product category depend on current store
-

1.0.18

(2016-09-06)

Improvements

- Export category with maximum level that related with product

Fixed

- Fixed an issue with category mapping
-

1.0.17

(2016-09-05)

Fixed

- Fixed possible issue with generation process
-

1.0.16

(2016-08-23)

Fixed

- Fixed an issue with delivery button
-

1.0.15

(2016-08-23)

Fixed

- Fixed a possible issue with filtration
-

1.0.14

(2016-07-18)

Fixed

- Fixed an issue with reports
 - CI
-

1.0.13

(2016-06-24)

Fixed

- Compatibility with Magento 2.1
 - Fixed an issue with validation dynamic attribute rules
-

1.0.12

(2016-06-17)

Fixed

- Fixed an issue with dynamic attribute values
-

1.0.11

(2016-05-25)

Fixed

- Fixed an issue with empty attribute value, if attribute contains numbers
-

1.0.10

(2016-05-25)

Features

- Implemented autocomplete for category mapping

Improvements

- Ability use dynamic variables in { % for % } cycle

Fixed

- Fixed an issue with dynamic attribute conditions (multi-select)
-

1.0.9

(2016-04-11)

Fixed

- Fixed an issue with menu
- ACL for dynamic variables

1.0.8

(2016-04-07)

Improvements

- Integration tests for Dynamic Variables
- Dynamic Variables
- Offset attribute for "for" cycle

Fixed

- Remove Root Category from categoryCollection method
- Fixed an issue with capture tag

Documentation

- Dynamic Variables
-

1.0.7

(2016-04-01)

Fixed

- Fixed an issue with cross-browsing ajax requests
-

1.0.6

(2016-04-01)

Fixed

- Fixed an issue with feed generation (not all products for feed with >100K products)
- Styles

Improvements

- Capture tag
- Ability to defined product ids for feed preview (stored in cookies)
- Dynamic Attributes
- Reports
- Added ability to export Related/CrossSell/UpSell products

1.0.5

(2016-03-14)

Improvements

- Added random param to export/progress url for prevent request caching
- Added ability to export Related/CrossSell/UpSell products
- Added 2 new filters: mediaSecure, mediaUnsecure
- Export all images with direct link (without cdn if defined)
- Updated "Save" button for Templates, Filters and Category Mapping
- Added tax rate resolver "{{ product.tax_rate }}"
- Clean feed history by cron (leave history for last 3 days)

Fixed

- Updated "Google Shopping" template
 - Fixed an issue with cron job scheduling
-

1.0.4

(2016-02-17)

Improvements

- Improved feed generation process by cron, plus added integration tests for cron job
- Improved feed history (CLI and manual export process)
- Added new filters inclTax, exclTax

Fixed

- Fixed an issue with broken link to Category Mapping in top menu
 - Fixed an issue with gallery images (for cycle)
 - CouplingBetweenObjects
 - Fixed an issue with rounding prices when apply filter inclTax, exclTax
 - Fixed an issue with trimming chars in {for} cycle
 - Fixed an issue with removing liquid filters durign change expression (xml)
 - Fixed an issue with mysql error at feed preview on empty catalog
 - Fixed an issue with exclude tax calculations
 - Fixed an issue with wrong js mapping (reports.js)
-

1.0.3

(2016-02-08)

Improvements

- Added new filters for urls: "secure" and "unsercure"
 - Added prices including tax
-

1.0.2

(2016-02-07)

Improvements

- Improved xml highlighting
- Split product resolve to few files depends on product type
- Added date filter
- Added ability to select associated products {{ product.associatedProducts }}

Fixed

- Integration tests
 - Fixed an issue with select/multiselect attribute values
 - Fixed an issue related with wrong loop length in liquid cycles
 - Fixed an issue with "No elements to pop"
 - Minor issue
-