

Product Labels Manual

Getting Started

Welcome to the **Product Labels** documentation.

You will find everything you need to set up your **Product Labels** so that you can support nearly any marketing strategy with attractive labels on your products' images.

Go ahead, dive in!

Firstly, please check out our extension in the [My Downloadable Products](#) section of our store. Learn [how to install extension](#), and proceed with [Quick Start](#), which will guide you to set up your product labels.

How to install the extension

1. Back up your store database and web directory.
2. Log in to the SSH console on your server and navigate to the root directory of the Magento 2 store.
3. Copy the 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_CatalogLabel` for enable extension.
5. Run the command `php -f bin/magento setup:upgrade` to install extension.
6. Run the command `php -f bin/magento cache:clean` for clean 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
```

Install the extension for Hyva

1. Run these commands

```
composer require hyva-themes/magento2-mirasvit-catalog-label
```

2. Enable installed Hyva modules:

```
bin/magento module:enable Hyva_MirasvitCatalogLabel
```

3. Update Magento database schema and data with the command:

```
bin/magento setup:upgrade
```

4. Compile the code of the installed extension:

```
cd /path/to/theme/web/tailwind/  
npm run build-prod
```

5. Run the command below to deploy static content and clean the cache:

```
cd /path/to/project/  
rm -rf pub/static/frontend/*; rm -rf pub/static/backend/*; rm -rf var/view_preprocessed/*  
php -f bin/magento setup:static-content:deploy  
bin/magento cache:flush
```

6. Apply required customizations to your theme as described [here](#)

Quick Start

Our Product Label extension is a simple yet powerful tool for marking your product according to your marketing and sale policy.

It is quite intuitive and works our-of-box, but there are some things to be tuned up.

1. Start with [Placeholders](#). Placeholders are blocks that group labels and describe how and where they should be displayed.
2. Proceed with [Templates](#). There you can create reusable styled templates for labels.
3. Create [Labels](#), which will display certain attribute values or combined conditions, allowing you to plan complex campaigns and mark promoted products.
4. Check your labels' appearance.
5. Check [Configuration tips and tricks](#) for configuration tips and managing to display labels in custom themes.

This should be a good start.

Refer to the corresponding sections of this manual to learn more.

Product Labels Configuration

General configurations of the Product Labels extension can be found at **Stores -> Configuration -> Mirasvit Extensions -> Catalog Label**.

This section features only one setting:

- **Flush dependent pages cache after creating new product** - When this option is set to **Yes**, all pages where this product are displayed or mentioned will be automatically flushed (i.e. their cached versions will be purged). It allows you to quickly update part of your store when a new arrival comes without the entire cache regenerating. It is extremely useful if you have a very complex store, and/or caching is a serious matter of productivity.
- **Apply labels for child products of configurable products** - If enabled, the extension will assign labels for child products of configurable products. Labels will be updated after swatch option(s) are selected for configurable products. Products filtered by the swatch attribute will display labels for corresponding child products. Reindex the Product Label Index after changing this setting.

Note

Enabling this option will increase the reindex time of the Product Label Index.

- **Discount output precision** - Number of digits displayed after comma for product discount variable in labels
- **Date format** - If not empty will format attributes with "Date" and "Date and Time" catalog input types
- **Don't use labels on pages** - list of full action name or URL patterns for pages, where labels shouldn't be displayed.

If you wish to fit your product labels to your theme, refer to the [Labels](#) section.

Manage Placeholders

Placeholders are the most basic building blocks of our extension. They should be created before any [Label](#) is created.

Placeholder are designed to group labels by positions and manage their appearance in the store.

All of them are located at **Marketing -> Promotions -> Product Labels -> Manage Placeholders** section. From there, you can edit, remove, or create a new Placeholder.

How to Create New Placeholder

Visit **Marketing -> Promotions -> Product Labels -> Manage Placeholders** and press the **Add New** button. You will be brought to the Creation Page, as shown on the screenshot.

General Information

Title *

Identifier

Is Active

 Yes

Limit of Labels

Leave empty or set 0 to disable limit

Labels Direction *

Select...

For multiple labels in one placeholder

Position *

Top Left

Top
Center

Top Right

Middle
Left

Middle
Center

Middle
Right

Bottom
Left

Bottom
Center

Bottom
Right

This page contains only one section, **General Information**, which consists of the following fields:

- **Title** - a sensical title of placeholder.
- **Identifier** - The identifier (code) of placeholder that is used to display labels, bound to this placeholder at required positions.
- **Is Active** - indicates whether this placeholder is active, and the bound labels are eligible for display.
- **Limit of Labels** - The number of labels that should be displayed in placeholder. If number of labels assigned to the placeholder is higher than the limit (n) only first n labels will be displayed (sorted by labels' sort order). Leave this field empty or set 0 to disable limitation.
- **Labels Direction** - describes how labels should be organized in the placeholder of more than one label assignd to the placeholder:
 - **Horizontal** - labels are shown in a row (horizontally).
 - **Vertical** - labels are shown in a column (vertically).
- **Position** - allows choosing where labels should be displayed. This field has options for two types of positioning:
 - **Automatically positioned** * - options for positioning labels over the product image.

Example

The Position is calculated from two directions: **vertical** and **horizontal** (not related to Labels Direction).

Vertical are:

- **Left**
- **Center**
- **Right**

Horizontal are:

- **Top**
- **Middle**
- **Bottom**

If you wish to position the label to the upper right corner, select **Top Right** option. Only one placeholder for each of these positions can be created.

- **Manually positioned** - allows to position label wherever you like by putting code snippet in coresponding place in template file. This type of labels has only one option - **Manual**. You can create as many manually positioned placeholder as needed. The code snippet for this placeholder that should be inserted in template file will be generated after the placeholder will be saved. For additional information regarding manually positioned labels check [Display Labels with manual position](#)

Manage Templates

Templates helps to create and reuse styled label templates which can be reused for as many labels as needed. Templates help to configure similar styled labels without duplicating same styles from label to label and speed-up the configuration of labels themselves.

Templates are not required part of the extension. Labels can be created without templates. In this case labes

will be rendered the same way they were rendered before Templates were introduced.

All templates are located at **Marketing -> Promotions -> Product Labels -> Manage Templates** section. From there, you can edit, remove, or create a new Placeholder. The extension provides a few preconfigured templates which can be used for labels and also edited or duplicated for creating new templates.

How to Configure Templates

To create a new template, press the **Add Template** button on the **Manage Templates** page.

Or you can edit or duplicate an existing template by choosing a corresponding option from the action column of a particular template.

The form for configuring the template is split into three sections:

- **General** - configuration data of the template

General

Internal Name *

test t2

Identifier

test-t2

HTML Template

```
1 <div class="t2">
2   <span>{{var label.title}}
3 </div>
```

List of variables:

{{var label.title}} - label text

{{var label.description}} - label description

{{var label.image_url}} - label image url

{{var label.image}} - label image

with label text

Styles

```
1 .t2 {
2   width: 75px;
3   height: 75px;
4   background-color: #f0f0f0;
5   display: flex;
6   justify-content: center;
7   align-items: center;
8   border-radius: 10px;
9   text-align: center;
10 }
```

this section contains the following fields:

- **Internal Name** - name of the template used in backend only.
- **Identifier** - templates code, will be generated from template's name if not set during creation of the template
- **HTML Template** - the HTML code of the template. The next variables can be used in this field:
- **{{var label.title}}** - label text
- **{{var label.description}}** - label description, can be used as value for title attribute
- **{{var label.image_url}}** - label's image URL
- **{{var label.image}}** - label's image element with label image as background and with label text. For more information regarding using this variable check [Using image variable in templates](#)
- **{{var label.url}}** - label URL

Note

If you want to use `<a>` tag inside label you need to wrap it in `<object>` tag to avoid layout issues on product listing pages

- **Styles** = Plain CSS for styling the template.
- **Data for testing (will not be saved)** - fields with sample data to check how the potential label will look if it will use current template.

Data for testing (will not be saved)

Label Title

Example Text

Label Description

Example Description

Label image URL

This section has the next fields:

- **Label Title** - title of the label (label's text)
- **Label Description** - description of the label (can be used as the value for the title attribute in HTML template)
- **Label Image URL** - the URL for the label's image.
- **Preview** - the section where the actual look of the template will be displayed. It updates automatically while fields of the form are edited.

Manage Labels

Product Labels are markers which are placed according to [Placeholders](#) they assigned to at Catalog and Product pages. Using these marks, you can easily promote your new product or boost sales during marketing campaigns.

All Labels are located at the **Marketing -> Product Labels -> Manage Labels** section. From there, you can edit, remove, or create a new label.

How to Create New Label

Visit the **Marketing -> Product Labels -> Manage Labels** section and press the **Add Label** button. Labels creation is split into two Stages, the first of which selects the Label Type, and the second gives details of their application.

The first stage has only one section - **General** - and consists of the following fields:

GENERAL INFORMATION

General

General

Internal Name *

Type *

Is Active

Active From

Active To

Appearance *

Visible In *

- **Internal Name** - a logical internal title of the label.
- **Type** - a type of label which needs to be created.
 - **Attribute** - a simple label that is displayed when the product has a selected attribute value.
 - **Rule** - a complex label that uses conditions to determine when a label needs to be applied.
- **Attribute** - list of attributes that can be used to create labels for. Applicable only if the **Type** field is set to **Attribute**
- **Is Active** - indicates whether the label is active and ready to apply.
- **Active From, Active To** - time bounds when a label should be applied. Very useful for planning Sales campaigns.
- **Appearance** - list of options of how the label should be configured and displayed:
 - **Product List Label** - configure and display label only for product listings such as category pages or widgets.
 - **Product View Label** - configure and display label only for product view page
 - **Same Label for List and View** - configure and display the same label design for product listings and product view pages.
 - **Separate Labels for List and View** - configure and display separate designs for product listings and product view pages.
- **Visible In** - stores where the label needs to be applied.
- **Customer Groups** - customer groups who are eligible to see labels.
- **Sort Order** - priority order in which labels are applied to products.

Once the first stage configurations are finished and the label is saved fields **Type** and **Attribute** no longer can be changed. Changing the **Appearance** field after configuring the label can result in design configurations loss.

The next stage contains detailed information about label displaying. The **General** section is displayed for both types of Label, the fields are the same as at the first stage.

Each label has a group of label's [design fields](#). The number of design fields' groups and the way they organized depends on values of the **Type** and **Appearance** fields.

Attribute-based Labels Setup

If **Attribute** was selected in the previous stage of **Type** and the **Attribute** field is selected as well in the second stage the **Gallery** section will appear.

GENERAL INFORMATION

General

Gallery

List of allowed product variables:

[br] - New Line

[price] - Price

[final_price] - Final Price

[special_price] - Special Price

[sku] - Sku

[stock_qty] - Stock Qty

[discount_amount] - Discount Amount

[discount_percent] - Discount Percent

[special_price_dl] - Days Left For Special Price

[new_days] - Product New Days

[attr | attributeCode] - Value of the product attribute with the

All-Weather

Cold

Cool

Indoor

Mild

Rainy

Spring

Warm

Windy

Wintry

The **Gallery** field will contain a bunch of fieldsets the number of which will be equal to number of options of the selected attribute. Each fieldset will contain one or two groups of the [design fields](#) depends on the value of the **Appearance** field.

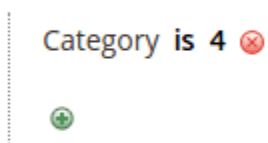
Rule-based Labels Setup

If **Rule** was selected as the **Type** in the previous stage, the second stage will feature two additional tabs: **Conditions** and **Design**.

At the **Conditions** tab, you will see a conditional block, which allows you to filter products to which labels should be applied, as shown on the screenshot:

Conditions (leave blank for all products)

If **ALL** of these conditions are **TRUE** :



These conditions applied for 14
product(s)

All conditionals should be enclosed in the global mode block. They have four possible global modes of applying, shown in the special header `If *[apply mode]* of these conditions are *[validation mode]*:`

Applying modes define, when rule shall be triggered:

- **ALL** - implies that the rule will be executed only when all conditions are strictly met;
- **ANY** - implies that the rule will be executed only when one or more (but not all) of conditions are met;

Validation modes define which result can produce each condition to be registered as "met":

- **TRUE** - implies that the conditions need to be valid.
- **FALSE** - implies that the conditions need to be invalid.

You can also define multiple nested mode blocks by selecting the **Conditions Combination** option. These modes allow for creating flexible condition sets to satisfy a policy of any complexity.

Once you have selected global mode (or left it as default), press the green (+) button, and select a condition type. There are three categories of available conditions:

- **Product**

Contains three base attributes which are used for identifying products:

- **Attribute Set** - a set of attributes defining a particular type of product (e.g. jewelry, for example). All these sets can be seen at **Stores -> Attributes -> Attribute Set**.
- **Category** - one or more categories where the product needs to be registered.
- **SKU** - direct value, or pattern (using **contains** operator), that needs to be in the product's SKU code.

- **Product Attribute**

Contains all attributes, defined at **Stores -> Attributes -> Product**, the most interesting of which are:

- **New** - allows to check whether the product is New.
- **Format** - a digital form of product (useful for downloadable products)

- **Product Additional**

Contains special attributes that can be used for building complex campaigns:

- **Created At** - checks how many days ago the product was added to the store.
- **Updated At** - checks how many days ago the product was last changed.
- **Percent Discount** - analyzes which discount was applied to this particular product.

Note

Only discounts that have been applied directly to the products are analyzed. They are created at **Marketing -> Catalog Price Rules**.

- **Price - Final Price** - is the final price, calculated after the application of all discounts and special price rules, which customers will actually pay.
- **Quantity** - the quantity of a product which is currently in stock. This is useful when you need to mark the product as sold out.
- **Set As New** - checks whether the product was set as new (including old products, returned as new arrivals).

At the **Design** tab will be one or two groups of the [design fields](#) depends on the value of the **Appearance** field.

Configuring Label's design

Each type of a label has the next group of fields to manage the desing of the label:

Image

Upload

Reset Label

Allowed file types: png, gif, jpg, jpeg.

Title

Variables can be used in this field

Description

Variables can be used in this field

Url

Placeholder

not configured	not configured	Test placeholder	Manual 1 (manual)
not configured	not configured	not configured	Manual 3 (manual)
not configured	not configured	not configured	

Template



- **Image** - uploads label which will be added to the product image.
- **Title** - sets title on the label. [Variables](#) can be used in this field.
- **Description** - sets label description. [Variables](#) can be used in this field.
- **Url** - sets URL key of label.
- **Placeholder** - assign the label to placeholder. Read more at the [Placeholders](#) section.
- **Template** - choose the template for label. Templates have previews which updating automatically while design fields are edited. Read more at the [Templates](#) section.
- **Style** - additional CSS styles that should be applied to the label.
- **Reset Label** - removes current label's design.

Using Variables

The extension provides you with the ability to use variables in labels' title and description.

The list of available variables:

- **[br]** - add new line
- **[price]** - product price
- **[final_price]** - product final price
- **[sku]** - product SKU
- **[stock_qty]** - product's quantity in stock
- **[discount_amount]** - absolute discount (in store currency)
- **[discount_percent]** - relative discount (in percent)
- **[special_price_dl]** - days left for special price
- **[new_days]** - days since the product was added to the store or since the date "Set Product As New From". The variable will return an empty string if the date "Set Product As New To" is less than the current date.
- **[attr|code]** - value of the product's attribute. Replace the **code** with the actual attribute code. For attributes of the type **multiselect** the variable will be replaced with all values of that attribute (comma-separated).

Applying labels

The extension has its own index to store label-product relations.

If the index is set to **Update on Save** these relations will be updated any time you save the label. The extension will reflect on changing products in the same way - during saving the product from the admin panel.

If the index is set to **Update by Schedule**, label - product relations update after changing products data will be handled by Magento's indexation. If labels are changed the reindex of label-product relations will be performed on a cron basis every 6 hours (full reindex of all created labels). If you'd like to apply label changes immediately, you'll need to press the **Apply Label** button on the label's edit page after saving the label. This button is only visible when the index is set to **Update by Schedule**

☰ Label "Bags Sale" ▾

← Back

GENERAL INFORMATION

General

Rule

Design

Internal Name * Bags Sale

Type * Rule ▾

Is Active Yes

Active From



Active To



Appearance * Same Label for List and View

Changing the appearance mode will configured label views of this label

Configuration tips and tricks

This section describes some common tips and tricks to help configuring the extension to fit the store requirements.

Updating to version 2.0.0

The major update of the extension includes interfaces and data structure changes as well as the changes to the way labels are outputting to the store front.

All previously configured labels should migrate automatically to the new version without issues.

However, after the upgrade some additional adjustments might be needed to make labels to be shown as they were before the upgrade.

In most cases adjustment may be required only in labels styles as they no longer are inline styles.

If your store used customization for displaying labels please check the [Display Labels within a Custom Theme](#) section of this page and update the code in your templates. Old customization code will not produce errors in

store front but will not display any labels.

Display Labels with manual position

There 2 types of placeholders that can be configured in the extension: positioned and manual.

In most cases positioned placeholders manages to display labels automatically.

If for some reason positioned labels are not displayed in the store front check the [Display Labels within a Custom Theme](#) section on this page.

Labels with manual position allows you to display labels at almost any place near product card. The only thing that required is to put the following code in the template file wherever the labels should be displayed.

Example

```
<?php
    echo $block->getLayout()->createBlock('\Mirasvit\CatalogLabel\Block\Product
        ->setProduct($_product)
        ->setPlaceholderByCode(placeholder_code)
        ->setType(view_type)
        ->toHtml());
?>
```

where:

- `view_type` should be 'list' or 'view' corespondingly to where this code is placed
- `placeholder_code` the identifier of the manually positioned placeholder
- `$_product` contains a current product object

Display Labels within a Custom Theme

Some themes have different layouts that display products in their own unique formats. In this case, you need to adjust templates manually. Take a look at the [real example](#) below.

Assuming that `$_product` contains a current product object, you will need to insert the following code to make our extension place positioned Labels.

Example

```
<?php
    echo $block->getLayout()
        ->createBlock('\Mirasvit\CatalogLabel\Block\Product\Label\Renderer
        ->setProduct($_product)
        ->setType(view_type)
        ->toHtml());
?>
```

where `view_type` should be 'list' or 'view' corespondingly to where this code is placed

A Real Example

Consider the standard template

Magento/Catalog/view/frontend/templates/product/list.phtml, which contains the following code:

```
<?php foreach ($_productCollection as $_product): ?>
    <?php /* @escapeNotVerified */ echo($iterator++ == 1) ? '<li class="item pr
    <div class="product-item-info" data-container="product-grid">
        <?php
        $productImage = $block->getImage($_product, $image);
        if ($pos != null) {
            $position = ' style="left:' . $productImage->getWidth() . 'px;'
                . 'top:' . $productImage->getHeight() . 'px;";'
        }
        ?>
        <?php // Product Image ?>
        <a href="<?php /* @escapeNotVerified */ echo $_product->getProductUrl()
            <?php echo $productImage->toHtml(); ?>
        </a>
        <div class="product details product-item-details">
            ...
    <?php endforeach; ?>
```

We will insert our label overlay code right before the closing anchor tag <a> which is a wrapper for product image:

```
<?php foreach ($_productCollection as $_product): ?>
    <?php /* @escapeNotVerified */ echo($iterator++ == 1) ? '<li class="item pr
    <div class="product-item-info" data-container="product-grid">
        <?php
        $productImage = $block->getImage($_product, $image);
        if ($pos != null) {
            $position = ' style="left:' . $productImage->getWidth() . 'px;'
                . 'top:' . $productImage->getHeight() . 'px;";'
        }
        ?>
        <?php // Product Image ?>
        <a href="<?php /* @escapeNotVerified */ echo $_product->getProductUrl()
            <?php echo $productImage->toHtml(); ?>

            <!-- Labels code for custom themes -->
            <?php
                echo $block->getLayout()
                    ->createBlock('\Mirasvit\CatalogLabel\Block\Product\Label')
                    ->setProduct($_product)
                    ->setType('list')
                    ->toHtml();
            ?>
        </a>
        <div class="product details product-item-details">
            ...
    <?php endforeach; ?>
```

This code will create a block which will overlap the image, creating a place for our label.

The same approach can be applied to other Magento 2 custom themes.

Using image variable in templates

The HTML Template field of labels template form allows you to use the next variables

- `{{var label.title}}`
- `{{var label.description}}`
- `{{var label.image_url}}`
- `{{var label.image}}`
- `{{var label.url}}`

While first three variables are self-explanatory the last one - `{{var label.image}}` - can be confusing.

The `{{var label.image}}` variable represents a code snippet of the ready-to-use image block with label text. So the value of this variable is a HTML code that uses `{{var label.title}}` and `{{var label.image_url}}` inside it and has some predefined inline styles.

The raw value of this variable looks like here:

```
<div class="label-image" style="background:url({{var label.image}}); background-size:{{var label.image_width}}px {{var label.image_height}}px; background-repeat:repeat-x; background-color:{{var label.image_color}};">
  <span class="label-title">{{var label.title}}</span>
</div>
```

where [image-width] and [image-height] are automatically calculated width and height of the image. Those are not variables and can't be used in the **HTML Template** field.

For further styling of the template that uses the `{{var label.image}}` variable, selectors like `div.label-image` and `span.label-title` can be used in the **Styles** field of the template edit form.

Note

If you want to use `<a>` tag inside templates you need to wrap it in `<object>` tag to avoid layout issues on product listing pages

Customizations for Hyva theme

In order to inject the Product Labels onto the product view you will need to override the **Magento_Catalog::product/view/gallery.phtml** template.

Add the below code after the gallery but still within the gallery container. The container should have CSS instruction `position:relative`. In Hyva theme it can be achieved by adding the `relative` class to the class list of the gallery container

```
<?php if ($labels = $block->getChildBlock('product.labels')): ?>
    <?= $labels->toHtml(); ?>
<?php endif ?>
```

In most cases product listings do not require additional adjustments. The only requirement is that the anchor tag `<a>` of the product item should have `position:responsive` CSS instruction as well.

If for some case in your custom theme labels are not displayed in product listings automatically it can be fixed by overriding the template file **Magento_catalog::list/item.phtml**

In `item.phtml` add the below code after the `` element but within the anchor `<a>` tag and add the class `relative` to the anchor tag

```
<?php if ($labels = $block->getChildBlock('labels')): ?>
    <?= $labels->setProduct($product)->setViewCode('list')->toHtml(); ?>
<?php endif; ?>
```

Command Line Interface

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

- `mirasvit:label:indexer --status` - Displays information about Product Label index
- `mirasvit:label:indexer --reindex-all` - Run full reindex of Product Label index
- `mirasvit:label:indexer --reindex-new` - Run reindex of Product Label index for recently created/updated products

How to upgrade an extension

To upgrade an extension, take the following steps:

1. Back up your store 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 the command `composer require mirasvit/module-cataloglabel:* --update-with-dependencies` to update the current extension with all 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. It will update all Mirasvit modules installed in your store.

4. Run the command `php -f bin/magento setup:upgrade` to install updates.
5. Run the command `php -f bin/magento cache:clean` for a clean cache.
6. 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 Extension

Temporary Disabling

To temporarily disable the extension, please follow the next steps:

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

Extension Removing

To uninstall the extension, please take the following steps:

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

Change Log

2.4.5

(2024-12-27)

Fixed

- Fixed the issue with displaying product variables on a label without a template
-

2.4.4

(2024-12-13)

Fixed

- Fixed the issue with the `[attr|attributeCode]` variables if they were added multiple times
-

2.4.3

(2024-12-10)

Fixed

- Fixed the issue with template preview when custom styles are applied
-

2.4.2

(2024-11-22)

Fixed

- Fixed the issue with discount calculation when catalog prices include tax
 - Fixed the issue with product variables on labels for child products of configurable products
-

2.4.1

(2024-11-14)

Fixed

- Fixed the issue with incorrect discount value in labels
-

2.4.0

(2024-11-06)

Improvements

- Improved and optimized reindexing time
-

2.3.11

(2024-10-24)

Fixed

- Fixed the issue when the "Apply labels for configurable products if one of the child products meets the conditions" configuration is enabled (TypeError: count(): Argument [#1]() (\$value) must be of type Countable|array, null given in mirasvit/module-cataloglabel/src/CatalogLabel/Block/Product/Label/Placeholder.php:172)
-

2.3.10

(2024-10-22)

Improvements

- Improved the "Apply labels for configurable products if one of the child products meets the conditions" configuration
-

2.3.9

(2024-10-11)

Improvements

- Added ability to display labels for configurable products if one of the child products meets the conditions

Fixed

- Fixed the issue with mass actions when applying a filter (all records are changed/deleted)
 - Fixed the issue with the "Price - Final Price" condition when taxes are enabled in the store
-

2.3.8

(2024-08-08)

Features

- Added ability to set the "Final Price" condition in a label rule
-

2.3.7

(2024-06-07)

Improvements

- Reindex improvements

Fixed

- Fixed the label positioning issue for products with multiple images on product view page
 - Fixed the issue with getting a discount percentage when the product price is set in a different currency
-

2.3.6

(2024-05-23)

Fixed

- Fixed the issue with double POST requests on saving labels (Firefox)
 - Fixed the issue with the mst_productlabel_label_rule_product table (indexing performed for all stores instead of only specified stores)
-

2.3.5

(2024-04-11)

Fixed

- Unnecessary empty placeholders when the 'Apply labels for child products of configurable products' option is disabled
-

2.3.4

(2024-03-21)

Fixed

- Fixed the issue with the "Customer Groups" configuration on product labels
 - Fixed the issue with CSS styles from label templates
-

2.3.3

(2024-03-14)

Improvements

- Security improvements
-

2.3.2

(2024-03-05)

Fixed

- Fixed the issue with labels not displaying on the homepage
-

2.3.1

(2024-02-23)

Improvements

- Ability to apply labels for simple products of configurable product
-

2.3.0

(2024-02-12)

Features

- Added the ability to hide labels on specified pages

Fixed

- Fixed the issue with labels on configurable products (the condition "Percent Discount" is set, and the "Display Product Prices In Catalog" setting is set to "Including Tax")

2.2.8

(2023-12-22)

Fixed

- Fixed the "Percent Discount" condition (multi-store)
-

2.2.7

(2023-12-08)

Improvements

- Run the "cataloglabel" cron every day at 00:45
 - Apply the "cataloglabel" cron only if the product label index is set to "Update by Schedule". Use the "mirasvit:label:emulate:cron" command to run the cron regardless
-

2.2.6

(2023-12-01)

Features

- Added the "Active From" and "Active To" columns to the "Manage Labels" admin grid

Improvements

- Optimized product label indexing

Fixed

- Fixed the issue with the label conditions applying (multiple stores)
-

2.2.5

(2023-11-16)

Fixed

- Fixed the issue with labels on bundle products that have discounts
-

2.2.4

(2023-10-27)

Improvements

- Prevent the overlay of product label preview on the admin panel
-

2.2.3

(2023-10-23)

Fixed

- Fixed the type error when using a product attribute in a label (some cases)
-

2.2.2

(2023-10-04)

Fixed

- Fixed error on labels listing when incorrect styles added to label
-

2.2.1

(2023-09-07)

Features

- Added the ability to set the date format for the "Date" and "Date and Time" attribute types

Fixed

- Fixed the issue with the "Active From" and "Active To" label settings
-

2.2.0

(2023-08-08)

Fixed

- Fixed the issue with conditions by tier price
-

2.1.9

(2023-07-31)

Fixed

- Fixed the issue with reindexing labels on Magento EE (giftcard products)
-

2.1.8

(2023-07-11)

Fixed

- Fixed the issue with saving labels (PHP8)
-

2.1.7

(2023-07-06)

Improvements

- Changed the type of the column 'style' in the table 'mst_productlabel_label_display' to allow saving large CSS
-

2.1.6

(2023-06-30)

Fixed

- Issue with error on Label edit page for labels of the type 'Attribute'
-

2.1.5

(2023-06-22)

Fixed

- Fixed the issue with error 'Notice: Undefined offset: 1' when labels not indexed yet
-

2.1.4

(2023-06-15)

Fixed

- The issue with the Save button after changing the Appearance field on the Label edit page
-

2.1.3

(2023-06-09)

Improvements

- Ability to apply label manually when Product Labels index is set to 'Update by Schedule'
-

2.1.2

(2023-06-08)

Fixed

- Issue with Labels relative URLs in multistore when store code is added to store URL
-

2.1.1

(2023-06-05)

Fixed

- indexer 'Update by Schedule'
-

2.1.0

(2023-05-30)

Fixed

- variable for label URL in templates
-

2.0.9

(2023-05-22)

Fixed

- Fixed the issue with labels' discount variable value (some cases)
-

2.0.8

(2023-04-21)

Fixed

- Fixed the issue with attribute variable not always returning correct value
-

2.0.7

(2023-04-20)

Improvements

- SVG images in labels

Fixed

- Fixed the issue with cron tasks (Cron Job cataloglabel has an error: Argument 1 passed to Mirasvit\CatalogLabel\Model\Observer::apply() must be of the type bool, object given)
-

2.0.6

(2023-04-18)

Fixed

- Fixed the issue with error when placeholder is deleted but its code is used for manual labels
-

2.0.5

(2023-04-12)

Fixed

- Fixed the issue with stock condition (multistock)
 - Fixed the issue with labels for products which not present in the default store
 - Fixed the issue with labels' Visible In setting not saved
-

2.0.4

(2023-04-10)

Fixed

- Fixed the issue with uploading label images
-

2.0.3

(2023-03-28)

Fixed

- Fixed the issue with errors related to incorrect styles
-

2.0.2

(2023-03-20)

Fixed

- Console command return value
 - PHP 8.2 compatibility
-

2.0.1

(2023-03-17)

Fixed

- Composer conflict with old hyva compatibility module
-

2.0.0

(2023-03-15)

Features

- Templates for labels
- Positioning labels through placeholders
- Manually positioned placeholders

Improvements

- Module structure changed
- New DB tables
- Labels output logic changed
- "Show labels in custom themes" code changed

Fixed

- Several small bugs fixed

Other

- Labels created in previous versions of the extension will be migrated to the new version automatically but may require additional adjustments
 - Old "Show labels in custom themes" code should not produce errors but no longer shows labels. Check the user manual for updated code.
-

1.3.15

(2023-02-15)

Fixed

- Fixed the issue with discount rule
-

1.3.14

(2023-02-08)

Fixed

- Fixed the issue with the discount rule for configurable products
-

1.3.13

(2023-02-06)

Fixed

- Fixed the issue with Labels massAction change status
-

1.3.12

(2023-02-01)

Fixed

- Fixed the issue with error after deleting labels
-

1.3.11

(2023-01-31)

Fixed

- Fixed the issue with stock quantity condition
-

1.3.10

(2023-01-30)

Improvements

- Added support of Magento 2.4.6

Fixed

- Fixed the issue with product quantity condition
-

1.3.9

(2023-01-20)

Fixed

- Fixed the issue with the condition 'Set as New'
-

1.3.8

(2023-01-17)

Improvements

- New variables [special_price_dl], [new_days], [attr|code]
-

1.3.7

(2023-01-16)

Fixed

- Fixed the issue with stock status rule
 - Fixed the issue with new table columns not added in the database during the upgrade of the module
-

1.3.6

(2023-01-11)

Fixed

- Fixed the issue with labels affecting products sorting order
 - Fixed the issue with error while trying to edit labels of the type rule (PHP8.1, Magento_PricePermissions)
 - Fixed reindexing issue
-

1.3.5

(2023-01-09)

Improvements

- Reindexing improved
 - Small code changes
 - Removed unnecessary empty labels in frontend
-

1.3.4

(2022-12-26)

Fixed

- Fixed the issue with the error while creating a product
-

1.3.3

(2022-12-16)

Fixed

- PHP8.1 compatibility issue
-

1.3.2

(2022-12-16)

Improvements

- Small code changes.
-

1.3.1

(2022-12-13)

Features

- Variables in label title and description.
-

1.3.0

(2022-12-05)

Improvements

- Code quality improved.
- Admin UI updated.
- Reindex performance improved.
- Additional console command for reindex.

Fixed

- Fixed the issue with duplicated labels on product listings
-

1.2.9

(2022-11-15)

Fixed

- Fixed the issue with reindex on large stores
-

1.2.8

(2022-11-02)

Fixed

- Fixed the issue with the error 'Division by zero'
-

1.2.7

(2022-10-20)

Fixed

- Fixed the issue with the error on reindex (preg_split(): Passing null to parameter #3 () of type int is deprecated)
-

1.2.6

(2022-10-14)

Fixed

- Fixed the issue with the error 'Deprecated Functionality: preg_match_all(): Passing null to parameter #2 of type string is deprecated' (PHP8.1)
-

1.2.5

(2022-10-12)

Fixed

- Fixed the issue with percent discount rule for configurable products
-

1.2.4

(2022-09-19)

Fixed

- Strpos error on format text function
 - Added composite index
 - Indexing optimization (performance improvement)
-

1.2.3

(2022-08-22)

Fixed

- Corrected getImageType call
 - Indexing issue (duplicates in index)
-

1.2.2

(2022-07-28)

Fixed

- Performance issues

Improvements

- Added the mst_cataloglabel index to handle Product Labels
- Cronjob for Label Rules indexing is running once a day now

- Label Rules indexing can be forced by running `mst_cataloglabel reindex`
-

1.2.1

(2022-06-20)

Improvements

- remove `db_schema_whitelist.json`
-

1.2.0

(2022-05-23)

Improvements

- Migrate to declarative schema
-

1.1.26

(2021-12-28)

Fixed

- Display labels on the catalog page
-

1.1.25

(2021-11-26)

Fixed

- `getDisplay` issue
-

1.1.24

(2021-11-25)

Improvements

- Speed up the `getDisplay()` method
-

1.1.23

(2021-08-31)

Improvements

- Added "Is Salable" product rule
-

1.1.22

(2021-03-02)

Fixed

- Apply percent discount rule issue
-

1.1.21

(2021-03-02)

Fixed

- Re-saving dates problem
-

1.1.20

(2020-10-19)

Fixed

- Small spelling fixes
-

1.1.19

(2020-07-30)

Improvements

- Support of Magento 2.4

Fixed

- missing product in ImageBuilder
-

1.1.18

(2020-06-18)

Fixed

- Missing products when attribute rule enabled

1.1.17

(2020-05-27)

Fixed

- Unable to save image
- Cannot instantiate abstract class issue
- Multiple labels display issue; added out of stock option t rules
- Invalid template file error in system.log

1.1.16

(2020-03-17)

Fixed

- Wrong rule processing when multistore inventory enabled

1.1.15

(2020-01-20)

Feature

- Display labels without images
-

1.1.14

(2019-05-29)

Fixed

- Position issue on product list
-

1.1.13

(2019-01-23)

Fixed

- M2.3. Product Label does not show on catalog page
-

1.1.12

(2019-01-03)

Fixed

- M2.1. Solved compilation issue
-

1.1.11

(2018-11-29)

Improvements

- M2.3 support
-

1.1.11

(2018-11-29)

Improvements

- M2.3 support
-

1.1.10

(2018-11-28)

Fixed

- support of magento 2.3
-

1.1.9

(2018-08-16)

Fixed

- Installation issue with area code
-

1.1.8

(2018-07-18)

Fixed

- Fixed an issue with not clickable link for Label on Product List page
-

1.1.7

(2018-07-13)

Fixed

- Fixed Percent Discount
-

1.1.6

(2018-07-09)

Fixed

- Properly display labels based on Percent Discount condition
-

1.1.5

(2018-06-02)

Fixed

- Fixed Percent Discount if Advanced Pricing->Special Price is given in percent
-

1.1.4

(2018-06-13)

Fixed

- Fixed an issue with unexpected label on product page
-

1.1.3

(2018-05-31)

Fixed

- Fixed an issue related to the rule cancellation while cron starts
-

1.1.2

(2018-05-21)

Fixed

- Fixed Attribute Gallery styles
-

1.1.1

(2018-04-26)

Fixed

- Fix big amount of memory usage
 - Fixed Percent Discount rule for bundle products
-

1.1.0

(2018-03-16)

Improvements

- Flush dependent pages cache after new product creating
-

1.0.16

(2018-03-09)

Fixed

- Fixed an error "Fatal error: Uncaught Error: Call to undefined method Magento\CatalogRule\Model\ResourceModel\Rule::calcProductPriceRule()"
-

1.0.15

(2018-01-29)

Fixed

- Fixed bottom label position for products that have a single image with no thumbnails displayed.
-

1.0.14

(2017-10-17)

Fixed

- Magento 2.2 compatibility

1.0.13

(2017-10-17)

Fixed

- Magento 2.2 compatibility
-

1.0.11

(2017-06-20)

Documentation

- Online User Manual updated
-

1.0.10

(2017-05-18)

Fixed

- Fixed usability issue: "Style" field made adjustable for "Manage Labels" Admin Panel page
 - Fixed "Notice: Undefined variable: percent" for some stores
-

1.0.9

(2017-03-29)

Improvements

- Added possibility to define additional CSS styles for labels from Admin Panel

Fixed

- Fixed design issue - labels removed from shopping cart page
-

1.0.8

(2017-01-12)

Fixed

- Fixed design issue
-

1.0.7

(2016-12-27)

Improvement

- Added product attribute "Set as New" to label conditions
-

1.0.6

(2016-12-06)

Fixes

- Fixed cron errors
-

1.0.5

(2016-10-18)

Improvement

- Updated docs
-

1.0.4

(2016-09-05)

Fixes

- Fixed an issue when badge description is breaking product list view
-

1.0.3

(2016-06-30)

Fixes

- Fixed an issue when cataloglabel is displaying in minicart after product was added to cart from product list
-

1.0.2

(2016-06-30)

Fixes

- Compatibility to Magento 2.1
-

1.0.0

(2016-02-17)

- Initial release