

Store Credit & Refund Manual

Getting Started

Welcome to the **Store Credit & Refund Documentation**.

Here, you will find everything you need to set up **Store Credit** and create a flexible policy for crediting and quick refunding.

Our extension allows you to convert customers' returns into store credit. Each order that opted for a refund can be easily turned to a special deposit on the customer's balance, and immediately used for shopping.

Go ahead, dive in!

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

How to install the extension

1. Back up your store database and web directory.
2. Log in to the SSH console of your server and navigate to the root directory of the Magento 2 store.
3. Copy the installation instructions from the page [My Downloadable Products](#) to the SSH console and press ENTER.

Note

As of Magento 2.3* the GraphQL module can be also installed. The command is `composer require mirasvit/module-credit-graph-ql`

1. Run the command `php -f bin/magento module:enable Mirasvit_Core Mirasvit_Report Mirasvit_Credit` for enabling the extension.
2. Run the command `php -f bin/magento setup:upgrade` to install the extension.
3. Run the command `php -f bin/magento cache:clean` to clear cache.
4. Deploy static view files

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

Install the extension for Hyva

1. Run the command

```
composer require mirasvit/module-credit-hyva:*
```

2. Enable installed Hyva modules:

```
bin/magento module:enable Hyva_MirasvitCredit
```

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

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

4. Compile the code of the installed extension:

```
bin/magento setup:di:compile
```

5. Run the command below to clean the cache:

```
bin/magento cache:flush
```

Quick Start

Our **Store Credit & Refund** extension for M2 is a simple yet powerful extension that allows customers to build complex policies with minimal effort. Most of its functions work out-of-box, and once you've installed it, you can refund your orders to the Store Credit balances.

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

1. You can control your customers balances using the [Balances](#) section, and quickly correct them using [manual transactions](#).
2. You can allow customers to easily replenish their balances with special [Credit Products](#), which you can create for your customers.
3. You can promote your products and earn your customers' loyalty by creating cashbacks and special deposits on purchases. Our [Earning Rules](#) can help you to construct them.
4. You can monitor the efficiency of your crediting and refund policy with our [Reports](#) and [Transactions Grid](#).
5. You can also tune up your Store Credit extension, using our [Settings](#) section in Configuration.

- Combine all features above and build a flexible crediting policy, which will make your store more intuitive and pleasant for your loyal customers.

This should be a good beginning.

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

Store Credit Settings

This section describes all settings that you can use for configuring your crediting and refunding policy.

They are located at **Sales -> Store Credit -> Settings** and divided into the following subsections:

- [General Settings](#)
- [Store Credit Calculation](#)
- [Email Notification Settings](#)

General Settings

Option	Description
	Global - means that customer balance stores in base store currency and all other currencies convert to the base currency, i.e., all currencies have the same balance.
	Per Currency - means that the customer has a separate balance for each currency.
Share Customer Balances	For example, you have a multistore and want the currency of store credit to be converted - Global option
	To merge non-base currency balances into the base currency balance, use the console command <code>bin/magento mirasvit:credit:merge-balances</code>
Enable Refund To Store Credit (the default value for credit memo)	Enables refunding to the Store Credit balance
Auto apply Store Credit for the shopping cart	Allow for automatically applying the maximum possible amount from the Store Credit balance to the cart
Enable Sending Credit to Customer	Enables/disables the possibility of sending credits to other customers. Important Note: After changing this option, you need to purge your cache.
Enable Quick Data Bar	Activates the display on the Transactions page of the store admin the bar with info on credits: total balance, received, spent, transactions. Choose the time period for the displayed data from the drop-down menu on this bar.

Store Credit Calculation

Option	Description
Include tax to Store Credit	Includes taxes, when applying a credit amount to the order total
Include shipping to Store Credit	Includes shipping fees (if any selected), when applying a credit amount to the order total

Option	Description
The order of Store Credit applying	Ability/ priority to change the order of Store Credit's applications

Email Notification Settings

Option	Description
Email Sender	Sets the email that will be set as the sender the of Store Credit's email notification. It is selected as one of standard, listed in Configuration -> General -> Store Email Addresses .
Enable balance update email	Allow for sending a notification to the customer, when his/her Store Credit balance is updated.
Template of Balance Update Email	Sets a template for a notification, which is enabled/disabled above. More info can be found in the How-To corresponding subsection .

Advanced

These options are useful for stores that have non-standard Order and Credit Memo numbering. They are used for correct parsing of the Order and Credit Memo numbers in the Store Credit transactions of the customer's account.

Option	Description
Order Pattern	PHP regular expression for stores that use non-standard order numbers
Credit Memo Pattern	PHP regular expression for stores that use non-standard Credit Memo numbers
Allow for selection of payment method for zero orders	Control the visibility of payment methods when credits are used.

Customers' Balances

Customers' Balance Grid

All customers' balances can be seen in one place, using the **Sales -> Store Credit -> Customers** grid.

The grid contains the following information:

- **Customer Name** - customer name.
- **Customer Email** - customer email.
- **Balance** - current Store Credit balance of the customer.
- **Updated At** - last activity on the customer's balance.

Each record is a shortcut to the corresponding customer account, where you can see more detailed information on store credit transactions.

Customer Balance Tab

Our extension allows you to access and manage all customer balance information directly from his/her account record using the additional **Store Credit** tab.

Veronica Costello

← Back

Delete Customer

CUSTOMER INFORMATION

Customer View

Account Information

Addresses

Orders

Newsletter

Product Reviews

Billing Agreements

Wish List

Store Credit

Balance Information

Current Balance:

Last Change:

Subscribed to email notifications:

Transaction #	Date
3	Dec 5, 2017, 11:01

This tab contains a summary of the balance under **Balance Information** label, where you can see:

- **Current Balance** - current cash sum deposited in the balance.
- **Last Change** - date of the last transaction related to this customer.

- **Subscribed to email notifications** - allows you to see whether a customer opted to receive email updates. The additional **Add New Transaction** button is a shortcut to a simplified [Manual Transaction](#) edit dialog, which allows you to quickly add or withdraw funds from the customer balance.

There's also a filtered Transactions Grid, which contains all transactions related to this customer.

Transactions

All **Store Credit** transactions are located in the **Sales -> Store Credit -> Transactions** section. Each of them provides the following information:

- **Transaction #** - ID of the transaction.
- **Customer Name** - the name of the customer, to whom the refund of Store Credit was issued.
- **Customer Email** - email of that customer.
- **Date** - date of transaction. The filter, available from the grid's header, allows you to show customer transactions only for the selected period.
- **Balance Change** - the amount added or withdrawn from the balance by the current transaction.
- **Balance** - total Store Credit balance of the customer.
- **Action** - action, by which the transaction was created. Below are the following possible cases:
 - **Refunded** - adding to the balance created using Credit Memo, with a Refund to Store Credit option.
 - **Earning** - adding to the balance as the result of triggering of an [Earning Rule](#).
 - **Purchased** - adding to the balance by purchasing a special [Credit Product](#).
 - **Refilled** - adding to the balance by purchasing a special virtual [Refill Product](#).
 - **Used** - withdrawal from the balance as payment to order.
 - **Manual** - manual transaction. It can be both adding and withdrawal. More on this case can be found [below](#).
- **Additional Message** - provides additional transaction information, i.e. *Order id*, *Credit memo id*, and shortcuts to appropriate documents.
- **Has Been Notified?** - displays if the balance update email has been sent to the customer.

Creating Manual Transaction

A manual transaction can be created both for adding and withdrawal from the balance. Generally, it is used for mass actions and correcting customer balances.

However, it can also be used as a powerful tool for promotional purposes - such as crediting customers for Black Friday, for example.

To create a manual customer transaction, perform the following steps:

1. Go to the **Sales -> Store Credit** section, and click the **Add New transaction** button.
2. **Store Credit Balance Change** - the sum of money that should be added or withdrawn from the balance.
3. **Additional Message** - additional information about a new transaction. It will be displayed on the Grid, so it should be logical.
4. **Currency** - specifies the currency of the transaction.
5. In the customer's list below, select the customers that are eligible for this action.
6. Save the transaction to apply it (if you have selected more than one customer, it will be propagated to each as different transactions).

Importing Transactions

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

Note

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

You can get a sample file by visiting the Transactions page 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 that it contains the following columns (each line of that file would consist of transactions):

- **transaction_id** - a unique ID of each transaction. Can be simple counter;
- **customer_name** - first and last name of the customer;
- **customer_email** - customer email;
- **updated_at** - date and time the transaction was updated;
- **balance_delta** - change on the credit balance;
- **balance_amount** - amount of funds available;
- **action** - type of the transaction (refill, refund, use);
- **message** - additional message sent alongside the transaction;
- **is_notified** - displays if the notification was sent;

- **website_id** - shall contain not actual Website ID, but its code, according to your M2 store view setup. *base* is the most common code used by default

Note

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

It 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 on the drop-down menu **Entity Type**, select **Store Credit Transactions**. You will see the following options:

- **Import Behavior** - defines which action should be performed. Currently supported:
 - **Add/Update** - adds or updates of transactions (if **transaction_id** matches);
 - **Replace** - replaces transactions regardless of their existence.
 - **Delete** - deletes all transactions with no IDs on the import file. Useful for cleaning up incorrect upload.
 - **Validation Strategy:**
 - **Stop on Error** - stops importing, if errors are found;
 - **Skip Error Entries** - upload data as valid data will be uploaded, while others will be dropped off.
 - **Allowed Errors Count** - used if **Stop on Error** is selected, and it determines the number of errors allowed.

- **Field separator**, **Multiple value separator** and **Fields enclosure** are CSV parameters, make them match your file format.

- **File to Import** - specifies the data file to import.
 - **Select File to Import** - drop here CSV file, that you had exported in the previous step;
 - **Images File Directory** - used, if links to pictures are used in the file exported. Typically it's not used at all.

Note

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

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

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

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

Note

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

Earning Rules

The Earning Rules allow you to award customers for placing an order with an additional monetary amount, which will go directly to his/her account and can be used immediately on the next placed order.

It is a powerful promotional tool that can be used for different actions - as you can see in the [Examples](#) subsection below.

Creating Earning Rule

To create an Earning Rule, go to the **Sales -> Store Credit -> Earning Rules** section, and perform the following steps:

1. Press the **Add New Rule** button, and fill in the basic fields:
 - **Rule Name** - a logical name of the rule.
 - **Websites** - select the store view.
 - **Is Active** - whether this rule should be activated after creation.
 - **Customer Groups** - groups of customers who are eligible for this Earning Rule.
2. Jump to the **Conditions & Actions** tab, and set the earning parameters in the **Actions** subsection:
 - **Cashback Type** - earning strategy. There can be two possible cases:

- **Amount per product** - the customer is awarded a fixed amount, deposited to his/her Store Credit balance.
- **Percent of item amount in order** - the customer is awarded an amount, equal to a percentage, calculated from the subtotal by a particular item.
- **Cashback Amount** - depending on a strategy that was selected in the previous option, it can contain either a fixed monetary amount or corresponding percentage of the subtotal.

3. If the customer should be awarded only on purchasing a particular product, you can use the **Conditions** tab to set these products.

- **How to use conditions in Earning Rules**

Let's take a look at this screenshot and describe the condition from top to bottom:

General Information

Conditions & Actions 

Conditions

If **ALL** of these conditions are 1

Category is 10 

SKU is one of

Search		Reset Filter
<input type="checkbox"/>	ID	
Any 	<input type="text"/>	
<input type="checkbox"/>	1	
<input checked="" type="checkbox"/>	2	
<input type="checkbox"/>	3	
<input type="checkbox"/>	4	
<input type="checkbox"/>	5	
<input checked="" type="checkbox"/>	6	
<input type="checkbox"/>	7	
<input type="checkbox"/>	8	

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

Applying modes define when rule shall be triggered:

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

Validation modes define which results 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 for creating a flexible condition set to satisfy a policy of any complexity.

Conditions are added using a green [+] button, each of which has the same scheme: [property] [operator] [check value]. There are two possible properties:

- **Category** - the category that the product should belong to.
- **SKU** - the SKU that the product should have

Operator is the type of check that should be performed to validate a rule. These 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 the check value is an array (should display as a drop-down menu), then the operator will search for **check value** in this array. If a value is the string or can be evaluated to the string, the operator will treat **check value** as a substring to locate.
- **does not contain** - the same as above, but with a negative result
- **is one of** - special operator, which works with a user-defined comma-separated array as **check value**. If this operator is used, then the condition is valid when the actual value of **property** exists in a defined array. If the property itself is a selection-type (SKU or Category, for example), then the dialog popup will appear to select elements of the user-defined array.
- **is not one of** - the same as above, but with a negative result

Check value is a value, to which the operator will be applied, as selected in the previous step. Both properties mentioned above allow for interactive selection.

Earning Rules Examples

- Deposit \$10 to the credit on each purchase

The customer will receive \$10 whenever he/she completes an order.

- General Information:
 - Is Active: Yes
 - Customer Groups: All
- Conditions & Actions:
 - Earning Type: Fixed
 - Earning Amount: 10

- **25% of each purchase cashback to the credit**

Whenever a customer places an order, 25% of his/her payment is deposited to his/her Store Credit balance.

- General Information:
 - Is Active: Yes
 - Customer Groups: All
- Conditions & Actions:
 - Earning Type: Percent of item amount in order
 - Earning Amount: 25

- **Deposit 10% of the amount spent for featured products to the Store Credit**

Whenever a customer buys promoted or new products, 10% of his/her payment is deposited to his/her Store Credit balance.

- General Information:
 - Is Active: Yes
 - Customer Groups: All
- Conditions & Actions:
 - Conditions: Category is one of 38 , 29 (ID's of categories **Sales** and **What's New**)
 - Earning Type: Percent of item amount in order
 - Earning Amount: 25

- **Deposit \$20 for each purchase of a product with blue coloring**

Whenever a customer purchases a configurable product with blue coloring, he/she receives \$20 to Store Balance.

- General Information:
 - Is Active: Yes
 - Customer Groups: All
- Conditions & Actions:
 - Conditions: SKU contains blue
 - Earning Type: Fixed
 - Earning Amount: 20

Note: Operator contains ignores the register, so SKU with both Blue and blue suffixes will be selected.

Credit product

Credit Products are special types of products that allow customers to deposit or refill funds in their balances.

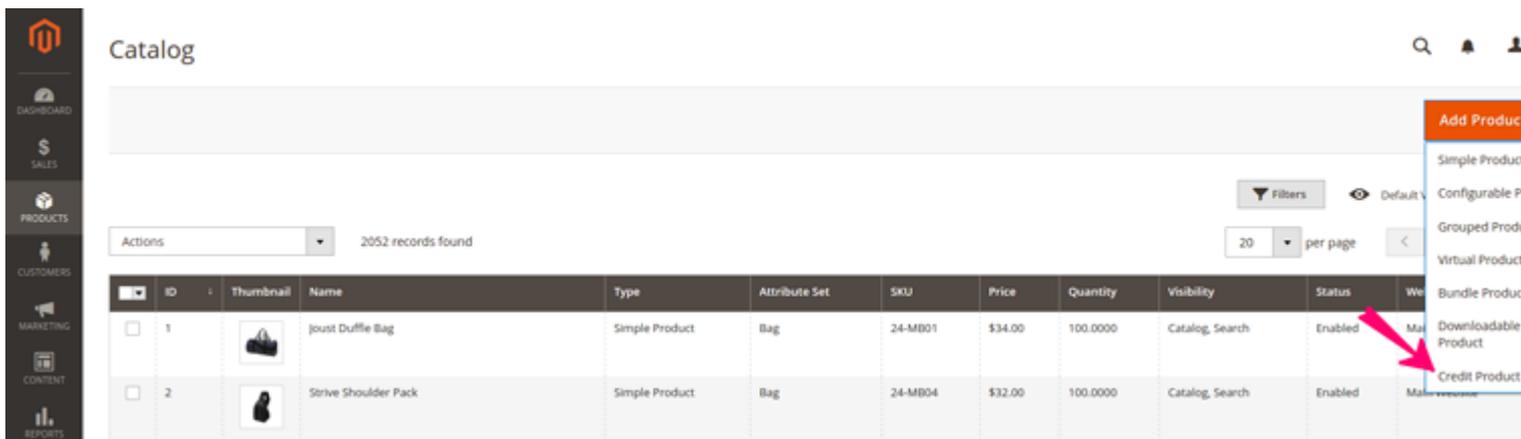
They consist of two types:

- **Credit Products** - allows customers to deposit money to the Store Credit balance at a price.
- **Refill Virtual Product** - allows customer to directly deposit a certain amount of money to his/her balance.

Their creation is discussed below with a set of ready-to-use [Examples](#).

Creating Credit Product

Go to **Products ? Catalog** and create a new **Credit Product** (take a look at the screenshot below):



1. Fill in the standard fields, such as **Product Name**, **SKU**, and **Quantity**.
2. Assign this product to the proper categories.
3. Proceed to the **Credit Price Options** subsection and select **Price Type**. There are three possible options:
 - **Single** - creates a simple product, which allows customers to deposit a certain amount to his/her balance. It triggers the following options:
 - **Price Options** - can be either **Fixed** or **Percent**.
 - **Price** - price that customer should pay to deposit credit funds. Suppose the **Fixed** option is selected, then the customer should pay exactly that amount. If **Percent** is selected, this field contains a percentage from the **Credits** value.
 - **Credits** - a monetary amount should be added to the Store Credit balance after order completion.
 - **Fixed** - creates a configurable product with a set of possible deposits. Each option has the same options, as in a **Single** price type.
 - **Range** - creates a product with a dynamic price that can be set by the customer. It triggers the following additional options:
 - **Price** - contains a conversion rate per 1 credit.
 - **Min Credit Amount** - minimal Store Credit funds that can be deposited to the balance.
 - **Max Credit Amount** - maximum Store Credit funds that can be deposited to the balance.
4. Save the product and make it enabled for purchasing.

Note

It is better to have a separate Category for all Credit Products. To create one, go to **Products -> Category**.

Creating Refill Virtual Product

Unlike Credit Products, which make a deposit at a certain price, Refill Virtual Product allows you to store funds to the Store Credit balance at no assigned price. After an order is complete, the purchase is completely sent to the Store Credit balance.

There can be only one product of this type.

To create such a product, go to **Products -> Catalog** and perform the following steps:

1. Press the **Add New** drop-down button, and select **Virtual Product**.
2. Add any appropriate properties, apart from the following that **should have** these values:
 - **SKU** - CREDIT
 - **Price** - 0
 - **Tax Class** - None
3. Jump to the **Customizable Options** subsection, and add the appropriate options, which will act as vouchers for depositing money. For detailed information on how to proceed, refer to [this section](#) of Magento 2 manual.
4. Save the product and make it enabled for purchasing.

Credit Products Examples

- **Credit product, which deposits \$50 to the balance for a fee of \$20**

The customer will pay \$70, and \$50 will be transferred to his/her balance on order completion.

Create a [Credit Product](#) with the following options:

- Credit Price Options:
 - Price Type: Single
 - Price Options: Fixed
 - Price: 70
 - Credits: 50

- **Credit product allows the customer to deposit custom money amount from \$100 to \$400 at a fee of 20%**

The customer will pay \$10 for each \$8, transferred to his/her Store Credit balance.

Create a [Credit Product](#) with the following options:

- Credit Price Options:
 - Price Type: Range
 - Price: 1 . 2
 - Min Credit Amount: 100
 - Max Credit Amount: 400

- **Create a set of vouchers for depositing \$100, \$200 and \$300**

The customer will select a voucher from the drop-down list, and on order completion, he/she receives the full purchase amount on Store Credit balance.

Create a [Refill Virtual Product](#) and create an option in the **Customizable Options** subsection:

- Option properties:
 - Option Title: Vouchers
 - Option Type: Drop-down
 - Required: Yes
- Option 1:
 - Title: \$100 Voucher
 - Price: 100
 - Price Type: Fixed
 - SKU: CREDIT-100
- Option 2:
 - Title: \$200 Voucher
 - Price: 200
 - Price Type: Fixed
 - SKU: CREDIT-200
- Option 3:
 - Title: \$400 Voucher
 - Price: 300
 - Price Type: Fixed
 - SKU: CREDIT-300

Here is a screenshot for your reference:

Refill Credit Product

Search Engine Optimization

Related Products, Up-Sells, and Cross-Sells

Customizable Options

Custom options let customers choose the product variations they want.

  Vouchers

Option Title * **Option Type ***

Vouchers Drop-down

Title *	Price *
 \$100 Voucher	\$ 100.00
 \$200 Voucher	\$ 200.00
 \$300 Voucher	\$ 300.00

Reports

Our extension allows you to generate reports about the Store Credit operations. It only provides general statistics, which can be seen at **Sales -> Store Credit -> Reports**.

The Report can be generated for the following time periods:

- **Day**
- **Week**
- **Month**
- **Year**
- **Custom period** using the date selector located at the top of the **Reports** page.

Each report contains the following data that are visualized both as chart and grid:

- **Time period** - dependent on the period, selected in the date selector, and can be **Day**, **Week**, and so on.
- **Delta** - the difference between amount of funds that was deposited and withdrawn during the selected period.
- **Received** - a total of funds that were deposited to the Store Credit balances
- **Spent** - a total of funds that were withdrawn from the Store Credit balances
- **Number of Transactions** - total number of transactions (regardless of type), made during a selected period.

These Reports are simplistic, yet they can be really handy in your crediting and refund policy evaluation.

How-To

This section describes the most common issues and their solution, consisting of the following subsections:

- [Customizing Email Template for Store Credit Balance Update](#)

Customizing Email Template for Store Credit Balance Update.

1. Go to **Marketing -> Communications -> Email Templates**, and press the **Add New Template** button.
2. Find the **Mirasvit_Credit** subsection in the **Template** field, and select the **Balance Update** option. Then press the **Load Template** button to load our basic template.
3. Customize the template as you need, and save it.
4. Go to **Sales -> Store Credit -> Settings -> Email Notification Settings** and in the **Template of Balance Update Email** field, select your new template.
5. Save configuration.

Command Line Interface

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

- `mirasvit:credit:import-export --export` - export all customer' balances to the `export.csv` file
- `mirasvit:credit:import-export --import --file import.csv` - import customer' balances from the `import.csv` file

GraphQL Objects Reference

Cart object

.

The `Cart` object can contain the following attributes:

Attribute	Data Type	Description
<code>credit</code>	<code>CartCredit</code>	Store credit information for cart

CartCredit object

Defines store credit information for cart

The `CartCredit` object can contain the following attributes:

Attribute	Data Type	Description
<code>amount</code>	<code>Money</code>	.
<code>is_applied</code>	<code>Boolean</code>	.

Customer object

The `Customer` object defines the customer name and address and other details.

The `Customer` object can contain the following attributes:

Attribute	Data Type	Description
<code>credit</code>	<code>Credit</code>	Store credit information for customer

Credit object

Contains customer store credit information

The `Credit` object can contain the following attributes:

Attribute	Data Type	Description
<code>balance</code>	<code>CreditBalance</code>	.
<code>transactions</code>	<code>[CreditTransaction]</code>	.

CreditBalance object

Contains store credit balance information

The `CreditBalance` object can contain the following attributes:

Attribute	Data Type	Description
<code>amount</code>	<code>Money</code>	Current balance
<code>currency_code</code>	<code>String</code>	Currency Code

CreditTransaction object

Defines store credit transaction information

The `CreditTransaction` object can contain the following attributes:

Attribute	Data Type	Description
<code>action</code>	String	.
<code>balance_amount</code>	Money	.
<code>balance_delta</code>	Money	.
<code>created_at</code>	String	.
<code>currency_code</code>	String	.
<code>is_notified</code>	Boolean	.
<code>message</code>	String	.
<code>transaction_id</code>	Int	.

Mutation object

.

The `Mutation` object can contain the following attributes:

Attribute	Data Type	Description
<code>applyCreditToCart</code>	<code>ApplyCreditToCartOutput</code>	Apply store credit to the cart

ApplyCreditToCartInput object

Defines the input to the run `applyCreditToCart` mutation

The `ApplyCreditToCartInput` object can contain the following attributes:

Attribute	Data Type	Description
<code>amount</code>	Float	The amount of Store Credit
<code>cart_id</code>	String!	The unique ID that identifies the customer's cart

ApplyCreditToCartOutput object

Defines the output for the `applyCreditToCart` mutation

The `ApplyCreditToCartOutput` object can contain the following attributes:

Attribute	Data Type	Description
<code>cart</code>	Cart!	Describes the contents of the specified shopping cart

How to upgrade extension

To upgrade the extension, take the following steps:

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

Note

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

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

4. Run the command `php -f bin/magento setup:upgrade` for installing updates.
5. Run the command `php -f bin/magento cache:clean` for clear cache.
6. Deploy static view files

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

Disabling Extension

Temporary Disabling

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

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

Extension Removing

To uninstall the extension, please take the following steps:

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

1.2.21

(2024-11-25)

Fixed

- Integrity constraint violation on customerCart graphql query when save available shipping rates

1.2.20

(2024-10-23)

Fixed

- Incorrect total calculation on graphql customerCart query when Store Credit is applied
-

1.2.19

(2024-07-19)

Fixed

- Added validation message in the Store Credit form in checkout
-

1.2.18

(2024-07-18)

Fixed

- Store Credit form is displayed in checkout if customer does not have credits on his account
-

1.2.17

(2024-02-22)

Fixed

- Return value of execute() must be of the type int, "null" returned when run credit console command
-

1.2.16

(2024-02-12)

Fixed

- Object of class Magento\Quote\Model\Quote\Item\Option could not be converted to int when create order with the credit product from admin
-

1.2.15

(2023-11-06)

Fixed

- Balance Update email is taken from the default store instead of the store where the customer was created
-

1.2.14

(2023-09-26)

Fixed

- Incorrect grandTotal calculation in Credit Memo if default Magento discount is applied
-

1.2.13

(2023-08-29)

Fixed

- Avoid prevent refund if getBaseTotalPaid is not rounded
- If order is not paid by Store Credit and Store Credit amount gets applied to CreditMemo, the order status does not change from COMPLETE to CLOSED when all paid amount is refunded

Improvements

- Added styles to Store Credit top link
-

1.2.12

(2023-07-04)

Fixed

- Allow entering only digital values to the Store Credit apply input in cart and checkout
- Added notification message for Credit Memo adjustment refund amount that exceed order amount
- customer.name variable did not work in emails
- Store Credit top link is not displayed on the cacheable pages

Improvements

- Compatibility with Order Management
-

1.2.10

(2023-05-16)

Fixed

- use_credit column was set to no when store credit is applied in the admin
-

1.2.9

(2023-04-19)

Fixed

- Added option to disable QuickDataBar
 - Call to undefined method Mirasvit\Rewards\Model\Config::getAdvancedSpendingCalculationMethod() when use Store Credit
-

1.2.8

(2023-02-22)

Fixed

- Earning rules give cashback based on the item row total including discount
-

1.2.7

(2022-12-12)

Fixed

- Credit discount is not applied to the bundle items if parent item qty > 1

Improvements

- Compatibility with Klarna Payment M245
 - Compatibility with Order Management Preview
-

1.2.6

(2022-11-07)

Fixed

- Credit discount is not applied to the bundle items if item quantity is greater than the parent item qty
-

1.2.5

(2022-09-12)

Fixed

- Compatibility with Pl_Mpgs
 - Wrong currency symbol in admin order create page
-

1.2.4

(2022-07-04)

Fixed

- Earning rules do not work for downloadable products
 - Store Credit amount is added to the Magento discount amount in the invoice on frontend
 - Store Credit discount is incorrect for taxable products when 'Including Tax' for prices is enabled
 - Store Credit amount is added to the Magento discount amount in the invoice
 - Currency code is set to global not as selected for manual admin transactions
-

1.2.3

(2022-05-20)

Fixed

- Refund to Store Credit
-

1.2.2

(2022-05-17)

Improvements

- update mirasvit/module-report dependency
 - Added api endpoint to get credit amount of the existing order
-

1.2.0

(2022-05-12)

Improvements

- Switched to declarative DB schema

1.1.16

(2022-03-10)

Fixed

- Price of store credit product for multi currency stores
 - Bundle item row total was not discounted lead to PayPal error 'To the totals of the cart item amounts do not match order amounts'
-

1.1.15

(2021-11-23)

Improvements

- Quick Data Bar
- PHP 7.1.0+

Fixed

- Amazon login popup is displayed for the registered customer when guest checkout is disabled
 - Translations for transaction actions in the customer account
-

1.1.14

(2021-09-06)

Fixed

- ItemPriceIncludingTax contains incorrect amount leads to invoice total 0.01
 - Klarna payment Order line totals do not total order_amount
 - When order is refunded Earned Credits are not refunded
-

1.1.13

(2021-07-16)

Fixed

- Wrong discount in invoice
 - Shipping discount not included in invoice
 - Incorrect currency in import balance transaction
-

1.1.12

(2021-04-19)

Fixed

- GraphQL: Canceling of the store credits does not work
-

1.1.11

(2021-02-26)

Fixed

- Issue when the row total in the backend is negative for items without tax
-

1.1.10

(2020-12-09)

Fixed

- Added composer dependencies for module-report
 - Button 'Send Credit to Friend' is not blocked after submitting and customer could send store credit to a friend a few times
 - Store Credit amount shouldn't be visible on create order page in the backend if order total is zero
 - The Store Credit link is displayed for the not loggedIn customers in some themes
-

1.1.9

(2020-11-13)

Fixed

- Add Option button does not work and remove button only works once in credit price options on create product page in admin
-

1.1.8

(2020-10-12)

Fixed

- double item discount in the "Items Ordered" table when MageWorx_ShippingRules enabled
 - Compatibility with m2.1.x. Invoice's "Grand Total" does not include credit's discount
 - Compatibility of earning rules with m2.1.x
-

1.1.7

(2020-10-09)

Improvements

- Compatibility with Cart2Quote module
 - Added StoreCredit line to the CreditMemo email layout
-

1.1.6

(2020-08-28)

Fixed

- Double item discount in the "Items Ordered" table when MageWorx_ShippingRules enabled
-

1.1.5

(2020-08-05)

Fixed

- Calculation of the creditmemo grandtotal when a coupon was used
 - Calculation of the Store Credit refund
-

1.1.4

(2020-07-30)

Improvements

- Support of Magento 2.4
-

1.1.3

(2020-07-22)

Fixed

- Apply of credit discount to the bundle products
-

1.1.2

(2020-07-03)

Fixed

- Actions::setActionFormName() must be an instance of Magento\Rule\Model\Action\Collection when add new Earning Rule when create new rule
-

1.1.1

(2020-06-23)

Fixed

- Issue when credit discount does not apply to invoice totals
-

1.1.0

(2020-06-18)

Fixed

Improvements

- Rename Earning Rules to Cashback rules
-

1.0.81

(2020-06-04)

Fixed

- Earning credits for the "Percent of item amount in order"
-

1.0.80

(2020-05-29)

Fixed

- credit memo spelling mistake in the transaction comment
-

1.0.79

(2020-04-24)

Fixed

- Earning credits for "Percent of item amount in order"

Improvements

- Add option "Allow for selection of payment method for zero orders" that control visibility of payment methods when credits were used.
 - Compatibility with Klarna_Checkout 9.0.0
-

1.0.78

(2020-04-24)

Fixed

- Compatibility with IE11
 - Doubling rows in the backend totals
-

1.0.77

(2020-03-18)

Fixed

- Calculation of shipping amount
-

1.0.76

(2020-03-16)

Improvements

- Frontend UI improvement
-

1.0.75

(2020-02-05)

Fixed

- Changed credit field label for the creditMemo
 - Display of currency code on creditmemo for single currency order
 - Issue when entered amount of credits to return reset to 0 on creditmemo
-

1.0.74

(2020-01-23)

Fixed

- Reset address information on quote saving
 - Unable to spend credits in the backend order
 - Error "The product instance type in options group is incorrect"
-

1.0.73

(2019-12-23)

Fixed

- Display of Store Credits amount in customer account for non base currency store with global balance
 - Sending of emails for customers with non-ASCII chars in the name
-

1.0.72

(2019-12-16)

Fixed

- Conditions in earning rules
- Compatibility with AW checkout
- Call to a member function `getStore()` on null in console command "merge-balances"

Improvements

- Added options to parse order and credit memo in transaction comments
-

1.0.71

(2019-11-21)

Fixed

- Credits calculations for shipping tax
-

1.0.70

(2019-11-13)

Fixed

- GraphQL moved to dedicated module (mirasvit/module-credit-graph-ql)
-

1.0.69

(2019-10-31)

Fixed

- Earning credits with earning rules
-

1.0.68

(2019-08-28)

Fixed

- Issue when store credit transaction does not create for multi shipping orders
-

1.0.67

(2019-08-19)

Fixed

- Display of purchased store credits on order view page
-

1.0.66

(2019-08-09)

Fixed

- Issue when store credits do not apply to grand total when Apply Tax Before Discount enabled
-

1.0.65

(2019-07-01)

Improvements

- Compatibility with Aheadworks One Step Checkout
-

1.0.64

(2019-06-10)

Fixed

- fixed error "PHP Fatal error: Interface 'Magento\Framework\GraphQL\Query\ResolverInterface' not found in ..."
-

1.0.63

(2019-06-07)

Fixed

- Error "Invalid method groupFactory:\Mirasvit\Credit\Model\ProductOptionCredit"
 - Restore amount of Store Credits for partial creditmemo
-

1.0.61

(2019-04-08)

Improvements

- CLI command for import/export balances
-

1.0.60

(2019-03-20)

Fixed

- Creation of backend order with Credit products
 - Issue "unable to create invoice for refill product"
-

1.0.59

(2019-03-18)

Fixed

- Wrong amount of credit discount displays after switch to another currency
-

1.0.58

(2019-02-21)

Fixed

- Issue when "Partial credits do not applied to invoice discount"
 - Currency in order cancel transaction email
 - Error "Table 'credit_transaction' does not exist ..."
 - Balance amount displays in wrong currency
-

1.0.57

(2019-01-28)

Fixed

- Issue when "Invoice grand total differ from order grand total"
-

1.0.56

(2019-01-11)

Fixed

- Issue with subscription
-

1.0.55

(2019-01-04)

Fixed

- Error "Call to a member function getStoreId() on null in Model/Quote/Item/AbstractItem.php"
 - Error "Fatal error: Method Magento\Ui\TemplateEngine\Xhtml\Result::__toString() must not throw an ..."
 - Credits apply twice to invoice
 - Creditmemo grandtotal does not subtract returned credits
 - Currency symbol in credit transactions
-

1.0.54

(2018-12-24)

Fixed

- Issue with Reports
-

1.0.53

(2018-11-30)

Fixed

- Tax calculation if credit order set before tax
-

1.0.52

(2018-11-29)

Fixed

- Compatibility with Magento 2.3.0
-

1.0.51

(2018-11-08)

Improvements

- Styles
-

1.0.50

(2018-11-07)

Fixed

- Issue "Invalid type for value: Expected Type: "float"."

Improvements

- Added ability to change the order of store credit applying
-

1.0.49

(2018-10-23)

Fixed

- Fixed multicurrency issue

Improvements

- Added option "Share customer balances"
-

1.0.48

(2018-10-09)

Fixed

- Creditmemo partial return with "Do not refund applied Store Credit"
-

1.0.47

(2018-10-02)

Fixed

- Validation for partial credits
-

1.0.46

(2018-08-30)

Improvements

- Unable to close orders totally paid with Store Credits ()
-

1.0.45

(2018-08-22)

Improvements

- Added option that allows to not return applied credits to customer on credit memo
-

1.0.44

(2018-08-22)

Fixed

- Issue when credit form displayed for guest customer in checkout
- Fatal error: Uncaught TypeError: 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
- Auto assign new customers for transaction emails

Improvements

- Added possibility to use custom amount of credit points
-

1.0.43

(2018-03-07)

Fixed

- Fixed ACL Issue not allowing access to Credit Settings with custom Role

Improvements

- Added possibility of hiding Send Credits block on frontend
-

1.0.42

(2018-02-27)

Improvements

- Compatibility with latest version of Mirasvit module reports
-

1.0.41

(2018-02-02)

Fixed

- Compatibility with Amasty checkout
- Credit block does not show on Paypal preview page

1.0.40

(2018-01-09)

Fixed

- Issue with wrong currency in admin (default currency insted base currency)
-

1.0.39

(2018-01-05)

Fixed

- Issue when credit section was updated twice
-

1.0.38

(2017-12-18)

Fixed

- Compatibility with 2.2.x
 - Partial return using PayPal
-

1.0.36

(2017-10-04)

Fixed

- Compilation
-

1.0.35

(2017-09-27)

Fixed

- Compatibility with Magento 2.2.0
-

1.0.34

(2017-09-19)

Fixed

- Fixed numerous bugs
 - Compatibility with Magento 2.2.0rc
-

1.0.33

(2017-09-01)

Fixed

- Shipping calculation
-

1.0.31

(2017-07-14)

Features

- Added ability to exclude shipping and tax amount from Store Credit payment
-

1.0.30

(2017-06-27)

Features

- Added ability to sell Store Credits

Fixed

- An issue with credits not being cancelled for orders in backend
 - Creditmemo with partial return
-

1.0.29

(2017-03-24)

Fixed

- Fixed an issue with updating balance in top menu
-

1.0.28

(2017-03-17)

Improvements

- Ability to auto apply store credit for shopping cart

Fixed

- Fixed an issue with credit registration (paypal express checkout)
-

1.0.27

(2017-03-01)

Improvements

- Added option "Enable Refund To Store Credit (default value for credit memo)"
-

1.0.26

(2017-02-27)

Improvements

- Report version 1.2.*

Fixed

- Fixed an issue with number validation
-

1.0.25

(2017-01-20)

Fixed

- ACL
-

1.0.23

(2017-01-19)

Improvements

- Changed appearance for refill section
 - Implemented ability to build reports by transactions
-

1.0.22

(2016-12-15)

Features

- Added Earning Rules feature

Improvements

- Added store credit block to order creation page

Fixed

- When order was not created, transaction was not removed
- In some cases back url is wrong

1.0.21

(2016-09-15)

Fixed

- Checkout. Payment methods does not update after cancellation of Store Credit
-

1.0.20

(2016-09-14)

Fixed

- Wrong cart total for zero costs products
-

1.0.19

(2016-09-12)

Fixed

- Compatibility with Magento 2.1
-

1.0.18

(2016-07-28)

Fixed

- Fixed an issue when unable to add first product to cart ()
-

1.0.17

(2016-07-19)

Fixed

- Fixed an issue when credit does not apply to order
-

1.0.16

(2016-07-01)

Fixed

- Fixed an issue with using credit for virtual products
-

1.0.15

(2016-06-24)

Fixed

- Compatibility with Magento 2.1
-

1.0.14

(2016-05-24)

Improvements

- Added credit information to pdf files (invoice and creditmemo)
-

1.0.13

(2016-05-23)

Improvements

- Added store credit information to order email
-

1.0.12

(2016-05-17)

Fixed

- Fixed an issue with PayPal payments (The totals of the cart item amounts do not match order amounts)
-

1.0.11

(2016-05-03)

Fixed

- Fixed an issue with Incorrect dependency during compilation
-

1.0.10

(2016-04-29)

Improvements

- Added ability to import/export transactions
 - Added redirect to customer edit page after adding new transaction
 - Added link to customer edit page to transactions grid
-

1.0.9

(2016-04-26)

Fixed

- Fixed currency conversion
-

1.0.8

(2016-04-18)

Features

- Added total balance on admin transactions page

Improvements

- ACL
- Frontend styles
- Ability to purchase additional credits (refill balance)
- Styles

Fixed

- Fixed issue with DI
- Fixed an issue with js error on cart page
- SCR2-53 - Added sequence Magento_Sales
- SCR2-53 - Fixed "Current balance is used in each order in multishipping"
- composer requirements

Documentation

- Added "Refill store credit" section
-

1.0.7

(2016-03-24)

Fixed

- Fixed an issue with message formatting
-

1.0.6

(2016-03-24)

Improvements

- Code Refactoring
 - Added Store Credit block to Checkout page
 - Show current balance in My Account (dropdown)
 - Support multi-shipping checkout
-

1.0.5

(2016-03-16)

Improvements

- Ability to force email in "Send Credit to Friend"

Fixed

- Removed wrong layout file
-

1.0.3

(2016-03-13)

Improvements

- Code refactoring
- Updated email template
- Improved translations

Documentation

- Updated installation steps
-

1.0.2

(2016-03-03)

Fixed

- Fixed an issue with sql error, when create new customer via backend
-

1.0.1

(2016-02-05)

Fixed

- Fix problem with wrong menu code
-

1.0.0

(2016-01-29)

- Initial stable release
-