

AffStackDev Store Credit & Gift Cards Complete Customer Reference

AffStackDev | Release 1.0.7 | Documentation revision 2026-10-21.1

Edition date: October 21, 2026.

Audience: Store owners, administrators and agency staff.

About this reference

AffStackDev Store Credit & Gift Cards for WooCommerce lets you issue store credit, sell and deliver gift cards (emailed or printed and shipped), accept eligible value at checkout, manage referral rewards and administer balances, reports and recovery. Store credit and gift cards keep separate customer balances by currency while using the plugin's protected transaction ledger.

This reference covers every current capability and configuration for store owners, administrators and agency staff, including what shoppers and recipients can do. Use the contents and feature index to find the setup, controls, result and recovery steps. The companion Master Technical Reference specifies exact fields, hooks, contracts and stored evidence for developers and integrators.

What's new in 1.0.7

Release 1.0.7 follows 1.0.5 and adds nine features: refund an order as store credit from WooCommerce's refund panel ([Refund an order as store credit](#)); card designs the buyer chooses from ([Card designs](#)); physical gift cards shipped with the order ([Physical gift cards](#)); holder gift cards that whoever holds them can spend, made in batches for a card printer ([Holder gift cards](#)); customers adding money to a gift card they hold ([Add money to a gift card](#)); a printable card with a QR code for every gift card and credit ([Print a gift card](#)); customers sending balance to someone else when you allow it ([Send balance to someone else](#)); cashback paid into store credit ([Cashback](#)); and credit spent in the currency a shopper switched to with WPML or FOX, where gift cards, cashback and refunds as credit are converted too ([Shoppers in another currency](#)). Holder gift cards, sending balance and cashback are off at the start. Releases 1.0.3 to 1.0.5 were corrections listed in the plugin's changelog; 1.0.6 was not released. Everything else in this section came with 1.0.2 and 1.0.0.6a.

Release 1.0.2 follows the launch release 1.0.0.6a. It has three corrections: store credit used by an order that was paid entirely with credit comes back when the order is marked Refunded; a credit or gift card code that pays nothing in the cart is taken off the cart with a notice instead of stopping the checkout; and **Reports** and a credit's **Activity ledger** show the source, event and reason in words (for example *Used by an order* and *Order status: Processing*) instead of internal keys. The guide built into the plugin now describes the current release, and version numbers have three parts from this release on. Everything else in this section came with 1.0.0.6a.

Release 1.0.0.6a was the AffStackDev launch release of Store Credit & Gift Cards; WordPress showed it as version 1.0.0.6.1. It contains every change since 1.0.0.5.

Store credit and gift cards are payment in Refer a Friend. Paying with store credit or a gift card no longer lowers the friend offer's minimum eligible spend, and paying the rest of a qualifying order with credit counts as the friend's payment (see [Set up Refer a Friend](#)).

A used invitation clears itself. After the friend's qualifying order is placed or paid, the invitation clears, so later ordinary checkouts work without **Remove invitation**; a payment retry of the same order keeps it. Accepting an invitation again reuses the session's unused friend coupon instead of creating more (see [Invite a friend and use an accepted invitation](#)).

Referral credit stays with its owner. A referral reward can be spent only by the account or email it was issued to, even if its coupon is later edited.

Clearer checkout refusals. When Refer a Friend refuses a Checkout Blocks order, the shopper sees the same customer message as in the classic checkout instead of a server error.

Saved credit applies once. Applying a saved credit that is already on the cart, for example from a second signed-in device, changes nothing and no longer blocks the next checkout.

Your license and updates. Each Store Credit & Gift Cards license covers one live (Production) site and one test copy of it; an installation set to Staging, Development or Local uses the license's staging place. AffStackDev's current plans do not require a Suite license, and AffStackDev Suite needs no key of its own. Download the plugin from **My Account** → **Downloads** on affstackdev.com; after activation, WordPress offers new releases through protected updates.

Contents

- [What's new in 1.0.7](#)
- [Feature index](#)
- [Install and open Store Credit & Gift Cards](#)
- [Use the standalone workspace or AffStackDev Suite](#)
- [Read runtime and integration status](#)
- [Merchant administration overview](#)
- [Open the merchant workspace](#)
- [Choose a display mode and protect unsaved work](#)
- [Read the Dashboard](#)
- [Find and inspect a credit](#)
- [Issue one store credit](#)
- [Adjust, stop, archive or restore a credit](#)
- [Refund an order as store credit](#)
- [Cashback](#)
- [Save settings and read integration status](#)
- [Use Reports and read-only reconciliation](#)
- [Review liability by currency, age and expiry](#)
- [Export credits and ledger history](#)
- [Prepare and commit a bulk CSV](#)
- [Resolve merchant errors](#)
- [Create a gift card product](#)
- [Buy for yourself or another recipient](#)
- [Add money to a gift card](#)
- [Payment and gift delivery](#)
- [Stop a purchase before refunding](#)
- [Receive and use a gift without an account](#)
- [Print a gift card](#)
- [Link an emailed card to My Account](#)

- [Read balances and activity](#)
- [Send balance to someone else](#)
- [Resolve gift purchase and account errors](#)
- [What credit value represents](#)
- [Shoppers in another currency](#)
- [Storage availability and preserved financial records](#)
- [Separate Store Credit and Gift Card balances](#)
- [Issuing credit and keeping the original identity](#)
- [Reading the financial history](#)
- [Order reservations and settlement](#)
- [Restoring credit after a refund](#)
- [Adjust, expire, revoke or archive](#)
- [Tax budgets and eligible shipping](#)
- [Stable references and protected evidence](#)
- [Reconciliation findings](#)
- [What reports count](#)
- [Personal-data exports and retained records](#)
- [Connecting compatible local products](#)
- [Partner payouts issued as reusable credit](#)
- [Recovering a ledger or integration error](#)
- [Using gift cards and store credit at checkout](#)
- [Apply a code and understand its value](#)
- [Saved gift cards and credit in Cart/Checkout Blocks](#)
- [When an order holds and uses credit](#)
- [Coordinated quote or companion checkout](#)
- [Acknowledging an already settled order](#)
- [Per-item allocation evidence for integrations](#)
- [Shipping-enabled credit](#)
- [Protected advanced orders](#)
- [Cancellation, refunds and reopening orders](#)
- [Paying a manual subscription renewal](#)
- [B2B status and consumer referral rewards](#)
- [Checkout problems and recovery](#)
- [Set up Refer a Friend](#)
- [Choose which offers can combine](#)
- [Invite a friend and use an accepted invitation](#)
- [Release, spend and review referral rewards](#)

- [Referral processor availability](#)
- [Shared processing and integration-only controls](#)
- [Referral history, removal and privacy](#)
- [Licensing, setup, and existing customer value](#)
- [What Commerce Core does](#)
- [Read shared records and technical details](#)
- [Download reports and individual order records](#)
- [Customer eligibility and pricing integrations](#)
- [Shared cart lines and protected merchandise](#)
- [Accepted order history and publication recovery](#)
- [Diagnostic reports and integration interfaces](#)
- [Retained commerce records and safe updates](#)
- [Choosing a supported feature provider](#)
- [Shared configuration, pauses and ownership](#)
- [Fresh evaluation and short-lived permission](#)
- [One business operation, one retained identity](#)
- [Reconciling and recovering unfinished work](#)
- [Retiring a business policy safely](#)
- [Discovering checkout credit participation](#)
- [Shared operation troubleshooting](#)
- [Find the complete Store Credit & Gift Cards references](#)
- [Glossary](#)

Feature index

Capability or reference	Section
Accepted order history and publication recovery	Open reference
Acknowledging an already settled order	Open reference
Activate or replace the installation license	Open reference
Add money to a gift card	Open reference
Adjust, expire, revoke or archive	Open reference
Adjust, stop, archive or restore a credit	Open reference
Apply a code and understand its value	Open reference
Apply or remove account credit	Open reference
Automatic checks and retry behavior	Open reference
B2B status and consumer referral rewards	Open reference

Capability or reference	Section
Buy for yourself or another recipient	Open reference
Cancellation, refunds and reopening orders	Open reference
Card designs	Open reference
Cart calculation limits	Open reference
Cashback	Open reference
Checkout problems and recovery	Open reference
Choose a display mode and protect unsaved work	Open reference
Choose the value and limits	Open reference
Choose which offers can combine	Open reference
Choosing a supported feature provider	Open reference
Commercial facts without price changes	Open reference
Connecting compatible local products	Open reference
Coordinated quote or companion checkout	Open reference
Create a gift card product	Open reference
Customer eligibility and pricing integrations	Open reference
Diagnostic reports and integration interfaces	Open reference
Discovering checkout credit participation	Open reference
Download reports and individual order records	Open reference
Draft, publishing and committed	Open reference
Every program setting	Open reference
Export credits and ledger history	Open reference
Find and inspect a credit	Open reference
Find the complete Store Credit & Gift Cards references	Open reference
First setup, upgrades and data retention	Open reference
Fresh evaluation and short-lived permission	Open reference
Holder gift cards	Open reference
Install and open Store Credit & Gift Cards	Open reference
Interpret license states and deadlines	Open reference
Invite a friend and use an accepted invitation	Open reference
Issue one store credit	Open reference
Issuing credit and keeping the original identity	Open reference

Capability or reference	Section
Legacy B2B and Dynamic Pricing	Open reference
Licensing, setup, and existing customer value	Open reference
Link an emailed card to My Account	Open reference
Loading, empty lists and fallback behavior	Open reference
Merchant administration overview	Open reference
One business operation, one retained identity	Open reference
Open and read the License screen	Open reference
Open the merchant workspace	Open reference
Order reservations and settlement	Open reference
Partner payouts issued as reusable credit	Open reference
Paying a manual subscription renewal	Open reference
Payment and gift delivery	Open reference
Per-item allocation evidence for integrations	Open reference
Persistence restrictions	Open reference
Personal-data exports and retained records	Open reference
Physical gift cards	Open reference
Prepare and commit a bulk CSV	Open reference
Print a gift card	Open reference
Protected advanced orders	Open reference
Protected plugin updates	Open reference
Qualifying products and independent credit restrictions	Open reference
Read balances and activity	Open reference
Read runtime and integration status	Open reference
Read shared records and technical details	Open reference
Read Store Credit & Gift Cards through AffStackDev AI	Open reference
Read the Dashboard	Open reference
Reading a checkout that is still open	Open reference
Reading the financial history	Open reference
Receive and use a gift without an account	Open reference
Reconciliation findings	Open reference
Reconciling and recovering unfinished work	Open reference

Capability or reference	Section
Recover a manual download	Open reference
Recover a missing email	Open reference
Recover an interrupted activation or changed site	Open reference
Recover an interrupted shared reward	Open reference
Recovering a ledger or integration error	Open reference
Referral history, removal and privacy	Open reference
Referral processor availability	Open reference
Referrer, buyer and historical evidence	Open reference
Refund an order as store credit	Open reference
Release this site's remote activation	Open reference
Release, spend and review referral rewards	Open reference
Resolve gift purchase and account errors	Open reference
Resolve merchant errors	Open reference
Respond to connection messages	Open reference
Restoring credit after a refund	Open reference
Retained commerce records and safe updates	Open reference
Retirement and removing an ineligible referral	Open reference
Retiring a business policy safely	Open reference
Review liability by currency, age and expiry	Open reference
Review or retry a purchase	Open reference
Save settings and read integration status	Open reference
Save settings and use the editor without JavaScript	Open reference
Saved gift cards and credit in Cart/Checkout Blocks	Open reference
Saved-order certificates	Open reference
Schedule a gift	Open reference
Send balance to someone else	Open reference
Separate Store Credit and Gift Card balances	Open reference
Set up Refer a Friend	Open reference
Shared cart lines and protected merchandise	Open reference
Shared configuration, pauses and ownership	Open reference
Shared operation troubleshooting	Open reference

Capability or reference	Section
Shared processing and integration-only controls	Open reference
Shipping proof and order changes	Open reference
Shipping-enabled credit	Open reference
Shoppers in another currency	Open reference
Stable references and protected evidence	Open reference
Stop a purchase before refunding	Open reference
Storage availability and preserved financial records	Open reference
Stored information and support details	Open reference
Suite is optional unless your plan explicitly requires it	Open reference
Tax budgets and eligible shipping	Open reference
Understand the email	Open reference
Update verification and recovery	Open reference
Use a block product template	Open reference
Use Reports and read-only reconciliation	Open reference
Use the standalone workspace or AffStackDev Suite	Open reference
Using gift cards and store credit at checkout	Open reference
What Commerce Core does	Open reference
What completion and rollback mean	Open reference
What credit value represents	Open reference
What reports count	Open reference
When an order holds and uses credit	Open reference
Which activities need new-value permission	Open reference
Who qualifies and what becomes fixed	Open reference

Install and open Store Credit & Gift Cards

Store Credit & Gift Cards manages issued store credit, purchased gift cards, email gifting, referral rewards and the records of how those balances are spent or restored. It works with WooCommerce. You can use its own administration menu or open it inside AffStackDev Suite.

Use WordPress 6.7 or newer, PHP 7.4 or newer and WooCommerce 10.0 or newer. Upload the product's installer ZIP through **Plugins → Add New → Upload Plugin**, then activate it. Use the installer supplied for the release; the outer source-and-documentation handoff is not the WordPress upload archive. Keep the existing `affstackdev-store-credit-for-woocommerce` plugin directory when updating the same product.

If WooCommerce is absent or too old, activation stops with an explanation. Install or update WooCommerce and activate again. If the transactional ledger cannot be created or verified, activation stops with a database message. Correct that reported database condition before repeating activation; the plugin's ledger is necessary for credit operations.

Administrators and Shop Managers receive the `manage_store_credit` capability during activation. Custom roles need the appropriate capabilities assigned by their administrator. Some merchant operations also require permission to manage WooCommerce; the relevant controls describe those requirements.

First setup, upgrades and data retention

New installations start with Enhanced UI on, multiple eligible credit or gift-card codes allowed, and new credits permitted to combine with ordinary coupons by default. Existing saved choices are retained. Open Settings to choose the current options, License to manage the plugin's entitlement, and the relevant merchant or product screen to issue or sell value.

When the plugin version changes, it refreshes the customer-account endpoint and rewrite rules once and stores the new version marker. Ledger schema checks run separately. Each site in a multisite network has its own current-site setup and financial data; the plugin does not run a single activation loop over every existing site.

Deleting the plugin normally preserves balances and financial evidence. Reinstalling the matching product can use the retained data. A developer-only destructive-uninstall constant can permit removal of the two main ledger tables, but existing referral-history setup prevents that deletion branch. This is not an administration checkbox. The plugin does not provide a general erase-everything command, and uninstall does not automatically remove every associated product, coupon, order or licensing record. See the technical lifecycle for the exact conditions.

Use the standalone workspace or AffStackDev Suite

Without an accessible compatible Suite route, open **Store Credit & Gift Cards** in the WordPress administration menu. With AffStackDev Suite, open **AffStackDev → Store Credit & Gift Cards**. The same product screens, permissions, data and business rules apply. Suite does not take ownership of credit balances or gift-card orders.

The integration checks that your account can access both the Suite parent menu and the product submenu before relying on the Suite route. It can retain standalone access when that route is unavailable. A compatible Suite can also show the Store Credit & Gift Cards summary card on its dashboard.

Enhanced UI is an optional navigation wrapper over normal PHP screens. If the product's local preference is off, it uses page loads. If Suite's global Enhanced UI is off, the product follows that restriction even when its own preference remains on. No Suite, or an older compatible Suite without the global helper, lets the local preference decide. The forms and underlying authorization still apply in either mode.

Suite is optional for the plugin's ordinary setup, and AffStackDev's current plans do not require a Suite license. A license policy that requires Suite entitlement is still supported; [licensing](#) explains the resulting states and recovery. Merchant controls and their behavior are listed in [merchant administration](#).

Read runtime and integration status

Open **Store Credit & Gift Cards → Settings → Integration status**. This card shows:

- **Commerce Core runtime:** the selected shared runtime's status, host and version.

- **Checkout connection:** whether the exact supported credit participant can be discovered.
- **Referral history reader:** the fixed read-only continuation label; this card does not inspect the shared referral controller.
- **Business-policy retirement:** the state reported by the B2B integration.

The plugin includes Commerce Core 0.1.0.2c. Core needs 64-bit PHP 8.1+, WordPress 6.7+ and WooCommerce 10.0+. Native credit administration still supports PHP 7.4. A compatible active product may supply a newer selected Core; that does not transfer ownership of your balances or business rules.

registered means the expected checkout interface is available. It does not mean a checkout completed or a balance changed. held means the available descriptor did not exactly match the required contract.

unavailable means the required runtime or interface could not be read. Use the current supported product packages and [checkout recovery](#) to resolve the relevant condition.

Opening this card does not run a transaction or collect financial effects. The plugin also has a shared referral adapter, used only after the explicit migration and claim-binding steps described in the referral chapter. Historical adoption and program migration do not happen automatically. Missing shared eligibility evidence can hold new referral rewards while preserving already issued balances. The [referral chapter](#) explains which evidence a new reward requires.

Store Credit does not add a separate detailed Core administration screen. When a compatible Suite host supplies one, its panels and authorized record downloads are described in the [Core reference](#). Those optional host tools are separate from this product's passive status card.

Merchant administration overview

Use the merchant workspace to issue and inspect credit, record balance changes, review reports and liability, export financial records and commit reviewed bulk CSV batches. The sections below explain every control and how to check the saved result when a request is interrupted.

Open the merchant workspace

Open **Store Credit & Gift Cards** in WordPress administration. If AffStackDev Suite hosts the product, open its Store Credit & Gift Cards workspace instead. The same nine sections are available: Dashboard, Credits, Issue Credit, Gift Cards, Refer a Friend, Reports, Operations, Settings and License. Suite's optional summary card shows active credits, available value and reserved value; it does not change balances.

You need Store Credit management permission to use the workspace. Issuing a new credit, using Operations or restoring credit for a partial refund also requires WooCommerce management permission. New issuance and positive adjustments require the corresponding current license permission. A visible shortcut does not override these checks: the destination explains when the account lacks permission. [License controls](#) explain how to resolve entitlement restrictions.

On narrower screens, expand **Store Credit & Gift Cards section** to choose a section. Forms and advanced-setting disclosures work with ordinary browser controls; many tables become labeled cards, while other tables can scroll horizontally. The current section is marked in the navigation. Gift-card product creation is covered under [Gift Cards](#), and the referral chapter covers Refer a Friend.

Choose a display mode and protect unsaved work

Enhanced UI changes product pages in place and adds customer/product search selectors. PHP mode uses ordinary WordPress page reloads and text-based customer/ID fields. Both use the same server validation and ledger operations. Change the mode in [Settings](#). If Suite has Enhanced UI off, its choice overrides the local On preference until Suite allows it again.

With JavaScript available, leaving changed forms prompts you to discard unsaved work. Saving one form does not silently discard another changed form. The workspace also checks for newer edits before replacing a screen with a response. While a financial request is pending, wait for its result before leaving or submitting again.

If you see **This request may have completed**, reload and check the saved credit or activity ledger before repeating the action. A lost response can follow a successful ledger change. The browser blocks further writes on that uncertain screen until you reload. An error loading a read-only page leaves the current screen in place. A prompt to leave does not save a form, and cancelling departure leaves the current edits intact. With JavaScript disabled, rely on ordinary form submission and save before leaving; the server's permissions and financial checks still apply.

Operations uploads, exports, template downloads, commits and discards use ordinary native requests even in Enhanced UI. Bulk actions can return to the native Store Credit & Gift Cards page rather than the Suite shell. Turning Enhanced UI off also saves by a normal page reload immediately.

An initially opened filtered/detail URL can lose its filters or credit ID from the address bar when Enhanced UI initializes, even while that result remains displayed. Use the screen's own filters and View links to establish an in-place address, or use PHP mode when you need to retain a direct filtered URL. Back and Forward are guarded against silently discarding changed forms.

Read the Dashboard

Dashboard shows current Outstanding liability, Available to spend and Reserved balances, followed by Issued and Redeemed over 30 days. It also shows Active recipients, Orders using credit over 30 days and Live capabilities. The last value counts currently reported capabilities; it is not a license count or confirmation of a completed checkout.

Use **Issue store credit**, **View balances**, **Open reports** and **Merchant operations** to open the corresponding workflow. A + after Active recipients means the bounded count may not include every recipient. Dashboard and Reports use the current store-currency label; after a store-currency change, use [Operations liability](#) to review historical currencies separately.

Find and inspect a credit

Credits includes both store credit and purchased/gifted instruments. It shows 25 records per page. Use All, Active, Reserved, Exhausted, Expired, Revoked or Archived to select a view. Choosing a view clears the other filters. Expired means a stored expiry is due or the credit is marked expired. The Reserved view includes any credit with value held for an order, including an inactive credit; the row's State can therefore still say Expired. Reserved value cannot be spent again while that order is in flight.

Control	How to use it
Search	Enter a credit/coupon ID, code, customer name, login or email. Search matches at most 200 coupon IDs; use a more specific search when needed
Source	Any source, Manual, Compensation, Refund alternative, Loyalty, Promotion, Giveaway, API / import, Gift card purchase, Gift card for recipient, Partner payout or Other
Created from	Include records created from the start of the selected date in the store timezone
Created through	Include records through the end of the selected date
Expires by	Select the latest expiry date to include
Apply filters	Run the selected filters
Clear	Remove search/source/date filters while keeping the selected view
Page links	Continue through the result while keeping the filters

Referral credits remain available through All/search and the referral workspace even though the Source selector has no separate customer-referral choice. Enter real dates; an invalid date format does not provide a valid date filter.

The list displays Code and credit ID, Recipient, Original amount, Available, Reserved, net Redeemed, State and Expiry. **View** opens the credit. **Advanced coupon** opens its WooCommerce coupon record in a new tab. A missing coupon can leave a ledger record visible as Coupon unavailable; use the read-only reconciliation report before making changes. Editing a WooCommerce coupon's face value is not a way to adjust its ledger balance.

The detail page shows the code and recipient, all four money figures, Source, Created, Expires, the latest issuance-email status/attempt, and the saved Tax / shipping mode. A bound account displays the customer's current name/email; a deleted account appears as Customer #ID. Email-bound credits show their allowed email address. Back to Credits returns to the unfiltered list.

Activity ledger displays the latest 100 entries, including date, event, available/reserved changes, balances after the event, order/refund, actor, reason and note. Positive changes carry a plus sign. System is shown when no current user name is available. Order links open the native WooCommerce order editor. Entries are retained financial history: corrections append another entry instead of editing old ones. Download the ledger CSV when you need older entries.

Issue one store credit

Open **Issue Credit**. Fill the form, select **Review credit**, then select **Issue store credit** after checking the review. Value is created only at confirmation. **Cancel** returns to a fresh empty form; it does not preserve the old form for editing.

Field	Choice and starting value
Registered customer	Initially empty. Search for an existing customer; that account takes priority over an external email
Registered customer in PHP mode	Enter an existing customer ID, exact email/login, or a name that identifies exactly one user. If several users match, use an exact identifier

Field	Choice and starting value
External recipient email	Initially empty. Use a valid address when the recipient has no selected WordPress account; do not place an unregistered address in the customer lookup
Amount	Required positive store-currency value; the form uses 0.01 increments
Reason / source	Starts Manual; alternatives Compensation, Refund alternative, Loyalty, Promotion, Giveaway and Other
Internal note	Optional internal record of why the credit was issued
Customer message	Optional message included in the issuance email
Expiry date	Empty means no expiry; otherwise use a future date. Individual issuance expires at the end of that date in the store timezone
Allow this credit to combine with ordinary coupons	Starts from the saved default in Settings; the choice is saved on this credit
Send the customer issuance email after the ledger commit	Starts checked; clear it to create the credit without sending the issuance email

The amount is stored in the store's configured monetary precision. The service rounds finer input to that precision and requires at least one minor unit. Notes/messages are sanitized and saved up to 2,000 bytes; keep them concise, particularly with multibyte characters. The external email field is not an additional recipient when a registered customer is selected.

Expand **Advanced restrictions** to set these optional limits:

Field	Behavior
Allowed products	Search products/variations in Enhanced UI, or enter their IDs in PHP mode. Blank means no product allowlist
Allowed categories	Choose product categories in Enhanced UI, or enter their IDs in PHP mode. Blank means no category allowlist
Manual code (optional)	Leave blank for a random 128-bit code, or supply a unique code. WooCommerce normalizes the supplied coupon code

PHP ID lists accept commas, spaces or semicolons. Use real positive IDs and keep each list within 200 entries; later restrictions beyond the issuer's 200-ID limit are not saved. The enhanced category selector lists at most 200 categories, including empty categories. If the category you need is not listed, use its known ID in PHP mode. Product/category restrictions work together with the credit's other eligibility checks.

Expand **Advanced redemption: tax and shipping** to choose the saved spending policy:

Field	Choices and effect
Tax budget	Default Tax-exclusive discount budget (existing behavior) counts the merchandise discount. Tax-inclusive discount budget counts that discount plus its tax reduction against the balance

Field	Choices and effect
Shipping	Default Do not cover shipping. Use remaining credit for eligible shipping allows unused value to discount an eligible shipping package

Fees are not covered. Advanced tax-inclusive or shipping use requires the supported per-line, noncompound WooCommerce tax calculation. Use checkout for advanced credit rather than native order coupon recalculation. These choices remain attached to the issued credit; later defaults do not change it. Advanced amounts are limited to 100,000,000,000 minor units, equivalent to 1,000,000,000.00 with two decimal places. [Saved value rules](#) explain the calculation and eligibility in detail.

The review shows Recipient, Amount, Source, Expiry, ordinary-coupon compatibility, email choice and tax/shipping labels. It carries your restrictions, notes, message and manual code forward but does not list those fields in the review summary. Check them before selecting Review. If you need to change them from the review, cancel and enter a corrected form.

After confirmation, the success screen shows the code and **Open credit / Issue another**. An enabled email is sent or queued for retry only after the ledger commit. Email failure does not mean the credit does not exist. Check the existing credit and use its resend control rather than issuing another balance. A validation error returns the entered form so you can correct the indicated field.

Adjust, stop, archive or restore a credit

Open a credit's detail to use **Credit controls**. These actions preserve the old ledger entries. Review the current balance and order reservations before submitting.

Control	What it does and what to enter
Adjust available balance → Record adjustment	Enter a positive amount to add available value or a negative amount to remove it, a required Reason and optional Internal note. Zero is refused; you cannot remove more than available or adjust revoked/archived credit. Adding value needs current permission to create value
Expire credit now... → Confirm expiration	Expand the confirmation, keep or edit the required reason merchant_expire, then confirm. Blocks new spending and preserves reservations/history. Shown only when stored state is not expired/revoked/archived
Revoke credit... → Confirm revocation	Expand the confirmation, keep or edit required reason merchant_revoke, then confirm. Remaining available value becomes unspendable. A live order reservation or archived state prevents revocation
Archive credit... → Confirm archive	Available only when available and reserved amounts are both zero. Makes the zero-balance credit inactive while preserving its history
Resend issuance email	Forces a delivery attempt for the existing credit, including when original sending was off, the email is disabled or it was already sent. Creates no value and does not queue a retry if this forced attempt fails. Check email delivery if it cannot send
Advanced WooCommerce coupon editor	Opens the existing coupon in a new tab; it is not a replacement for the ledger controls

Reasons are stored as compact keys; the adjustment reason field allows 64 characters in the form. Use the Internal note for a fuller explanation. There is no Undo, Unrevoke or Unarchive button here. Adding value to an expired credit does not extend its expiry. If expiration reports an error, check both its coupon expiry and ledger state through Reports: the coupon change can have happened before the separate ledger operation failed.

For a partial refund, use the original order/refund's linked credit context. A valid context adds **Restore original store credit**, showing the WooCommerce refund, credit still consumed from this account and Maximum allowed here. It appears only for a refund belonging to that order and credit actually consumed by it. Ordinary partial refunds do not automatically restore credit.

Enter **Amount to restore** and the required **Reason** (initially `partial_refund_restore`), then select **Restore original store credit**. The starting amount is the current maximum. The system limits restoration to both this account's still-consumed credit and the refund amount not already restored across credits. For example, with 40.00 still consumed, a 30.00 refund and 12.00 already restored for that refund, you can restore at most 18.00 here. If there is no allowance, no further restoration button appears.

Compare any cash/gateway refund before restoring credit; this operation returns original credit, not cash. It requires both management permissions and checks the current allowance again. If a repeated request is refused after an uncertain response, inspect the ledger first: a successful previous restoration may already have reduced the remaining allowance. Do not create a new balance to compensate for a result you have not checked.

Refund an order as store credit

From release 1.0.6 a paid order can be refunded as store credit instead of money. Open the order in WooCommerce, select **Refund**, choose the lines, quantities and amounts exactly as for any WooCommerce refund, then select **Refund ... as store credit** (the button shows the amount, for example *Refund \$10.00 as store credit*) and confirm.

What happens:

Part	Result
The WooCommerce refund	Recorded as WooCommerce's <i>Refund manually</i> records it: the refunded lines and quantities, restocking when <i>Restock refunded items</i> is ticked, the order's refunded total and, when nothing is left to refund, the status Refunded. Nothing is sent back through the payment gateway. The refund's reason reads <i>Refunded as store credit</i> , after your own reason when you typed one; WooCommerce shows that reason to the customer.
The store credit	One credit of the refunded amount, with the source Refund alternative . It belongs to the order's customer account; for a guest order it is tied to the billing email, and the customer spends it with its code and that email, like any emailed credit. It has no expiry and no product or category restriction; it may be combined with ordinary coupons when New credits may combine with ordinary coupons by default is ticked in Settings.
Records	The order gets a note that links the credit. The credit's screen shows Given for with the refund and the order; its history shows <i>Order refunded as store credit</i> .
The customer	WooCommerce's refunded-order email says that the refund was given as store credit, not paid back to the payment method, with the code and where to find it.

Two choices sit next to the button:

Choice	Starting value and effect
Email the store credit to the customer	Ticked. The usual store credit email is sent once WooCommerce has kept the refund. Clear it when the refunded-order email is enough.
The store credit may also pay tax and shipping	Cleared: the credit pays for products, like a credit you issue with the default rules. Ticked: it is saved with the tax-inclusive budget and may pay eligible shipping with what is left (see Tax budgets and eligible shipping).

The button is disabled, with the reason under it, when the order is not paid (for example on hold waiting for a cheque), when it is in another currency than the store, when it has neither a customer account nor a valid billing email, or while the license does not allow new credit. An order that sold gift cards follows [Stop a purchase before refunding](#) first; until its cards are stopped and revoked, WooCommerce refuses this refund as it refuses any other.

Exactly once

Each refund gives exactly one credit. Pressing the button again with the same screen is refused by WooCommerce's own check ("Error processing refund. Please try again."), so a double click cannot refund twice; reload the order first. If WooCommerce recorded the refund but the credit could not be issued (the order note says why), choose **Issue the store credit of a refund made as store credit** in **Order actions** and apply: it issues that refund's credit with the same reference and never a second one.

Orders paid with store credit

The button refunds what WooCommerce can refund, which is the part paid with money. For the credit the order used, the usual rules apply: a full refund gives it back by itself, and on a partial refund you decide (see [Cancellation, refunds and reopening orders](#)). For example, an order of 30.00 paid with a 10.00 credit and 20.00 by card, refunded in full as store credit, leaves the customer with a new 20.00 credit and their 10.00 credit back: 30.00 in all.

Deleting the refund

If you delete that refund on the order, its credit is revoked when nobody has used, held or changed any of it, and the order note says so. Otherwise nothing is changed and the note asks you to review the credit.

Only staff who can manage Store Credit, manage WooCommerce and edit orders see the button (administrators and shop managers at the start).

Cashback

From release 1.0.6 the plugin can give cashback as store credit. In **Settings**, under **Cashback**, tick **Give cashback as store credit on paid orders of signed-in customers** and set:

Setting	Starting value	Meaning
Cashback percent	2	More than 0 and at most 100, with at most two decimals

Setting	Starting value	Meaning
Minimum order for cashback	0	Counted like the cashback itself (below); 0 means any order
Most cashback per order	0	0 means no limit
Days to wait before adding it	14	0 to 365, counted from payment
Cashback expires after (days)	0	0 means the cashback credit does not expire
Products without cashback (IDs)	Blank	Product or variation IDs, separated by commas
Categories without cashback (IDs)	Blank	Product category IDs, separated by commas

Select **Save settings**. A wrong value is refused with its reason, and the cashback settings stay as they were.

How cashback is counted

When a signed-in customer's order is paid (processing or completed; cash on delivery once completed), the settings of that moment are saved with the order; changing the settings later does not change it. The cashback is the percent of what the customer paid for products: each product line after its discounts and after store credit or gift cards used on it, less what was refunded on that line. Tax, shipping, fees, gift cards, top-ups and the excluded products and categories do not count. A refund entered as an amount without lines counts against the products first. The result is rounded down to the cent. Below the minimum there is no cashback; above the maximum it is the maximum. Example: products of 30.00 and 7.77 at 3 % give 1.13 (3 % of 37.77 is 1.1331).

When it is added

After the waiting days the cashback is added once, as store credit for the customer's account: source *Cashback*, the message "Cashback for order #...", the usual store credit email, and the expiry you set. The order notes say what was planned, added, cancelled or taken back, and the customer's order page in My Account shows the cashback and its date. Guests, business customers and applicants under review (with AffStackDev B2B; also an order whose customer B2B cannot confirm as a consumer) and orders that earn nothing get none. If the license does not allow new credit when it is due, the cashback waits and is tried again the next day.

Refunds and cancellations

Before the cashback is added, a cancellation or a full refund cancels it, and a partial refund lowers it. After it was added, a refund takes back the part the refund removed (*Adjusted*, with the reason *Cashback taken back after a refund*), but only what is still unused: the credit never goes below zero, and the order note says how much was already used.

Save settings and read integration status

Settings begins with the [canonical documentation downloads](#). Its merchant preferences are:

Setting	Default and effect
Enhanced UI	On. Uses in-place navigation and enhanced selectors when Suite also permits it. Off uses PHP page loads; both preserve the same credit rules

Setting	Default and effect
Allow multiple store credit or gift card codes on the same order	Checked. Allows the supported multiple-code experience; uncheck to limit that behavior
New credits may combine with ordinary coupons by default	Checked. Sets the starting choice for the next single-credit form; existing credits keep their saved choice
Attach the printable gift card (PDF) to the gift card email	Unchecked. Attaches the card's PDF to gift card emails; see Print a gift card
Allow holder gift cards (printed cards that anyone holding them can spend)	Unchecked. Lets gift card products and batches make holder gift cards; cards already made keep working when it is unticked; see Holder gift cards
Customers may send balance to someone else	Off. Or Gift card balance, or Gift card and store credit balance; see Send balance to someone else
Cashback	Off. Its settings are described in Cashback

Select **Save settings** to persist the choices for the site. The screen shows Effective mode and explains a Suite override when present. If an externally changed or invalid stored preference makes the selected radio differ from the effective mode, explicitly choose and save the intended mode. Bulk CSV's blank `allow_coupons` value always defaults to Yes; it does not inherit the single-credit checkbox default. Gift-card product policies are set on each product.

Integration status displays Commerce Core runtime, Checkout connection, Referral history reader and Business-policy retirement. It is a status display, not a connection editor or checkout-completion receipt. The referral history reader is read-only and does not automatically take over rewards. Core coordination needs its stated WordPress/WooCommerce/PHP requirements; native credit administration has its own supported requirements. Missing eligibility evidence can hold new referral rewards without deleting existing balances.

Gift Cards is always available in the customer account area, including a zero balance. Store Credit appears when that customer has an available store-credit balance. Credits are issued in the store currency, and existing instruments retain their tax/shipping choices. Manual renewals require WooCommerce Subscriptions and a manual-renewal subscription; automatic renewals are not supported.

Use Reports and read-only reconciliation

Reports offers **7 days**, **30 days**, **90 days** and **365 days**, initially 30. It shows current Outstanding liability, Available to spend and Reserved, plus Issued, Redeemed, Restored, Expired / revoked value, Active recipients, Orders using credit and Associated order value. Period activity follows the selected range; current balances and recipients are snapshots. Reports can remain cached for up to five minutes and refresh after supported committed credit events.

Issuance sources lists Source, Credits and Issued value for up to 20 source groups. **Recent credit activity** lists the latest 20 entries with Date, Credit, Event, Order and Reason. Select a credit ID to inspect its detail. Empty panels state when the selected period has no issuance or activity.

Active recipients examines at most 500 active accounts; Associated order value examines up to 200 redeemed orders. A + identifies a count/value that reached its sampling limit. Associated order value is WooCommerce order value connected to orders using credit, not revenue caused by this plugin. These summaries use the

current store-currency label; use Operations to keep historical currencies separate. Expired/revoked figures represent recorded historical events, not a cash payout or automatic write-off.

Opening Reports runs **Read-only reconciliation**. It displays Accounts checked, Order links checked, Issues found and Mode: Read only, followed by up to 20 issue messages with credit links. It checks a bounded sample of up to 100 accounts against coupon, transaction and live order evidence. It does not repair records or prove that every account is healthy. Follow the exact issue's [reconciliation guidance](#); avoid editing financial history to silence a warning.

Review liability by currency, age and expiry

Open **Operations** for accounting visibility. It requires both Store Credit and WooCommerce management permissions. Its first metrics show current-store-currency Outstanding liability, Credits with liability, Expiring in 30 days and Other recorded currencies.

Each recorded currency then has its own section. Outstanding includes available value only while the credit is active and unexpired, plus all reserved value because it may belong to an order already in flight. Unlike currencies are never summed or converted. If the store currency changed, historical balances remain under their original currency.

View	Interpretation
Credits with liability	Number of accounts contributing a positive liability amount
Liability aging	Current liability grouped by account creation age: 0–30, 31–90, 91–180, 181–365 and 366+ days; it is not age since last use
Expiry exposure	Active instruments expiring in Next 30 days, 31–90 days, 91+ days, or with No expiry
Inactive reserved review	Reservations whose stored account state is inactive; review their original order settlement/release
Expired remaining evidence	Remaining available amounts past expiry, excluding revoked/archived accounts; historical evidence that is not currently available liability
Liability by source	Currency, Source, number of contributing Records and Outstanding, for up to 250 nonzero source groups

Expiry buckets do not always add to the total: inactive or already-expired reservations can still be liability but fall outside future-expiry buckets. A date-expired account still stored as active may also be absent from Inactive reserved review. The expiry and aging views are read-only; they do not expire, extend or revoke anything. Currency totals are current aggregate reads, while the source table is limited to 250 groups without a further-page control.

Export credits and ledger history

In **Operations** → **CSV exports**, select **Export credits CSV** for all credit records, or choose **30 days**, **90 days**, **365 days** or **All ledger history** and select **Export ledger CSV**. The ledger selector starts at 90 days. Credits-list filters do not restrict these exports.

The credit CSV includes:

Columns	Contents
credit_id, coupon_id, code	Ledger ID, WooCommerce coupon ID and redeemable code
owner_type, customer_id, recipient_email	Customer/email/unbound ownership, user ID where applicable, and current user email or first allowed email
status, source, currency	Effective state, issuance source and original currency
original, available, reserved, redeemed	Decimal money values; available may be retained unspendable evidence on an inactive credit
expires_at_gmt, created_at_gmt, updated_at_gmt	UTC expiry, creation and last update; blank expiry means none
allow_coupons, tax_mode, shipping_mode	Current coupon-combining choice and saved value policy

The ledger CSV includes transaction_id, credit_id, type, currency, delta_available, delta_reserved, resulting_available, resulting_reserved, order_id, refund_id, actor_user_id, reason_code, note and created_at_gmt. It retains signed changes and internal transaction evidence. Missing order/refund/actor values are blank; dates are UTC.

Downloads stream directly to your authenticated request rather than becoming public upload files. They contain codes, customer email addresses and internal financial history, so store and share them with the appropriate access restrictions. Files use UTF-8 and CSV quoting; text that resembles a spreadsheet formula is prefixed with an apostrophe. That prefix is intentional. Exports scan live records in batches; a busy store's changing data is not frozen into a single point-in-time snapshot. Keep each currency separate in your own accounting.

Prepare and commit a bulk CSV

Bulk issue credits is in Operations. Start with **Download CSV template**, replace its two example data rows, and save a CSV. The template's customer ID 123 and customer@example.com are real data values, not comments: leaving them in can issue credit if they pass validation.

The file must be 1 byte to 512 KiB and contain at most 250 data rows. Required column names are reference, customer_id, recipient_email and amount; both recipient columns must be present, but each row uses exactly one. Additional supported columns are listed below. Do not add other columns. Keep comma-containing ID lists or multiline messages properly CSV-quoted.

Column	What to enter
reference	A unique, permanent reference for the intended credit, 1–100 letters/numbers or ., _ , : , - ; case matters
customer_id	Existing WordPress user ID, leaving recipient_email blank; otherwise blank/zero
recipient_email	A valid email, leaving customer_id blank/zero
amount	Positive amount in the current store currency
source	Blank becomes api_import; alternatives compensation, refund_alternative, loyalty, promotion, giveaway, manual, other
expiry_date	Blank for no expiry, or future YYYY-MM-DD. Bulk expiry is the end of that date in UTC, unlike the individual form's store timezone

Column	What to enter
allow_coupons	Blank means Yes, even if the single-credit default is off
send_email	Blank means No; explicitly enable delivery when intended
tax_mode	Blank or tax_exclusive, or tax_inclusive
shipping_mode	Blank or none, or remaining
internal_note	Optional internal note, shortened to the text limits
customer_message	Optional email message, shortened to the text limits
product_ids	Optional products/variations, at most 200 tokens separated by commas, spaces or semicolons; use positive integer IDs without leading zeros
category_ids	Optional product-category IDs with the same limits
manual_code	Optional unique coupon code, maximum 191 bytes after normalization; blank generates a code

Boolean columns accept Yes/No, True/False, 1/0, On/Off and Y/N without case sensitivity. Blank uses that column's default. Bulk notes/messages are shortened to 1,000 characters where multibyte support exists, otherwise 1,000 bytes; the issuer's 2,000-byte storage cap also applies. Product/category IDs must exist. Repeated references and repeated manual codes in the same CSV are errors. Bulk cannot create purchased, gifted, partner-payout or customer-referral provenance; use the corresponding product/payment, partner or referral workflow.

Select the file and **Preview CSV**. No credit is issued by preview. The preview lasts 30 minutes and belongs to the administrator who uploaded it. It shows valid/error counts, the valid rows' face-value total, all detected errors, and Row, Reference, Recipient, Amount, Source, Email and Expiry. The total can include already-issued references, so it does not necessarily equal newly created value. Product/category restrictions, tax/shipping settings, notes/message and manual codes are not expanded in the preview table; inspect them in the CSV first.

Fix every error and upload the corrected file. A preview with errors has no Commit button. **Discard preview** removes that preview without creating value. It is not a way to reverse a completed batch.

When the preview is correct, tick **I reviewed this batch and understand that committing it creates real Store Credit value**, then select **Commit bulk issuance**. Current issuance permission is checked again. Rows run independently: some can succeed while another fails because a customer, code, restriction or store state changed after preview. Successful rows remain committed. The result shows new credits, existing idempotent retries and failed rows, with up to 50 failure messages. Results also expire after 30 minutes.

Keep the original reference when retrying the same intended credit. A committed reference resolves the existing credit rather than adding value, provided its protected recipient, amount, currency, source, expiry, restrictions, coupon compatibility, tax/shipping rules and any supplied manual code still match. A new reference creates a new intended issuance. References are shared across administrators on that site, so coordinate their naming.

Re-uploading a committed reference does not update its notes/message, change its email choice or send an email again. Use the existing credit's resend control for delivery. Correct rows that failed before commitment and re-upload with the original references; do not change an already-issued amount under the same reference. If the browser or request stops before showing results, inspect existing credits and retry those original references. There is no whole-batch undo.

Resolve merchant errors

Use the field-specific message to correct missing recipients, ambiguous customer lookups, invalid amounts, nonfuture dates, missing products/categories or duplicate codes. For an ambiguous customer name, choose an exact user ID, login or account email; use External recipient email when no account exists. For CSV errors, retain the exact supported headers, remove duplicate references, keep within 250 rows/512 KiB, and fix every listed row before committing.

An expired or unavailable bulk preview must be uploaded again by the administrator who will commit it. A missing confirmation requires checking the batch and ticking its confirmation checkbox. A permission error in Operations applies to both viewing and changing tools: the account needs Store Credit and WooCommerce management permission. License restrictions on new value are handled in License.

If a refund restoration exceeds the current maximum, check previous restorations and the original order/refund before changing the amount. If revocation is blocked by a reservation, resolve the original order's settlement or release; do not erase the hold. A zero-balance requirement for Archive means both available and reserved must be zero. If email fails, resend delivery of the existing instrument rather than creating more credit.

For an uncertain write or interrupted bulk commit, inspect saved balance/history before another request. The technical error lookup lists exact codes/messages and [ledger reconciliation](#) explains financial evidence checks. These tools report and preserve existing evidence; they do not silently roll back a committed financial action.

Create a gift card product

Open **AffStackDev** → **Store Credit & Gift Cards** → **Gift Cards**. The **Sell gift cards** section explains the sales model and provides **Create draft gift card** and **Open WooCommerce products**. Each paid unit creates a separate gift card with its own reusable code and balance. Changes to a product do not rewrite cards already issued.

Choose **Create draft gift card** to open an unpublished WooCommerce product named **Gift Card**. It starts as a Simple product with Virtual selected, Tax status None, regular price 25 and stock management off. Edit its ordinary WooCommerce name, description, product image, visibility and other catalog settings, configure the Gift card tab, then publish when ready. Creating the draft does not make it available for purchase.

For an existing product, use the native WooCommerce product editor. Select **Simple product**, enable **Virtual**, set a positive regular price and choose **Tax status** → **None** in General. Select **Gift card**, then open its **Gift card** tab. Variable products and gift-card variations are not supported. The plugin does not silently change native product type, tax status or price to make a product valid. The sale must remain nontaxable even when the issued gift balance is allowed to cover tax on later purchases. A printed card that is shipped is the one exception to Virtual: see [Physical gift cards](#).

Access requires both the store-credit management permission and WooCommerce management permission. Creating a draft also requires permission to edit products; saving an existing product requires permission to edit that product. The independent React product editor has no separate gift-card integration; use the native editor that displays the Gift card checkbox and tab.

The **Gift card products** list shows Product, Status, Value / recipient, **Edit product** and **View**. It includes enabled published, private, draft and pending products, with 20 per page and Previous/Next navigation. **Configuration needs review** means the saved gift settings could not be read; open the product, correct its settings and save. A normal product image is catalog artwork. From release 1.0.6 the pictures a buyer chooses from are the product's [Card designs](#); there is no live email or card preview.

Choose the value and limits

Gift card setting	Starting value	How to use it
Gift card value	Gift card equals product price	Choose one of the four modes described below.
Fixed face value	25	Used by Separate fixed face value; the customer still pays the native product price.
Preset values	25, 50, 100	Up to 20 distinct positive values, separated by commas; customers choose one.
Custom minimum	5	Smallest permitted customer-entered amount.
Custom maximum	500	Largest permitted amount; cannot be below the minimum.
Custom increment	1	Allowed steps counted from the minimum.
Recipient	Optional gift recipient	Choose Purchaser only, Optional gift recipient or Gift recipient required.
Maximum units per product / order	10	Set 1–20. Split lines for the same product share this cap; all gift products together are capped at 20 units per order.
Expiry after issue (days)	0	Zero means no expiry; otherwise 1–3650 days. The actual clock begins at payment or scheduled delivery, whichever is later.
Scheduled delivery	Off	Allows a future delivery choice only when a recipient email is supplied.
Maximum scheduled-delivery window (days)	365	Set 1–3650 days from the time the buyer chooses the gift.
Gift card tax budget	Tax-exclusive discount budget (existing behavior)	Controls which tax amounts can be covered when the card is spent; see Value rules .
Cover shipping	Do not cover shipping	Controls shipping coverage at redemption under the same value rules.
Allowed redemption product IDs	Blank	Optional product or variation IDs, separated by commas; limits spending rather than buying this gift product.
Allowed redemption category IDs	Blank	Optional product-category IDs; limits spending.
Combine with coupons	On	Allows the issued gift balance to combine with ordinary coupons when used. It does not allow coupons to discount the gift-card purchase.
Design pictures	Blank	Up to 12 Media Library pictures the buyer chooses from; see Card designs .
Top-ups	Off	Lets customers add money to a gift card they hold through this product; see Add money to a gift card .

Gift card setting	Starting value	How to use it
Physical card (shipped)	Off	A printed card shipped with the order; see Physical gift cards .
Also email a physical card	On	Used only for a physical card: also send the gift card email when the card is issued.
Holder card (anyone holding it can spend it)	Off	Used only for a physical card, with holder gift cards turned on in Settings: the card has no recipient and is never emailed; see Holder gift cards .
PIN under a scratch-off	On	Used only for a holder card: the card gets a 6-digit PIN entered after its number.

Gift card equals product price charges and issues the current native Woo product price. A valid native sale price is honored. **Separate fixed face value** charges that price and issues the configured value: a product priced at 25 with face value 30 gives one 30-value card for 25, or two separate 30-value cards for a quantity-two charge of 50. **Customer selects a preset value** uses the selected amount as both charge and card value. **Customer enters a value** also charges and issues the chosen amount, but enforces the minimum, maximum and increment.

Custom increments start at the minimum. With minimum 5, maximum 20 and increment 3, valid values are 5, 8, 11, 14, 17 and 20. Enter decimal amounts without a currency symbol, thousands separator, sign or exponent. Use no more decimal places than the store currency supports. Zero and negative amounts are invalid. Even preset/custom products need a positive base regular price so the native product remains purchasable.

Restriction lists allow at most 200 submitted IDs each. Duplicate IDs are collapsed and zero is ignored. They are typed ID fields, not new searchable product selectors; use actual product/category IDs. For **Gift card tax budget**, choose **Tax-exclusive discount budget (existing behavior)** or **Tax-inclusive discount budget**. For **Cover shipping**, choose **Do not cover shipping** or **Use remaining credit for eligible shipping**. Tax-inclusive or shipping-covering cards also have a calculation limit of 100,000,000,000 minor currency units, equivalent to 1,000,000,000.00 for a two-decimal currency. Keep the face value within that limit; an excessive value can fail when the paid purchase is issued. The complete tax, shipping, product and category eligibility rules remain attached to each issued card. Older cards retain their original rules when you change the product.

Save settings and use the editor without JavaScript

Save or update the product after configuring it. Invalid amounts, missing settings, unsupported type or tax settings produce an editor error and do not save a new gift configuration. Turning off **Gift card** stops treating the product as a gift-card sale while retaining its previous configuration for later editing; it does not revoke previously issued balances.

With JavaScript, selecting Gift card opens its tab. The tab hides fields that do not apply to the chosen value mode or recipient mode without clearing their stored values. Switching back restores them. Changing Woo's product type does not automatically disable the opt-in, but unsupported types are still rejected when saving or buying.

Without JavaScript, select Gift card and save to reveal the settings. General and the enabled gift settings remain available in the PHP page. Complete all required settings, including the stored amount values used by other modes, because server validation still checks them. Use the native Woo save/update control; there is no separate gift-card save button.

Card designs

From release 1.0.6 a gift card product can offer pictures. In the product's **Gift card** tab, under **Card designs**, **Choose pictures** opens the Media Library: pick up to 12 pictures and select **Use these pictures**. Their IDs fill **Design pictures** in the order you picked them; without JavaScript, type the IDs separated by commas. The first picture is the standard design. Save the product. Only Media Library pictures are accepted: a 13th picture, or an ID that is not a picture, is refused when you save.

On the product page the buyer sees **Card design** with each picture and its name (the picture's alternative text, else its title) and picks one; the first is chosen at the start. A storefront that adds the card through the Store API without a choice gets the standard design. The choice belongs to the cart line (**Card design** in the cart and on the order) and to the frozen purchase, and is saved on each card when it is issued. The gift card email shows the picture above the card, **My Account** shows it on the card, and the [printable card](#) carries it.

If you later delete a picture from the Media Library, cards that used it simply show no picture; nothing else changes, and the product's **Design pictures** drops that picture at its next save. Changing the designs changes what new buyers see; cards already issued keep their picture. Top-ups and products without pictures show no design choice. Pictures in landscape format work best: the product page shows them at 8:5 and the email at most 560 pixels wide.

Physical gift cards

From release 1.0.6 a gift card can be a printed card that you ship. Leave **Virtual** off in General, so WooCommerce asks for a shipping address and charges shipping as for any shipped product, and tick **Physical card (shipped)** in the Gift card tab. Tax status stays None. A virtual product with Physical card, or a shipped product without it, is refused when you save. Scheduled delivery and **Top-ups** do not apply to a physical card product and are switched off for it.

The card is issued when the order is paid, exactly like an emailed card: for its value only (not the shipping), with its own code, tied to the buyer's account or email, or to the recipient's email when the buyer entered one. The product page tells the buyer that the card is shipped and works with the recipient email (or their own when they leave it empty). **Also email a physical card** (ticked at the start) also sends the usual gift card email when the card is issued; untick it when the printed card should be the surprise. Print the card from the order, **Purchased gift cards** → **Printable card (PDF)** next to each card (see [Print a gift card](#)), and put it in the parcel.

A printed card is used like any gift card: with its code and the email it is tied to, or from the account it belongs to. A printed card that works for whoever holds it, without an email (as sold at a shop counter), is a [holder gift card](#). Refunds follow [Stop a purchase before refunding](#).

Holder gift cards

From release 1.0.7 a gift card can work for whoever holds it, like cash: a holder gift card has no recipient and no email, and anyone who has its card number (and PIN) can spend it. Holder gift cards are off at the start. In **Settings**, tick **Allow holder gift cards (printed cards that anyone holding them can spend)** and select **Save settings**. Unticking it later stops new holder cards (sales, batches and activations); cards already sold or activated keep working.

A holder card's number is GC- followed by 16 characters in four groups, for example GC-7K3M-Q9TD-2XWR-H5PN; the letters I, L, O and U are never used. With a PIN, the shopper enters the number and the 6-digit PIN together in the coupon field, for example GC-7K3M-Q9TD-2XWR-H5PN-482913. Spaces instead of dashes, lower case and the letter O for a zero are accepted. A number entered without its PIN is refused with a hint to add the

PIN. Any billing email works, and so does guest checkout when your store allows it. A customer who links the card in My Account (**Link an emailed gift card or credit**, with the number and PIN) makes it theirs: from then on only that account can spend it.

Selling holder cards online: On a gift card product with **Physical card (shipped)**, tick **Holder card (anyone holding it can spend it)** and, for a PIN, **PIN under a scratch-off** (ticked at the start). The product page then has no recipient fields, and a gift message is printed on the card. Each paid unit becomes one holder card; no gift card email is sent. Print each card from the order (**Purchased gift cards** → **Printable card (PDF)**): the PIN and the QR code, which holds the PIN, are in one box; cover that box with a scratch-off label before you put the card in the parcel. A product with Holder card but without Physical card is refused when you save.

Cards for a card printer: Open **Gift Cards** → **Holder gift cards**, and under **Make a batch** fill in **Batch name**, **Number of cards** (1 to 500), **Value of each card** (leave it empty to enter the value when each card is sold), **PIN under a scratch-off** and **Expires after activation (days)** (0 means the cards do not expire). Select **Make batch**. No value is issued yet: a card holds nothing until it is activated, so a lost stack of unsold cards is worth nothing. In **Batches**, each batch has:

Download	What it is
Numbers for the printer (CSV)	One row per card: batch, card position, card number, PIN, the code to enter, the value, the state and the QR code address. Card printers take such a file for variable-data printing. The file holds the PINs: keep it as safe as the cards.
Card sheet (PDF) , 100 cards per file	10 business-card-size cards per page with light cut lines (3.5 × 2 in on Letter for stores in the United States and Canada, 85 × 55 mm on A4 elsewhere): store name, value (when the batch has one), card number and, for PIN cards, the PIN and the QR code in a box marked Scratch here . Print it on business-card stock and cover the boxes with scratch-off labels. Void cards are left out.

Activating a card when it is sold: under **Activate a card when it is sold**, enter the **Card number** (with or without its PIN), the **Value sold** (only when the batch has no value) and, if you like, a **Note (staff only)**, then select **Activate card**. The card becomes a gift card with the source *Holder gift card* and can be spent at once; its expiry counts from now. Activating the same card again changes nothing. For a batch with one value, **Activate cards** from-to activates up to 50 cards at a time, for example for an event. Activating issues new value, so it needs a license that allows new credit.

Lost or stolen cards: **Void unused cards**, with its confirmation, voids every card of the batch that has not been activated: they can never be activated. An activated card is a normal credit: find it in **Credits** (its recipient shows as *Holder card (anyone holding it)*) to see its balance and history, or revoke what is left.

Buy for yourself or another recipient

Open the product page using **Choose gift card options**. Gift-card products do not use an archive's one-click AJAX add-to-cart shortcut. Choose the gift value if a preset or custom amount is required; otherwise the page states each card's value.

For **Purchaser only**, no recipient field is offered. A signed-in self-purchase belongs to the order's customer account; a guest self-purchase uses the final billing email. For **Optional gift recipient**, leave the email blank to buy for yourself or enter the recipient. **Gift recipient required** cannot be purchased without that email. A recipient does not need an account to receive a gift.

When gift delivery is allowed, the form offers **Gift recipient email**, **From (optional)** and, if enabled for the product, **Deliver on (optional)**. The sender appears in the recipient's email. A sender or scheduled delivery requires an actual recipient email. The optional **Gift message** is customer-facing and will appear in the delivery email; do not enter private merchant or order notes there.

The browser limits sender text to 120 characters and the message to 1,000 characters. The server also limits sanitized sender text to 240 bytes and message text to 2,000 bytes, so text using multibyte characters may reach the server limit earlier. Email must be valid as entered. If the form or nonce is stale, reopen the product page and choose the options again.

One cart line carries one recipient, sender, message and delivery choice. Quantity two creates two independent cards for those same details. Add the product separately for a different recipient or message. The product's cumulative unit limit still applies across those lines, and the order cannot exceed 20 gift cards overall. Each card retains its own code and can fund eligible orders until its balance is spent or it expires.

Cart details show **Gift card value (each)**, the recipient or **Yourself**, optional **From** and **Gift delivery**, and the **Card design** when the product has designs. The gift message is not repeated in that cart summary. Order details use gift-card labels while preserving the saved original purchase information.

Coupons cannot lower a gift-card purchase price. A separate fixed face value or a positive native sale price can establish the merchant's offer; ordinary coupons and external price overrides cannot reduce the frozen charge afterward. Other eligible merchandise in a mixed cart may still receive its valid discounts. New gift purchases also require the product's licensing conditions to permit sales; see [Licensing feature gates](#).

If a product's settings, native price, supported type or store currency changes while it is in the cart, remove the affected line and add it again. Different custom values on separate lines are supported. Invalid rows remain removable; checkout does not silently replace their purchase choices.

Use a block product template

Cart and Checkout Blocks can continue to handle checkout. If the product template does not include the classic add-to-cart form hook, open **Block template setup** in the product's Gift card settings and copy its product-specific shortcode. Put `[afsd_store_credit_purchase product_id="123"]`, with the correct product ID, in a Shortcode block in place of the ordinary add-to-cart control.

The shortcode supplies the same validated options, quantity and **Add gift card to cart** button. It appears only for an enabled, published, purchasable, in-stock product. The quantity field shows a maximum of 20, but a lower configured product maximum still applies. Custom storefront clients can provide these options through Woo's authenticated Store API extension; the exact fields are in the technical purchase reference. An ordinary block add-to-cart button does not itself create the recipient and schedule fields.

Add money to a gift card

From release 1.0.6 customers can add money to a gift card they already hold. Tick **Top-ups** on a gift card product (it can be your normal gift card product); its value settings decide the amounts, for example **Customer enters a value** with your minimum, maximum and increment. A physical card product cannot take top-ups.

In **My Account**, each of the customer's own gift cards without an expiry date shows **Add money**. It opens the first published product with Top-ups, with that card chosen: the page names the card by its masked code, shows its available balance and asks only for the amount (no recipient, sender, message, delivery date or design). The cart line shows **Amount added (each)** and **Top-up for gift card** with the masked code, and the order keeps them.

The money is added when the order is paid, under the same rule as a gift card purchase (processing or completed; cash on delivery once completed), exactly once per unit: the card's history shows *Topped up* with the order. The card keeps its own rules (what it can pay for) and stays without expiry. Payment retries and repeated status changes add nothing more. A top-up is refused for a card of another account, a card with an expiry date, a revoked, archived or expired card, store credit (only gift cards take top-ups), and through a product without Top-ups. Like a paid gift card, a paid top-up is added even while the license waits for activation.

Before refunding a top-up order, use Stop a purchase before refunding: the stop takes the added money back out of the card (*Top-up taken back*) while it is unused, and the card stays active with its earlier balance. If the money was already used or is held by another order, the stop says so and the refund stays blocked until that is resolved.

Payment and gift delivery

Gift value is created after the order reaches accepted paid conditions: processing or completed, with Woo's paid state and paid date. Pending, on-hold and failed orders do not issue value. Cash on delivery waits for completed; processing alone does not establish that cash has been collected. Custom order statuses do not substitute for these conditions. Record offline payment as collected only when it actually has been received.

Checkout saves and verifies the original purchase plan before the payment gateway is called. It freezes value, recipient/owner, restrictions and delivery details. Later catalog edits do not change that existing order's gift promise. Changing its paid line amounts, quantities, product identity, billing email or customer identity can stop issuance for review. A paid order without its original validated plan is not automatically rebuilt, and manually adding a gift product to an already paid order does not create valid original purchase evidence.

When a gift email already belongs to a WordPress user at plan creation, the card is intended for that account and its email is resolved at delivery. Other gifts remain bound to the recipient email. Registering an account later does not automatically move an already email-bound card into it; use **Link to my account** as described below.

Every order item and unit has a permanent issuance reference. Repeated payment notifications or retries reuse an already issued card rather than create a second balance. If only some units were issued before a failure, their value remains intact and a retry can issue missing units. Issuance and email delivery have separate status and retry paths: a failed email does not cancel the card.

Schedule a gift

When the product permits it, choose **Deliver on (optional)** in the displayed store timezone. The browser's local timezone does not override it. Leave it blank for delivery after payment is accepted. The time must be a real future date/time within the product's allowed window and must include a recipient email. A time normalized by the server, such as an invalid calendar date or a nonexistent local time, is rejected.

A future scheduled gift creates neither its card value nor its initial email before it is due. The saved schedule is converted to UTC, so a subsequent store timezone change does not move the promised moment. If payment is accepted after the chosen time, issuance can proceed once payment qualifies. The worker rechecks the same original evidence and stop/refund conditions when delivery becomes due.

The system prefers Action Scheduler and falls back to WP-Cron. Keep the site's scheduled-action/cron runner working. The chosen time is the earliest permitted delivery time; worker delays and mail transport can make receipt later. Expiry is calculated from the later of payment time and scheduled delivery time, not from a delayed worker's eventual execution. A delayed run does not extend the expiry automatically. If the calculated

expiry has already passed before a missing card can be issued, issuance is refused for review; the merchant must reconcile that purchase.

Review or retry a purchase

Open the Woo order's **Purchased gift cards** panel. **Issuance status** shows progress, and an error appears when review is needed. Links identify each issued card, its status, item and unit. Open a linked card for its balance, history and **Resend issuance email** control.

The usual purchase states are awaiting payment, scheduled delivery, issued, stopped, stopped and refund-ready, and needs review. A successful retry may create no new cards because every unit already exists. The system makes up to three scheduled issuance retries after an exception, with waits of approximately 5, 10 and 15 minutes after successive failed attempts. This is separate from the email retry queue.

After correcting a transient storage or service failure, use **Retry gift card issuance (paid orders only)** in **Order actions** and click Apply. The action requires the relevant store-credit, Woo management and order-edit permissions. A retry does not resend an already attempted delivery email, reconstruct missing original paid evidence, change the gift value or resume a stopped order. If an issued coupon is missing, review its existing ledger before creating any replacement. Preserve the WordPress authentication salts with database backups, because the saved purchase evidence is signed with them.

Stop a purchase before refunding

Before any refund on an order containing a gift-card purchase, including a partial refund or a refund of other merchandise in that order, open **Purchased gift cards**. Select **I confirm permanently stopping ALL purchased gift cards on this order. I will request the cash refund separately**. Then select **Permanently stop and revoke purchased gift cards (no cash refund)** in **Order actions** and apply it.

This permanently stops remaining issuance for the whole order and attempts to revoke its issued gift cards. It does not request a gateway refund, restore spent value or erase transaction history. A card reserved by another order must be released or resolved through that order. Positive net redeemed value also blocks refund readiness until the financial situation is properly reconciled. Review every linked card if some revocations fail; successful revocations and the permanent stop remain in effect.

For money added to a card (top-ups), the stop takes that money back out of the card instead of revoking it, while the money is unused; the card stays active. Used or held top-up money blocks refund readiness in the same way.

When the order is stopped and refund-ready, request the normal WooCommerce cash refund separately. Readiness requires the issued cards to be revoked, with no live reservation and no positive net redeemed value. A paid order missing its original gift evidence is blocked for reconciliation.

Attempting the Woo refund first permanently stops future issuance but does not silently revoke active cards. The refund can therefore be refused while the order remains stopped. Abandoning or failing the cash refund does not undo that stop. Cancelling or refunding the order, or later moving it out of cancelled/refunded status, also does not authorize fresh replacement issuance. There is no resume button; use a new order for a replacement purchase.

This guard operates on WooCommerce's normal refund-creation path. Refunds made directly in a processor dashboard, chargebacks or an integration bypassing that path need separate review. A refund of only the ordinary merchandise in a mixed order still follows the same whole-order gift-card rule.

Receive and use a gift without an account

The recipient receives a code at the supplied email; no account is needed merely to receive the gift. For an email-bound card, enter the code in the coupon field at cart or checkout and use the recipient email as the billing email. Guest redemption is available only when the store itself permits guest checkout. An account-bound card requires signing in to its owner account. The code does not make the balance transferable to an unrelated email or account.

Gift-card and store-credit delivery share **WooCommerce → Settings → Emails → Store credit / gift card issued** and retain its existing saved email settings. The inherited Woo email settings control delivery and saved subject/heading overrides. Default gift wording is **You have a gift card at {site_title}** and **Your gift card is ready**; other credit uses the corresponding store-credit wording. The plugin does not add a separate branded gift-image designer or live email preview.

Understand the email

Gift-card emails include original gift value, full code, expiry when present, tax/shipping policy text, optional sender and customer gift message, and applicable product/category names. If the available gift balance now differs from its original value, the email also shows **Available balance**. A resent partly spent card therefore keeps its original face value while showing its current available value. Product/category name lists are limited to the first 20 saved IDs of each kind; the complete stored restrictions still apply.

How to use your gift card explains the owner or recipient-email requirement. An account-bound card links **View Store Credit & Gift Cards**. An email-bound gift links **Add this gift card to your account (optional)**; other email-bound credit uses **Claim this credit to your account**. Both offer **Shop now**. Opening the claim link alone does not link the card; sign in and submit its code.

The HTML email uses Woo's header/footer and a simple text-based gift-card panel. The plain-text edition carries the same core value, code, policy, message and links without HTML markup. Internal order notes and private merchant notes are not included. The optional customer message is intended for the recipient. No automatic copy button is supplied; select and copy the visible code normally.

Recover a missing email

First check the order's payment and scheduled-delivery state. A gift scheduled for the future has not yet been issued or emailed. For an issued card, open its merchant detail to inspect the last email status and attempt time, check the recipient account/email and mail configuration, then use **Resend issuance email** if appropriate. See [Credit controls](#).

Failed ordinary sends have a separate queue with up to three retries, approximately 5, 10 and 15 minutes after successive failures. A recorded sent result suppresses ordinary retry. Disabling the Woo email or choosing not to send when issuing ordinary credit prevents its normal automatic delivery; a deliberate merchant resend uses the forced path. A forced resend does not itself queue retries if it fails.

Resending never issues more value, resets expiry or reactivates a revoked card. It uses the current owner email or current email restriction and includes the same existing code. A sent status means the mailer accepted the send, not that the message reached the inbox. If the send succeeded but its status could not be saved, a later retry can send another email for the same card; it still does not create another balance.

Print a gift card

From release 1.0.6 every card can be printed. The PDF is one page (Letter for a store based in the United States or Canada, A4 elsewhere) with your store's name, the card's picture when it has one, *Gift card* or *Store credit*, its value, its code, a QR code, the amount available when printed if some was used, its expiry, the sender and message of a gift, what it may be spent on, how to use it and your store's address.

Who	Where
The customer who owns the card	My Account → the card → Download as PDF
Staff	The credit's screen → Download printable card (PDF) ; on an order that sold gift cards, Purchased gift cards → Printable card (PDF)
The recipient of a gift card email	When Attach the printable gift card (PDF) to the gift card email is ticked in Settings (it is not at the start), the gift card email carries the PDF

Nobody else can download a card: the customer link works only for the signed-in owner, and the staff link needs Store Credit management. The file is made when it is downloaded or attached and is not kept on the server; its name holds the store's name and the card's number, never the code. Text in scripts written right to left or with joined letters (for example Arabic or Hebrew) is left out of the PDF; the email carries it.

The QR code opens your store's cart with the code after a "#" in the address. That part of an address is never sent to the server, so the code does not reach server logs or analytics. In the cart, the store applies the code with WooCommerce's own coupon check; if the cart is empty, it says that the code is ready and applies it when the customer opens the cart again with products in it, in the same browser tab. A code tied to an email still needs that email as billing email at checkout.

Link an emailed card to My Account

Sign in to an account using the recipient email, then open **My Account** → **Gift Cards** or **Store Credit & Gift Cards**. Under **Link an emailed gift card or credit**, enter its full code and choose **Link to my account**. An account is optional for a card that can be redeemed using the recipient billing email; linking makes its remaining balance available in the account area.

The account must have the exact recipient email, the code must belong to this store's gift-card/store-credit system, and the card must not already belong to someone else. A card reserved for an order cannot be linked until that order is finished or cancelled and its reservation is resolved. Revoked or archived cards cannot be linked. Refresh and retry if the claim request has expired.

Linking preserves the code, value, expiry, restrictions and transaction history. It replaces the email restriction with account ownership, so a later legitimate change to the account email does not disconnect the card. It does not transfer value, generate a new card or restore expired/spent value. An expired or empty card may be linkable but remains expired or empty. Repeating a successful link to the same account is harmless when no reservation or other blocking state is present.

If saving cannot be confirmed, refresh the account page and review the existing card rather than creating a replacement balance. There is no customer unlink or ownership-transfer control. Registering an account with a matching email alone does not claim older email-bound cards automatically.

Read balances and activity

The account menu is **Gift Cards** when there is no available store-credit balance and **Store Credit & Gift Cards** when store credit is available. The gift-card destination remains available even at zero balance. Purchased and gifted instruments appear as Gift Cards; other sources appear as Store Credit. Classification uses their saved source, so older valid balances do not need to be reissued.

Balance cards separate gift cards from store credit and keep different currencies separate. Gift Card balance shows the store currency even at zero. Store Credit balance appears for currencies with available store credit. Reserved value is not available to spend again; expired or revoked value is not counted as available. Email-only cards that have not been linked are absent from account totals.

Buy a gift card links eligible visible published gift products when available. **Your gift cards** and **Your store credit** show each card's type, status, full code, available balance and currency, expiry or **No expiry**, and any amount reserved for an order. **Use at cart or checkout** opens the cart; it does not automatically apply the code. Depending on your settings, a card also shows **Download as PDF** ([Print a gift card](#)), **Send to someone** ([Send balance to someone else](#)) and, on gift cards, **Add money** ([Add money to a gift card](#)). Select and copy the full code if you want to use the ordinary coupon field.

The page lists up to the latest 100 available cards/credits, while totals include all eligible cards linked to the account. **Past cards and credit activity** expands older/inactive entries and a bounded recent-activity list. Each activity row identifies the kind, transaction type, local date and related order when present. The displayed amount is unsigned; use the transaction type to understand whether it represents issuance, redemption or another change. Reservation movements can appear without an amount because they only move existing value between available and reserved.

This recent list is not a complete downloadable account ledger. If balances cannot be loaded, refresh and try again; the page displays a read error rather than inventing a zero balance. Expired, revoked or other inactive cards retain their history. See [Ledger accounts](#) for the detailed eligibility rules.

Apply or remove account credit

On the classic cart and checkout, signed-in customers can use **Available Gift Cards** and **Available Store Credit**. Up to 50 eligible account-owned instruments in the store currency are offered, with masked reference codes and available amounts (in another currency, also what each pays there; see [Shoppers in another currency](#)). Choose **Apply** to use Woo's normal coupon validation or **Remove** to take a selected card out of the cart. Applying a card does not complete an order or transfer ownership; eligibility and final payment rules still apply.

Refresh if the action cannot be verified. A card not available to the signed-in account cannot be applied through this chooser. You can also enter a permitted code in the ordinary coupon field. Blocks have their corresponding account panel and controls, described under [Blocks](#); the underlying restrictions remain the same.

The account, claim and classic cart controls use normal forms and remain usable without customer-side gift JavaScript. Narrow screens stack the cards and controls, and long account codes wrap. The cart chooser's masked reference is not a full code; obtain the full code from the account card or delivery email when using manual coupon entry.

Send balance to someone else

From release 1.0.6 customers can pass value on when you allow it. In Settings, **Customers may send balance to someone else** offers **Off** (the start), **Gift card balance** or **Gift card and store credit balance**. Refer a Friend rewards and partner payout credits can never be sent.

In **My Account**, each card that may be sent shows **Send to someone**. The form asks for the **Amount** (at most what the card has available, with the store's decimals), **Their email address** and **Message (optional)**. Select **Send**. In one step the amount leaves the sender's card (*Sent to someone else* in its history) and a new card is created for the recipient (*Received from another customer*): a gift card from a gift card, store credit from store credit, with the same product and category rules, combining rule, tax and shipping rules, expiry and picture. When the email belongs to a customer account, the new card belongs to that account; otherwise it is tied to that email. The recipient gets the usual email with their own code, the sender's name and the message.

A send cannot be taken back. The same form sent twice (a double click) moves the value once. Refused: more than the card has available, zero, more decimals than the store uses, the sender's own email, another customer's card, and a card that may not be sent. A customer can send at most 10 times a day. Sending creates no value, so it also works while the license waits for activation. A sent card's source is *Gift card sent by a customer* or *Store credit sent by a customer*. In Reports the period's issued total leaves it out, because no value was created; Issuance sources lists it under its own source.

Resolve gift purchase and account errors

For a rejected amount, recipient or delivery choice, correct the field named in the message and submit the current product form. For changed cart evidence, remove the gift and choose its options again. For a paid purchase that needs review, preserve its original order and issued cards, resolve the stated storage, locking or evidence problem, and retry only the missing issuance. A retry does not bypass a permanent stop or replace missing original paid evidence.

For account linking, use the recipient account and full code, refresh an expired form, and resolve an existing reservation or different owner before trying again. For a refused refund, complete the whole-order stop/revoke process and reconcile any reserved or net-spent value before requesting cash separately. Exact developer error identifiers and messages are listed in the gift purchase and claim error reference.

What credit value represents

Store Credit and Gift Cards use the same reusable balance system. Each instrument has its own WooCommerce coupon code, original value, available value, reserved value and transaction history. The original coupon amount does not decrease when credit is used; the ledger records the changing balance. A code without a funded ledger account cannot spend value.

Available means value not already reserved for an order. **Reserved** means value held for an existing order and unavailable to another checkout. **Redeemed** means value consumed by an order, less any value later restored. Neither a recorded balance nor a credit coupon means money has been paid through a cash gateway.

Amounts are stored in the currency's smallest configured units. For a two-decimal currency, 1,235 units means 12.35. Amounts with extra decimals round at the first discarded digit: 12.345 becomes 12.35 and -12.345 becomes -12.35. An issue amount that rounds to zero is rejected. The store's currency precision controls display

and conversion, so do not change the meaning of existing saved amounts by treating them as another currency or precision.

New credit is issued in the store currency; issuing never converts currencies (spending can: [Shoppers in another currency](#)). If an integration pays a commission as credit, that integration must finish its currency conversion and freeze the amount before asking Store Credit to issue it. Technical callers can find the exact bounds in money normalization.

Shoppers in another currency

From release 1.0.7, store credit and gift cards can be spent while a shopper has switched the shop to another currency with one of these currency switchers:

- WPML Multilingual & Multicurrency for WooCommerce (WPML's WooCommerce plugin; it works on its own, without the rest of WPML).
- FOX - Currency Switcher Professional for WooCommerce, with its **Is multiple allowed** setting on **Yes** (customers pay in the currency they selected). With **No**, FOX only shows prices converted and the order is paid in your store's currency; credit can then be spent while the shop shows your store's currency.

There is nothing to set up in Store Credit. Credit is still kept in your store's own currency, the one under **WooCommerce** → **Settings** → **General**: balances, reports, exports, emails and My Account show it in that currency, with that currency's decimals, whatever currency the shopper is looking at.

At checkout the credit is converted with the switcher's exchange rate for the shopper's currency. WPML's price rounding rules are not applied to credit.

- What a credit can pay is rounded down to the shopper's currency, and what that payment costs the credit is rounded up to your store's currency, so a credit never pays more than it holds. With 1 USD = 0.9 EUR, a 20.00 USD gift card pays EUR 18.00. A 10.01 USD card pays EUR 9.00 and costs 10.00 USD; the 0.01 USD left is too little to pay anything in EUR (the shopper is told so) and can still be spent in USD.
- Currencies with other decimals work the same way: with 1 USD = 150 JPY, 7.77 USD pays ¥1,165.
- The rate is the one in force when the order is placed. If the switcher's rate changes after the shopper applied the credit, checkout uses the new rate for the prices and the credit alike.
- In the cart, each available credit shows both amounts, for example \$20.00 (€18.00).
- The order note gives what was reserved in your store's currency, what it pays of the order and the rate, for example "Store Credit reserved \$20.00 across 1 credit instrument(s). It pays €18.00 of this order, converted at the currency switcher's rate of 0.9 (1 USD = 0.9 EUR)." The rate, both currencies and both amounts are also saved with the order's credit line.

Cancelling such an order, or refunding it fully, gives back exactly what the credit paid, in your store's currency. For a partial refund, **review credit restoration** (linked from the order note) shows the refund in the order's currency and in your store's currency at the order's rate; the most you can restore is that converted amount, rounded the same way as the payment, so refunding exactly what the credit paid gives back exactly what it cost.

Value created from an order in another currency is in your store's currency too, converted at the rate saved with that order at checkout:

What	On an order in another currency
Buying a gift card or adding money to one	The card holds its value in your store's currency and costs that value converted at the switcher's rate, rounded up: a \$25.00 card costs €22.50, a \$10.01 card €9.01. The product page gives the value in your store's currency (preset values also show their converted price), and a custom amount is entered in your store's currency. The cards are issued after payment as usual.
Cashback	The order's products are counted converted into your store's currency at the order's rate (rounded down), then the percentage applies: 10% of a €27.00 order is \$3.00.
Refund an order as store credit	The refund is converted into your store's currency at the order's rate, rounded down: refunding €9.00 gives \$10.00 of store credit. The order note and the refund email give both amounts.
Refer a Friend	Its offers and rewards work only in your store's currency.
AffStackDev B2B quotes	Store credit on a quote paid in another currency is refused at checkout.
A currency the switcher has no exchange rate for	Credit and gift cards are refused: "This store credit or gift card can be used in USD, the currency of this store: switch the shop to USD to use it." and "Gift cards are sold in USD, the currency of this store. Switch the shop to USD to buy one."

Credit is added to an order in another currency only at checkout. When you change such an order on WooCommerce's order screen (for example by applying or removing a coupon), the credit keeps the rate saved with the order and what it holds for the order follows its new amount. The order screen works in your store's decimals: on an order in a currency with other decimals (JPY in a USD store, say), the credit then pays nothing and what it held is given back to it.

Other currency switchers: with any other switcher, credit can be spent only in your store's currency, as before 1.0.7. Aelia Currency Switcher is not supported yet.

Storage availability and preserved financial records

Credit issuance and spending require both local ledger tables to be available and use the InnoDB transactional engine. The plugin checks its schema version and missing tables during setup/update, creates or updates the required definitions through WordPress, then verifies storage. It does not silently convert an unsupported database engine.

A manager may see **Store Credit ledger unavailable**: with an explanation that issuance and redemption are disabled. Ask the site administrator to resolve the database/table issue, then let setup verify it again. Deleting financial records or making another coupon is not a substitute for restoring the ledger.

Each account update records a new version and transaction. Order operations lock affected accounts in a consistent order, and a single order can use at most 20 instruments through the ledger. Concurrent requests must use the current saved balance; a stale checkout can be refused and need recalculation. The service does not allow negative available or reserved balances.

History is retained through adjustments, expiry, revocation and refunds. Corrections are new events, so an administrator can see how the current value arose. WordPress/WooCommerce still own customer details,

products, coupon settings and orders. Exact database fields and transaction boundaries are in the technical schema reference.

Separate Store Credit and Gift Card balances

The type comes from the instrument's saved source. Purchased or gifted value, and gift card balance another customer sent, appears as **Gift Card**. Manual credit, compensation, refund alternatives, loyalty, promotions, partner payouts, referral rewards, cashback and store credit another customer sent appear as **Store Credit**. Older purchased/gifted instruments are categorized the same way without moving or issuing their money again.

Account balance totals scan all eligible account-owned instruments and keep each currency separate. They count available value only, excluding reservations and unspendable expired/revoked/archived value. Recent history and checkout cards are deliberately shorter views: account lookup is bounded to 100 instruments, checkout's safe list to 50, and recent activity normally shows 20 events with a supported maximum of 50. A history/list limit does not truncate the separate complete balance totals.

Ownership is rechecked from the current WooCommerce coupon before account data is shown. A gift issued only to an email address is not automatically an account-owned balance. The recipient uses the [claim workflow](#) to link the existing instrument to the correct account; claiming does not duplicate money. Another person's code or email does not grant account access.

Displayed condition	Meaning
Active	The recorded instrument is available for its allowed uses, subject to positive available value and current checkout checks.
Reserved	An otherwise active instrument has value held by an order. Any remaining available value is still separately counted.
Exhausted	Available and reserved value have both reached zero.
Expired	Its saved expiry has passed or it was explicitly expired. New spending is blocked.
Revoked	A manager has disabled spending while preserving the recorded value and history.
Archived	The zero-balance instrument is retained as history.

The public code-free balance summary uses recorded status with expiry correction; the customer's reserved display is a separate presentation. A displayed balance still has to satisfy ownership, currency, product, coupon-combination and checkout rules before it can be used.

Checkout-safe records show a masked reference such as ••••-A19F, available amount, currency, expiry, type and whether it is applied. They do not expose full codes, email, notes or internal operation references. Recent account events show balance changes and associated order/refund references. Use the [account controls](#) for the actual view and Apply/Remove actions.

Managers can find instruments by credit ID, coupon ID, exact code or customer/email search in [Credits](#). Searches are bounded: up to 200 results, up to 20 matched users and limited coupon candidates per user/email. A missing search result alone is not proof that no credit exists. Administrative listings can filter status, live reservations, source and dates; their active/expired filters take the current expiry into account. Technical integrations use the account reader contract.

Issuing credit and keeping the original identity

Use [Issue Credit](#), the [bulk workflow](#), a paid [gift-card purchase](#) or a supported integration. These controls supply the underlying issuer with a recipient, positive amount and stable operation reference. The ledger applies the source's [license permission](#); fulfillment of an already paid purchase is a different obligation from creating new discretionary credit.

Issuance choice	Effect and restriction
Customer account	Must still exist. When selected, it takes precedence over recipient email and the coupon is bound to that account.
Recipient email	A valid email is required when no customer account is selected. The instrument uses its native email restriction.
Amount/currency	Positive after currency rounding; current store currency only.
Source/reason	Records why value was created. Sources include compensation, refund alternative, loyalty, promotion, giveaway, manual, other, import, purchased, gifted, partner payout, customer referral, cashback, and gift card or store credit sent by a customer.
Code	A valid unused manual code or a generated code with cryptographically random content. Duplicate codes are refused.
Combine with coupons	Allowed by default; disabling it sets the coupon's individual-use behavior.
Expiry	Optional; a new issue needs a future date/time. The saved ledger mirror uses UTC. Each form explains its displayed timezone.
Product/category limits	Restrict eligible products through WooCommerce. The internal issuer retains at most 200 IDs per list; the form/import may enforce a stricter validation rule.
Tax/shipping policy	Defaults to tax-exclusive merchandise credit with shipping excluded. Advanced choices are described below.
Internal note	Financial/administrative note; not a recipient message.
Customer message	Message for the recipient. Notes/messages have a 2,000-byte stored limit after cleaning.
Send email	Enabled by default in the ordinary issuer; an owning integration may intentionally handle its own notice.

Issuance writes the coupon first, then creates its funded account and original issue transaction. The account starts with all issued value available and no reservation. Saved source, recipient, policy and restrictions belong to that instrument; later changes to a product/default do not rewrite it.

Retries must keep the original operation reference and financial choices. A matching completed issue is returned as the same existing instrument. Changed amount, currency, source, recipient, combination policy, product/category restrictions, expiry, tax/shipping policy or manual code is rejected. Generic retry does not replace saved notes or email options. A previously future expiry that has now passed can also stop a repeated issue request before its receipt is read; inspect the original instrument instead of assuming another issue is needed.

A notification or email problem can occur after money was committed. The issuer checks the saved account and exact issue transaction, preserves confirmed funded credit, and invokes its email recovery where appropriate. If the outcome cannot be proved, it retains the evidence for review. Do not delete the coupon or create replacement value merely because the request returned an error. The [delivery chapter](#) explains retries/resending of the same instrument; it does not issue another balance.

Generated codes normally begin ASD-SC- followed by four groups of eight random hexadecimal characters. An installed integration can change the permitted prefix in code; there is no implied customer setting for changing the random generator. Full issuance inputs, replay checks and metadata are in the issuer contract.

Reading the financial history

History event	What happens to value
Issue	Creates the original available balance.
Reserve	Moves value from available to an order hold.
Release	Returns a live order hold to available.
Redeem	Consumes a hold after the order is accepted; it does not deduct available a second time.
Restore	Returns previously consumed credit to available.
Adjustment	Records an authorized increase/decrease or a zero-value expiry-setting change.
Expire/revoke	Blocks new spending while preserving the value/history.
Archive	Retains a zero-balance instrument as history.

For example, a 100.00 issue followed by a 30.00 reservation leaves 70.00 available and 30.00 reserved. Settling that order leaves 70.00 available, no reservation and 30.00 net redeemed. A later 10.00 restoration leaves 80.00 available and 20.00 net redeemed. Original value remains 100.00 throughout.

Transactions retain both changes and resulting balances, currency, time and relevant order/refund/actor references. Notes and reasons help explain intentional corrections. Net redeemed subtracts restorations; it is not the same as gross redemption activity for a report period. Revoking an 80.00 recorded balance does not automatically erase it or turn it into a cash refund.

Compatible plugins can receive a post-commit notification containing safe IDs and monetary changes. A repeated operation may repeat a notification for an existing transaction, so consumers must recognize its transaction ID and avoid another financial effect. Email or external bookkeeping success is separate from the ledger commit. Technical event fields explain what integrations can rely on.

Order reservations and settlement

Checkout calculates a preview, then validates the order and reserves its exact credit allocation against the current ledger. Another order cannot spend the reserved part at the same time. Up to 20 instruments can participate in the ledger operation, and their updates must succeed together.

An existing reservation can be recognized only when it matches the exact account/order amount and is still live. An old hold that has already been released or redeemed is not a reusable permission to reserve again. A failed or uncertain checkout must inspect its original order evidence instead of creating a replacement reservation to hide the error.

When the owning checkout lifecycle accepts the order, redemption consumes the hold without a second deduction from available. A cancellation that releases an unused hold moves it back to available. Exact status transitions, payment timing and restored-order reactivation rules are in [the order lifecycle chapter](#). These ledger actions themselves do not edit product lines or decide that a payment succeeded.

An authorized order recalculation can change a reservation by the difference. Increasing a hold needs enough currently spendable value; reducing it releases the difference. An order can legitimately move from 5.00 reserved to 3.00 and back to 5.00, with separate history for each transition. Equal targets need no event. The calling order workflow checks whether recalculation is allowed; the ledger does not grant permission merely because an order ID is known.

Shared checkout integrations use the same original database transaction and connection-bound evidence, deferring notifications until the coordinator commits. Discovery of a participant alone does not authorize payment or spending. See [checkout transactions](#) and the ledger hold reference for exact proof and concurrency requirements.

Restoring credit after a refund

Restoration returns previously consumed credit to its original account. It does not send cash through a payment gateway. Use the supported [merchant credit controls](#) and order lifecycle; a partial cash refund does not by itself authorize arbitrary extra credit.

An explicit restoration must name credit consumed by that order. Each amount is limited by what the order still owes back to that instrument. When it is attached to a WooCommerce refund, the refund must belong to the same order, and all new restorations together must fit that refund's amount after earlier credit restorations for that refund.

For example, if the refund is 25.00 and 10.00 of credit has already been restored against it, a new combined restore can be at most 15.00. It is also limited by each instrument's still-consumed value. A larger request is refused rather than silently changed to another amount.

The automatic full-remaining restoration path can return all credit still consumed by an order even if its cash refund amount is zero. This matters when credit covered the whole order. An explicit partial allocation follows the tighter refund-total limit. The operation reference distinguishes a repeat from a later, separately reviewed top-up.

After an interrupted or repeated restore request, inspect the existing transaction and current remaining consumed value. Because current bounds are checked before some replay lookups, a completed partial retry can report that the requested amount now exceeds what remains. That response is not a reason to change the operation reference or issue another coupon. The original history shows whether value was returned.

Restoring value does not override expiry, revocation or archival status. The record can receive a balance restoration while remaining unspendable pending the appropriate merchant review. Gift-card purchase refunds have a separate whole-order stop/revoke process described under [purchases](#); returning a purchased card's sale price is different from restoring credit spent on merchandise. Exact conditions are in the refund contract.

Restoring credit an order used is different from refunding the money part of an order as new store credit; for that, see [Refund an order as store credit](#).

Adjust, expire, revoke or archive

Control/service	Effect and restrictions
Adjust available value	Adds or subtracts a nonzero amount with a reason and stable reference. Cannot make available value negative or change reserved value. Revoked/archived records cannot be adjusted. A positive increase needs current adjustment permission; a negative correction has different license treatment.
Revoke	Blocks new use without deleting value or history. Live order reservations and archived records prevent revocation.
Expire	Blocks new spending while preserving balances. Existing reservations can still settle truthfully. Archived records cannot be expired.
Archive	Allowed only when both available and reserved balances are zero. Preserves the instrument and transaction history.
Change coupon expiry	Updates the ledger's expiry mirror as a zero-value audited configuration event. It does not issue money.

Use [credit controls](#) for exact permissions, confirmations and required fields. These actions are not interchangeable with refunds or balance deletion. There is no general undo button that removes financial history.

An exhausted account can become active when value is legitimately restored or added. An expired record normally stays expired through monetary changes. Updating/removing its native coupon expiry recalculates date-based status and can make it active again when it has value and is not revoked/archived. Revoked and archived status remain protected by that synchronization. An unchanged expiry creates no new history event.

Expiry uses the saved time as well as recorded status, so a reached deadline can block spending before a manager opens the record. If an update reports concurrent change, refresh the instrument and review the current state before submitting again. Use the original operation reference after uncertainty; exact low-level replay rules are in lifecycle operations.

Tax budgets and eligible shipping

An instrument's type and its redemption policy are separate choices. A Gift Card can retain its saved tax/shipping restrictions just as Store Credit does. Older instruments without policy metadata remain tax-exclusive merchandise credit with shipping excluded.

Choice	What consumes the credit budget
Tax-exclusive discount budget	The net merchandise discount. WooCommerce may also lower tax because the merchandise price was discounted.
Tax-inclusive discount budget	The net merchandise discount plus its actual rounded tax reduction.
Do not cover shipping	No credit allocation to shipping.

Choice	What consumes the credit budget
Use remaining credit for eligible shipping	Remaining budget can reduce an eligible chosen shipping rate after merchandise allocation.

For an eligible item with a simple 10% tax rate, an 11.00 inclusive budget can fund a 10.00 merchandise discount and 1.00 tax reduction. Tax-exclusive credit uses 10.00 of its budget for that same net discount. WooCommerce performs the tax calculation. This is a discount budget, not a new payment gateway or a promise that tax stays due on the original undiscounted price.

Rounding can leave a small remainder when the next permitted discount increment would exceed the budget. The balance is retained. Tax-exempt customers and non-taxable items remain tax-free. The plugin does not change the store's tax settings automatically.

Advanced tax-inclusive or shipping policy requires normal WooCommerce item calculation, per-line tax rounding and valid noncompound rates from 0% through 100%. Subtotal rounding and integrations replacing the specified item-tax/discount calculation are refused. An unsupported setup needs a compatible integration or a supported policy choice, not a manually edited amount. Advanced calculations are bounded to 100,000,000,000 minor units; the major-unit equivalent depends on the currency precision.

Shipping coverage never makes excluded merchandise eligible. Credit-sale products are excluded to prevent buying more value with that same value. The shipping integration controls package eligibility, selected-rate changes and signed order proof; see [shipping credit](#). Do not infer total ledger spending from the merchandise coupon's net discount alone when inclusive tax or shipping is involved.

Issuance/product screens save these choices on each instrument. A later default/product change does not silently rewrite existing credit. The technical policy reference explains the exact saved fields, calculation boundaries and tax-context handling.

Stable references and protected evidence

Supported integrations keep one stable reference for one issuance or payout. The local database serializes competing requests for that reference with a five-second lock wait. If it is busy or unavailable, check the original receipt and retry the same identity after the condition is resolved. Replacing the reference can describe another financial operation.

Shipping and payout evidence is signed against the site's WordPress authentication salt. A changed payload, invalid signature or changed/unavailable signing configuration prevents it being accepted as the original evidence. Preserve the original records and restore the correct site context with the administrator. Editing a receipt or signing new replacement data is not financial recovery.

The signature confirms local data integrity; it does not independently prove a customer's entitlement, correct tax policy, payment or commission approval. Each owning workflow still validates those conditions. These locks and signatures are implementation services, not new customer settings. See technical evidence rules.

Reconciliation findings

The reconciliation report is read-only. It compares the recently updated accounts it checks with WooCommerce coupons, latest transaction results and related order evidence. It never repairs money or rewrites a coupon automatically.

Finding	What needs review
Negative balance	Available or reserved value is below zero.
Coupon mismatch/missing	The coupon cannot be loaded or no longer has the correct credit type.
Face-value drift	The coupon's original amount differs from the account's original issue value.
Expiry drift	An active account and its coupon no longer agree on expiry.
Missing/drifted history	No latest financial event is returned, or its resulting balances differ from the account.
Missing order	An order still has consumed/reserved value but WooCommerce cannot load it.
Invalid advanced evidence	Saved tax/shipping proof no longer matches the order.
Missing order credit evidence	The order lacks a matching private coupon-item record for its consumed/reserved account.

The service normally checks 100 accounts and accepts a limit from 1–200. Related order evidence is bounded to 100 account/order pairs, and investigation stops around its issue threshold. The result reports counts and truncation flags, so an empty issue list does not claim that every historical account was inspected. A storage failure can also make evidence unavailable; establish database health before concluding that records were deliberately removed.

Use [merchant reports](#) to run/read the result. Resolve the underlying cause through an authorized workflow; do not overwrite the ledger or issue replacement value simply to make a finding disappear. Developers can match exact issue codes in reconciliation.

What reports count

Reports offer 7, 30, 90 or 365 days, with 30 as the default. Period dates use UTC. Results can be cached for five minutes; committed ledger events normally invalidate the period caches. Reporting does not supply checkout's balance authority.

Measure	How to interpret it
Available	Current active, unexpired unreserved value, independent of the chosen report period.
Reserved	Current held value across all account states.
Outstanding	Available plus reserved under those definitions.
Active credit count	Current active, unexpired instruments.
Active recipients	Distinct account owners or first recipient email from a bounded active-account scan.
Issued	Positive issue value created within the period.
Redeemed	Gross value consumed through redemption events within the period.

Measure	How to interpret it
Restored	Value returned through restoration events within the period, reported separately from redeemed.
Expired/revoked	Available value recorded at those status events; repeated events can count the same account again. This is not a unique-account liability calculation.
Orders used	Distinct orders with a redemption event in the period.
Associated order value	Current Woo totals for the loaded related orders. It is not revenue proven to have been caused by credit.

Source summaries show at most 20 source groups. Recent events show the latest 20. Associated order value uses at most 200 newest order IDs, and active-recipient counts scan at most 500 active accounts; truncation flags expose these limits. A boundary flag can also be true when the number exactly equals the limit.

The general report totals follow the store-currency ledger model and do not perform FX conversion or partition every amount by currency. If the store's currency changed while historical value remains, use the original records' currency and the appropriate exports instead of assuming a combined report is a converted total. Customer balance views do show currencies separately.

Report totals, customer history and [liability reporting](#) answer different questions. Use [reports](#) and [exports](#) for their controls, and the technical calculations for exact fields and bounds.

Personal-data exports and retained records

Use WordPress **Tools → Export Personal Data** for an authorized, confirmed privacy request. Store Credit & Gift Cards provides separate exporters for balances/recipient messages and transaction history. WordPress controls confirmation and delivery; an ordinary shopper cannot call these administrative exports as a public balance lookup.

The credit export includes credit ID, currency, original/available/reserved values, recorded state, source, creation/expiry, redemption policy, recipient message and only the last four code characters. History includes event/reason, value changes and resulting balances, related order/refund IDs and UTC time. Full redeemable codes, private financial notes and raw signed proofs are not included in these automated exports.

Export permissions are WordPress's personal-data export permission. The erasure service uses its corresponding erasure permission. Candidates are processed in pages of 50 instruments or 100 transactions, with current coupon ownership rechecked before output. A changed owner or unmatched email is excluded rather than exposing another person's record.

WordPress **Erase Personal Data** removes recipient messages from matched issued credit only. It explicitly retains balances, ownership, transaction history, internal financial notes, signed purchase/order evidence and payout references needed for outstanding value, refunds and reconciliation. This is not complete anonymization. Pending gifts, signed gift/order details and other retained data require a separate merchant retention review under the store's disclosed policy.

If an exporter/eraser reports failure, check the privacy request, permissions and database availability before retrying. The service does not claim completion on error. Some earlier messages may already have been removed, while financial records remain preserved. The plugin adds suggested privacy-policy text describing these behaviors. Exact WordPress callback shapes and error codes are in privacy contracts.

Connecting compatible local products

The plugin exposes a local PHP contract for compatible products. It describes available implementations, provides safe credit summaries and routes, and delegates exact checkout/referral contracts. These are developer interfaces, not an unrestricted editor, a remote bridge or permission for another product to rewrite balances.

A safe credit summary gives the credit/coupon IDs, currency, original, available, reserved and net-redeemed amounts, status, source and expiry. It omits the code, customer email and private notes. The saved policy reader provides tax and shipping rules. An empty summary can mean the coupon is not this type or its account is unavailable; it is not authority to create replacement value.

Capability flags indicate implemented manual issuance, customer accounts, Classic/Blocks checkout, gift-card sales/gifting, tax-inclusive/shipping policy, partial restore, multiple instruments, supported manual subscription renewals, checkout participation/settlement/allocation and referral interfaces. Each still has its own permissions, configuration and evidence requirements. Automatic subscription renewals are not supported. A listed manual-renewal provider must satisfy the [renewal contract](#).

Checkout participant discovery, settlement acknowledgement and cart/order allocation are separate interfaces. Observing an allocation does not calculate or reserve money; finding a participant is not proof that a transaction can be paid. Referral program/continuation readers are passive, while migration, adoption and execution are separate explicit operations under the [referral provider contracts](#). The consumer referral program does not enroll customers as affiliates merely because the interface is available.

The public administration URL helper can direct an authorized user to Dashboard, Credits, Issue, Gift Card products, Referrals, Reports or Settings, optionally a coupon detail. Creating a link does not grant access. The marketing URL is a separate filterable product-page destination. Normal integration calls stay inside the WordPress installation; Store Credit does not require a plugin-to-plugin HTTP bridge.

The technical public contract lists the exact version, every method, safe fields and independent protocol versions. The implementation flag saying a consumer payout adapter is not included means Store Credit does not contain another plugin's commission/payout UI. A compatible installed AffiliPartnerCS product can supply that side.

Partner payouts issued as reusable credit

A supported payout integration can ask Store Credit to issue the already reviewed net payout as reusable account credit. The payout product owns partner consent, recipient linkage, commission liability, fees, bonuses, currency conversion and finance approval. Store Credit owns the resulting instrument and its financial receipt. This does not initiate a cash payment.

A finance user needs both Store Credit and WooCommerce management permissions, the relevant license permission, and an explicit current-user review of the original payout reference. The companion also enforces its own finance permission and batch controls. Scheduled/background work may reconcile existing receipts but cannot initiate a new payout through this reviewed API.

The payout must already name an existing customer account, a positive amount in current store-currency minor units and a fingerprint of the frozen batch. It keeps one permanent origin/reference across retries. The public request supports a recipient message up to 2,000 bytes, combine-with-coupons and email preferences; these

Boolean preferences default on. It does not accept caller-selected expiry or advanced tax/shipping overrides. Issued payout credit is tax-exclusive with shipping excluded.

A repeated matching request returns the original credit receipt. Changing the recipient, amount, currency, message, options or frozen batch under that same reference is refused. If the request times out or bookkeeping is interrupted, read the original receipt before any further payment. A committed matching receipt means credit exists even if the caller never received its success response. Later spending or revocation does not make the original issuance disappear.

The receipt identifies the credit, coupon, transaction, original amount, currency, status and fingerprints without exposing a full code, email or private note. A missing/unverifiable receipt during a storage problem is not proof that another payout is safe. Keep the original batch and reference for reconciliation; do not mint replacement credit or send cash to compensate for uncertainty.

In the compatible AffiliPartnerCS workflow, that product freezes its own batch and recipient, checks its payout/finance conditions, and stores the request before calling Store Credit. It intentionally disables the Store Credit issue email because AffiliPartnerCS owns the completed payout notice. After verifying the receipt, it records AFSD-SC:<credit ID>:<transaction ID> and closes its commission records. If credit was issued but commission closure stopped, recover the remaining AffiliPartnerCS bookkeeping rather than issue again. Historical fixed-cart coupon settlements are separate evidence and must not be converted into another account-credit payment.

These controls belong to the installed payout companion. Store Credit supplies the producer and passive receipt interfaces described in reviewed payouts.

Recovering a ledger or integration error

Problem	Appropriate recovery
Storage unavailable or nontransactional	Restore the required local tables/engine and original financial evidence before issuing or spending.
Invalid amount, currency or policy	Correct the current authorized request; preserve the original values for a completed or uncertain operation.
Existing code or conflicting issuance reference	Find the original coupon/receipt. Do not change the key merely to create a duplicate instrument.
Balance changed or reservation no longer live	Recalculate through the owning checkout/order workflow and inspect its original evidence.
Concurrent ledger change	Reload current state before another intentional operation.
Restore exceeds remaining consumed/refund value	Check previous restores and remaining limits; a repeat may already have succeeded.
Revocation/archive blocked	Resolve live reservations or remaining value through their supported workflows first.
Expired/revoked value after restoration	Review status separately; returning value does not automatically authorize spending.
Signed shipping/payout evidence invalid	Preserve original records and restore the correct site/signing context; do not fabricate a replacement proof.

Problem	Appropriate recovery
Payout busy, unknown or not confirmed	Reconcile the same origin/reference and original batch before any new payment decision.
Post-commit listener/email failure	Inspect the committed issue transaction and recover notification for that same credit.
Privacy operation incomplete	Verify the WordPress request, permission and database, then retry without assuming complete erasure/export.

Provide support with the account/order/transaction references and the visible error. The technical error lookup preserves exact implementation messages and returned privacy codes, while the relevant feature chapter explains its state and recovery boundary.

Using gift cards and store credit at checkout

Gift cards and store credit reduce eligible WooCommerce purchases through the coupon system. Apply the instrument you want to use, review the updated totals, and complete the normal payment step for any remaining amount. Applying a code to a cart does not immediately consume its balance: the store holds the required value when it creates the order, then records use when the order reaches processing or completed.

Both classic checkout and WooCommerce Cart/Checkout Blocks use this flow. Credit cannot buy products that the store has marked as credit-sale products. It does not cover ordinary fees. Shipping is covered only when the individual card or credit was issued with that permission. Existing balances remain subject to their ownership, currency, expiry and product restrictions even if new-credit creation is unavailable under the store's license.

The store's [customer credit pages](#) explain where to find saved instruments and how emailed gift cards are claimed. [Merchant controls](#) cover issuing, reviewing and restoring balances.

Apply a code and understand its value

Enter the full code in WooCommerce's coupon field, or choose an account-assigned instrument from the saved-credit panel when available. Review the cart after applying it. The store allocates credit after earlier native coupons, across the remaining eligible merchandise in WooCommerce's item order. Removing the coupon from a cart removes that proposed discount; the balance is not spent merely by viewing it.

Requirement	What to do
Correct owner	Sign in to the account assigned to the instrument. An administrator account cannot spend another customer's assigned credit.
Emailed recipient	Enter the exact recipient billing email at checkout. A guest preview may show the discount before the email is entered, but final submission still checks it.
Correct currency	Use the instrument in its own currency. It is not converted when the cart currency changes.
Available value	The instrument must be active, unexpired and have usable value. Another pending order may already hold part of its balance.
Permitted products	Check its product, category, sale-item and other coupon restrictions. Credit-sale products are excluded.

Requirement	What to do
Multiple codes	The merchant's multiple-credit setting decides whether more than one credit instrument can be used in the same order.

The amount consumed depends on the instrument's saved value policy. A **tax-exclusive discount budget** pays the eligible net merchandise reduction; any resulting tax reduction is additional savings. A **tax-inclusive discount budget** pays both that net reduction and its associated tax reduction. Shipping permission is a separate setting on the instrument. These policies are set when the store issues or configures the credit, not selected by the shopper at checkout.

For example, a 10.00 tax-exclusive credit applied to 19.99 of eligible merchandise leaves 9.99 of merchandise. Tax, shipping and fees are still handled according to the cart and the instrument's policy. For merchandise priced at 10.00 before 20% tax, a 6.00 tax-inclusive budget can reduce net merchandise by 5.00 and tax by 1.00. A 6.00 tax-exclusive budget instead pays a 6.00 net reduction, with a further 1.20 tax reduction. Exact results depend on the store's rates and WooCommerce's line rounding.

Tax-inclusive or shipping-enabled credit requires normal supported WooCommerce tax calculations and per-line rounding. The store cannot use it with compound/invalid rates, subtotal tax rounding, or certain custom totals integrations. If checkout identifies one of these conditions, use an eligible instrument or have the merchant resolve the configuration. Repeatedly pressing Apply does not bypass the restriction.

Saved gift cards and credit in Cart/Checkout Blocks

When you are signed in, the normal WooCommerce Cart or Checkout Block can show a saved-credit panel immediately above the block. It appears outside the order-summary coupon area. Depending on your instruments, the title is **Your Gift Cards** or **Your Store Credit & Gift Cards**. Gift cards appear before store credit, with group balances, each available amount, and a masked reference.

1. Find the instrument by its type, masked reference and available amount.
2. Select **Apply gift card** or **Apply store credit**.
3. Wait while the selected row says **Updating...**. The server recalculates the cart, then the row shows its applied status and **Remove**.
4. Review the updated totals. To undo the cart selection, choose **Remove** and wait for the recalculation.

The panel does not automatically apply saved funds, reveal full redeemable codes or create a credit. Its buttons send only the selected instrument ID and action; the server checks that the current account owns it. Unsaved address fields are not deliberately overwritten by the update request. During an update the panel disables its buttons, and a second panel cannot start a competing request. Status and error messages are announced for assistive technology, and button labels identify the type and masked reference.

Loading, empty lists and fallback behavior

An empty saved-credit list means the current account has no instruments returned as available for this panel. It is different from a loading error. Check the account's ownership assignment, currency, expiry and available value; an email-only gift card may still need its emailed code or the supported claim process. Entering a matching billing email does not make someone else's account-assigned credit visible.

A missing response first displays loading, then after eight seconds suggests refreshing or entering the code. If the script never starts, the static fallback remains. A lost login asks you to sign in. With JavaScript disabled, use the normal supported code-entry flow and enable JavaScript if you need the saved-credit buttons. Refreshing or signing in does not issue replacement value.

The main panel is for the usual parent WooCommerce Cart/Checkout blocks. It is not added to order-pay, order-received, admin or arbitrary third-party checkout layouts. Some WooCommerce layouts can provide a secondary discount-area panel when the main host is absent. Merchants do not need to edit the saved Cart or Checkout page merely to add this plugin's normal panel. If an optimization layer serves old assets, refresh its relevant cache without clearing balances or order data.

A failed Apply message can indicate ownership, restrictions, currency, unavailable value or an invalid action. Review the message and cart, then reapply only after the cause has been resolved. The button cannot override normal coupon rules.

When an order holds and uses credit

At final submission the store rechecks the instrument, recipient, currency, tax/shipping policy and available balance. This matters when a balance changes after the cart was first displayed, including another order using the same funds. If the amount is no longer available, recalculate the cart and review the new total before submitting again.

Once the order has been created successfully, its required amount is reserved. That portion is unavailable to another order. The same order can use its own reservation during a supported retry; it does not need to reserve a second copy. Pending, on-hold and failed payments retain the reservation. A failed payment status alone does not return the value to the general available balance.

When payment/order handling moves the order to processing or completed, the held amount is recorded as used. Cancellation returns a hold, or restores already-used credit through the order lifecycle. A discarded checkout order can release its hold through WooCommerce's exception handling. If a release or validation fails, the merchant must review the order's notes and ledger rather than issue a replacement simply because a checkout message appeared.

An edited or previously restored order can need a reviewed removal and reapplication of credit. Returning to an old order-pay link does not create missing financial evidence. If the message says the order's payment evidence needs review, stop retrying that link and contact the merchant with the order number.

Coordinated quote or companion checkout

A compatible local checkout owner can include Store Credit in one coordinated database transaction. The owner controls the order or quote conversion; Store Credit controls its balances. The customer still uses the normal permitted credit and payment flow. There is no customer switch to enable this developer integration.

Before committing the checkout, the owner must prove that the exact order, customer, currency, instruments and held amounts still match. If that proof fails, the owner must finish its rollback or resolve an uncertain outcome before allowing payment. A companion cannot safely bypass the hold or create a replacement reservation after it has already committed its own conversion.

What completion and rollback mean

Successful completion makes the coordinated reservation durable, then releases its queued reservation notifications. A rolled-back attempt discards those queued messages and removes the attempt's in-memory credit markers; it does not issue a compensating refund for value that never committed. The integration protects the original database connection and exact order object, so a changed connection or mismatched order stops the attempt.

If a process ends after the database commit but before a notification is delivered, the notification can be missing even though the financial record exists. Merchants should use the order and ledger to determine what happened, not the arrival of a message. The companion remains responsible for its quote, stock and other records.

Reading a checkout that is still open

A compatible developer integration can read the held allocation while the coordinating transaction is still open. This is only an observation, not permission to make another charge or spend another balance. It is deliberately unavailable if the connection, order or financial proof cannot be pinned reliably. That read does not itself cancel or alter the attempt.

The exact local APIs and owner obligations are in the technical transaction contract.

Acknowledging an already settled order

A compatible companion can ask whether an order's original shared credit reservation was fully redeemed. This helps it acknowledge an existing result without reserving or spending the funds again. It is not a gateway payment receipt and does not authorize a second payment, refund or reward.

This read requires the order to be processing or completed, its original shared binding to match, and its complete original reserve/redeem history to remain verifiable. It holds when the checkout is still open, the provider is missing, the order changed, the history includes restoration or release, or required evidence is missing. Older settled orders without the bound original scope are not silently upgraded from their current editable metadata.

The result is valid as an observation at the time it is read. A companion must read again for a later acknowledgement. Customers and merchants do not edit private metadata to make it pass; resolve the order or integration issue instead. See the technical settled-reader contract.

Per-item allocation evidence for integrations

Store Credit can report how an observed checkout allocated credit across product lines, including associated tax. A compatible local integration may use that information to understand the original checkout. It does not replace the ledger or authorize more spending, and it is not a separate merchant report or checkout control.

Cart calculation limits

The cart reader needs a complete unchanged calculation in the current request. It supports at most 500 product rows and 20 credit instruments, with normal supported WooCommerce pricing and tax calculations. It records the actual before/after line amounts, instrument policies and restrictions while using hashes instead of exposing full codes, email addresses or tax-location addresses.

This optional reader supports only carts whose applied coupons are this plugin's credit instruments and whose credit does not cover shipping. A cart mixing other coupons, using shipping-enabled credit, custom line-price adjustments or unsupported tax rounding can still have its normal supported credit workflow, but cannot obtain this stricter allocation certificate. An unavailable certificate is not evidence that the customer's balance was lost.

Saved-order certificates

When the original checkout meets those conditions, the plugin can sign a certificate linking its cart rows to saved order items and original reservation records. A later read requires that certificate and the same original financial scope. It can distinguish a zero-credit calculation, a reserved amount, an uncommitted shared reservation and a settled amount.

Editing protected order facts, changing the instrument policy, restoring funds, or losing the signing salts can make the original certificate unavailable. The plugin does not reconstruct one from an arbitrary edited order. Preserve the store's database and WordPress authentication salts together when making or restoring backups.

Persistence restrictions

Saved allocation reads require WooCommerce's native legacy order storage or HPOS. Custom order data stores, oversized or malformed evidence, conflicting metadata and uncertain ledger rows are held. Any refund child, including a partial or trashed refund, makes the original saved allocation unavailable to this reader. This does not itself reverse the order or erase its ledger history.

The technical reference explains the exact allocation fields and limits. A merchant should not change internal certificate fields, transaction rows or signing data to force a companion's acknowledgement to succeed.

Shipping-enabled credit

When an instrument permits **Use remaining credit for eligible shipping**, the store first applies it to eligible merchandise, then uses the remaining budget on eligible chosen shipping rates. It reduces the shipping price itself; there is no extra negative fee. The shipping method label adds **store credit applied**.

A package qualifies only when it contains at least one paid physical product and all its paid physical products meet the instrument's product restrictions. A free or virtual product alone does not qualify a package. Free shipping needs no credit. Only the chosen rate is discounted, and one remaining budget is shared across packages. When several instruments can cover shipping, the plugin uses its stable account-ID order. At most 20 shipping packages are supported.

Tax-exclusive credit consumes the net shipping reduction. Tax-inclusive credit consumes the net reduction plus its associated tax reduction. For example, 6.00 of remaining tax-inclusive credit against shipping priced at 10.00 plus 20% tax can reduce shipping by 5.00 and tax by 1.00, leaving 5.00 plus 1.00 tax. The actual calculation follows WooCommerce's rates and rounding.

A custom shipping method must use a tax calculation the plugin can reproduce. If checkout reports an unsupported shipping tax calculation, remove the shipping-enabled instrument or ask the merchant to choose a supported configuration or method. Recurring subscription shipping is excluded.

Shipping proof and order changes

The plugin privately records the original and discounted shipping amounts, method, taxes, order and instruments. This keeps a reduced rate from being copied to another order as though it were a fresh discount. It does not expose redeemable codes or private proofs in public shipping-rate data.

When a credited coupon is removed during order recalculation, an affected shipping line returns to its original quoted amount and tax. Editing a shipping line while its credit remains applied is refused if the saved proof no longer matches; remove and reapply the credit through supported checkout. Ordinary shipping lines without Store Credit evidence are not rewritten by this process.

Order totals may show **Store credit included in shipping:** followed by **already deducted; not an extra discount.** That amount is explanatory: it is already reflected in shipping and must not be subtracted again.

Protected advanced orders

Tax-inclusive and shipping-enabled instruments save a private proof of their allowed merchandise, tax and shipping amounts. Before payment, the store compares that proof with the customer, currency, product lines, coupons and shipping details. This prevents a later edited order from spending a different amount under an old proof.

If an advanced order has been edited, remove credit before changing its protected amounts, recalculate, and reapply through supported checkout. Do not copy a discounted shipping line or private credit fields into another order. If the order has already been paid or refunded, use the merchant's lifecycle and restoration controls instead of attempting to rebuild its original proof.

The order may be held when its totals no longer match, even if the visible coupon code still looks correct. Give the merchant the order number and displayed message. A support review should use the order notes, live credit policy and ledger; it should not overwrite signatures or issue duplicate funds.

Cancellation, refunds and reopening orders

Situation	Credit behavior
Pending, on-hold or failed payment	The order retains its reserved value. Contact the merchant if the order should be cancelled.
Processing or completed order	The reserved credit is recorded as used.
Cancelled order	Held value is released; value already used is restored through the credit ledger. Review notes identify failures needing attention.
Full WooCommerce refund	The plugin automatically restores the remaining consumed credit for that order.
Order marked Refunded with no cash left to refund (an order paid entirely with credit)	WooCommerce creates no refund in this case. The plugin restores the consumed credit when the status changes to Refunded, exactly once (from 1.0.2).
Partial WooCommerce refund	The merchant chooses the credit amount to restore. The plugin adds links and a note; it does not assume a proportional split.

Situation	Credit behavior
Refund as store credit (from 1.0.6)	The refunded amount becomes a new store credit for the customer; nothing goes back through the gateway. Credit the order used follows the full and partial refund rules above. See Refund an order as store credit .
Cancelled/refunded order reopened	Old restored-credit coupon lines are removed and remaining coupons are recalculated. Credit is not automatically spent again.

A credit or gift card code that pays nothing in the cart (other discounts already cover everything, or none of its products are in the cart) is taken off the cart with a notice; its balance does not change (from 1.0.2).

For a partial refund, use the [refund-linked credit controls](#). The available amount is limited by both the instrument's remaining consumed value and the refund-linked restoration allowance. A cash refund and a credit restoration are different actions; confirm the intended combination before applying them. Gift-card purchase refunds have their own [purchased-value protection](#), separate from restoring credit used to pay for an order.

A failed automatic action adds a review note. Resolve that note and the financial history before treating the refund or cancellation as fully reconciled. Reopening the order removes old credit discounts but does not prove that an earlier restoration succeeded. The plugin retains a bounded record of removed credit lines and keeps the immutable financial history.

For an unpaid editable order, a valid recalculation can adjust or clear its existing hold to match its credit items. Paid orders and invalid advanced proofs cannot be treated as ordinary cart edits. If the order is no longer payable under its original credit evidence, the merchant must review and reapply it through the supported process.

Paying a manual subscription renewal

This plugin does not automatically debit store credit for scheduled recurring payments. It does not copy a prior order's private credit evidence or discounted shipping into a subscription template or renewal order.

A supported manual renewal can use credit when you sign in as the renewal's customer and pay your own saved pending, on-hold or failed renewal. All related subscriptions must belong to you and use manual payment, and the optional subscription provider must expose the required renewal APIs. Normal currency, ownership, expiry, product and balance checks still apply.

Initial subscription, switching and recurring-template carts are excluded. A renewal created with copied credit coupon or shipping evidence is rejected rather than spending that old evidence again. Contact the merchant to correct that renewal; copying or editing hidden fields is not a payment method.

B2B status and consumer referral rewards

Store Credit's B2B integration distinguishes consumer referral eligibility from spending an existing legitimate balance. B2B status can prevent a new referral invitation, discount or reward obligation while leaving previously issued credit available under its normal redemption rules.

The store remembers when B2B has been present. If that previously expected owner disappears, new consumer referral work does not automatically become eligible. A missing or uncertain decision holds the affected operation until the owner or integration is available again.

Referrer, buyer and historical evidence

The referrer and buyer are checked separately. A current referrer decision must identify that exact registered customer as a final eligible consumer. When the referral obligation is frozen, its accepted referrer fact is retained within the signed claim, so a delayed fulfilment uses the required historical proof rather than pretending the buyer's status proves the referrer's status.

Once B2B is present or expected, the buyer's order also needs its committed historical audience record. Today's customer status is not a substitute for a missing order record. Missing or malformed proof produces an unknown/review state; it does not silently award a new promotion. The [referral chapters](#) describe those qualification and reward workflows.

Retirement and removing an ineligible referral

An accepted B2B retirement proof confirms a narrowly defined owner-policy retirement. It does not grant positive consumer eligibility, erase the store's remembered owner expectation or restore promotion permission by itself. Existing balances are not revoked or relabelled by this check.

When B2B supplies a valid ineligible cleanup context, Store Credit can remove only its own matching signed referral discount and clear its referral session context. It does not remove unrelated coupons or alter account balances. If the discount cannot be identified safely or removal cannot be confirmed, the cart cleanup is held for review.

Checkout problems and recovery

Message or situation	Recovery
Sign-in or ownership error	Use the assigned account, or enter the emailed code with its exact recipient billing email where applicable.
Currency changed	Return to the instrument's currency or remove it. The plugin does not convert balances.
Balance changed before checkout	Recalculate the cart, review any other pending order holding funds, and submit only after the new totals are clear.
Only one credit code allowed	Remove additional instruments or ask the merchant about the store setting.
Unsupported tax/shipping calculation	Use a supported policy/configuration or remove that instrument. Custom totals cannot be forced through by retrying.
Shipping/order evidence changed	Remove and reapply credit through supported checkout; ask the merchant to review an existing paid/refunded order.
Shared payment evidence needs review	Give the merchant the order number. Do not keep retrying a stale payment link or create duplicate replacement credit.
Saved-credit panel remains loading	Refresh, check sign-in and current assets, or use the supported full-code coupon field. Missing data is not the same as a zero balance.
Full refund/cancellation review note	Merchant checks ledger and order history before concluding the value was returned.

Message or situation	Recovery
Partial refund	Merchant explicitly chooses the intended credit restoration using the refund-linked controls.
Companion allocation/settlement unavailable	Resolve the identified order, original-proof or integration condition. These read contracts do not repair evidence or authorize a financial retry.
B2B referral decision unavailable	Restore the required owner/current or committed evidence; existing earned balances remain separate.

Under **Tools → Site Health → Status**, the Store Credit storage check reports transactional-engine and required-index availability. It is read-only: it does not repair balances or demonstrate that a disputed checkout succeeded. Developers can also detect other installed credit coupon types; this does not import or adopt another extension's balances. Technical support can use the exact error lookup without exposing full codes, recipient details or private proofs.

Set up Refer a Friend

Refer a Friend lets a customer invite a new customer and earn store credit when the friend's purchase qualifies. You choose whether the friend receives credit after that purchase or a discount on it. This is a separate customer program: participants do not need an affiliate application, business onboarding, tax profile, or professional payout minimum.

Open **Store Credit & Gift Cards → Refer a Friend**, or the Store Credit product workspace in AffStackDev Suite. Managing the program requires both Store Credit and WooCommerce management permissions. The page contains one program with a permanent identity. You can revise its settings, but revisions do not reset its customer eligibility or lifetime claim counts.

Referrals start disabled. Open the administration workspace so its referral storage can initialize. If the page reports **Referral storage is unavailable**, resolve the storage problem before enabling the program. Store Credit can process standalone referrals. If AffiliPartnerCS is active, its compatible customer-referral adapter must be available; otherwise new standalone invitations pause to avoid competing processors. On a previously unconfigured program, a compatible AffiliPartnerCS adapter is selected in the form automatically; save your chosen settings to publish them.

To configure the program:

1. In **Program and rewards**, enter the program name and select the **Referral processor**.
2. Choose **Store credit after qualifying purchase** or **Discount on first qualifying purchase** under **Friend receives**.
3. Set both positive reward amounts, the eligible merchandise minimum, time periods and lifetime limits.
4. Choose eligible products/categories and the permitted combinations of offers.
5. Configure the independent spending restrictions for each earned-credit side and add customer terms.
6. Select **Enable customer referrals**, then **Save program**. Read the saved confirmation.

Searching catalogs does not save the program. If another administrator changed it while your form was open, reload before saving. Previously accepted invitations retain their accepted rules; existing order obligations retain their original reward terms and processor. New settings govern later acceptances. Pausing the program blocks new qualifying capture while preserving already accepted purchase obligations and earned balances.

Every program setting

Control	Default and choices
Enable customer referrals	Off. Enables new referral capture when all dependencies and permissions are available.
Program name	Refer a Friend ; up to 120 bytes of plain text. An empty name returns to this default.
Referral processor	Store Credit — standalone customers or AffiliPartnerCS — customer referrals (not affiliates) .
Friend receives	Credit after qualifying purchase, or discount on first qualifying purchase.
Friend amount	1000 minor units by default: 10.00 in a two-decimal currency. Must be positive.
Referring customer credit	Independently set positive amount; also defaults to 1000 minor units.
Minimum eligible merchandise spend	2500 minor units by default: 25.00 in a two-decimal currency. May be zero, but the order must still have a positive eligible merchandise amount.
Days after payment before credit release	14 days; allowed 0–365. Cash on delivery also waits for Completed status.
Invitation window	30 days; allowed 1–365. The shopping session must remain available throughout.
Friend credit expiry days	0 means none; allowed 0–3650. Unused when the friend receives an immediate discount.
Referrer credit expiry days	0 means none; allowed 0–3650. Independent of the friend's expiry.
Lifetime referral limit per referring customer	25; allowed 1–10,000 accepted claims.
Lifetime total program limit	1000; allowed 1–1,000,000 accepted claims.
Additional customer terms	Up to 4000 bytes of plain text; line breaks are retained.

Amounts use the store currency and its decimal precision. Enter a nonnegative decimal number without currency signs, grouping commas or scientific notation. Each amount supports at most 1,000,000,000 minor units; both rewards must be greater than zero. There is no percentage reward, zero-sided reward, per-tier reward, separate currency selector or automatic currency conversion in this form. With an immediate friend discount, the referrer still earns later credit, but the friend does not receive a second credit.

Expiry is measured from the frozen reward release time, based on the original paid date and hold period. Processing delays do not restart the expiry period. If a promised reward expires before it can first be issued, it requires merchant review.

Lifetime limits count accepted checkout claims, including unpaid, cancelled and reviewed claims. They do not count only successful rewards. Cancelling a referral, editing the program or choosing another processor does not restore the friend's one-time eligibility or reset these limits.

Qualifying products and independent credit restrictions

Expand **Eligible products and categories** to set **Only these products**, **Exclude these products**, **Only these categories**, and **Exclude these categories**. An empty inclusion list leaves otherwise eligible merchandise available. If both inclusion lists are populated, an item must match both. Exclusions take precedence. Each list supports up to 100 selected entries.

A parent product selection also matches its variations. An exact variation selection matches that variation. Category selections use assigned category IDs and do not automatically include child categories; select the required child categories separately. Gift-credit/credit-sale products and subscription products are excluded from referral qualification. Shipping, tax and fees are outside the eligible merchandise total; a zero-value reward line contributes nothing.

The two credit-spending sections contain independent **Only these reward products/categories** and exclusion lists. These determine where the issued friend's or referrer's credit can later be used. They do not inherit the qualifying purchase's product list automatically. Set both sides deliberately. The friend-credit section has no spending effect when the friend receives an immediate first-purchase discount. That discount is allocated only to the qualifying merchandise.

The minimum is measured **after other discounts and before this program's friend discount**. For example, 40.00 of eligible merchandise reduced by another 5.00 offer gives 35.00 toward the minimum. A 10.00 friend discount then leaves 25.00 of eligible merchandise payment. At least one minor unit must remain after the friend discount. A completely discounted eligible merchandise purchase does not qualify, even when you enter a zero minimum.

Store credit and gift cards are payment, not discounts. Merchandise paid with them still counts toward the minimum, and paying the rest of a qualifying order entirely with credit satisfies the positive-payment requirement. For example, 40.00 of eligible merchandise with a 10.00 friend discount and 30.00 paid with a gift card counts 40.00 toward the minimum with 30.00 of payment. New-customer, self-referral and lifetime limits are unchanged.

Choose which offers can combine

The form has three separate sets of rules: **Qualifying purchase: combines with**, **Spending the friend's earned credit: combines with**, and **Spending the referrer's credit: combines with**. Each has the following six categories, initially set to **Allow none**.

Category	What you can select
WooCommerce coupons	Coupon types or individual published coupons. Choosing a type applies to each offer of that type. Generated friend-discount coupons are omitted from this picker.
Catalog sale prices	Products or categories for native Woo sale-price offers. The picker can include products that are not currently on sale.
Existing store-credit sources	Manual, compensation, refund alternative, loyalty, promotion, giveaway, other, API import, purchased, gifted, partner payout and customer referral credit. This setting no longer has an effect: store credit and gift cards are payment and never block a combination; each credit's own spending rules still apply.

Category	What you can select
Free Gifts & BOGO	Offers identified by its compatible adapter and catalog.
AffiliPartnerCS offers / attribution	Offers or attribution identified by its compatible adapter and catalog.
Other registered promotion adapters	Offers explicitly supplied by a compatible integration. This does not automatically include every third-party pricing plugin.

For each category choose **Allow all supported**, **Allow none**, **Only selected**, or **All except selected**, with up to 100 selections. Every active offer must satisfy its own category. Allowing one coupon does not approve a second coupon or an unrelated promotion. Unknown, missing or malformed adapter evidence does not become permitted through **Allow all supported**. A detected negative fee is also an unidentified offer and prevents referral compatibility.

These permissions do not override another coupon's individual-use restriction or another plugin's own promotion rules. An immediate friend discount is a Woo coupon internally, so another promotion that prohibits coupons can still refuse its offer. Referral settings determine whether the resulting purchase can qualify; they do not force other products to grant discounts or gifts.

Use **Find products, coupons and promotions → Search catalogs** to update the lists. Search retains selected entries and unsaved form values, but you must select **Save program** to publish them. Search results are bounded: refine the name if an item is absent. Product lists return up to 35 products, categories up to 35 terms, coupon search up to 40 matching posts, and each integration catalog up to 60 results. Search text is limited to 80 bytes. Saved selections remain visible even when their current label is unavailable, shown as **Saved selection: ID**.

With Enhanced UI enabled, each list also has **Search this catalog → Find**. It preserves selected items while refreshing results. If this search fails, use the ordinary **Search catalogs** form or retry. The complete settings form works with Enhanced UI disabled. If a selected adapter disappears, its saved restrictions remain; restore the adapter before assuming that surviving coupons or attribution can qualify.

Invite a friend and use an accepted invitation

A signed-in eligible customer opens **My Account → Refer a Friend**, reads the current terms and selects **Create my share link**. Creating the link is an explicit action; viewing the account page does not create it. The link contains a random token, not the customer's email or account number. **Copy link** copies it, with manual selection available if the browser cannot copy. **Share** appears on devices supporting browser sharing.

The `[afsd_refer_a_friend]` shortcode can place the same account panel on another page. Anonymous visitors receive a sign-in link. The account menu remains available when the program is enabled or the customer has referral history. When new referrals are paused or the account is ineligible, the panel explains that status and keeps existing reward history and Store Credit access available. A revoked sharing link has no customer reactivation button.

The friend opens the link, reads the reward/minimum/timing/compatibility summary and additional terms, and selects **Accept invitation and shop**. Merely opening the link does not assign a referral or issue value. An unavailable link instead offers normal shopping. The friend may buy as a guest, but must supply a valid billing email at checkout. A known matching account receives account-bound earned credit; otherwise the friend reward is bound to the original email.

An accepted invitation belongs to that Woo shopping session and expires after the configured window. It is not cross-device tracking. If the session disappears, reopen the link and accept it again. Reaccepting an identical, still-current invitation reuses it; switching or renewing invitations too quickly can require waiting one minute. A shopper can have one accepted invitation in this session; accepting another replaces the earlier context. An invitation is used once: after its qualifying order is placed or paid, it clears itself, so the friend's later ordinary orders check out normally without **Remove invitation**. If a payment attempt for that same order fails, the invitation stays for the retry.

In immediate-discount mode the program creates and applies a session-bound coupon as eligible items are added. The visible label is **Refer a Friend**. Copying its internal code does not make it a general bearer discount. In credit mode there is no immediate discount or automatic use of existing Store Credit. Both modes display a cart/checkout panel with the promised friend benefit, minimum and **Remove invitation**.

Removing the invitation removes its current applied friend discount. Removing that discount through the coupon control also removes the invitation. Cart and Checkout Blocks update the panel from the server's accepted-invitation status; this status does not promise that the eventual order will qualify. The customer can remove an incompatible invitation and continue ordinary shopping.

Who qualifies and what becomes fixed

The friend must be a new purchasing customer. Checks use both billing email and an identified account, including guest-order history. Prior processing, completed or refunded purchases, and orders with a recorded paid date even if their status later changed, count as previous purchases. A customer cannot refer their own account or use the referring account's email.

Both buyer and referrer must be eligible consumer participants when the invitation/order is accepted. If B2B policy says the customer is ineligible, or a previously expected policy service is unavailable, new referral qualification is held. Removing the B2B plugin or retiring a policy does not create permission. Once the order is accepted, reward settlement relies on its retained historical audience decision and the referrer's frozen eligibility evidence rather than replacing them with today's account classification.

Checkout also checks currency, eligible products, minimum, positive merchandise remainder, permitted offer combinations, first purchase and lifetime limits. It reserves one original order and friend identity for the program. A failed/unpaid or later cancelled claim still retains that identity. A simultaneous second checkout cannot claim another reward for the same program/customer.

The accepted order retains the reward amounts, recipient identities, processor, restrictions and financial fingerprint. Editing its customer, billing identity, currency, merchandise, coupons, shipping, fees or financial totals can require review. Ordinary payment/status notifications do not change its reward identity. A pay-for-order attempt can validate an existing original referral but cannot attach a new invitation to that old order. Use the original checkout to qualify the purchase.

Release, spend and review referral rewards

Credit rewards release only after the original order is processing/completed, has a paid date, and reaches its hold time. Cash on delivery requires Completed. Pending, on-hold and failed purchases do not release rewards. Background processing checks the original order and evidence again before issuance.

In credit mode the referrer's and friend's credits are separate instruments. Their issue references remain fixed. If one side was committed before processing stopped, a retry finds that original credit and handles only the

remaining work. The pair cannot be spent until the claim reaches **Rewarded**. A delivery email for the first side can therefore arrive while the overall claim is still pending. Email delivery uncertainty does not justify issuing replacement value.

Issued referral credits use the usual Store Credit account/email access, with the original side's product and offer restrictions. They apply to eligible merchandise under tax-exclusive credit rules and do not cover shipping. Later program edits do not loosen existing credits. An expired, changed, incomplete, cancelled or reviewed referral reward can be refused even when its coupon code still exists.

Customers see **Your referral rewards**, their own promised amount/currency and status. Waiting, pending and issuing all appear as **Pending**. Immediate friend benefits are labelled **First-purchase discount**. Referrers see no friend's email or order contents. The history is paginated in groups of 20 and links to **View and use your store credit**. Guest-bound history does not automatically become account-bound when a new account is created later.

Merchant state	What it means
pending	The checkout obligation exists, but payment or qualification processing remains outstanding.
waiting	The paid purchase is within its hold period.
issuing	Original reward issuance or its receipt confirmation is incomplete.
rewarded	Required reward sides are complete; normal credit and frozen-rule checks still apply when spending.
revoking	Cancellation is stopping spending and reconciling original credits.
cancelled	The referral has been cancelled; eligibility and financial history remain.
review	Changed evidence, repeated processing failures or used/reserved/uncertain rewards need merchant attention.

Open **Referral activity and reward review** to see each referral/order link, referring customer, frozen rewards, provider, state and reason. The Woo order also displays **Customer Refer a Friend** with a link to **Review referral rewards**. The table has **Retry / reconcile** and **Cancel referral** with **Confirm cancel/revoke**.

Use **Retry / reconcile** after restoring the original processor, storage or unchanged order evidence. It follows the same original reward references. A reviewed issuance can resume only for a repaired processing failure or a restored original financial fingerprint, with the original paid, unrefunded order and processor available. Other review reasons cannot be overridden by repeatedly pressing Retry. Background exceptions retry with increasing delays; after five failures the record requires review. A missing processor pauses pending processing instead of silently using another processor.

Any cancellation or refund of the qualifying purchase, including a partial refund, stops both rewards. Pending rewards are cancelled. Issued rewards can be revoked only when unused and unreserved. Used, reserved, missing or uncertain rewards enter review; the program does not create a negative balance or charge unrelated credit. **Cancel referral** affects referral rewards only and does not issue a cash refund. The cancelled claim continues to count toward lifetime limits.

If a cancelled/refunded record still needs reconciliation, **Retry / reconcile** retries the cancellation path. It does not resurrect qualification. There is no proportional referral reward adjustment for a partial refund and no

automatic replacement claim. Preserve the original record so the merchant can resolve its actual balance and receipts.

Referral processor availability

The **Referral processor** selects the preferred processor for new invitations. Store Credit's standalone mode needs no AffiliPartnerCS installation. When AffiliPartnerCS is present, use its compatible customer-referral adapter; the referral page reports a missing, incompatible, unlicensed or expired adapter rather than assuming that an installed plugin can process rewards. Store Credit's own new-value permission must also be available.

For ordinary, unmigrated claims, the processor is fixed when the purchase is accepted. If you choose another processor later, earlier pending claims retain the original processor and reward references. Restore a missing original processor to resume those claims. Previously accepted standalone claims do not become affiliate claims merely because AffiliPartnerCS is subsequently installed.

Shared programs use the supported controller resolved for the retained definition. The following section explains the explicit migration and original-claim requirements. These are separate from ordinary affiliate commission payouts or conversion of an affiliate payout into store credit.

Shared processing and integration-only controls

The **Referral processing** card explains the current processor and whether new capture is ready or held. A displayed **Shared reward controller** is informational: availability and permission are checked again for each reward. A processor's presence, stored preference or displayed owner is not permission to issue credit.

When the required shared runtime, original database writer, licensing, currency/precision and compatible adapters are available, the page offers **Use shared settings for this program** and **Save shared program**. This is the program's explicit opt-in shared integration. Select the checkbox and button to migrate its configuration. If activation was interrupted, repeat the offered original operation to complete it. Until activation is confirmed, capture remains held; deleting its state is not a supported recovery step.

Shared settings preserve the permanent program and complete behavior: reward amounts, customer/product/offer rules, time periods, limits, currencies and original reward references. The current supported controller can be Store Credit or AffiliPartnerCS according to Core selection and the saved preference. Store Credit continues to own the journal and balances. An accepted invitation may keep earlier terms while the current supported controller changes.

Migration does **not** automatically transfer historical referrals. An authorized integration can explicitly adopt an eligible original claim only after verifying its exact signed snapshot, original audience/processor qualification, order, existing side receipts and native financial protection. There is no historical-adoption control in this screen. Adoption preserves recipients, amounts, original provider identity and all issue/reversal references. It cannot manufacture missing old qualification from a customer's current status.

Pausing **Enable customer referrals** stops new capture while retaining accepted purchase obligations. A separate Core integration can hold all shared processing or archive a definition without cancelling those obligations. It can also manage supported executor retirement/restore. Those controls are not additional checkboxes in the referral form. Retiring a processor does not move unadopted native claims or grant consumer eligibility; a shared successor must satisfy the complete original contract.

Shared execution requires the store's native supported MySQL writer and an original claim lock so a stopped connection cannot later create an unnoticed duplicate. If the page reports financial-fence or recovery-service unavailability, restore those services before retrying. A changed store currency or saved precision can hold execution. Retained configuration and receipts remain available for supported review; they are not automatically converted to a new monetary basis.

Recover an interrupted shared reward

Use the same **Retry / reconcile** action for a shared reward. The system first checks whether the original side already committed. If its outcome is uncertain, it keeps the original reward on hold. Only an exact no-effect result, with the original writer excluded and the required Core recovery support available, can rearm that same operation. Its definition must still match, and the purchase must still be paid, unrefunded and due.

Messages distinguish a missing recovery service, unavailable original financial records, an ineligible accepted purchase, changed program definition, and an outcome still awaiting reconciliation. Keep the original claim and reward records while resolving the stated cause. Changing the processor, creating another claim or issuing a separate manual reward would not reconcile the uncertain original operation.

Integrations also have readonly program, original-claim and receipt interfaces. Ordinary continuation reads include the original customer references and report observed issue/reversal records; they do not authorize another effect. Missing receipts and unknown email delivery are not promises that no value was issued. The stronger shared receipt reader validates original financial and Core evidence. See the technical referral contracts for these local PHP interfaces; they are not public customer endpoints.

Referral history, removal and privacy

Store Credit retains referral links, accepted claims, signed reward terms and event history separately from its balance ledger. Normal plugin removal preserves these records and balances. Once referral storage exists, the legacy destructive uninstall switch is also blocked from dropping the related financial data. Deactivation is neither transfer nor deletion, and there is no referral-screen **Start over** or lifetime-eligibility reset.

Signed referral history depends on the site's original authentication salts. Restoring records together with those salts preserves their verification. A changed or unverifiable snapshot is held for review instead of being silently rewritten. Removing an invitation from a cart does not delete its generated coupon post; the coupon remains unusable without the matching valid invitation session.

WordPress privacy export supplies the requesting customer's own referral role, state, reward amount, currency and creation date. It does not reveal a friend's email or order contents. Privacy erasure revokes sharing for a matching local account while retaining the link row, eligibility records and financial evidence. Its result reports that retention rather than claiming full deletion. Sharing revocation does not cancel already accepted purchase rewards.

This module records explicit Woo-session invitation acceptance and random sharing tokens. It does not collect a referral IP fingerprint or an automatic invitation-email list. Store Credit provides suggested privacy-policy text describing the retained referral records and their visibility.

Licensing, setup, and existing customer value

Store Credit & Gift Cards requires permission for new credit creation, new gift-card sales, positive balance adjustments, new payout credits, and new referral activity. Existing customer value remains serviceable:

licensing does not remove balances or block the retained redemption, release, restoration, reconciliation, read, export, privacy, and license-recovery paths. Those operations still apply their normal ownership, checkout, account, and financial checks.

A new installation has an initial 72-hour activation window, measured from its stored licensing setup time. It applies while there is no readable active or pending installation credential. Activate before that window ends. The window does not restart when you check the license or deactivate an activation; any remaining portion still uses the original setup time. Once a credential is active or pending, follow its validation/recovery state instead.

The license can contain restrictions on new work. Read-only or hard-lock policies prevent new licensed writes. A lock policy does not erase existing balances or remove the retained recovery tools. A plan may also allow only specific groups of new operations. Restoring the license does not change balances or complete a financial operation by itself; return to the relevant workflow and review its result.

Suite is optional unless your plan explicitly requires it

Store Credit works independently of the optional AffStackDev Suite interface. AffStackDev's current plans do not require a Suite license, and the Suite needs no key of its own. Merely installing or removing Suite does not turn an independent Store Credit plan into a Suite-dependent plan.

If the accepted Store Credit license explicitly requires a licensed Suite, the product checks Suite's current public permission. The License screen then shows **Required Suite license verified** or **Required Suite license could not be verified**. A missing, incompatible, expired, or held Suite assertion pauses new licensed work for that plan, including during the initial activation window. Existing customer balances and recovery remain serviceable while the dependency is restored.

Independent plans show **AffStackDev Suite available (optional)** or **Standalone — Suite is optional**. Suite's presence is a presentation capability, not a substitute for Store Credit's own license. Developers can find the exact supported assertion fields and timing rules in the Suite dependency contract; there is no form field where an administrator pastes that assertion.

Which activities need new-value permission

Activity	Licensing effect
Issue new credit or commit a reviewed bulk-issuance batch	Requires new issuance permission. The Issue screen can show License activation required.
Increase a balance	Requires positive-adjustment permission. Negative corrections remain subject to the ordinary ledger checks without this new-value license gate.
Sell a gift card	New gift-card cart additions and checkout checks require sales permission. A hold affecting a gift-card row prevents payment of that row; ordinary products do not become gift cards because of the license hold.
Fulfill an already accepted paid purchase or gift	Remains settlement of that purchase. This does not permit starting a new gift-card sale while sales permission is held.
Create affiliate payout credit	Requires payout-credit permission.
Accept a new referral invitation, create a referral link, or create a new referral reward	Requires referral permission, alongside program/provider and customer-eligibility rules.

Activity	Licensing effect
Redeem existing value or recover/reconcile an existing financial operation	Remains available under its normal evidence, ownership, balance, and workflow checks.

Permission is rechecked where new value is committed, not only when the administration form opens. A license change can therefore stop a new operation that was started earlier. An already-recorded result can be recovered without authorizing a second issuance. If a new operation is held, restore its required permission and inspect the workflow's existing result before retrying.

Open and read the License screen

In standalone navigation, open **Store Credit & Gift Cards → License**. When the product appears inside Suite, use its **License** tab. Normal access to the product administration requires Store Credit management permission. Plugin-update controls require the separate WordPress permission to update plugins.

The screen heading is **Store Credit & Gift Cards Licensing**. Its forms use normal WordPress submissions and remain usable independently of the product's enhanced interface.

Screen section	What it shows
Summary	Product license, Suite integration, Plan, Environment, and applicable policy guidance. A missing plan is a dash.
Installation identity	Product code, Installation ID, Site, and License service hostname. Review the site and environment before activation.
Validation	Last attempted check, Next automatic check, Last accepted check, Offline deadline, Term deadline, and Last error code. All displayed times are UTC; missing times show a dash.
Activation panel	Activate Store Credit & Gift Cards, or Replace or recover activation when an active credential exists.
Protected updates	Readiness label and explanation, last plugin-update check, and the update-check button when permitted.
Release this site activation	Remote deactivation acknowledgment and button, shown while an active credential exists.

An accepted activation updates Last accepted check without setting a new Last attempted check; that attempt field records validation and deactivation requests.

Policy warnings may also appear elsewhere in WordPress for Store Credit managers, with an **Open License** link. A warning can describe the initial activation window, a dependency, a read-only restriction, or a product-license condition. Use the License screen and the relevant workflow together: a license status is not a balance, payment, or referral result.

Activate or replace the installation license

1. Check the displayed Site and Environment.
2. Enter the **Complete Store Credit & Gift Cards license key** in the blank password field.

3. For an installation without an active or pending credential, choose Production, Staging, Development, or Local. The default follows the WordPress environment setting, otherwise Production.
4. Select **Activate license**.
5. Review the resulting license status and next check.

The form accepts a key up to 128 characters, and the submitted key must fit the client's 128-byte limit after processing. Copy the complete issued key. The complete key is used for the explicit activation request and is not stored afterward. Store Credit generates and encrypts its own installation credential for later checks.

Environment selection is disabled while an active or recoverable pending credential exists. There is no ordinary License-screen editor for the stored installation ID, site identity, or service address. To change the licensed environment, first resolve any pending activation and release the active binding when appropriate, then choose the environment for activation again. The License Server owns entitlement and activation-slot decisions.

An active installation uses the **Replace or recover activation** panel, with the same **Activate license** button. The existing active credential is retained while a replacement is being resolved. A successful accepted response promotes the new credential; a failed replacement does not by itself revoke the old accepted entitlement.

Secure storage is required. Store Credit uses Sodium encryption where available, otherwise supported OpenSSL encryption. If neither is available, activation stops. Secure-randomness or local-storage failures also require administrator/hosting attention before another activation attempt can complete.

Recover an interrupted activation or changed site

An activation request may have reached the server even if its reply did not reach the browser. Store Credit retains an encrypted pending credential for that exact attempt. While it exists, the activation fields are disabled and the screen tells you to use **Check now** before entering another key. Due recovery can also run on an eligible visit or scheduled check.

A valid candidate becomes active. When a candidate is definitively rejected but an older active credential exists, Store Credit clears the candidate and can return to validating the older credential. A manual check can perform that older-token check immediately; an automatic check schedules it separately. If there is no older active credential and the server says the license is unavailable, the pending candidate can remain while the entitlement is resolved.

A changed site URL does not silently create a new installation identity. Activation, validation, and manual or automatic deactivation all require the current site to match the stored binding. Resolve the migration before retrying; the deactivation button is not a way to release the original site's binding from an unresolved moved copy.

The credential is also tied to WordPress security salts and stored identity/service details. Changing those can make it unreadable. Restore the correct storage context with the site administrator rather than generating a different identity to hide the warning. A local accepted-state integrity error is different: the encrypted credential can remain readable, allowing **Check now** to obtain a new accepted state.

A storage error means Store Credit could not confirm a required request intent or attempt guard was saved locally. Address the site's option-storage problem before retrying. The guarded remote request is not sent when that confirmation fails. The screen has no reset action that promises to free an old remote activation.

Interpret license states and deadlines

Product license state	Meaning and recovery
Activation required	No active credential. Activate, and note whether the initial setup window still applies.
Site deactivated	Remote deactivation was confirmed. Activate again if required.
Site identity changed	Stored and current site identities differ. Resolve the migration/binding.
License credential needs recovery	The encrypted credential cannot be read. Investigate salts, identity, service details, and cryptography.
Activation recovery pending	A candidate exists without an active credential. Use Check now or allow due recovery to run.
Local license state needs recovery	The accepted local cache failed its integrity check. Use Check now with the retained readable credential.
Validation required	A credential exists without accepted current validation. Check now.
System clock needs attention	Local time moved backward by more than five minutes from the last acceptance. Correct the clock and validate.
Activation missing	The server did not find this exact active installation. Review its identity and server activation.
License could not be authenticated	The server rejected the current credential or product scope. Review the Store Credit key/product binding.
License unavailable / License needs attention	The server explicitly denied current entitlement. The public response does not reveal whether it expired, was suspended, was revoked, or has another restriction.
License validation expired	The accepted term or connection allowance ended. Restore the entitlement/connection and validate again.
Licensed	A current product validation is accepted. Action-level feature restrictions can still apply.
Licensed — deactivation pending	Release was requested but not confirmed. The old permission remains bounded by its existing deadline.
Licensed — term grace	The server has permitted a term-grace interval. Review the license before its deadline.
Licensed — connection grace	The latest check could not reach the service. Previously accepted permission lasts only to its fixed limit.
Licensed — term and connection grace	Both apply; the earlier deadline wins.
Licensed — validation overdue	A check is late, but accepted permission remains within its hard deadline. Check manually or let an eligible visit/scheduled run retry.

A successful accepted response normally schedules another check around 12 hours later, with up to 15 minutes of variation. The connection allowance is at most 72 hours from acceptance; an earlier license-term or server-grace deadline wins. Repeated connection failures do not restart it. For example, a Monday noon acceptance

with a Wednesday noon term deadline permits at most 48 more hours without a newer accepted response, even though the connection window alone would last 72 hours.

The initial activation window and the connection allowance are separate: the first runs from stored setup time without a readable active/pending credential; the second comes from an accepted server validation. Neither overrides explicit restrictions. The License screen shows the product state plus separate Suite/policy guidance, so read both before concluding that a new operation is permitted.

Automatic checks and retry behavior

Check now appears when an active or pending credential can be read. It recovers a pending activation first where appropriate. If remote deactivation is pending, the same button retries that deactivation; its label remains Check now.

Automatic checks use WordPress scheduled events, eligible administrator pages, and ordinary read-only browsing. They avoid cart, checkout, account actions, WooCommerce endpoints, APIs, form submissions, AJAX, command-line requests, feeds, and trackbacks. Administrator activity recovery needs Store Credit management or WordPress settings-management permission. Multiple participating AffStackDev products share one activity-triggered licensing operation per request, so they may recover over successive visits.

Connection/local failures back off to approximately 15–16½ minutes after the first failure, 60–65 minutes after the second, and six hours after later failures. A server request to wait can extend a shorter availability retry up to six hours. Overdue scheduled events are preserved rather than repeatedly pushed forward by initialization. A running licensing operation can produce a “try again shortly” error instead of overlapping another operation.

Front-end recovery finishes the browser response first on supported FastCGI hosting. Other hosts can incur up to the automatic request's four-second network timeout on an eligible visit. Explicit license actions allow a longer request. A missed scheduled event is not by itself a denial: compare accepted deadlines and use Check now when needed. Repeated manual submissions do not repair rate limiting or extend accepted permission.

Release this site's remote activation

1. Resolve any pending activation first with **Check now**.
2. Open **Release this site activation**.
3. Check **I understand this releases the remote activation slot**.
4. Select **Deactivate license on this site** and review the resulting status.

A connection failure does not prove the slot was released. Store Credit retains the release intent and retries the same request identity through Check now or a due check. It removes the credentials only after an acceptable server acknowledgment. A pending release has no separate cancel button; Check now continues that intent.

Confirmed deactivation retains the stored installation identity and product data. It does not delete balances, transactions, coupons, orders, referrals, payouts, or WooCommerce records. It clears active/pending credentials, the pending operation, and the license-validation schedule. Any unexpired part of the original setup window remains tied to its original creation time rather than starting anew.

Disabling or uninstalling the WordPress plugin is different from remote license deactivation. Neither operation releases the server activation slot. The uninstall routine retains licensing/update options; financial records have their own retention rules described elsewhere in this reference. Use the explicit License workflow when remote release is intended.

Protected plugin updates

Store Credit uses its installation credential for protected updates; it does not retain the complete root key as an update workaround. The **Protected updates** panel shows these readiness states:

Label	Meaning and next step
Waiting for token-authenticated updates	The current accepted license response has not advertised this update method. Validate and review server support.
Release trust key required	A trusted release key, PHP Sodium signature verification, or PHP ZipArchive is unavailable. Ask the administrator to restore the required host/trust configuration.
Store Credit license required	The update prerequisites are present but current combined product/Suite permission and a readable active credential are required. Resolve the license/dependency. The initial setup window alone does not qualify.
Protected updates ready	WordPress can check for signed releases and request an authorized package. Installation still follows WordPress controls and scheduling.

With permission to update plugins, select **Check for plugin updates**. This performs a fresh release check. Ordinary checks reuse a result for up to 15 minutes, including a cooldown after a failure. The last check displays **Up to date**, **Update available**, or **Update check needs attention**, with a UTC timestamp. A license check and a release check are separate operations.

Use **WordPress → Plugins** to install an offered update or enable/disable automatic updates. Store Credit does not enable automatic installation for you. Only newer stable-channel releases are offered. Manual and automatic protected updates use the same authorization and verification path. A read-only or hard-lock license restriction does not by itself prevent updating when the current combined permission and token still qualify; a kill switch or unmet explicit Suite dependency prevents update readiness.

Update verification and recovery

The updater requires an authorized offer for the current installation and supported WordPress, PHP, and WooCommerce versions. It verifies the release signature, complete downloaded archive, embedded proof, and extracted files, and checks authorization again immediately before replacement. A changed or withdrawn offer requires another current check. The package selector shown to WordPress contains no reusable download authorization.

An invalid signature, unexpected/missing file, unsafe path, conflicting filename, identified symbolic link, oversized package, checksum mismatch, or mismatched version header stops protected installation. The exact format and limits are in the technical archive reference.

The updater also checks already-installed signed files before replacing them. Intentional local modifications should be reviewed with the responsible administrator before replacement. When both proof files are absent, an unsigned installation may receive its first signed update; only one missing proof file is an integrity failure. The installed check covers the signed inventory and does not treat every extra local file as a modified signed file.

Problem	Recovery
License/update readiness incomplete	Open License and restore the required permission, token, dependency, or host support.

Problem	Recovery
Connection or rate limit	Restore connectivity or wait before trying again. Repeated requests do not create authorization.
Offer or authorization changed	Check again and use the currently authorized offer.
Platform requirements not met	Review the release's WordPress/PHP/WooCommerce minimums.
Temporary download storage unavailable	Ask the host to restore writable temporary storage.
Signature, archive, or extracted-file verification failed	Obtain the valid current release and resolve the reported problem; do not bypass verification.
Installed signed files changed	Review and preserve intentional changes before replacing the installation.

The License screen stores a bounded last-check result. WordPress supplies its installation and rollback reporting; Store Credit does not add a separate automatic-failure history or dismissal control in this panel. A rejected protected package does not authorize replacement of the installed plugin.

Stored information and support details

Licensing requests send the product, installation ID, stored site URL, environment, and primary activation type. Activation also sends the complete key and a generated installation credential. Later validation, deactivation, and release checks use the installation credential. Release checks additionally identify the installed version and stable channel. Requests verify HTTPS and do not follow redirects.

The local configuration keeps the original setup time and site/installation binding. Credentials are encrypted with keys derived from the site's WordPress security salts and bound details. Accepted state stores plan, feature/capability information, server term dates and limits, check/retry times, and error details. Update cache/status keeps release/check information rather than a complete root key, installation token, or download-grant secret.

When seeking support, start with the displayed site/environment, product and Suite status, UTC check/deadline times, and last error code. Do not share complete root keys or decrypted credentials. The technical error lookup explains exact local failures and recognized remote denial/availability codes. A malformed or mismatched server response does not become new trusted permission.

What Commerce Core does

Commerce Core connects compatible products to shared customer context, cart-line identities and accepted order records. Store Credit & Gift Cards includes the runtime and offers it for selection when its requirements are met. It needs 64-bit PHP 8.1 or newer, WordPress 6.7 or newer, and active WooCommerce 10.0 or newer. The included Core release is 0.1.0.2c; its technical version is 0.1.0.2.3. See [Runtime and integration status](#) for Store Credit's own Integration status card.

When several compatible products include Core, the highest eligible runtime version supplies the code for that request. Matching versions must contain matching listed runtime files; conflicting copies are held until the affected products are updated to a consistent release. An already loaded standalone Core is not loaded a second time. The Integration status card identifies the selected runtime status, host and version. The selected host is not automatically the owner of your prices, gifts, business policies or credit balances.

Core coordinates information and records. Your connected products still control their business features and licensing. Installing Core alone does not create a discount, assign a customer to a business program, issue credit, transfer a gift rule or add a new payment action. Features that need a product adapter stay unavailable until that compatible adapter is present.

Read shared records and technical details

Store Credit's own Settings screen displays the passive Integration status card described in [Runtime and integration status](#). The bundled Core also contains a detailed panel that a compatible host can expose. When AffStackDev Suite supplies that interface, open **AffStackDev → Suite Settings → Commerce Core**, then expand **Shared records and technical details**. Viewing that host panel and downloading its records requires permission to manage WooCommerce and access to the host. The forms work without JavaScript. Store Credit does not independently add this panel or register its download action.

Display	How to read it
Commerce Core version	The version of the runtime selected for this request. A compatible connected product may supply a newer runtime than Store Credit's included copy.
Foundation — Available	Core can read its local runtime configuration. Connected products still have their own readiness checks.
Foundation — Needs review	Core cannot read the required local record. Resolve the database or stored-record problem before relying on affected shared features.
Audience connection — Configured	Core remembers at least one product responsible for determining consumer-promotion eligibility. This is a retained requirement, not proof that the product is currently working.
Audience connection — Not configured	No audience owner has been configured in Core's retained record. Each product continues to apply its own ordinary consumer behavior and other conditions.
Order records	Count of accepted Core order snapshots. If the database count fails, the card can display 0; the downloaded status report distinguishes that failure from an actual empty history.
Audience providers	Currently registered owner names, their declared versions and registration state. This list does not itself run their eligibility checks.
Shared feature owners	The last saved feature assignment, with its feature ID, capability, owner and current recorded state. Products verify permission and availability again before applying a change.

The shared-feature state reads **Operation needs reconciliation or is in progress** when an operation is pending, **Enabled** for an enabled unarchived definition, or **Paused** otherwise. **No assigned owner** means no owner is recorded; **A shared record needs review** means the returned entry cannot be presented normally. If no shared adapters are available, the panel explains that none are registered for this request. The list displays the first 25 shared records. An integration can obtain additional records through the paginated status interface; there is no Next button in this panel. The [shared-feature chapters](#) explain ownership, editing and operations.

Respond to connection messages

Connections need attention can show more than one issue. Resolve each applicable condition rather than treating the Foundation card as approval for every feature.

Message concerns	What to do
Local records unavailable	Restore database access and readable Core storage. Reload the panel after repair. Existing records remain the basis for recovery.
B2B needs its Commerce Core reader update	Use a B2B integration that supplies the supported fresh reader. The older reader can retain business exclusions but cannot approve uncertain consumer promotions.
Dynamic Pricing has multiple audience callbacks	Have the owning integrations remove the competing adapter as part of their coordinated update. One audience decision path is required.
More than one expected audience owner	Review the competing integrations and their retained requirements. Core does not choose an owner by loading order or silently discard one.
Previously configured audience provider missing	Restore that product's supported integration or follow the explicitly supported policy-retirement procedure for the affected consumer. Simply deactivating a provider does not erase Core's expectation.
Previously configured licensing protection unavailable	Restore the expected protection integration. A missing reader does not turn protected or unknown merchandise into ordinary merchandise.

A compatible Suite host can also provide its own product connection report. Accepted retirement is a separate proof that integrations may explicitly consume. This bundled Core's ordinary audience and line-protection readers do not automatically clear their remembered requirements when a retirement proof is accepted.

Download reports and individual order records

Under **Records and support**:

1. Select **Download status report** to download `commerce-core-report.json` with current runtime, provider, connection, snapshot-count and shared-feature information, plus the latest retained diagnostic sample.
2. To investigate a specific order, enter its positive numeric **Order ID** and select **Download order record**. The server accepts an existing order ID of 1–13 digits, without a sign, decimal point, spaces or leading zero.
3. Keep the file with the relevant support or integration investigation. Downloading changes neither the order nor its journal state.

Both downloads use the same filename. Rename saved copies if you need to retain more than one report. An individual order download contains the journal's revision, state, fingerprint and recorded data. It can include a draft or an interrupted publication as well as a committed record. It does not convert an unfinished record into an accepted one.

Enter a valid order ID means the input format or order lookup failed. Correct the ID and confirm the order exists. **This order has no readable Core record** means there is no readable Core journal snapshot for that order. A real WooCommerce order can exist without a Core snapshot, for example when no compatible historical-audience provider was configured. The export does not substitute today's customer classification or an older B2B record. **Access denied** requires a user with WooCommerce-management permission. If a form's security verification expires, reload the Settings screen and use its fresh form.

Reports are administrative records. Status reports identify integrations and saved shared definitions; order reports contain the recorded order subject and line evidence. Provide the relevant file to the person authorized to investigate the issue.

Customer eligibility and pricing integrations

An audience owner determines whether the current shopper is eligible for consumer promotions. Core ties its decision to the store, shopper and current checkout context. Changes in login, cart contents or quantities invalidate the request's previous context. Currency, shipping choice, tax scope, variation attributes and applied coupons also contribute to that context. Core passes a digest rather than storing the shopper's address in these context fields.

Fresh decisions have a limited validity period of at most five minutes and must match the same request. Consumer eligibility requires a consumer classification. Invalid, missing, conflicting or expired evidence leaves eligibility unresolved; it is not a permission to apply a consumer promotion. Connected products may also have their own finality and business-rule requirements.

The **Configured** state is deliberately retained when an audience provider disappears. Removing the plugin does not make all shoppers consumers. If no owner has ever been configured, Core reports that no audience provider applies; the connected product decides how its own standalone behavior uses that result. Core does not expose a customer selector, manual audience override or force-eligible button.

Legacy B2B and Dynamic Pricing

The older B2B reader can preserve a known ineligible outcome. It cannot establish fresh eligibility for a consumer, even when its earlier result said eligible. Its historical reader also cannot prove that an order's business context was atomically finalized. Install the supported reader integration when the affected product requires fresh or accepted historical evidence.

Core's Dynamic Pricing adapter runs only when an audience dependency already exists, B2B is present, or the runtime configuration cannot be verified. It requires exactly one audience callback. It delivers fresh supported owner evidence to Dynamic Pricing; it does not calculate the discount itself. Competing callbacks or uncertain B2B evidence hold the affected consumer decision until the integration is corrected. Installing Core by itself does not turn a previously independent Dynamic Pricing setup into a new audience-dependent setup.

Commercial facts without price changes

A compatible product can attach a commercial fact to a cart line, including currency, quantity, tax basis, price source, lock state, an optional unit amount and the kinds of adjustments its contract recognizes. Refreshing that fact replaces the prior fact in the same owner/slot. Core records the information; the owning product must actually apply any permitted price or adjustment.

Stored facts can remain on an order after their current-request validity expires. They describe what was recorded and do not approve a later price change. Missing amounts are distinct from zero. For integration authors, value rules and commercial-fact fields give the exact decimal, field and size requirements.

Shared cart lines and protected merchandise

Compatible products can opt a cart line into Core identity. An untouched ordinary line does not receive an identity merely because Core reads it. Once a product claims or links a line, Core preserves its UUID, the

declaring product's reference and any declared relationships or facts. This allows connected products to describe a line consistently without changing the product itself.

A line can contain up to 16 provenance entries, 32 relationship entries and 16 commercial facts; Core cart operations support at most 200 lines. A product updating its own owner/slot replaces that entry rather than adding another identity. A relationship creates matching outgoing and incoming records on two different lines. Removing that particular relationship keeps the lines' identities and other records. These are integration operations, with no general claim/link/unlink buttons in Store Credit Settings.

If Commerce Bridge supplies the existing line identity, Core reads it as Bridge-owned and does not migrate it. While the Bridge cart reader is present, Core's own line writers stay read-only, including for a line that Bridge describes as unprotected. Core also refuses an unknown protection result. An integration must establish one supported writer rather than let two systems alter the same identity graph.

Core reports relationships pointing to a cart line that is no longer present; it does not silently remove them. Complete order snapshot creation requires every related UUID to exist in the order. Duplicate IDs, mixed Core/Bridge identities, malformed records or unresolved protection need correction by the owning integration. If checkout displays **A shared cart line needs review. Please refresh the cart before checkout.**, refresh and review the cart. If the same message returns, investigate the line-owning and protection integrations before repeating checkout.

Core's classic-checkout line hook stores the validated line record in WooCommerce order-item metadata. Blocks snapshot capture reads whichever Core line records actually exist on its order items; the bundle does not register a separate Blocks line-record writer. Normal formatted order-item metadata and normal WooCommerce REST order responses omit raw `_afsd_cc_` records. Authorized Core readers and exports remain the way to inspect their recorded evidence.

Accepted order history and publication recovery

An accepted Core snapshot preserves the owning provider's committed audience decision together with the recorded order and line context. It includes the recorded customer, currency, totals, payment-method identifier and line evidence. The order fingerprint binds those fields to a specific order state. A snapshot does not itself mean the order was paid or authorize a commission, reward or credit.

Automatic capture is available on classic and Blocks checkout only when exactly one expected audience owner has a registered historical-order reader. A current eligibility reader alone is insufficient. If no such owner is available, Core does not manufacture a snapshot. This means the Order records count need not match your total WooCommerce order count.

Once accepted, history remains readable even if the customer changes, the order is refunded, the provider is removed or the order's current totals change. Those later events do not rewrite the original snapshot. Its purpose is retained historical context, not a fresh entitlement for today's shopper. An older B2B record can be returned only as unverified legacy history; it does not become a committed Core record.

Draft, publishing and committed

State	Meaning and next action
Draft	Core has prepared matching order and owner evidence but has not finalized publication. A supported integration can continue using the current revision and original order fingerprint.

State	Meaning and next action
Publishing	Core saved the durable publication intent before confirming the corresponding order metadata. Preserve that intent and investigate the original record.
Committed	The exact order projection was confirmed and the journal finalized. The accepted record remains available for historical reads.

If the panel reports **An interrupted order publication needs review**, download the original order record. A user with WooCommerce-management permission can have the supporting integration invoke Core's reconciliation API with that exact order, revision and fingerprint. There is no reconciliation button in this panel. Reconciliation confirms that the intended order projection already exists and then finalizes the journal; it does not write a second projection or replace the original request.

If that projection is missing or differs, the operation remains pending. Passage of time does not authorize a takeover or discard the intent. Resolve the saved-record discrepancy through the owning integration. Do not create another order or use a different operation identity as a substitute for confirming the original result.

At checkout, **The order context needs review before payment** means Core could not verify the order fingerprint. **The accepted order context could not be saved. Please contact the store before retrying payment** means preparation or publication failed. Inspect the existing order and its retained Core state before retrying payment. A late failure can leave a publishing record that requires reconciliation rather than a fresh attempt. Technical procedures are in snapshot publication and snapshot recovery.

Diagnostic reports and integration interfaces

The status report includes diagnostic counters from the current request and the latest retained affected request. Core retains up to 32 distinct bounded code names per request, with each counter capped at 999. It saves one latest sample rather than an accumulating activity log. An ordinary “no audience provider” outcome is not stored as a persistent warning. A later request with no new relevant codes leaves the earlier sample intact, so compare its recorded time with the current problem.

Diagnostic samples contain code names, counts, Core version and recording time rather than raw provider error text or shopper details. The broader status and order reports contain their respective integration and order records. A report code identifies the failed boundary; it does not mean every previous step was rolled back. Runtime error recovery and [shared-operation recovery](#) explain how to interpret the retained state.

Developers use Core's local PHP interfaces to register readers, read current audiences, manage line declarations, prepare and reconcile order snapshots, and call the separate shared-definition, operation, retirement and checkout-participant protocols. There is no public HTTP audience endpoint. Most local interfaces rely on the calling integration to verify its own user permissions and request security. Core adds WooCommerce-management checks to its administrative downloads and specified management/reconciliation operations. The complete function reference identifies every facade, argument order and owning contract.

The selected runtime supplies the `affstackdev-commerce-core` translation domain. The included catalog is a template that translation packages can use. Core's status layout wraps long identifiers, stacks cards and download fields on smaller screens, and provides visible keyboard focus for its controls. Its included GPL text is available with the runtime files.

Retained commerce records and safe updates

Commerce Core keeps the records that let compatible products agree on a feature's current configuration, responsible product and unfinished operation. It does not become the owner of product prices, customer balances or payout ledgers. Orders continue to use WooCommerce's normal storage interfaces.

Store Credit includes Core 0.1.0.2c. Another compatible active host can provide the selected shared runtime. The available operation depends on that selected runtime and the owning products' supported interfaces, rather than on a release number alone.

Records belong to the original WordPress site. Core requires a supported 64-bit PHP environment and rejects attempts to reuse its active runtime after switching to another site. Its installer creates missing local records without rewriting an existing database schema. Network-wide installation through the Core installer is refused; activation must be handled separately for each site. An unavailable or unfamiliar record is preserved and held for recovery.

Each record has an identity, a revision, a state, its data and a fingerprint. A successful update must name the revision it actually read. If someone else saves first, the stale update fails instead of overwriting their work. A new record uses expected revision zero. For example, if two operators read revision 7 and one saves revision 8, the second must reload revision 8 before making a reviewed change. Saving at revision 7 again does not overwrite revision 8.

A fingerprint confirms the expected data content; it is not a payment receipt. Core can coordinate its records without guaranteeing that a separate financial service completed a payment. If a result is uncertain, keep the original reference and reconcile the owning service before taking another value-changing action.

Core also remembers which audience and line-protection services it has already depended on. Adding a requirement is persistent. Removing the product does not erase that requirement or make a previously protected benefit safe. There can be at most 16 expected audience owners. Repeated conflicting configuration writes eventually return a busy result; they do not silently drop an owner. Snapshot status counts report retained records by state, not the amount of money owed or paid.

These records have bounded, validated data. The journal accepts at most 128 KiB of canonical data per record; individual text values are at most 2,048 bytes, lists/maps at most 400 entries per level and nesting at most 12 levels. The responsible integration supplies supported fields and types. Merchants should use its current form or supported service instead of editing database rows or changing revision numbers manually. Exact storage contracts are in the technical journal reference.

Choosing a supported feature provider

A shared feature can run only through a product that explicitly supports its full configuration and is currently ready to perform the requested task. Products register their local adapters during normal initialization. This happens automatically through code supplied by those products; there is no screen for entering a PHP callback.

Provider registration identifies the owner, provider version, supported feature families and schema versions, preference priority, and separate availability, evaluation, execution and reconciliation functions. Up to 16 owners can register; an owner describes up to 16 feature families with up to eight schema versions each. These limits are integration boundaries, not a reason to divide a financial operation across several invented owners.

Core distinguishes checking or editing a feature, executing its effect and reconciling an earlier effect. A product may remain able to read an old receipt while it cannot perform new business. Both full feature support and

current task readiness must succeed. Missing callbacks, exceptions, a wrong site or conflicting registration keep the feature unavailable. A duplicate accepted owner can make the shared registry unavailable for the whole request; resolve the conflicting integration and begin a fresh request.

The ready preferred owner wins. If the preferred product is unavailable, Core considers other products that explicitly support the same definition. Among those, a smaller provider priority wins, with owner-name order breaking ties. For example, priority 10 precedes 200, but a ready preferred owner takes precedence over both. This is a technical adapter setting; it is not a ranking of merchant product discounts. A feature's own stored priority is separate from provider priority.

The provider overview reports registered or unavailable and its declared feature support. Registered does not mean licensed for this transaction, eligible to serve this customer or safe to issue money. Those checks remain with the product.

An authorized integration can retire a shared executor, or restore its availability, using the exact saved retirement-record revision and a reason. This requires WooCommerce management permission. Retirement prevents that owner from beginning a new authorized effect; it does not delete shared definitions, undo earlier effects or remove its ability to supply supported old-receipt reconciliation. A compatible alternative may become responsible after current authority is resolved. A pending operation is not forcibly transferred by this action. This executor control is separate from the [business-policy retirement procedure](#).

The technical provider contract defines the trusted local integration fields. It does not add a general provider editor to Store Credit.

Shared configuration, pauses and ownership

Shared configuration keeps one original feature identity while compatible products can edit or perform that feature under its supported contract. A product must understand the complete saved terms. Installing another product does not convert old terms or turn on a paused feature.

Configuration or record	Effect and restriction
Feature identity	The original unique ID is retained across edits, ownership changes and recovery.
Capability and semantics	Identify the feature family and exact meaning of its terms; existing records cannot change these identities.
Definition schema and payload	The owning product's complete supported field set. A replacement schema must be understood explicitly. Core does not infer missing settings.
Enabled	Allows execution when true and every other check succeeds. Turning it off retains the definition.
Archived	Holds execution while preserving configuration and history. It does not cancel a customer's accepted obligation.
Feature priority	Integer 0–9999 retained for the feature's own meaning; not the provider selection priority.
Preferred owner	Empty for normal supported selection, or the exact preferred supported product.
Creator and last editor	Preserve the original creator and identify the most recent changed definition.

Configuration or record	Effect and restriction
Definition revision	Advances when the complete configuration changes.
Ownership generation	Advances when configuration or selected ownership changes.
Record revision	Used to protect every saved replacement from a stale request. It is separate from definition revision and generation.
Pending operation	Holds the original operation reference, attempt token, expiry and evidence; blocks another feature operation or edit until resolved.

These fields are supplied through an owning product's form or an authorized local integration. Store Credit does not provide an unrestricted editor for arbitrary feature payloads. Configuration data must fit 48 KiB and the common text/depth limits.

An editor must currently support editing that exact definition. Saving a changed definition requires its exact record revision and no pending operation. It preserves the creator, advances definition revision and generation, clears the old owner and records the last editor. A save with identical terms leaves the row unchanged, but still checks edit authority and current revision. There is no shared-feature delete operation in this interface; disabling or archiving preserves the record.

When a compatible product asks who should act, Core checks whether the feature is enabled, unarchived and free of an unfinished operation, then selects a ready supported owner. If none is available, the saved feature remains held, potentially with an empty current owner. A missing owner never authorizes another product to guess the terms. Simultaneous ownership changes may return busy after three conflicting attempts; reload the current result.

The shared status view is read-only and requires WooCommerce management permission. It shows owner, generation, definition revision, enabled/archived state and whether work is pending. An unreadable feature is identified as unreadable rather than hidden as successful. Reporting is paginated: 25 records by default and at most 100 per request. A displayed owner is a stored assignment, not permission to execute now. For exact output fields see shared definitions and reporting.

Fresh evaluation and short-lived permission

A product evaluates a shared feature using its current configuration, current owner and exact transaction context. Core does not retain an old cart result as permission to apply it later. The selected product supplies the result, and Core checks authority again after that callback completes.

The returned authority claim lasts at most 60 seconds. It binds the site, feature, owner, ownership generation, definition revision and complete definition/context fingerprints. A changed customer or transaction context, changed owner, edited definition or expired claim invalidates the earlier evaluation. Obtain a fresh evaluation instead of editing or extending the old claim.

An unfinished operation has its own reserved authority and prevents ordinary resolution from supplying a competing owner. A local claim is not a public authentication token, product license, customer eligibility decision or financial receipt. The relevant product must still enforce each of those requirements.

Evaluation results are bounded to 16 KiB; context is at most 8 KiB. The owning product controls the supported fields. Integrators can use the evaluation and claim reference to preserve the complete envelope and correct data types.

One business operation, one retained identity

Every delegated operation uses a stable business reference and exact context. Core derives a stable operation and effect identity from that original reference and site. Keep it when retrying, moving between supported owners or investigating a failure. Creating a new reference to get around a pending result can create a separate effect.

Operation state	Meaning and next step
Pending	An attempt was reserved and its outcome is not yet confirmed. Reconcile the original operation.
Committed	The owner supplied a confirmed effect receipt. Repeating the original request returns the retained result without executing again.
Not applied	The owner supplied a confirmed no-effect receipt. Ordinary repeats return that result. Another attempt requires the separate explicit recovery conditions.
Unknown callback answer	The journal remains pending; unknown is not a fourth terminal state or proof that nothing happened.

Core reserves the feature before asking a product to perform its effect. The feature permits one pending operation at a time. The operation freezes the feature definition, context, owner and short-lived claim, with a unique attempt token and empty receipt. Other operations and edits remain held while the reservation is unfinished.

Immediately at its financial action boundary, the product must confirm that the original pending record, token, lease, owner, generation and definition revision still match. Core checks this before and again after callback-based readiness. Retiring the owner or losing its current availability can stop a stale attempt. The product still owns its transaction, money lock, durable deduplication and authoritative receipt; a Core check cannot replace those controls.

A product returns a confirmed applied or no-effect result with its receipt. Core saves that terminal result before releasing the feature reservation. If reservation release fails, replaying the terminal original result can retry that release safely. A thrown callback, malformed answer or unknown outcome preserves pending evidence. A storage error can happen after the owning product applied an effect, so do not equate an error with “nothing was paid.”

Passive lookup retrieves the original record without executing anything. It is useful for support and reconciliation, but its existence alone does not make a pending effect safe to repeat. The technical operation contract defines the exact callback and receipt fields.

Reconciling and recovering unfinished work

Reconciliation asks a supported authoritative reader what happened to the original operation. It never invokes the effect callback again. Core waits until the original reservation's lease expires before reconciling pending work. If the process stopped after reserving the feature but before inserting its operation row, Core can reconstruct that original pending row from the reservation, keeping the same claim and token.

The reconciler may be another supported product that understands the frozen feature and can verify its authoritative original receipt. It must report confirmed applied, confirmed not applied, or unknown. Unknown

keeps the hold. Expiry by itself does not prove absence: an earlier writer could still have applied the effect before the response was lost.

A new attempt is supported only through the explicit no-effect recovery API, when both the selected Core and the owning integration support it. The embedded Core does not create an automatic financial retry schedule or unrestricted Retry button for this API. The operator uses the original product's supported recovery workflow.

For explicit recovery, all of these conditions matter:

1. The original operation is confirmed **not applied**, its old claim has expired, and the request names its exact current operation revision, receipt and unchanged context.
2. The current feature's full definition and definition revision match the frozen original. Editing terms and then restoring their old values still changes the revision and blocks this recovery route.
3. An authoritative reconciler freshly confirms the same no-effect receipt and proves that every old writer is prevented from applying the old attempt. An empty table, elapsed timeout or manually typed receipt is insufficient.
4. A currently ready supported owner receives a different attempt token and bounded authority. The original business reference, effect identity, site and terms remain.
5. The old terminal attempt is archived before the original operation is advanced to pending again. Only the winning current revision can reach the effect callback, which must still enforce its own financial protections.

For example, after a provider confirms that reference `reward:order-42` was not applied, recovery can keep that same reference and its original frozen terms. Submitting `reward:order-42-retry` would be a different operation and is not the supported recovery procedure.

If recovery is interrupted before advancing the operation, the same request can resume an existing matching reservation without extending its expiry. An expired reserved attempt cannot be executed merely by resuming it; it must be reconciled. A failure after the new pending operation is saved leaves pending evidence. Reread that operation after an ambiguous result or conflict rather than repeatedly pressing retry or clearing another attempt's reservation.

Archived attempts retain their exact original receipts and payloads, linked by previous-attempt revisions. They are not overwritten or embedded recursively. Keep the archives, current row and financial owner's evidence together. If the current host lacks the required recovery interface or the original terms changed, leave the original case held for supported owner review; there is no force override. Exact requirements are in same-operation recovery.

Retiring a business policy safely

Business-policy retirement is a specific proof that an owner has no live policy left in an exact scope. It differs from retiring a shared executor. The two supported scopes concern current audience policy and protected-line policy. A compatible Suite host can expose B2B audience retirement controls. Store Credit reports and consumes accepted retirement evidence through its supported B2B adapter; it has no retirement form. Host controls do not expose every owner/scope or release the B2B product's business freeze.

Use the owning product's preparation procedure before deactivating it. The product must establish a durable freeze and confirm no live policy remains. Core records the exact owner, scope, site, request identity, policy revision, evidence reference and freeze reference. Preparation also records the current runtime configuration

fingerprint and a five-minute acceptance deadline. Core checks the proof's shape; the owner is responsible for the actual freeze.

Step	Requirement and effect
Prepare	Owner and its retirement guard are present. An authorized WooCommerce manager uses the current revision. The guard confirms no live policy and frozen state. Core writes Prepared evidence.
Deactivate owner	Follow the owner's supported procedure while keeping its freeze intact. Preparation alone does not remove the product.
Accept	Use a fresh request after deactivation, before five minutes expire. The guard and both site/network-active plugin entry must be absent, with the exact Prepared revision and unchanged runtime fingerprint.
Read accepted proof	A compatible consumer freshly verifies the accepted original record. It does not substitute the management display or an error.
Cancel	An authorized exact-revision write revokes the record. The owner separately decides when its own resume conditions are satisfied.

An attestation can freeze owner-side work before Core's final save succeeds. If preparation fails, keep the freeze and check the actual stored result; do not assume the owner can resume. Likewise, an acceptance notification can fail after the Accepted record was saved. Reread persisted state instead of repeating a different operation on the assumption of failure.

Accepted proof does not expire simply because the original five-minute preparation deadline passes. It remains subject to its site, runtime fingerprint, absent owner and unrevoked state. Changed runtime expectations, returning owner or invalid data make it unusable. Retirement does not erase old audience/protection requirements, transfer business prices, change historical order evidence or grant consumer discounts. Ordinary Core readers still hold when their required provider is absent unless a compatible consumer explicitly uses the separate retirement proof under its own rules.

The management view separates **stored state** from **validity**. State can be None, Prepared, Accepted or Revoked. Validity can be none, current, expired, stale, owner present, revoked or unavailable. Prepared can be current while the owner is present; that does not make acceptance possible yet. The owner-present flag reflects the collected guard or retained plugin basename, not an unrestricted scan of every possible owner.

Revocation reporting also distinguishes not attempted, confirmed and unconfirmed. This is what the current request observed; a later stored revision may differ. A read/write error must not be shown as confirmed revocation.

Cancellation always writes and advances the revision, even if the record was already revoked. Expired or stale proof can be cancelled, but corrupt or wrong-site records remain held for recovery. If no record exists, cancelling at revision zero creates a cancellation-only marker. That marker is not retirement proof; it prevents a delayed earlier preparation from appearing after cancellation. A later legitimate preparation requires the newer revision and new authentic owner evidence.

When an owner returns, Core attempts to revoke its previously accepted audience and protection records. It preserves Prepared records so the normal prepare → deactivate → accept sequence can work. Failed revocation remains unconfirmed: the owner's durable freeze must stay until all relevant cancellations succeed under its own write lock. A saved cancellation does not itself release that lock or resume business policy. Do not delete

the journal or reset a revision to bypass a failed cancellation. The technical retirement contract contains exact fields and permission boundaries.

Discovering checkout credit participation

Core can advertise the currently registered Store Credit reservation participant to a compatible checkout coordinator. This discovery is independent of ordinary audience providers and shared feature owners. It does not reserve or spend credit.

The supported descriptor has exactly six values: schema 1, owner `affstackdev-store-credit`, capability `store_credit_reservation_participant_v1`, integer revision 1, and the owner's callable `enlist` and `order-verification` functions. Compatible plugins supply this descriptor in code during checkout-participant collection at initialization priority 40. There is no customer field for choosing a callback or entering an alternate protocol revision.

Registration must finish before another plugin resolves the participant. A duplicate attempt for the supported owner makes the participant conflicted for the request, even if the second submission is identical or follows a malformed first attempt. A thrown collection callback discards partial registration. Records are not saved across requests, and stale unavailable callbacks are discarded rather than reused.

Connection status	Meaning and response
Registered	Descriptor and callable functions are available. The actual transaction still needs its owning products' checks.
Not ready	Collection has not finished; the integration must resolve later in initialization.
Missing	No accepted supported descriptor was registered; restore the owning integration.
Unavailable	Required function is not callable; restore the compatible implementation.
Conflict	Competing supported-owner registrations occurred; correct them before a fresh request.
Failed	Collection could not complete; investigate the registering service.
Site mismatch	The active Core belongs to another site; use the original supported site context.

Reading status does not invoke either financial function. Resolving a participant checks the exact owner, capability and integer revision, then returns a detached descriptor for the trusted caller. It does not validate an order's financial proof, verify payment, mark an order paid or provide a substitute participant. B2B owns coordination, Store Credit owns its ledger and receipt evidence, and both products' actual transaction contracts remain necessary. Use [checkout coordination](#), [B2B integration](#) and the technical participant contract for their respective boundaries.

Shared operation troubleshooting

Symptom	Appropriate response
Record or runtime cannot be verified	Restore the original site's compatible runtime/storage. Preserve unfamiliar schema and corrupt evidence for recovery; do not overwrite it as empty.

Symptom	Appropriate response
Save conflicts or reports busy	Reload the current revision and review concurrent changes before saving.
No supported feature owner	Check the exact definition, product availability, enabled/archive state, adapter registration and current permissions. A displayed old owner is not a grant.
Evaluation or claim expired	Obtain a fresh current evaluation; do not edit the token or extend its expiry.
Operation pending or outcome unknown	Keep the original reference and context. Reconcile through its authoritative owner after the lease boundary; do not submit a new financial operation.
Confirmed not applied but retry held	Review exact operation revision/receipt, unchanged definition and fresh no-effect proof. Explicit supported recovery is required.
Archived attempt cannot be read	Keep the original record/history intact and restore the compatible reader; do not manufacture replacement history.
Retirement preparation fails	Keep the owner's freeze, inspect current proof/revision and complete the owning product's supported preparation.
Acceptance is unavailable	Confirm owner deactivation, five-minute preparation window, unchanged runtime and exact revision. Reread actual state after ambiguous results.
Cancellation or automatic revocation is unconfirmed	Keep the freeze and inspect persisted state under the owner lock. Cancel at the current revision; do not assume it succeeded.
Retirement was accepted but benefits remain held	Acceptance is separate evidence, not a customer eligibility grant or an automatic clearing of provider requirements. Follow the consuming product's supported policy.
Checkout participant is missing/conflicted	Restore the exact compatible owner and correct registration timing/duplicates. Discovery failure does not authorize bypassing credit checks.

Provide support with the original site, feature/business reference and current error code. Keep the original journal contents intact while the owning integration resolves the case. The technical error lookup maps every returned code to its specific recovery boundary.

Find the complete Store Credit & Gift Cards references

Open **AffStackDev** → **Store Credit & Gift Cards** → **Settings** → **Documentation**. Choose **Download customer reference Markdown** for the complete task reference or **Download technical reference Markdown** for the detailed developer and agency reference. Both files belong to the installed Store Credit & Gift Cards release. Word and PDF editions of the customer reference are on affstackdev.com.

Your account needs the `manage_store_credit` capability, normally granted to Administrator and Shop Manager on activation. The downloads remain available when Enhanced UI is off or the product is using PHP administration mode. Downloading a reference changes no settings and performs no commerce operation.

Recover a manual download

If a saved download link is rejected, reopen Store Credit & Gift Cards Settings and use a fresh link. If the file is unavailable, restore the matching Store Credit & Gift Cards package so both canonical files are present in its docs directory. A permission message means the signed-in account lacks `manage_store_credit`; a file-read error means the installed document or its filesystem permissions need attention. The Documentation card downloads Markdown. Use the Word or PDF edition on affstackdev.com when you prefer those formats.

Read Store Credit & Gift Cards through AffStackDev AI

When the separate AffStackDev AI 0.1.0.11a or a later compatible version is installed, its plugin documentation area can identify Store Credit & Gift Cards's exact installed version, feature reference and two current manuals. Store Credit & Gift Cards includes a declaration that points to the same Markdown files downloaded from Store Credit & Gift Cards Settings. The declaration identifies each manual's technical or customer audience and its authored release. It does not give an AI assistant permission to change your store.

After installing an updated Store Credit & Gift Cards package in the existing plugin folder, open Store Credit & Gift Cards in the AI plugin documentation area and choose **Check this plugin again**. Check **Installed plugin version**, then select the references labeled **Installed package**. That check rereads the installed files; it does not install Store Credit & Gift Cards or fetch replacement documentation. If Store Credit & Gift Cards is excluded from owner AI, use its **Include this plugin for owner AI** control when you want those owner-facing reference tools to read it.

Under **Previous manuals & automatic checks**, enable **Continue using previous documentation when current manuals are unavailable** and choose **Save documentation preference** if you want eligible retained references to fill missing documentation. This preference applies across installed plugins. A **Previous manual** records its earlier documentation version. A readable current customer or technical manual takes precedence over previous manuals for that audience. Use the **Installed package** pair for the current Store Credit & Gift Cards release. A missing or mismatched feature reference is recovered by reinstalling the matching installer from My Account → Downloads and checking it again. A large feature reference may require opening Store Credit & Gift Cards's individual documentation screen to complete inspection, and its full feature list is retrieved in pages. After a completed check, the inventory can reuse its saved summary.

Plugin manuals and reviewed site knowledge have separate refresh and sharing controls. Updating Store Credit & Gift Cards replaces its installed references; it does not change an earlier AI reply or automatically approve a site knowledge source for a customer-facing channel.

Glossary

Term	Meaning
Instrument	One issued store-credit or gift-card account with its own code, owner, currency, rules and ledger.
Store credit	Value classified as credit, including merchant issuance, compensation and rewards.
Gift card	Purchased or gifted value, or gift card balance a customer sent, presented separately from store credit.
Ledger	Durable transactions recording issuance, reservation, spending, restoration and other balance changes.

Term	Meaning
Available balance	Value currently usable after active reservations and the instrument's rules.
Reservation or hold	Value set aside for one operation before final settlement.
Settlement	Finalizing the original reserved operation with its retained identity and evidence.
Idempotency reference	An operation identity that prevents a repeated request from issuing or spending value twice.
Claim	Depending on context, associating a gift card with an account or establishing a referral operation's authority.
Redemption	Applying eligible issued value through WooCommerce's supported credit transport.
Tax budget	Whether value consumption counts only the merchandise discount or also its tax reduction.
Allocation	Evidence assigning an already accepted credit effect to individual checkout or order items.
Reconciliation	Resolving an operation from its saved evidence before retry or recovery.
Core	Bundled shared commerce coordination runtime, separate from the plugin's financial ledger.
Owner	The product or component authorized to decide a particular business effect.
Snapshot	Facts retained at a decision point for subsequent replay, comparison or recovery.
Definition	Versioned configuration accepted by a supported shared feature contract.
Retirement	Explicit release of a defined owner's authority while preserving historical evidence.
Enhanced UI	Optional JavaScript administration layered on the plugin's PHP screens.
Nonce	WordPress action-bound request token checked in addition to the user's capability.

Use Store Credit with an owner AI assistant

Store Credit & Gift Cards supplies reviewed actions to AffStackDev AI 0.1.0.12 and compatible later releases. An authorized connected administrator can ask the assistant to find records, inspect the saved settings and prepare changes. You review the exact proposal in WordPress and approve it before applying it. Existing Store Credit menus remain available.

Open the owner AI action catalog and select the Store Credit and Gift Cards provider. If the plugin is excluded from owner AI, include it through the AI plugin's native inclusion control. Action availability also depends on your current WordPress permissions and the owning WooCommerce, licensing or referral services. Updating an installed plugin refreshes its available action definitions on the next request. An old proposal must be prepared again after its provider or reviewed state changes.

Review and apply a request

1. Describe the intended task and recipient. Names, usernames, email addresses, product titles and SKUs can resolve records. "Me" means the WordPress account that authorized the connection. Choose among returned candidates when a name is ambiguous.
2. Ask the assistant to inspect the current record and prepare the change. Inspection and preparation do not issue value, send an email or save the requested settings.
3. Open the project review in WordPress. Check the identified recipient, currency, amount, restrictions, dates and every listed effect. Issuance requires an explicit choice about sending its email. Gift-purchase and referral recovery use their original saved terms.
4. Approve the exact project in the authenticated native review screen, then explicitly apply it. Approval alone does not perform the change. Reject an unwanted proposal instead of applying it.
5. Read the saved result and follow its native editor link when further work is needed. A completed receipt is historical evidence of that operation; inspect the current record to see later changes.

For example, ask "Prepare 25 of store credit for me without sending an email." Confirm the recipient and store currency before approving. Continue that original project to avoid duplicate issuance. A separately approved project is a new issuance even with identical text. Without a manual code, the native random generator creates one on apply.

Available administration tasks

Area	Available through reviewed actions
Records and reporting	Find customers, products, categories and credits; inspect ledger activity, gift purchases and referrals; read reports, reconciliation findings, CSV export pages, settings, license status and Commerce Core reports.
Credit administration	Issue credit, process a CSV batch, adjust available value, expire, revoke, archive an empty balance, resend an issuance email and restore original credit for an existing partial refund.
Gift cards	Create the native Gift Card draft, configure its Store Credit sale fields, retry an eligible paid purchase and stop further issuance while revoking eligible cards.

Area	Available through reviewed actions
Referrals	Change program terms, qualifying products, reward restrictions and offer combinations; enable shared program processing; retry or cancel an existing accepted claim through its original owner.
Site preferences	Save Store Credit display mode, multiple-credit behavior, default coupon combining and supported issuance-email settings.
Licensing	Inspect status, check a saved activation, release an activation and check protected update metadata. Enter or replace license keys and select the activation environment in the native secure license form.
Privacy	Read this plugin's privacy exports; remove messages from selected credits or revoke referral sharing for an already confirmed WordPress erasure request. Financial and eligibility records remain retained.
Preview management	Reject an AI project, or discard your uncommitted native CSV preview using the preview token from its URL.

The gift-card action changes Store Credit sale settings. General WooCommerce product editing and publication, cash refunds, advanced coupon editing, and PHP email-template files retain their native owners and controls. Completing a Store Credit action does not attest that those separate operations happened. The privacy actions cover this plugin's callbacks; WordPress still owns the complete personal-data request and its delivery.

Batches and interrupted operations

AI CSV proposals review up to 20 rows at a time. Supply up to 15,000 bytes of CSV text, or upload through Store Credit Operations and copy the `bulk_batch` token from the preview URL. Inspect that private preview to obtain its fingerprint, then select rows with `row_offset` and `row_limit`. The native 250-row/512 KiB and 30-minute preview limits still apply. Changed, expired or other-administrator previews are refused. Keep each original row reference. Exports return ten rows per page and a continuation value. Follow all pages before treating an export as complete. Lookups also identify whether more results remain.

A purchase stop can finish with financial review still required. Reserved cards remain unchanged; other eligible cards can be revoked, while spent value remains in history. A partially stopped purchase is not refund-ready. Referral cancellation similarly retains a review state when a reward is spent or reserved. Read the exact reviewed outcome and saved result.

If a project becomes unknown or interrupted, open that original project and reconcile it. Reconciliation observes the saved result and original evidence. It does not resend an email, issue another credit or repeat a financial change. An unresolved operation can block another AI operation on the same target. Restore the missing service or evidence and use the native record for review; do not create fresh requests merely to get around the uncertain one.

Email results report mailer acceptance or the existing automatic retry queue. They do not establish inbox delivery. Explicit resending is a separately reviewed action. When a target, permission, license, dependency or proposal changes, prepare and review a fresh proposal as directed.