

Reward Points + Referral program Manual

Getting Started

Welcome to the **Reward Points + Referral Program** documentation.

Here you will find everything you need to set up your **Reward Points** service and create a flexible reward policy, which can cover almost any your marketing strategy.

Our extension allows customers to earn points for different actions, earn additional points for so-called thresholds (on reaching some level of sales amount, for example), and even earn from sharing links on most popular social networks.

Earned points can be spent on orders, as a dynamic discount, and even used to pay order completely.

Go ahead, dive in!

Firstly, please, find our extension in [My Downloadable Products](#) section of our store. Learn [how to install extension](#), and proceed with [Quick Start](#), which will guide you to setup your reward policy.

How to install extension

How to install the extension using composer

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

Deploy static view files

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

How to install the extension manually

1. Backup your store's database and web directory.
2. Download archive from [My Downloadable Products](#).
3. Unzip the extension locally.
4. Copy unzipped folder to the root directory of Magento store.
5. Run command `php -f bin/magento module:enable Mirasvit_Core Mirasvit_Report Mirasvit_Rewards` to enable the extension.
6. Run command `php -f bin/magento setup:upgrade` to install the extension.
7. Run command `php -f bin/magento cache:clean` to clean the cache.
8. Deploy static view files

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

Quick Start

Our **Reward Points & Referral Program** extension for M2 is a simple, yet powerful extension, that allows customers to build complex reward policies with a minimum efforts. The most of its functions work out-of-box.

However, if you intend to create a complex policy, you will need a fresh quick start to help you build your program.

1. Reward Policy begins with creation of at least two rules - one for [earning](#) points, and one for [spending](#) them.

Important Note: While creating the rule, you should fill in the fields "Rule Name" and "Display Name":

"Rule Name" is the name for inner usage and will be displayed in your store back end.

"Display Name" is the rule description for your customers. It will be displayed in the client's account, thus bringing a client the valuable information on the rule (way) to earn (for earning rules) or to spend (for spending rules) the points on your store.

The main rules of points earning with a client's account are gathered in "My Reward Points" menu, in the "My Points" tab.

Rules of points earning with the help of social media are displayed in the "My Reward Points" menu, in the "Share & Save" tab.

2. Earning rules can be defined as one of the following strategies:
 - [Product Rule](#), which allows to earn points for buying specific products;
 - [Cart Rule](#), which allows to earn points for a whole cart subtotal, including shipping and handling fees;
 - [Behaviour Rule](#), which allows to earn points for conducting specific actions (for example, review placement);

- [Custom Behaviour Rule](#), which allows you to define your own behaviour actions;
3. Spending rules allows you to define, at which rate points should be spent.
Important Note: Without at least one such a rule, our extension "does not know", how points should be converted to discount, and our spending block in Cart and Checkout won't display.
 4. You can also define your own notification policy, by creating custom messages at [Notification rules](#), which will appear in customer's account and cart. These messages are powerful tool for promoting your reward policy.
 5. Choose [social network buttons](#), which will appear on your products pages, and use [Behaviour Rules](#) to make customers earn for sharing.
 6. Explore extensive conditions in earning, spending and notification rules, to create flexible and powerful reward policy. Our extension can support almost any plan you will develop.
 7. Use Magento's Customer Groups and build with them customer ranks. Then, using our [Behaviour Rules](#) action **Assign Customer to a Group** build milestone rules to uplift customers in ranking system.
 8. Observe rewards statistics frequently with our [Reports](#), that allow you to see - how points are earned and spent daily and by different customers groups, making your Rewards policy flexible and responsive to customer's actions.
 9. Build up your [Referral Program](#) and extend your customer base greatly.

This should be a good beginning.

Refer to appropriate sections of our manual to have more info.

Configuration Settings

Go to **Stores > Settings > Configuration > Mirasvit Extensions > Reward Points**. You will see there all Reward Points settings (except [Social Buttons](#), which have their respective section), divided into the following sections:

- [General Settings](#)
- [Notification Settings](#)
- [Referrals](#)
- [Display Options](#)
- [Advanced Settings](#)

General Settings

| Option | Description |
|-----------------|---|
| Point Unit Name | Name of unit for points (e.g. Reward Point(s), Gold Coin(s)). |

| Option | Description |
|---|--|
| Reward points expire after, days | Number of days after which points will be expired. If you don't need this feature, leave this field empty. |
| Approve earned points on invoice | If enabled, earned points will be added to the customer's account after creating a new invoice for the order. |
| Approve earned points on shipment | If enabled, earned points will be added to the customer account after creating a new shipment for the order. |
| Approve earned points if order has status | If order has one of the selected statuses from the list, earned points will be approved. |
| Delay usage of rewards points for, days | If enabled, earned points will be added to the customer account after X days. This option works only for product or cart earning rules. |
| Cancel earned points after order refunding | If enabled, earned points will be canceled after creating a new credit memo for an order. |
| Restore spent points after order refunding | If enabled, spent points will be restored after creating a new credit memo for an order. |
| Include discount for Earning Rules | If enabled, allows to include discount to calculations of earned points (applies only to Cart Rules) |
| Include tax for Earning Rules | If enabled, allows to include tax to the amount, from which earned points are calculated |
| Include tax for Spending Rules | If enabled, allows to include tax to the amount, eligible for paying by points (applies only when Stores -> Configuration -> Sales -> Tax -> Calculation Settings -> Apply Customer Tax is set to Before Discount). |
| Allow to earn points from shipping charges | If enabled, earned points will be added to the customer's account on shipping charges. |
| Allow to spend points for shipping charges | If enabled, customer is able to spend earned points on shipping charges. |
| Allow to place orders with grand total = 0 | If enabled, customer is able to place orders and earn points with a grand total = 0. |
| Allow to display product points as currency equivalent | This option allows customers to see points awarded for particular product as their potential money equivalent. It is calculated from rates defined in the spending rules, and assumes that points - earned for this product - were spent with the highest possible rate. This calculation, however, does not include conditions defined in spending rules - as they can be applied only in the cart - so this option shows only the approximate money amount . |
| Allow to display maximum amount of possible points for configurable product | This option allows customer to display points amount on pages of configurable products. Amount is calculated as if customer picked options with maximal values. |

Notification Settings

| Option | Description |
|----------------------|---|
| Email Sender | Email which will be used to send email messages to customers. |
| Balance update Email | Template for balance update notification. |
| Points expire Email | Template for points expiration notification. |

| Option | Description |
|--|---|
| Send a notification before points expire (in days) | Number of days before which the extension will a send points expiration notification to the customer. |

Tiered Rewards Program

| Option | Description |
|-------------------------------|---|
| Switch between tiers based on | Allows to select the way to switch between tiers. |
| Use orders in status | Enable only if "Switch between tiers based on" is selected as "sum of spent \$". This option defines what orders are included in "sum of spent \$" |
| Include Tax | Enable only if "Switch between tiers based on" is selected as "sum of spent \$". This option defines if the tax is included in "sum of spent \$" |
| Include Discount | Enable only if "Switch between tiers based on" is selected as "sum of spent \$". This option defines if the discount is included in "sum of spent \$" |
| Include Shipping | Enable only if "Switch between tiers based on" is selected as "sum of spent \$". This option defines if the shipping is included in "sum of spent \$" |

Referrals

| Option | Description |
|---------------------------|---------------------------------------|
| Invitation Email Template | Template for invitation of referrals. |
| Enable Referral System | Enables referral system. |

Display options

| Option | Description |
|---|--|
| Show rewards points menu on frontend | If enabled, Reward Points menu item will appear on Magento menu. |
| Show rewards points on category page | If enabled, Reward Points will display not only on product pages, but also in product cells at Category pages. |
| Show number of points on the product page | If enabled, customer will see approximate amount of points they wil earn for buying the product |

Advanced Settings

| Option | Description |
|---------------------------|---|
| Round earning points down | As points can not be fractional, this option allows you to control how to round them: to the nearest lowest or highest number |
| Custom Behavior Events | Allows you to add custom behavior rules using this format: rulecode, rulename . For more details refer to the Custom Behavior Events section |

Social Networks Settings

Our Reward Points extension allows you to place social buttons on Product and Categories pages, and even more - using [Behaviour Rules](#) you can make your customers earn points for sharing your products in social networks.

You can configure, which buttons shall be shown at **Stores > Settings > Configuration > Mirasvit Extensions > Social Networks Settings**:

- [Facebook Settings](#)
- [Twitter Settings](#)
- [Google Plus Settings](#)
- [Pinterest Settings](#)
- [AddThis Settings](#)

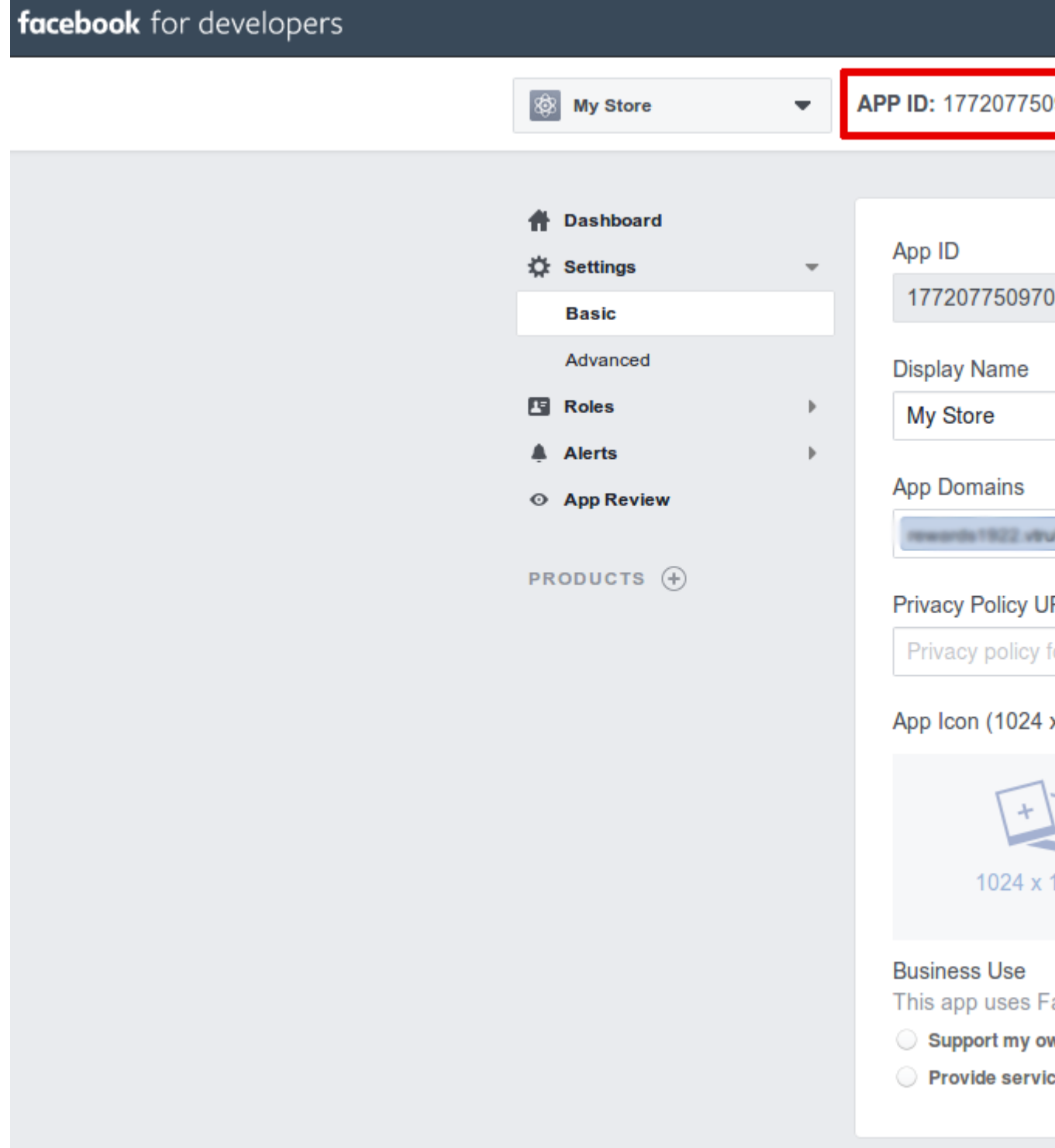
Facebook Settings

- **Facebook App ID** - This option is used, when rule for earning from Facebook Like or Share is created. This rule uses Facebook API for tracking customer activity, and for that Facebook Application ID is needed.
- **Show Facebook Like button** - If option is enabled, the Facebook Like button will be shown on the product and category pages.
- **Show Facebook Share button** - If this option is enabled, additional Facebook Share button will appear, allowing customers earn from sharing page on Facebook.

Important Note: An useful tutorial for creating Facebook Application can be found [here](#). Facebook Apps themselves are created [here](#).

- **How to Create Facebook Application.**
 - Having logged in to Facebook, go to [Facebook App Center](#). Press **Add a New App** button. Supply two base property:
 - **Display Name** - a sensical name of your App. For consistency reason name it just like your store's name.
 - **Contact Email** - which email shall be connected with this application. It shall be the same, as your Facebook account.
 -

When App is created, you will receive an ID - it will be displayed on the top, as shown below:



- Proceed to **Settings** -> **Basics** and add additional properties to make App work:
 - **App Domains** - enter here your store's base URL (e. q. `mystore.com`)
 - **Category** - Business and Pages
- Save your app, and enter your App ID to the **Facebook App ID** at **Stores** -> **Settings** -> **Configuration** -> **Mirasvit Extensions** -> **Social Networks Settings** -> **Facebook Settings** section.

Twitter Settings

- **Show Twitter Follow button** - If option is enabled, the Twitter Follow button will be shown on the product and category pages.

Google Plus Settings

- **Show Google Plus +1 button** - If option is enabled, the Google Plus +1 button will be shown on the product and category pages.

Pinterest Settings

- **Show Pinterest pin button** - If option is enabled, the Pinterest pin button will be shown on the product page.

AddThis Settings

AddThis is a special network service, which allows to place on your store almost any social network button. However, these buttons are not tracked by our rules, so they can be used only for promotional purpose.

- **Show AddThis** - If option enabled, AddThis buttons will be shown on the product page.
- **AddThis Code** - AddThis code that should be inserted in the template.

Important Note: An useful tutorial for creating AddThis buttons can be found [here](#);

Display options

- **Show Social Buttons block on category page** - If enabled, Social Buttons block will be shown on the catalog page.

Transaction Email Templates

In the email templates of order, invoice and creditmemo you can use following variables:

- `{{var order.getRewardsEarnedPoints()}}` - amount of earned points for order
- `{{var order.getRewardsSpentPoints()}}` - amount of spent points for order
- `{{var order.getRewardsSpentAmount()}}` - amount of money spent

Note

For a partial invoice or creditmemo this variables will contain amount for the whole order.

Earning Rules

Earning Rules define ways for customers to earn points.

To create a new Earning Rule, go to the **Marketing > Promotions > Reward Points > Earning Rules** and press the button **Add New**.

Edit page of an earning rule is very complex and differs depending on type of the rule, so each type has its own subsection here:

- [General Information \(common part\)](#)
- [Product Rule](#)
- [Cart Rule](#)
- [Behaviour Rule](#)

If you wish to create custom earning rule, please, refer to [Custom Behavior Rules](#) section of this manual.

General Information

These options are common to all types of earning rules.

- **Rule Name** - name of the rule. Used only for internal purposes.
- **Display Name** - is the rule description for your customers. It will be displayed in the client's account, thus bringing a client the valuable information on the rule (way) to earn the points on your store.
- **Type** - the type of rule. There are following types of rules:
 - **Behaviour Rule** - points can be earned for customer's actions.
 - **Product Rule** - points can be earned for buying specific products.
 - **Cart Rule** - points can be earned for cart amount or contents.
- **Is Active** - rule status.
- **Active From, Active To** - period of rule activity.
- **Customer Groups** - customer groups for which rule can be applied.
- **Stop further rules processing** - if enabled, no other rules will be applied, if this one applies. Please, see important note below for details.
- **Priority** - order which will be used for rule sorting before applying.

Note

Important: It is common case, when different policies shall be applied on the mostly same conditions (or the same event), but with some differences - which require applying only one rule per order. Then **Priority** and **Stop further rules processing** options shall be used for organizing their execution order.

Example

For example, regular customer shall receive 10 points, but VIP shall receive 100.

Let's have two rules for that - "Regular" and "VIP" In this case VIP rule shall have higher priority, and has option **Stop further rules processing** enabled. Then, when VIP customer will buy, the VIP rule will apply, but Regular will not (since option **Stop processing...** is set). When Regular customer buys, VIP rule will be skipped, and Regular rule will apply normally.

Each rule has four possible global modes of applying conditions in respective **Conditions** tab, shown in special header `If *[apply mode]* of these conditions are *[validation mode]*:`

Applying modes define, when rule shall be triggered:

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

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

- **TRUE** - implies. that conditions should be valid.
- **FALSE** - implies, that conditions should be invalid.

These modes allow to create flexible condition sets to satisfy policy of any complexity.

Product Rules

These rules award points for buying a particular product. This type of rule is recommended when products are presumably simple and have fixed prices.

If you wish to include shipping and handling fees to points calculation, but still filter products, you need to use [Cart](#) rule with **Product attribute combination** mode.

Conditions

This section define either single condition or their set, which trigger the rule. They all must be placed inside global mode definition, as described in [General](#) subsection.

Each condition has the same scheme: `[property] [operator] [check value]`

- **Property** is the product attribute, that shall be checked;
- **Operator** is the type of check, that should be performed to validate rule. There are the following operators available:
 - **is** - arithmetic operator of equality (==)
 - **is not** - arithmetic operator of non-equality (!=)
 - **equals of greater then** - arithmetic operator >=
 - **equals or less then** - arithmetic operator <=
 - **greater then** - arithmetic operator >
 - **less then** - arithmetic operator <
 - **contains** - special operator, which works differently. If check value is an array (should display as drop-down menu), then operator will search for **check value** in this array. If value is string, or can be evaluated to string, operator will treat **check value** as a substring to find.
 - **does not contains** - the same as above, but with negative result
 - **is one of** - special operator, which works with user-defined comma-separated array as **check value**. If this operator used, then condition is valid, when actual value of **property** exists in defined

array. If property itself is a selection-type (SKU or Category, for example), then dialog popup will appear to select elements of user-defined array.

- **is not one of** - the same as above, but with negative result
- **Check value** is value, to which will apply operator, selected in previous step.

Below you will find a short list of all conditions, currently supported by our extension:

- **Product Properties, available by default**

Generally, their list resembles properties, that are set for Products on particular store. If you will create a custom field for Product, or use extended properties list, they also shall be available in rules.

- **Attribute Set** - product attribute setup, created at **Catalog -> Attributes -> Attribute Sets**
- **Category** - category, under which this product is listed. When you need to check for more than one category, use **is one of** operator in condition.
- **Product Type** - allows to check for certain type of product, placed to the cart. Can take the following values:
 - Simple;
 - Virtual;
 - Bundle;
 - Downloadable;
 - Grouped;
 - Configurable.
- **Quantity** - quantity of items of this product, placed to the cart;
- **SKU** - SKU of product. Selection-type property.
- **Pattern** - pattern, used in this product. Included here as an example of extended product property.
- **Performance Fabric** - extended property, used only for fabric-type products. Included here as an example of extended product property.
- **Activity** - area of usage of particular product
- **Sale** - whether product is on sale. In fact is a boolean property, but is a selection Yes/No type. Also included as an example of extended property.
- **Size** - size of particular item of product. Is a numeric value, and can be cast to a string.
- **Base Image** - base image of the product
- **Base Image Size (bytes)** - base image size
- **Small Image** - small image of product
- **Small Image Size (bytes)** - size of base image of a product
- **Thumbnail** - image, that is used in search or category page display cell for a particular product
- **Thumbnail Size (bytes)** - size of the image above.

Additionally, there is a special condition - **PHP Condition** - which allows you to create your own complex conditions.

Here is the universal expression for any product attribute:

```
$object->getData([ATTRIBUTE_CODE]) == [ATTRIBUTE_VALUE]
```

`object` is an actual product, which currently checked by the rule. So you can virtually evaluate in the left part of this expression any combination of product attributes. But the right shall contain valid value to compare result with, otherwise condition won't work;

Actions

This section contains action, that shall be executed, when conditions on previous sections are met.

- **Customer Earning Style** - strategy, which will be used for points awarding
 - **Give X points to customer** - customer is given with a fixed amount of points;
 - **For every Y, give X points** - points calculated from product's cost, which is divided into chunks. Since it is Product Rule, calculations is performed separately for each product in cart.
- **Number of Points (X)** - number of points (integer number), that shall be awarded according to picked strategy. It also serves as minimal default currency amount, that can be converted to points.
- **Step (Y)** - used only when **For every Y, give X points** strategy was selected. Contains a size of one chunk, into which product's cost is broken.
- **Maximum Distributed Points** - maximum number of possibly earned points for a current rule triggering.

Notification

- **Message** - this message will be displayed on product page, if the rule is valid for the current product

Cart Rules

Award points for the total amount of actual money spent on the cart. This type of rule is recommended when using tier or special prices (with taxes, dependent from shipping country, for example), which can dynamically alter the actual product sell price.

Conditions

Cart Rules has more complex condition subsystem. Although it has the same modes, as Product Rule, described [above](#), it features three more special modes, that are selected as regular conditions, but transforms to a nested containers:

- **Product attribute combination** - allows you to set the product attribute combination. Attributes list is the same, as in **Product Rule**. It has the following scheme: If an item is [validation mode] in the cart with [applying mode] of these conditions true:
 - *Validation mode* can be either **FOUND** or **NOT FOUND**, which defines, whether checked product should exists in Cart;
 - *Applying mode* can be either **ALL** or **ANY**, their meaning is the same as in previous rule type: whether all conditions in this container apply, or one or more of them.
 - Additionally to product properties set, this mode provides three additional properties regarding Cart content:
 - **Price in Cart** - allows to check product, that match conditions, price from cart. Useful, when discounts or custom prices are used;
 - **Quantity in Cart** - allows to check product quantity in cart.
 - **Row Total In Cart** - Total cost of product, that match conditions, in current Cart. In fact, are calculated as multiply of two above properties.
- **Product subselection** - allows you check the cart item attributes. It also is a container and has the following scheme If [selection mode] for a subselection of items in cart matching [applying mode] of these conditions.
 - *Selection mode* defines, which primary parameter is checked - product quantity (**total quantity**) or its row total (**total amount**). It shall be supplied with numeric value.
 - *Applying mode* can be either **ALL** or **ANY**, their meaning is the same as in previous rule type:

whether all conditions in this container apply, or one or more of them.

◦ Container can have the same conditions, as above mode.

- **Conditions combination** - allows you to break rule into logical parts, each of which can be validated separately. It is extremely useful, when **ANY** applying mode is used for a whole rule and complex conditions required. It creates just a nested block of `If *[apply mode]*` of these conditions are `*[validation mode]*` type, as described in [General](#) section.

If no additional mode is selected, the following properties are available:

- **Subtotal** - subtotal of cart, including shipping and handling fees. It also can contain discount, if **Stores -> Configuration -> Sales -> Tax -> Calculation Settings -> Apply Customer Tax** is set to **Before discount**.
- **Total Items Quantity** - total items quantity in current Cart.
- **Total Weight** - sum of weights of all products, contained in Cart.
- **Payment Method** - payment method.

Note

If payment method is not selected - this condition will return false. Therefore, this condition forces earning only in checkouts, or when payment selection is available directly from Cart.

- **Shipping Method** - shipping method. Note above also applies here.
- **Shipping Postcode** - postal code, where products shall be sent. Note above also applies here.
- **Shipping Region** - shipping region. Note above also applies here.
- **Shipping State/Province** - state, where product should be delivered. Note above also applies here.
- **Shipping Country** - country, where product should be delivered. Note above also applies here.

Actions

- **Customer Earning Style** - strategy, which is used for points calculation.
 - **Give X points to customer** - customer is awarded with fixed points amount.
 - **Give X points for every spent Y** - points are calculated from chunks, into which Cart subtotal breaks.
 - **Give X points for every Z quantity** - points are calculated from total quantity of items in current Cart.
- **Number of Points (X)** - quantity of points, that should be awarded for each step. It also serves as minimal points, that can be awarded.
- **Step (Y)** - used only when **Give X points for every spent Y** strategy is selected, and defines size of one chunk, into which subtotal should be divided. It also serves as minimal currency amount, which will be awarded with points.
- **Quantity Step (Z)** - used only **Give X points for every Z quantity** strategy is selected. If it's not empty, then items quantity will be divided into chunks of this size, and points will be calculated from them.
- **Earn Maximum** - maximum number of possible earned points for the current rule.

Example

For example, you can create the following cart rules in your store:

- Buy 3 products within one order and receive 100 points.
- For every \$1 spent, receive 1 point.

- For every product ordered, receive 100 points.

This section also contains a special subsection **Apply the rule only to cart items matching the following conditions**. It is a shortcut for a condition of **Product attribute combination** mode (see above).

Behaviour Rules

These rules triggers only on certain customer's actions, such as writing a review or placing an order.

Conditions

This section allows to pick up **Event**, which corresponds with customer's action, and bind to it current rule:

- **Customer signs up in store** - signing in to a store. Useful for awarding points, that should be used in the very first order.
- **Customer places order** - triggered, when customer places an order, and it is completed (e. q. invoiced and shipped).
- **Customer refer a friend** - customer sends an invite to a friend.
- **Customer writes a product review** - customer adds a review, and it is approved by staff.
- **Customer has a birthday** - this rule is executed by crontask, which starts at 00:00 of each day on every customer. If customer has birthday, rule will be executed at that time.
- **Customer is not active for a long time** - should be set in days. This rule also is executed at 00:00 by crontask.
- **Customer joins the affiliate program** - customer joins to affiliate program (Mirasvit_Affiliate)
- **Newsletter sign up** - customer opts to receive newsletter.
- **Facebook Like** - customer likes product page on Facebook.
- **Twitter Tweet** - customer tweets product page on Twitter.
- **Google+ Like** - customer places product pages on Google Plus.
- **Pinterest Pin** - customer places a pin for a product on Pinterest.
- **Referred customer signs up in the store** - referred customer (referral) signs up to the store. For more information refer to [Building Referral Program](#) section.
- **Order from a referred customer** - referred customer had placed an order, and it is completed. For more information refer to [Building Referral Program](#) section.
- **Custom events** - list of custom events that were created using this [Custom Behavior Events](#)

Actions

- **Number of points to give (X)** - number of points that a customer receives if the rule is applied.
- **Maximum number of earned points for one customer per day** - sets limit of earned points for one customer per day. Set 0 or empty to disable limit.
- **Assign customer to Group** - allows to move customer to the desired Group, if rule is triggered. It is extremely useful for maintaining customer ranks program for example, or building referral program.

There's also a subsection **Apply the rule only if the following conditions are met**, where additional conditions can be set. Here is the list of currently supported properties:

- **Is a Referee** - checks, whether customer has invited customers to a store, and they had became registered customers. It is extremely useful for building Referral Program.
- **Is a Referral** - checks, whether customer is a referral (e. q. was invited by someone)
- **Group** - checks, to which group belongs this particular customer.
- **Lifetime sales** - calculates total of all orders, placed by current customer, and compares with value

supplied.

- **Number of Orders** - calculates total number of orders, placed by current customer, and compares to value supplied.
- **Is subscriber to newsletter** - checks, whether customer is subscribed to newsletter.
- **Number of Reviews** - calculates total number of approved reviews by current customer, and compares to value supplied.
- **Number of referred friends signups** - calculates, how many of invited customers became registered customers (e. g. became referrals).
- **Number of referred friends orders** - whole number of orders, placed by all referrals, invited by current customer.
- **Sum of referred friends orders** - total money amount of all orders, placed by all referrals, invited by current customer.
- **Number of referred friends that ordered at least once** - number of referrals, invited by current customer, which placed at least one order.

Notification

- **Message in the rewards history** - a message, that will be added as description of transaction, created by this rule.
- **Message for customer notification email** - a message, that will be sent to customer via email. There shall be only sensical message, not the whole template. That message will be inserted to the **Balance update Email** template, which can be set in [Settings](#).

Example

For example, you can create the following behaviour rules in your store:

- Write a review and receive 10 points
- Refer a friend and receive 10 points when your friend makes their first order

More examples can be found at [Earning Rules Examples](#) section.

Spending Rules

Spending Rules are much simpler than earning, since they only define a rate, at which points can be converted to discount.

Note

Points can be spent by customer using special block, that will appear at Cart and Checkout pages, when the following conditions are met:

- Options **Marketing -> Reward Points -> Settings -> Display** are enabled;
- At least one Spending Rule is defined and activated;
- Customer has enough points on his balance;
- Customer and Cart fit the conditions of Spending Rule(s).

To create a new Spending Rule, go to **Marketing > Promotions > Reward Points > Spending Rules** and

press the button **Add New**. Rule edit page, that will appear, is divided into three sections:

General Information

- **Rule Name** - name of the rule. Used only for internal purposes.
- **Display Name** - is the rule description for your customers. It will be displayed in the client's account, thus bringing a client the valuable information on the rule (way) to spend the points on your store.
- **Is Active** - rule status.
- **Active From, Active To** - period of rule activity.
- **Customer Groups** - customer groups for which rule can be applied.
- **Stop further rules processing** - if enabled, no other rules will be applied.
- **Priority** - order which will be used for rules sorting before applying.

Conditions

Spending Rule applies only to Cart as a whole, so conditions in this section resemble Cart conditions. Refer to [Cart Earning Rule](#) section for detailed description of them.

Actions

This section defines, how points are converted into discount.

- **Points spending style** - defines basic strategy, which is used for points conversion.
 - **Flexible** - allow to spend points in proportion to the **For each spent X points**. For example, for rule where customer gets \$1 discount for 10 point customer can spend 15 points and get \$1.5 discount.
 - **Fixed** - allow to spend points strictly for a multiplication of the **For each spent X points**. For example, for rule where customer gets \$1 discount for 10 point customer can spend only 10 or 20 or 30 or ...
- **For each spent X points** - defines minimal quantity of points, that can be traded for discount. If **Flexible** strategy is used, then it used as proportion base.
- **Customer receive Y discount** - defines quantity of currency, for that should be traded quantity of points (property above).
- **Spend minimum** - minimal quantity of points, that may be spent on one Cart. Used mainly for **Flexible** strategy.
- **Spend maximum** - maximal quantity of points, that may be spent on one Cart.

Note

To allow a customer to spend their points, you must create at least one spending rule. Otherwise our spending blocks in Cart and Checkout won't appear.

More examples can be found at [Spending Rules Examples](#) section.

Notification Rules

Notification Rules allow you to setup different messages for customers, which notify customers about possible earning/spending points.

To create a new Notification Rule, go to **Marketing > Promotions > Reward Points > Notification Rules** and press the button **Add New**, and you will see edit page, divided into three sections:

General Information

- **Rule Name** - name of the rule. Used only for internal purposes.
- **Is Active** - rule status.
- **Active From, Active To** - period of rule activity.
- **Customer Groups** - customer groups for which the rule can be applied.
- **Stop further rules processing** - if enabled, no other rules will be applied.
- **Priority** - order which will be used for rules sorting before applying.

Conditions

Notification Rule uses current persistent Cart, so conditions in this section resemble Cart conditions. Refer to [Cart Earning Rule](#) section for detailed description of them.

Actions

- **Show message on** - list of pages where you would like to show this notification message.
- **Message** - text of message to show.

More examples can be found at [Notification Rules Examples](#) section.

Custom Behavior Events

Using this feature you can add your own ways for customers to earn points.

1. To create a new Custom Behavior Rule, go to **Marketing > Promotions > Reward Points > Settings > Advanced Settings**.
2. In the field **Custom Behavior Events**, add a new rule in this format: **rulecode, rulename**. You can add more rules, each on a new line.
3. Insert the following code in the place where points should be charged:

```
$eventManager = $this->objectManager->get('Magento\Framework\Event\ManagerInterface');  
$eventManager->dispatch('mirasvit_rewards_behavior_rule_process', ['code' => 'r
```

Example

For example, to create a rule for the event 'user name changed' you should insert in the field **Custom Behavior Events**: `customer_change_name, Change customer name rule`

and insert the following code:

```
$eventManager = $this->objectManager->get('Magento\Framework\Event\ManagerInterface');  
$eventManager->dispatch('mirasvit_rewards_behavior_rule_process', ['code' => 'c
```

Note: the variable **\$customer** should be defined

General Information

Use API to attract customers and increase sales rate on your store. From now on, you have the possibility to simultaneously use Reward Points Extension from external tools that work on separate servers and use other programs, including mobile applications. Using Reward Points, users can get rewards for the particular actions made out of a store.

For example: If you use a mobile app for your store, then a buyer can obtain additional points in this app.

Reward Point extension is available to partner programs, works with tools that interact with your customers directly.

To get the full list of available REST APIs check file Rewards/etc/webapi.xml. You can check some examples in [Using REST API](#) section

Using REST API

REST API is an standard interface, which Magento 2 supports for accessing and manipulating data from external tools and services.

It uses CURL PHP library for sending requests, and token-based authentication. It means, that in your REST PHP script you need to pass login and password, and in return you will receive one-time token, which will be used for all subsequent requests until expired.

Token can be generated using the following code:

```
$userData = array("username" => "admin", "password" => "sZ62LdQsJunn5ZRG");
$ch = curl_init("http://store.com/index.php/rest/V1/integration/admin/token");
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($userData));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-Type: application/json", "O
$token = curl_exec($ch);
```

This token can be used for accessing any data, which covered by REST API.

Our extension allows to manipulate transactions with REST API. Result of each REST request will be JSON-encoded string, which can be decoded using `json_decode` function.

Here are some snapshots, which will help you in most cases:

- **How to get Transaction Data**

To retrieve particular transaction, you need to pass transaction ID. Concat it to the REST URI, as shown on example below:

```
$transactionId = 1;
$ch = curl_init("http://store.com/index.php/rest/V1/rewards/transaction/"
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "GET");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-Type: application/json"
$result = curl_exec($ch);
```

- **How to get all Transactions for a Customer**

To retrieve particular transaction, you need to pass customer ID. Insert it to the REST URI, as shown on example below:

```
$customerId = 1;
$ch = curl_init("http://store.com/index.php/rest/V1/rewards/customer/" . $
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "GET");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-Type: application/json"
$result = curl_exec($ch);
```

This expression will return a JSON-encoded array of all transactions, which belong to selected customer.

- **How to update some field in existing Transaction**

To update an existing transaction, you need to provide a transaction ID and a field with updated value. They shall be provided as json-encoded parameter CURLOPT_POSTFIELDS.

Note

Transaction ID in data and REST URL **shall match**, otherwise action will return error.

```

$transactionId = 1;
$data = [
    'transaction' => [
        'id' => $transactionId,
        'comment' => 'some updated text',
    ]
];
$headers = [
    "Content-Type: application/json",
    "Content-Length: " . strlen(json_encode($data)),
    "Authorization: Bearer " . json_decode($token),
];
$ch = curl_init("http://store.com/index.php/rest/V1/rewards/transaction/");
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
$result = curl_exec($ch);

```

You can add as many fields, that you wish to update, to the \$data array. Result of this code will contain fields, that were actually changed, filled. Untouched will be equal to null.

- **How to create a Transaction**

To create transaction, you need to provide all fields of transaction record, similar to above. Here is a minimalistic example:

```

$data = [
    'transaction' => [
        'customer_id' => 1,
        'amount' => 23,
        'comment' => 'REST API Example',
        'code' => 'api_transaction',
        'expires_at' => ('2019-01-15'),
    ]
];
$headers = [
    "Content-Type: application/json",
    "Content-Length: " . strlen(json_encode($data)),
    "Authorization: Bearer " . json_decode($token),
];
$ch = curl_init("http://store.com/index.php/rest/V1/rewards/transaction");
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
$result = curl_exec($ch);

```

Result will return a JSON-encoded representation of newly-created Transaction.

- **How to remove a Transaction**

You can delete a transaction using REST API. For that you will need just to know a transaction ID.

```
$transactionId = 1;
$ch = curl_init("http://store.com/index.php/rest/V1/rewards/transaction/");
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "DELETE");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-Type: application/json"));
$result = curl_exec($ch);
```

• How to get earn points for product page

You can also get amount of earn points for product page using REST API. For that you will need to use endpoint `/V1/rewards/products/points/get` with parameters:

| Name | Description |
|------------|---|
| sku | Product SKU. |
| price | The price of the product. |
| customerId | Current customer ID, 0 for guest customers. |
| websiteId | Website ID. |
| tierId | Tier ID. |

```
$data = [
    'sku'          => '240-LV05',
    'price'        => 23.53,
    'customerId'  => 1,
    'websiteId'   => 1,
    'tierId'      => 2,
];
$headers = [
    "Content-Type: application/json",
    "Content-Length: " . strlen(json_encode($data)),
    "Authorization: Bearer " . json_decode($token),
];
$ch = curl_init("http://store.com/index.php/rest/V1/rewards/products/points");
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
$result = curl_exec($ch);
```

You can get amount of earn points for several product using REST API. For that you will need to use endpoint `/V1/rewards/products/points/multiplicity` with parameter `productInfo`. `productInfo` - is array of data for each product

```
$data = [
    'productInfo' => [
        [
            'sku'          => '240-LV05',
            'price'        => 23.53,
            'customerId'  => 1,
            'websiteId'   => 1,
            'tierId'      => 2,
        ],
        [

```

```

        'sku'           => '240-LV06',
        'price'        => 15.77,
        'customerId'   => 1,
        'websiteId'    => 1,
        'tierId'       => 2,
    ],
]
];
$headers = [
    "Content-Type: application/json",
    "Content-Length: " . strlen(json_encode($data)),
    "Authorization: Bearer " . json_decode($token),
];
$ch = curl_init("http://store.com/index.php/rest/V1/rewards/products/points");
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
$result = curl_exec($ch);

```

These snapshots are enough to satisfy almost any task, including import or export of transactions.

Examples of Earning Rules

In this section we explain the possible configurations of typical rules.

Behavior Rules

- **Give 5 points for a sign up in the store**
 Customer signs up in the store and receives points
 - Type: Behavior Rule
 - Is Active: Yes
 - Customer Groups: All
 - Event: Customer signs up in store
 - Number of points to give (X): 5
 - Conditions: None
 - Message in the rewards history: Give points for a sign up in the store

- **Refer friends and receive 10 points for their sign up in the store**

Customer refers any number of friends

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Referred customer signs up in store

- Number of points to give (X): 10
- Conditions: None
- Message in the rewards history: Points for your referral signing up in store

- **Refer 5 friends and receive 15 points when they order**

Customer refers any number of friends but get points only for the first 5 friends.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Order from referred customer
- Number of points to give (X): 15
- Conditions:
 - If ALL of these conditions are TRUE:
 - Number of referrals who ordered at least once is 5
- Message in the rewards history: Earned points for referrals' orders

- **Give 50 points if customer is inactive for more than 20 days**

If customer is inactive for more than 20 days they will receive 50 points.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Customer is inactive for a long time
- Number of Inactive Days: 20
- Number of points to give (X): 50
- Conditions: None
- Message in the rewards history: We miss you

- **Write a review and receive 100 points**

Customer will get points for each approved review in the store.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Customer writes a product review
- Number of points to give (X): 100
- Conditions: None
- Message in the rewards history: Points for product review

- **Write max 2 reviews per day and receive 100 points**

Customer will write any number of reviews but will get max 200 points for two reviews per day.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Customer writes a product review
- Number of points to give (X): 100
- Maximum number of earned points for one customer per day: 200

- Conditions: None
- Message in the rewards history: Points for product review

- **Give 100 points for Birthday**

Customer will get 100 points on their birthday each year.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Customer has a birthday
- Number of points to give (X): 100
- Conditions: None
- Message in the rewards history: Points for birthday!

- **Give 15 points for Google+**

Customer will share page via Google+ and get points.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Google+ Like
- Number of points to give (X): 15
- Conditions: None
- Message in the rewards history: Points for Google+

- **Give 15 points for pinning to Pinterest**

Customer will pin page to Pinterest and get points.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Pinterest Pin
- Number of points to give (X): 15
- Conditions: None
- Message in the rewards history: Points for a Pinterest Pin

- **Give 15 points for 3 Tweets in day**

Customer will share page via Twitter and get points.

- Type: Behavior Rule
 - Is Active: Yes
 - Customer Groups: All
 - Event: Twitter Tweet
 - Number of points to give (X): 15
 - Maximum number of earned points for one customer per day: 45
 - Conditions: None
 - Message in the rewards history: Points for Tweet
-

Give 15 points for a Facebook Like

Customer will like page on Facebook and get points.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Facebook Like
- Number of points to give (X): 15
- Conditions: None
- Message in the rewards history: Points for Facebook Like

Give 15 points for a Facebook Share

Customer will share page via Facebook and get points.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Facebook Share
- Number of points to give (X): 15
- Conditions: None
- Message in the rewards history: Points for Facebook Share

Give 150 points when 5 referred friends register on the website

Customer refers any number of friends but gets points only for first 5 registrations.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Customer signs up
- Number of points to give (X): 150
- Conditions:
 - If ALL of these conditions are TRUE:
 - Number of referrals registered is 5
- Message in the rewards history: Earned points for referred friends registering

Use friend invitation and receive 30 points for the first order

Customer uses referral link and places their first order.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Event: Customer places order
- Number of points to give (X): 30
- Conditions:
 - If ALL of these conditions are TRUE:
 - Is Referral is 1
 - Number of Orders is 0
- Message in the rewards history: Earned points for invite

- **Give 20 points if customer places 4 orders or if they reach \$500**

Customer will get 20 points for placing 4 orders or if they reach \$500.

- Type: Behavior Rule
- Is Active: Yes
- Customer Groups: All
- Conditions:
 - If Any of these conditions are TRUE:
 - Number of Orders is 4
 - Lifetime Sales equals or greater than 500
- Number of points to give (X): 20
- Message in the rewards history: Earned points for number of orders

Product Rules

- **Receive 10 points for every \$100**

Customer will get 10 points for every \$100 spent in the store.

- Type: Product Rule
- Is Active: Yes
- Customer Groups: All
- Customer Earning Style: For every Y, give X points
- Number of Points (X): 10
- Step (Y): 100

- **Receive 1 point for every \$10 spent in a category**

Customer will get 1 point for every \$10 only in the specified category (Fitness Equipment with ID = 5).

- Type: Product Rule
- Is Active: Yes
- Customer Groups: All
- Conditions:
 - If ALL of these conditions are TRUE:
 - Category is 5
- Customer Earning Style: For every Y, give X points
- Number of Points (X): 1
- Step (Y): 10

- **Receive 1 point for every \$10 spent in a group of categories**

Customer will get 1 point for every \$10 spent only in a specified group of categories (Fitness Equipment ID = 5 and Watches ID = 6).

- Type: Product Rule
- Is Active: Yes
- Customer Groups: All
-

Conditions:

- If ALL of these conditions are TRUE:
 - Category is one of 5, 6
- Customer Earning Style: For every Y, give X points
- Number of Points (X): 1
- Step (Y): 10

Cart Rules

- **Pay with Cash On Delivery and receive a bonus of 100 points**

Customer will pay using the "Cash On Delivery" method and get 100 points.

- Type: Cart Rule
- Is Active: Yes
- Customer Groups: All
- Conditions:
 - If ALL of these conditions are TRUE:
 - Payment Method is Cash On Delivery
- Customer Earning Style: Give X points to customer
- Number of Points (X): 100

- **Order 3 or more products and receive an additional 10 points for every \$100**

Customer will add 3 or more products to the shopping cart and get 10 points for every \$100 spent.

- Type: Cart Rule
- Is Active: Yes
- Customer Groups: All
- Conditions:
 - If ALL of these conditions are TRUE:
 - Total Items Quantity equals or greater than 3
- Customer Earning Style: For every Y, give X points
- Number of Points (X): 10
- Step (Y): 100

Examples of Spending Rules

In this section we explain the possible configurations of typical rules.

- **For every 10 points give a \$1 discount**

Customer will get \$1 discount for every 10 points spent.

- Is Active: Yes
- Customer Groups: All
- For each spent X points: 10

- Customer receive Y discount: 1

- **For Retailer Customer Group: for each 10 points give a \$5 discount**

Only users from the Retailer Customer Group can get this discount.

- Is Active: Yes
- Customer Groups: Retailer
- For each spent X points: 10
- Customer receive Y discount: 5

- **Apply rule for only 4 specified products and for every 10 points give a \$1 discount**

Customer will get a \$1 discount for every 10 points spent, but only for products with SKU: 24-MB01, 24-UG01, MH01-L-Black, MJ04-L-Purple

- Is Active: Yes
- Customer Groups: All
- For each spent X points: 10
- Customer receive Y discount: 1
- Conditions:
 - If ALL of these conditions are TRUE:
 - SKU is one of 4-MB01, 24-UG01, MH01-L-Black, MJ04-L-Purple

- **Restrict use if less than 100 points and do not allow to use it for more than 30% of order cost**

Customer will get \$1 discount for every 10 points spent. But the minimum point amount which they can use in the Shopping Cart is 100 points. Also, they can't use more than 30% of the order cost

- Is Active: Yes
- Customer Groups: All
- For each spent X points: 10
- Customer receive Y discount: 1
- Spend minimum: 100
- Spend maximum: 30%

Examples of Notification Rules

In this section we explain the possible configurations of typical rules.

- **My Reward Points Description**

Rule will show a notification message in the "My Reward Points" section in the customer account.

- Is Active: Yes
 - Customer Groups: All
 - Show message on: Customer Account > My Reward Points
 - Message: For each spent \$10.00 you earn 10 Reward Points.
-

Refer a friend and receive points

Rule will show notification message in the "My Referrals" section in the customer account.

- Is Active: Yes
- Customer Groups: All
- Show message on: Customer Account > My Referrals
- Message: Refer a friend and receive points for their first 3 orders!

Order 3 or more products and receive double points!

Rule will show notification message in the Shopping Cart.

- Is Active: Yes
- Customer Groups: All
- Show message on: Cart Page
- Message: Buy 3 or more products and receive double points!

Reports

Reports in our extension breaks into two separate menu actions:

- **Refresh Statistics**, which is used for keeping statistics up to date, and concentrating data for displaying.
- **Reports**, which is used for visualization of data.

Both of these actions can be selected from menu drop-down in **Marketing -> Reward Points** section.

How to generate Reports on Rewards Policy

- If you did not gather statistics recently, go to **Marketing -> Reward Points** and select **Refresh Statistics**. This will force data gathering and after that will redirect you to the **Reports** page, updated with current data;
- Select a **Filters** on buttons bar, to show chart in regard to the following parameters:
 - **Period** - defines period, for which chart is built, and table shown;
 - **Customer Group** - defines customers' group, which is taken into account;
 - **Store View** - defines storeview, where activity is shown;
 - **Chart Step** - defines, which step is made for each row/chart bar.
- Click **Apply** and chart will be created.

Importing Transactions

Our extension allows data importing from other extensions using CSV data format.

Note

Unfortunately we can not add download sample file due to Magento restrictions.

You can get sample file simply by adding manual transaction at **Marketing -> Reward Points -> Transaction**, and then exporting it using **Export -> CSV** action of the Grid.

How to Import Transactions data

1. Extract data from your Reward Points extension to the CSV file, so it contains the following columns (each line of that file would be transactions):
 - **transaction_id** - a unique ID of each transaction. Can be simple counter;
 - **website_id** - shall contain not actual Website ID, but its code, according to your M2 storeview setup. *base* is the most common code, used by default.
 - **amount** - amount of points, that shall be uploaded by current transaction.
 - **amount_used** - amount of used points. Typically is set to 0. If there will be nonzero value, then difference between this and above will be uploaded.
 - **comment** - comment on current transaction.
 - **code** - code of action (upload in this case). It is recommended to pick up the same code for all uploaded transactions, so if upload was incorrect, transactions can be easily removed.
 - **is_expired** - whether uploaded points are expired. Useful, when uploading legacy transactions to maintain balance history.
 - **is_expiration_email_sent** - whether expiration email is sent. Typically 0. It does not really affects upload.
 - **expires_at** - date of expiration for uploaded in this transaction points.
 - **created_at** - date of actual transaction creation. Extremely useful for uploading legacy transactions.
 - **updated_at** - date of last transaction update. Typically should be equal to above.

Note

If you're importing data from Mirasvit Reward Points extension, you can use special [Migration script](#).

This will extract your data directly into `'/tmp/points_export.csv'` file. Feel free to change this SQL to suit your needs.

2. Go to **System -> Data Transfer -> Import**, and in drop-down menu **Entity Type** select **Reward Points Transactions**. You will see the following options:
 - **Import Behavior** - defines, which action should be performed. Currently supported:
 - **Add/Update** - adds or updates transactions (if **transaction_id** matches);
 - **Replace** - replaces transactions regardless of their existence.
 - **Delete** - deletes all transactions, whose ids are found in import file. Useful for cleaning up incorrect upload.
 - **Error Handling** (not labeled):
 - **Stop on Error** - stops importing, if errors found;
 - **Skip Error Entries** - upload data as is, valid data will be uploaded, others just dropped.

- **Allowed Errors Count** - used if **Stop on Error** selected, and defines number of allowed errors
- **Field separator**, **Multiple value separator** and **Fields enclosure** are CSV parameters, make them match your file format.
- **Select File to Import** - drop here CSV file, that you had exported in previous step;
- **Images File Directory** - used, if links to pictures are used in export file. Typically do not used at all.

Note

Magento 2 Import function can have restriction of the file size. By default it's only 2M. The best way to increase it - open your `.htaccess` file, and enter the following options (let us make limit of 40M):

```
php_value upload_max_filesize 40M    php_value post_max_size 40M
```

Select appropriate options, and press **Check Data**. This will conduct data validation.

3. After validation is finished, you will see under options section message with errors found, and if data validation passed (errors quantity did not exceed **Allowed Errors Count** parameter), **Import** button appears. Press it to import transactions.

Note

You can also upload only customer's balances. Just treat them as transactions - one per customer.

Building Referral Program

Our extension provides means for creation a small yet flexible referral program. Here is how it works:

1. Customer invites to the store other customer, using **My Referrals** section of his account. There's two ways to do it:
 - Send an invitation email, using **Send Invitations** form;
 - Send (or share in social network, for example) so-called **Referral Link**, displayed at **Your referral link** label.
2. Each invited customer receives unique link, that he should use for visiting. Our link contains hidden data, which are picked up by our controller, and recorded as potential referral section;
3. If customer registers during the same session, he is registered as referral. If he does not, he will be treated like regular user after session expiration - unless he uses referral link again (then another potential record for him will be created);
4. When referral registers, event **Referred customer signs up in the store** is triggered. If [Behaviour Rules](#) were binded to it, they will trigger.
5. When referral places an order, event **Order from a referred customer** is triggered. If [Behaviour Rules](#) were binded to it, they also will trigger.
6. You can also create other [Behaviour Rules](#) for other events, and connect them to referral policy, using **Is a Referral** special condition.
7. Combine different types of rules to create flexible referral program and extend your customer base with corresponding awards.

How To

Contents

- [How can I make customer earn points from buying specific products](#)
- [How can I prevent rules from multiply?](#)
- [How can I set up rule to award both referral customer, and it's referee](#)
- [How can I create so-called 'threshold' and 'milestone rule'](#)
- [How can I award customer with 'gift' points, that he can use on his first order](#)

How can I make customer earn points from buying specific products

It can be done in two ways - using Product or Cart rules.

If you use **Product** rule:

- Jump to **Conditions & Actions** tab, and add one (or several) conditions as follows:
 - If you need to restrict earning to some product Categories, use **Category** condition from **Product Attributes** subsection in drop-down list;
 - If you need to restrict earning to some particular product, use **SKU** condition from the same subsection;
- If you need to filter more than one product or category, then use one of following approaches:
 - Change **ALL** to **ANY** at master condition **If ALL of these conditions are TRUE** (this will make rule trigger, when at least one condition applies), and create a number of conditions for each category or product SKU;
 - Place to the condition comma-separated list of values, and select operator **is one of** instead of **is**.

If you use **Cart** rule:

- If you need to control not only products, but also their quantity:
 - Jump to **Conditions** tab and add a new condition. In drop-down list select **Product subselection**. It will produce nested conditions block with a master condition, which filters quantity of products selected;
 - In this nested block set up quantity, that should trigger the rule. If you need to award customer for buying 3 items of some product - then nested block master condition should contain **If total quantity equals or greater than 3 for a subselection of items in cart matching ALL of these conditions**;
 - Add to this nested block Category, SKU, or other attributes of products, that should trigger the rule - like described for Product rule;
- If you need just to control product presence in the Cart:
 - Jump to **Actions** tab, and create a new condition in **Apply the rule only to cart items matching the following conditions** section;
 - Add appropriate product attributes, that should trigger the rule - as described above.

Note

If you use Cart rule, do not use simultaneously both of described approaches - they will conflict. Create instead two Earning rules - you may also see next case to properly trigger them;

How can I prevent rules from multiply? For example: if there's two rules for products, and customer added to cart both - but only one rule should trigger.

- Jump to **General Information** tab and assign to these rules priorities according to product's promotional significance. Say, you have products A and B, and if they both in cart - reward for B should be applied. Rule for product A thus should have priority 20, and for A - 10;
- Set option **Stop further rules processing** to **Yes**. This will prevent rules with same or lower priority from triggering, even if their conditions apply.

How can I set up rule to award both referral customer, and it's referee (e. q. customer, who invited him).

It can not be done via one earning rule, you will need two of them - one for Referral, and the second for Referee.

Rule for Referral can be created this way:

- Create a new **Earning Rule** of **Behaviour** type at **Reward Points -> Earning Rule**;
- Jump to the **Conditions** (or **Conditions and Actions**) tab, and add new condition. In drop-down list, in **Customer** section select **Is Referral** condition and set it to **Yes**;
- Add other conditions - for example, if you need to set awards for certain order number, refer to *milestone rule* case, below in this section;
- Jump to **Actions** section, set points to earn, save and activate rule.

Rule for Referee can be created only using **Behaviour** type of Earning Rule:

- Create a new **Earning Rule** of **Behaviour** type at **Reward Points -> Spending Rule**, and jump to the **Conditions and Actions** tab;
- If you wish to award customer, when referral registers, pick event **Referred customer signs up in store**;
- If you wish to award customer, when referral places an order, pick event **Order from referred customer**;
- If you also need to award referee for a certain number of order, placed by referral - or make other limitations regarding referral - use conditions from **Referral Customer** subsection. For example, if you need to award points only for first referral's order, you need **Referral: Number of Orders** condition;
- Save and activate rule.

How can I create so-called 'threshold' and 'milestone rule', which will award customer with additional points on reaching sales amount or certain orders number

To award customers with additional points depending of their achievements, you need to use either **Cart**, or **Behaviour** rule. Both of them support conditions, that allows checking of customer's properties. Simply add

them to the **Conditions** tab to make rule threshold or milestone.

Here is the list of conditions, that can be used for the most cases:

- Award only with first order. Use condition - **Number of Orders** - and set it to **0**. It does not apply only to **Behaviour** rules, which use **Customer places an order** event - in this case you need to set **1** (because this event triggers after order is placed);
- Award customer for certain X-th order (say, 10-th). Use the same **Number of Orders** condition, but it shall be equal to X - 1 (for **Customer places an order** shall be equal to X);
- Award customer for reaching certain X points on his balance. Use two conditions **Balance points amount** - for lower limit, and upper limit (if applies).
- Award customer for spending X money in store. Use two conditions **Lifetime Sales** - for lower limit, and upper limit (if applies).

Not only customer overall achievements can be checked, but particular purchase as well. Use for that additional conditions, located under **Cart Attributes**, for rules of **Cart** type.

How can I award customer with 'gift' points, that he can use on his first order.

Customer can receive points immediately after registration using only **Behaviour** rule. Here is how it can be done:

- Create a **Behaviour** rule at **Reward Points -> Earning points** and on **Conditions & Actions** pick up event **Customer signs up in store**;
- Set **Customer Earning Style** to **Give X points to customer**, and set points amount to receive;
- If you wish to suppress frontend notification for this rule, add **Signing up** event at **Reward Points -> Settings -> Display -> Disable Behaviour Notifications**;
- Add other conditions, if you need to - for example, a check for Referral (see case above);
- Save and activate rule.

This rule will make customer receive points on registration stage, and they can be already spent on first order.

Migration from Magento 1 to Magento 2

Below are examples of script, which may be used for transferring data from Mirasvit Reward Points for Magento 1 to Mirasvit Reward Points for Magento 2.

To use script you need a help of PHP developer. Script may contain bugs and may not transfer all necessary data. So you may need to fit it for your needs. Migration service is not a part of our support service. If you need a help of our team, you'll need to buy a [migration service](#).

Click on the name of export script below to download:

- [mirasvit_rewards_m1_export.php](#) - script for exporting Rewards data to CSV file.

How to perform Migration

Follow these steps to correctly perform migration:

1. Run standard Magento 2 migration, and make sure, that customers and orders were transferred from your M1 store to M2;
2. Make sure, that Reward Points for Magento 2 is [installed](#) on your M2 store.
3. Modify `mirasvit_rewards_m1_export.php` script to match your setup. Typically, you just need to change `$exportFileName` variable for storing exported data (**full path is required**). Make sure, that this file **is not exist**.
4. Place `mirasvit_rewards_m1_export.php` to the root of your M1 store, and run it either:
 - From browser by opening URL
`http://store_m1.com/mirasvit_rewards_m1_export.php`, or
 - From SSH/Console: navigate to your store's root and run command
`php -f mirasvit_rewards_m1_export.php`
5. If script returned success, file with name, set at `$exportFileName`, will be created. Transfer it to your workstation.
6. Run step 2 of [Import](#) procedure.
7. Check migrated data.
8. When migration is completed, **remove migration script**.

Troubleshooting

- In some cases import script can crush on timeout or memory overloading error (due to large quantities of data). In this case just restart import script by reloading browser page. It will skip previously uploaded data, and proceed from the last unprocessed record.

How to upgrade extension

To upgrade the extension follow these steps:

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

Note

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

Run instead `composer update mirasvit/*` command. It will update all Mirasvit modules, installed on your store.

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

Deploy static view files

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

Disabling the Extension

Temporarily Disable

To temporarily disable the extension please follow these steps:

1. Login 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_Rewards` to disable the extension.
3. Login in to the Magento back-end and refresh the store cache (if enabled).

Extension Removal

To uninstall the extension please follow these steps:

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

Change Log

2.3.23

(2019-08-15)

Fixed

- Country condition for earning rules
 - Translation for expiration date
 - Display points for tier prices on product page
 - Issue when rewards form does not show in cart/checkout
-

2.3.22

(2019-08-07)

Fixed

- Compatibility with Amasty_ShippingTableRates
- Added condition "Additional Payment Method" to earning cart rules
- Prevent update of purchase data after order was created
- Error "Call to a member function getId() on array in Rewards/Model/Cron/Tier.php"
- Issue when total's spend row hides on cart update
- Active From/To fields saving for all rules
- Error "ReferenceError: rewardsCurrentUrl is not defined"

Improvements

- "activate transaction" cron task
-

2.3.21

(2019-07-18)

Fixed

- Possible division by zero
 - Points not loading on catalog page if redis enabled
 - Translation for product page
-

2.3.20

(2019-07-12)

Fixed

- Compatibility with extension Bss OrderAmount
- Calculation for fixed amount steps of earning rules
- Call to a member func on bool on the product page when Amasty_CommonRules is enabled

Improvements

- Display of product points using ajax
-

2.3.19

(2019-06-25)

Fixed

- Issue when Klarna payment method disappears after points were applied
 - Issue when unnecessary added to behavior notification emails
 - Behavior rule condition "Is Referral"
-

2.3.18

(2019-06-14)

Fixed

- Issue for Magento EE v2.2.5 when rewards discount is reset on place order (apply customer tax before discount)
-

2.3.17

(2019-06-12)

Fixed

- Points amount on product page for tier price
- Earn points do not display in checkout if spend points is not allowed
- Condition "Discount Used" in spending rules

Improvements

- Compatibility with Klarna Checkout
-

2.3.16

(2019-05-31)

Fixed

- Points discount for "tax applied after discount"
 - Facebook Like button loading for m 2.3.1 and remove private content for social buttons
-

2.3.15

(2019-05-27)

Fixed

- Points rounding on product page
 - Values of condition "Payment Method" for earning rule
-

2.3.14

(2019-05-24)

Fixed

- Points rounding on product page
- Points name on customer order view page

Improvements

- Added `{{order_increment_id}}` to earning rule history message
-

2.3.13

(2019-05-20)

Fixed

- Compatibility with Amasty Shipping Rates
 - Applying of reward points by "enter" button in checkout
-

2.3.12

(2019-05-15)

Improvements

- Compatibility with Klarna extension v6.1.0, Klarna Core v5.1.0
- Display format of earned points in backend on order view page

Fixed

- "Minimum Order Amount" does not consider Rewards discount
 - Display of rewards discount for multi currency stores
 - Issue when messages on product page hide very fast
-

2.3.11

(2019-05-03)

Improvements

- Added ability to delay usage of rewards points
 - Improve tier update cron job
-

2.3.10

(2019-04-23)

Fixed

- Allow to use percent in monetary step of spending rule
-

2.3.9

(2019-04-19)

Fixed

- Remove rewards expiring notification subscription from Newsletter Subscription page (left only on My Reward Points > History page)
-

2.3.8

(2019-04-17)

Fixed

- Label "Rewards Discount" overrides coupon discount label in cart
 - Points does not applies to Grand Total if tax does not include in spending rules
-

2.3.7

(2019-04-10)

Fixed

- Display of tier progress bar for rtl theme
 - Compatibility with m 2.3.1
 - Display Rewards discount in backend order creation form
-

2.3.6

(2019-03-27)

Fixed

- Points calculations for min/max price of bundle product
 - Issue when condition "Number of reviews" includes pending reviews
 - Spend maximum calculates incorrect
 - Remove unnecessary rewards calculations
-

2.3.5

(2019-03-21)

Fixed

- Compatibility with Magecomp_Surcharge
 - Added Rewards discount to "Checkout Totals Sort Order"
 - Display of Rewards discount in backend print pdf
-

2.3.4

(2019-03-19)

Fixed

- Display of rewards discount in backend and emails
 - Rewards information for SOAP API order request
-

2.3.3

(2019-03-15)

Fixed

- Display of rewards discount for order
-

2.3.2

(2019-03-14)

Fixed

- Translation for the date range in transaction table
- Compatibility with Aheadworks_OneStepCheckout

Improvements

- Added new api:
 - /V1/rewards/products/points/get
 - /V1/rewards/products/points/multiplicity
-

2.3.1

(2019-03-12)

Fixed

- Rewards discount does not includes in Paypal total
 - error "Notice: Undefined index: int NOT NULL DEFAULT 0" for Magento 2.3.x
 - Js error "Mirasvit_Rewards/js/product/view.min.js net::ERR_ABORTED 404 (Not Found)" on backend order creation page
-

2.3.0

(2019-03-07)

Fixed

- Issue when rule save wrong data if message starts from variable
- Condition "Coupon used is no" does not work in spending rules
- Js error on product page for custom product types
- Points calculations on paypal preview checkout

Improvements

- Moved rewards discount to totals
 - Compatibility with AwStoreCredit and AmastyGift extension
 - Compatibility with Magecomp Paymentfee v1.0.5
 - Added new api:
 - V1/rewards/product/points
-

2.2.34

(2019-02-07)

Fixed

- Points calculations for bundle product with special price
- Fixed points calculations for bundle product with special price
- Added new api:
 - V1/rewards/:cartId/apply/:pointAmount

Improvements

- Added ability to save "Logo" of Tier per storeview
 - Added ability to save "Front Name" of Spending Rule per storeview
-

2.2.33

(2019-02-06)

Fixed

- Disable coupon show success message after applying
 - Bundle description does not update when no product rule
 - Compatibility with Amasty Free Gift
 - Product amount in minicart does not update correctly
 - Added new api:
 - V1/rewards/transactions
 - V1/rewards/purchase/:orderId
 - V1/rewards/balances
 - V1/rewards/balances/:customerId
 - V1/rewards/mine/purchase/:orderId
-

2.2.32

(2019-01-23)

Fixed

- Update Facebook Like script
-

2.2.30

(2019-01-22)

Fixed

- Encoding in field min/max for spending rule(percent amount applies as a number)
 - Validation of min/max for spending rule
-

2.2.29

(2018-12-26)

Fixed

- Error when session stores in Redis and rewards notification returns error message
- Points calculations for earning rule 100:100
- Tier points calculates incorrect on Magento 2.2.7
- Compatibility with Reward Points, Magecomp Surcharge and Aheadworks Store Credit

Improvements

- Added ability to upload logo to tier
 - Moved rewards shipping discount from item to address shipping discount
-

2.2.28

(2018-12-12)

Fixed

- Error on cart page "Cannot read property 'replace' of undefined"
 - Compilation error on Magento 2.1.x
 - Images in tier description
-

2.2.27

(2018-12-10)

Fixed

- Customer created from checkout get points(Product Rule) for order
- Points calculations for tax based on "Unit Price"
- Reward notification display wrong amount of earned points
- Related products on product page have the same amount of points as main product
- Issue when points do not show for product with price less \$1 and qty more then 1
- HTML tags in earning message on cart page

Improvements

- Added translation for earning rule "Display Name"
-

2.2.26

(2018-12-04)

Fixed

- Some minicarts do not update products amount when product was removed from cart page
 - Wrong number of referrals on account page
 - Missing earning rule conditions
 - Compatibility with Reward Points, Magecomp Surcharge and Aheadworks Store Credit
 - Issue when Cart rule condition "Total Items Quantity" does not work
 - Unable to export customer grid with tiers
-

2.2.25

(2018-11-29)

Fixed

- Issue when tier calculation process includes spend points
 - Compatibility with Magento 2.3.0
-

2.2.24

(2018-11-27)

Fixed

- Unable to use "Zero Subtotal Checkout" with Reward Points(started from 2.2.20 version)
- Earned points do not cancel on order cancellation
- Email translation variables
- Issue "tier submenu Delete does not work"
- Issue when reward points do not calculate correct for configurable product options with minimum qty greater then 1
- Issue when points do not change on catalog page for different options

Improvements

- Added ability to translate product rule notification
-

2.2.23

(2018-11-22)

Fixed

- Issue when rewards discount does not apply to totals
 - Issue with the incorrect points calculating for the configurable product options on the catalog page
-

2.2.22

(2018-11-21)

Fixed

- Points calculations on product page when tier and special prices are set
 - Points calculations on product page when catalog price rule price displays
-

2.2.21

(2018-11-19)

Fixed

- Tier update events triggered only for the first update
- Points calculation on product page after update qty amount
- Fixed error "Exception message: Notice: Undefined index: 1.0000" during customer registration

Improvements

- Compatibility with MageComp Surcharge
 - Compatibility with Aheadwords Store Credit
-

2.2.20

(2018-11-13)

Fixed

- Notice: Undefined offset ... Rewards/Observer/BundlePriceConfig.php
- Compatibility with Faonni_Price
- Unnecessary loading of Facebook script on product page
- Incorrect transaction amount for expiration emails.

- Wrong amount of points for product tier price on product page
- Option "based on sum of spend \$" does not take into consideration value of option "take into account only last"

Improvements

- Added calculation options for tier upgrade process based on spend amount of \$
 - Compatibility with Amasty Gift Cart. Wrong max allowed points.
 - Added events to behavior rule: "Customer tier up" and "Customer tier down"
-

2.2.19

(2018-10-30)

Fixed

- Compatibility with Faonni_Price extension
- Error "fjs does not exist"
- Tax calculation for bundle products

Improvements

- API. Added "created_at" field to transactions response
 - API. Added min/max points information to totals info
-

2.2.18

(2018-10-29)

Fixed

- Points calculation when rule limited with maximum
 - Added option "Show social Button block on category page"
 - Fatal error for out of stock configurable products
-

2.2.17

(2018-10-26)

Fixed

- Compatibility with some 3rd party extensions, mini-cart is not updated with main cart
 - API. Added rewards information to totals
-

2.2.16

(2018-10-25)

Fixed

- Points calculations for webapi
 - Calculation points for simple product's options
-

2.2.15

(2018-10-23)

Fixed

- Points calculations on product page for advanced prices
- "Application ID" does not display after save

Improvements

- Added new API:
 - GET /V1/rewards/mine/balance
 - GET /V1/rewards/mine/transactions
 - POST /V1/rewards/mine/apply
-

2.2.14

(2018-10-17)

Fixed

- Rounding for Earning points
-

2.2.13

(2018-10-12)

Fixed

- ReferenceError: rewardsFacebookApiVersion is not defined
- "Subtotal" condition in rules
- Points calculation for multi currency store for bundle and configurable products
- Fixed earning points calculations for option "Apply Customer Tax"
- Amasty Shipping Rates do not show in backed during order creation

- Error "Missing argument 1 for Mirasvit\Rewards\Model\Spending\Rule::getSpendPoints()"
 - Tier description and emails for storeview
 - Tier description for storeview
-

2.2.11

(2018-09-26)

Improvements

- Added ability to switch between tiers based on sum of spent \$
- Added option "After login to account redirect a customer to My Reward Points section"
- Added validation for block modifications
- Added condition "payment method" to 2.2.6(Magento removed it from conditions)

Fixed

- Points on product page displays according to current currency instead of base currency
-

2.2.10

(2018-09-11)

Fixed

- Installation
 - Styles for option "Force to apply styles"
 - Compatibility with Ebizmarts_SagePaySuite
 - Error "Call to a member function getID() on null in Service/Customer/Tier.php:169"
-

2.2.9

(2018-08-31)

Fixed

- Fatal error: Uncaught Error: Cannot instantiate interface Mirasvit\Rewards\Api\Repository\TierRepositoryInterface ..."
-

2.2.8

(2018-08-29)

Fixed

- Newsletter unsubscribe does not deduct points
- Displaying of Share and Referral tabs in account

Improvements

- Added social buttons on category page
-

2.2.6

(2018-08-28)

Fixed

- Sending multiple emails using the TransportBuilder class causes an Zend_Mail exception
- Update FB API version to 3.1
- Points spending style "Fixed"
- Points calculations when earn points approved on status "Pending" and spend points less then "For each spent X points"
- Actions of Cart Rules

Improvements

- Added FB API version number to config
 - Added "All Groups" to rule settings
-

2.2.5

(2018-08-17)

Fixed

- Displaying of Rule's history message for multiple store view
-

2.2.4

(2018-08-16)

Fixed

- Ability to save history and email messages for Behavior Rules for store view
- Product review message
- Social buttons

- Backend create order error - "Warning: Division by zero"
 - Tax excluded twice on product page
-

2.2.3

(2018-08-08)

Fixed

- Issue with newsletter subscription by cron tier
-

2.2.2

(2018-08-01)

Fixed

- Compilation error
 - Customer attribute "Rewards Tier" is required
-

2.2.0

(2018-07-31)

Feature

- Tiers

Fixed

- Issue when "Minimum Order Amount" was ignoring
 - Issue when rule "Display Name" showing for all groups
-

2.1.34

(2018-07-20)

Fixed

- Added points for the order when customer created an account at the end of checkout
-

2.1.33

(2018-07-09)

Fixed

- Displaying points for bundle product
 - Guest address is not saving in checkout
-

2.1.32

(2018-07-03)

Fixed

- Points calculations for Newsletter Subscription
 - Points for tier price on product page
 - Compatibility with other Twitter plugins
-

2.1.31

(2018-06-07)

Fixed

- Product page error "Uncaught SyntaxError: Unexpected end of JSON input"
 - Expiration email sending
 - Points calculations for multi currency stores
-

2.1.30

(2018-05-18)

Improvements

- Added api for transactions
-

2.1.29

(2018-05-04)

Fixed

- Cached social block does not display "Pin it" button
- Compatibility with Aheadworks_OneStepCheckout
- Referral link for stores in subfolder

Improvements

- Added option that allows to display pending rewards transactions
 - Added Prices to product rule
 - Added "Lifetime Spent Points" condition to behavior rules
-

2.1.28

(2018-04-26)

Fixed

- Compilation error
-

2.1.27

(2018-04-25)

Fixed

- Date filter does not work in transaction grid
 - Unable to add points in backend to several customers
 - Call to a member function getEntityId() on null
-

2.1.26

(2018-04-13)

Fixed

- Issue when inactive period does not save
- Error with message "There is no information about associated entity type "customer_group""
- Color does not change on product page
- Unnecessary calculations for bundle products

Improvements

- Added option that adds Rewards style to pages if theme ignore it

2.1.25

(2018-04-03)

Fixed

- Behavior rules with order conditions
-

2.1.24

(2018-03-16)

Fixed

- Calculation of minimum points
 - Calculation of earning points for virtual cart
 - Calculation of spending points for option "Include tax for Spending rules"
 - Compatibility with Swissup Firecheckout
-

2.1.23

(2018-03-16)

Fixed

- Report Plugin
-

2.1.22

(2018-03-12)

Fixed

- Uncaught TypeError: Argument 1 passed to `Magento\Quote\Model\Cart\Totals::setExtensionAttributes()` must be an instance of `Magento\Quote\Api\Data\TotalsExtensionInterface`, instance of ...
-

2.1.21

(2018-03-07)

Fixed

- Miltiorders for one cart
-

2.1.20

(2018-03-07)

Fixed

- Fixed issue with incorrect email sender in multistore configuration
-

2.1.19

(2018-03-06)

Fixed

- DI Compile (reports module)
-

2.1.18

(2018-03-05)

Improvements

- Allow to earn points for joining affiliate program (Mirasvit_Affiliate)
-

2.1.17

(2018-03-01)

Fixed

- Miltiorders for one cart
 - Compatibility with Taxjar SalesTax module
 - Points are not assigned by "sign up" rule for Magento 2.2.2
-

2.1.16

(2018-02-20)

Fixed

- Fix error during setup:di:compile
-

2.1.15

(2018-02-19)

Fixed

- Incorrect calculation of base discount (Grand Total (Base) and Grand Total (Purchased) are different)
 - Add customer conditions to cart and product earning rules
-

2.1.14

(2018-02-09)

Fixed

- issue with wrong report version requirement
-

2.1.11

(2018-02-09)

Fixed

- Error "Cannot read property 'sectionLoadUrl' of undefined" that appear on slow connection in checkout
 - Compatibility with Amasty checkout
 - Error "The requested Payment Method is not available" for checkout with Grand Total equal to zero
 - Fixed magento 2.2.2 issue <https://github.com/magento/magento2/issues/12993> 'Argument 1 passed to Magento\Quote\Model\Cart\Totals::setExtensionAttributes() must be an instance of Magento\Quote\Api\Data\TotalsExtensionInterface, instance of Magento\Quote\Api\Data\AddressExtension given'
 - Added a formkey and max number of allowed invites to referral invitation form
-

2.1.10

(2018-01-31)

Fixed

- Points are not calculating correctly when changing shipping in cart
-

2.1.9

(2018-01-23)

Fixed

- In some cases, incorrect reward points discount calculation. Module rounds discount amount to integer and does not allow to have a fractional discount amount. Affects versions from 2.1.6.
 - Registration rule is triggered on customer saving. If customer was created before registration rule and customer does not have transaction for this rule he takes points for registration rule on reset password
 - In some cases Twitter returns "Connection timed out"
-

2.1.8

(2018-01-05)

Fixed

- Issue when rewards section was updated twice
-

2.1.6

(2017-12-27)

Fixed

- Social buttons on the account page were incorrectly displayed
- In some cases, when coupon and points were used at the same time, order total was negative

Improvements

- Integration with Plumrocket SocialLoginPro
-

2.1.5

(2017-12-20)

Fixed

- Fixed an issue with slow category page load time. If a category page has a lot of configurable products with many sub-products, it took a long time to load the page.
-

2.1.4

(2017-12-19)

Fixed

- In Magento 2.2.2, if customer attempts to register, he sees the error "PHP Fatal error: Uncaught Error: Call to a member function getId() on null in .../app/code/Mirasvit/Rewards/Observer/BehaviorCustomerRegisterSuccess.php:64"

2.1.2

(2017-11-30)

Improvements

- Compatibility with Magento 2.2.1
-

2.1.1

(2017-11-28)

Improvements

- Compatibility with Aheadworks onepagecheckout
-

2.1.0

(2017-11-20)

Fixed

- Points calculation on the bundle product page
 - Code improvements
 - Translation
-

2.0.18

(2017-11-15)

Fixed

- Transaction grid
 - Earning points for sign up events
 - Styles for social buttons
-

2.0.17

(2017-11-02)

Fixed

- Issue with bundled price observer
-

2.0.16

(2017-10-25)

Improvements

- Added Base Price, Final Price and Special Price to spending rules
- Added condition "Discount Used" to spending rules

Fixed

- Rules assigning to website
-

2.0.15

(2017-10-18)

Improvements

- Added referral group to behavior rule condition for event "Customer signs up in store"

Fixed

- Rules assigning to websites
-

2.0.14

(2017-10-13)

Fixed

- Fixed twitter url
 - Translation
-

2.0.12

(2017-10-10)

Improvements

- Social button
-

2.0.11

(2017-10-09)

Fixed

- Date validation
 - Points for "Not Visible Individually" products
 - Tax calculation
 - Acl for "Transaction"
-

2.0.10

(2017-09-28)

Improvements

- Magento 2.2 Compatibility Fix
- Allowed to change Customer's Group on Behaviour Rule triggering
- Added Is Referee condition to Behaviour Rule
- Improved documentation

2.0.9

(2017-09-25)

Improvements

- Added earning rule of behavior type for subscription on push notification

Fixed

- Fixed an issue with actions rule
-

2.0.8

(2017-09-21)

Fixed

- Bugs
-

2.0.7

(2017-09-19)

Improvements

- Rewritten documentation
-

2.0.6

(2017-09-18)

Improvements

- Added conditions "Coupon Used" and "Coupon Code" to Spending Rule and Cart Earning Rule

Fixed

- Solved XSS issue
 - Reward Points for Shipping Tax
-

2.0.5

(2017-09-11)

Fixed

- Single spending rule with payment method
 - Compatibility with Magento 2.2.0rc
-

2.0.4

(2017-09-06)

Fixed

- Points displaying on product page
-

2.0.3

(2017-08-31)

Fixed

- "Customer signs up in store" for checkout
 - Points for orders mass actions
 - Rule saving for multistore
 - Fixed loading of unnecessary social scripts
-

2.0.2

(2017-08-21)

Fixed

- Fixed referral sender for multistore
-

2.0.1

(2017-08-18)

Fixed

- Notification rule saving
-

2.0.0

(2017-08-16)

Improvements

- Refactoring
-

1.1.49

(2017-08-16)

Final version for Magento 2.0.x

1.1.46

(2017-08-03)

Fixed

- Referral sign up event
-

1.1.45

(2017-07-21)

Fixed

- Updating of Reward points on product view page
-

1.1.44

(2017-07-18)

Fixed

- Behavior rules event "Customer signs up in store"
 - Translation
-

1.1.43

(2017-07-12)

Fixed

- Different issues with Behavior Rules

Improvements

- Added option that allows to include/exclude discount from Earning rules
 - Added rewards variables (getRewardsEarnedPoints(), getRewardsSpentPoints(), getRewardsSpentAmount()) to email templates of order, invoice, creditmemo
-

1.1.42

(2017-06-30)

Fixed

- Different issues with Earning Rules
-

1.1.40

(2017-06-26)

Fixed

- Calculation of points for Spending Rules with "Spend minimum"
-

1.1.37

(2017-06-14)

Fixed

- Prevent errors during refresh statistic
-

1.1.36

(2017-06-13)

Fixed

- Rule conditions
-

1.1.35

(2017-06-12)

Fixed

- Displaying of Reward points for grouped products

Improvements

- Displaying of spending points in order in backend
-

1.1.34

(2017-06-07)

Fixed

- Rule conditions
- Points cancelation in backend order
- Translation for transaction's auto comments.

Improvements

- Added option that allows to control how to round earning points
-

1.1.33

(2017-05-25)

Fixed

- Calculation of used points
- Compatibility with Mageplaza one step checkout
- Calculation of Earning Product Rule for discount

Improvements

- Added option that allows to display the highest possible amount of points for configurable product
-

1.1.32

(2017-05-06)

Fixed

- Composer php
-

1.1.31

(2017-05-03)

Fixed

- Default value for "Subscription to Points Expiring Notification"
-

1.1.30

(2017-04-18)

Improvements

- Added ability to unsubscribe from points expiring notification
 - Added "Points spending style" option to Spending Rule
 - Allow to use percent in "Customer receive Y discount" field in spending rule
-

1.1.29

(2017-04-06)

Improvements

- Compatibility with PSP MultiSafepay

Fixed

- "Sign Up for Newsletter" rule for guest
-

1.1.28

(2017-03-17)

Fixed

- Multi store support for email templates
-

1.1.27

(2017-03-09)

Fixed

- Reports
-

1.1.25

(2017-03-06)

Improvements

- Reports
-

1.1.24

(2017-02-20)

Fixed

- Moved "Display options" settings from global to store view settings
 - Renamed option "Show rewards points on frontend" to "Show rewards points on category page"
 - Added option "Show rewards points menu on frontend"
 - Fixed share messages
-

1.1.23

(2017-02-10)

Features

- Added Facebook share button
- Added option to hide rewards menu from frontend
- Added new condition "Is Rewards Points used" to earning cart rule
- Added ability to import/export rewards transactions
- Added options to configure include/exclude tax for Earning and Spending rules
- Added ability to add notification to product earning rule

Fixed

- Socail urls for multistores
 - Category condition's operator "is" for product rule
 - Replaced old Facebook tag with valid html5 tag
 - SmartWave Porto theme compatibility
 - An issue when "Product's earning points do not display on Home page "
 - UTF characters in "Point Unit Name"
 - "Customer is not active for long time" condition
 - Minicart's total and item number do not update correctly
 - Actions for cart Earning Rules
 - An issue when "Product points as currency equivalent" displays in checkout
 - Points for Payment rule method
 - Compatibility with theme Infortis Ultimo
 - Included Fixed Product Tax in rewards spending points
 - Compatibility with MageCheckout Onestepcheckout v2
-

1.1.22

(2016-12-05)

Fixed

- Cart rule condition "Payment method"
 - Wrong calculation for Spending rules' "Spend minimum"
-

1.1.21

(2016-11-29)

Fixed

- Displaying of Rewards points for grouped and bundle products

Features

- Added earning behavior rule event "Customer places an order"
 - Allowed display on Product pages points as money equivalent
-

1.1.20

(2016-11-18)

Fixed

- Calculation of Spending rules

1.1.19

(2016-11-08)

Fixed

- Tables formatting for small screen or cell phone
-

1.1.18

(2016-11-08)

Features

- Added Rewards block to admin create order form

Fixed

- minor bugs
-

1.1.17

(2016-11-04)

Fixed

- Minicart total and item number do not update correctly
 - Social js are not loading deferred
-

1.1.16

(2016-10-24)

Features

- Added ability to show AddThis buttons in Rewards social block

Fixed

- trigger_recollect caused infinite loop
-

1.1.15

(2016-10-13)

Fixed

- Compatibility with Bss/DeferJS
- Compatibility with onestepcheckout
- Earning rule conditions do not display all attributes
- Magento page cache js error
- Conflict with Amasty Rules
- Undefined variable in social buttons block
- Integration with MageCheckout Onestepcheckout v2

Features

- Added Rewards block to checkout page
 - Added integration with AddThis
 - Added ability to create custom behavior rules
-

1.1.14

(2016-09-09)

Fixed

- Conflict with Amasty Rules
-

1.1.13

(2016-09-07)

Fixed

- Conflict with Amasty Rules
 - Issue when js error appears, when social buttons are disabled
-

1.1.12

(2016-09-05)

Fixed

- Issue when cart rules with "Fixed Amount Discount for Whole Cart" show error in cart
-

1.1.11

(2016-08-31)

Improvements

- Compatibility with 2.1.1
-

1.1.10

(2016-08-05)

Features

- Added settings "Show number of points on the product page"
-

1.1.9

(2016-08-03)

Fixed

- Issue when reward points are not being deducted after using them through Paypal
- Issue when earning rule not working for referral

Features

- Added integration with MageCheckout Onestepcheckout
-

1.1.8

(2016-07-06)

Features

- Added integration with MageStore Onestepcheckout
-

1.1.7

(2016-07-06)

Improvements

- Added new conditions "Is referral"
 - Added variables for Notification Rule message
-

1.1.6

(2016-06-24)

Improvements

- Support of Magento 2.1.0
-

1.1.5

(2016-06-21)

Fixed

- Issue during discount coupon codes apply
-

1.1.4

(2016-06-13)

Features

- Show message about product rule points for grouped products
-

1.1.3

(2016-06-08)

Fixed

- Incorrect points calculation when we remove items from the cart
 - Incorrect URL in spending rules grid (effects from 1.0.13)
-

1.1.2

(2016-06-02)

Fixed

- Customer sign up earning rule does not allow save customer in backend
 - If a rule is disabled in Earning Rules/Spending Rules, you have no way to find it back
 - Error when we give points for newsletter subscription
-

1.1.1

(2016-06-01)

Fixed

- Styles for reward points on home page
-

1.1.0

(2016-05-31)

Improvements

- Show the amount of points able to earn in the product listing page
-

1.0.13

(2016-05-30)

Improvements

- Ability to filter grids by website
-

1.0.12

(2016-05-27)

Improvements

- Added website column to spending/earning/notification grid
-

1.0.11

(2016-05-17)

Improvements

- Improved twitter script loading
-

1.0.10

(2016-05-10)

Fixed

- Fixed formatting of referral invitation
-

1.0.9

(2016-05-06)

Improvements

- Improved notification rules messages for different currencies

Fixed

- Fixed "Pin It" button
-

1.0.8

(2016-04-26)

Fixed

- Fixed rules conditions issue
 - Fixed currency conversion
-

1.0.7

(2016-04-11)

Improvements

- Show current rewards points in my account dropdown

Fixed

- Issue with menu
 - Fixed issue with saving of rules if magento use custom locale
-

1.0.6

(2016-04-04)

Improvements

- Improved referral link

Fixed

- Fixed issue with rewards for tweets.
-

1.0.5

(2016-03-28)

Improvements

- Styles at shopping cart page
-

1.0.4

(2016-03-24)

Fixed

- Issue for not working condition "total items quantity"
 - Add protection from division on zero error
 - JS bug on in the social buttons
-

1.0.3

(2016-03-21)

Fixed

- When admin creates a new transaction in backend, he must be required to select a customer
- Dont show message when 0 points are earned
- Issue with saving of Conditions in Spending Rules

1.0.2

(2016-03-14)

Fixed

- Solved issues with notification emails
-

1.0.1

(2016-03-01)

Fixed

- Issues with PHP 7
-