

Rebuy Smart Cart Discount System Documentation

Overview

This system manages complex discount scenarios in the Rebuy Smart Cart, handling multiple types of discounts that can work independently or in combination. The system is deployed across 8 stores and handles:

Implementation Order & Use Cases:

1. **Quantity Updates:** Incrementing/decrementing line items (foundation functionality)
2. **Manual Discount Codes:** Percentage and fixed amount discounts via cart input
3. **URL-Based Discount Codes:** Same discount types but applied via URL parameters (combinable)
4. **Free Gift Implementation:** Threshold-based gifts with 0% or % discount variants (combinable)
5. **3-for-2 Promotions:** Manual codes, URL-based codes, and automatic Shopify discounts (combinable)
6. **Advanced Combination Scenarios:** Multiple discount types working together

System Capabilities:

- **Percentage/Fixed amount discounts** (manual codes or URL-based, combinable with correctly configured auto-discounts or gift codes)
- **Free gift promotions** (with threshold requirements, 0% or % discount variants, applicable manually OR via URL, combinable with correctly configured auto-discounts OR manual codes)
- **3-for-2 promotions:**
 - Manual codes
 - URL-based codes
 - Automatic Shopify discounts
 - **Combinable:** Auto 3-for-2 can combine with correctly configured manual/URL codes (percentage, fixed, or gift codes)
- **Complex combination scenarios:**
 - Auto 3-for-2 combinable with manual/URL order-level discounts
 - **Advanced edge case:** Auto 3-for-2 + URL discount + manual free gift code (triple combination)

Note: The most complex edge case involves auto 3-for-2 + URL discounts, where users can still manually add compatible free gift codes (0% or % discount variants), creating triple-layered discount scenarios. This requires discount codes to be configured as combinable in Shopify admin.

Important: Manual 3-for-2 codes and Auto 3-for-2 discounts cannot occur simultaneously - Shopify will only apply one since they perform the same function (reducing cheapest items to €0.00). Manual 3-for-2 codes are intentionally set as non-combinable to prevent conflicts.

Architecture Overview

Core Modules



File Structure

- 1. `custom-rebuy-event-manager.js` - Event coordination hub
- 2. `custom-cart-utils.js` - Shared utilities and cart operations
- 3. `custom-cart-callbacks.js` - Price display and DOM updates
- 4. `custom-three-for-two-handler.js` - 3-for-2 promotion logic
- 5. `custom-free-gift-handler.js` - Free gift management
- 6. **Liquid templates** - Settings and initialization

Discount Detection Logic

Priority Order (in `getActiveDiscount()`)

1. Manual Rebuy Discounts - Applied via cart discount input

```
javascript  
  
window.Rebuy?.SmartCart?.discount?.discount?.value
```

2. Shopify Checkout-Level Discounts - URL-based or automatic

```
javascript  
  
window.Rebuy?.Cart?.cart?.cart_level_discount_applications
```

3. Automatic Buy-X-Get-Y - Shopify's automatic 3-for-2

```
javascript  
  
// Detected by items with discounted_price = 0
```

Discount Types Supported

All discount types can be applied via **manual code input** OR **URL-based** application:

- **Percentage:** `(10)` (10% off) - Manual/URL
- **Fixed Amount:** `(12.99)` (€12.99 off) - Manual/URL
- **Free Gift (0% discount):** `(0)` with gift code validation - Manual/URL
- **Free Gift (with % discount):** `(10)` + gift code validation - Manual/URL
- **Manual 3-for-2:** Via validated codes in `(active3for2codes)` - Manual/URL
- **Automatic 3-for-2:** Shopify's native Buy-X-Get-Y discounts - Auto only

Discount Combination Rules

- **Product-level discounts** (Auto 3-for-2): Can combine with order-level discounts
- **Order-level discounts** (Manual codes, URL codes, Gift codes): Can combine with product-level discounts
- **Manual 3-for-2 vs Auto 3-for-2:** Cannot occur simultaneously - Shopify applies only one since both reduce cheapest items to €0.00. Manual 3-for-2 codes are configured as non-combinable.
- **Triple combinations:** Possible but require discount codes configured as combinable in Shopify admin
- **Result:** Everything matches and works together in supported combination scenarios

Main Processing Flow

`processCartItems()` Function

This is the central orchestrator in `custom-cart-callbacks.js`:

Function Responsibility

The `processCartItems` function is the central orchestrator for cart UI updates. It:

- Decides when to update the DOM based on cart state changes
- Handles visual price updates for different discount combinations
- Manages the display of discounted vs. original prices
- Coordinates with `ThreeForTwoHandler` for 3-for-2 scenarios
- **Handles quantity-based price updates:** Updates line item totals when quantities are incremented/decremented (not just single unit prices)
- **Works with all item types:** Sale items, reduced items, full price items all show correct totals based on quantity

javascript

```

async function processCartItems(cart) {
  // 1. EARLY EXITS
  if (!cart.items.length) return;
  if (!shouldProcessCart(cart)) return; // Signature-based change detection

  // 2. DETECT DISCOUNT SCENARIO
  const hasAuto3For2 = hasAutomatic3For2Discount(cart);
  const discountData = window.RebuyUtils.getActiveDiscount();
  const isZeroPercentGiftCode = /* Enhanced 0% gift code detection */;

  // 3. MAIN BRANCHING LOGIC
  if (hasAuto3For2) {
    // BRANCH A: Automatic 3-for-2 scenarios
    if (hasManualCode && !isZeroPercentGiftCode) {
      // COMBINATION: Auto 3-for-2 + Manual discount
      // Apply manual discount to non-free items only
      applyComboDiscount();
    } else {
      // Auto 3-for-2 ONLY - let ThreeForTwoHandler manage
      revertToRegularPrices();
    }
    return; // Early exit for all auto scenarios
  }

  // BRANCH B: Manual-only scenarios
  if (currentDiscount < 0) {
    // Regular discount codes - apply to all items
    applyRegularDiscount();
  } else {
    // No discount - quantity updates only
    // Updates line item totals when quantities change
    // Handles sale items, reduced items, full price items
    updateQuantitiesOnly();
  }
}

```

Key Scenarios & Behaviors

Scenario 1: Quantity Updates (Foundation)

- **Detection:** No active discounts detected
- **Handler:** `RebuyCartCallbacks`
- **Behavior:** Updates line item totals when quantities are incremented/decremented, maintaining correct pricing for sale items, reduced items, and full price items (displays total per line item, not unit prices)

- **Purpose:** Foundation functionality that works regardless of discount state

Scenario 2: Manual Discount Codes (Percentage/Fixed)

- **Detection:** `currentDiscount < 0` and not a 3-for-2 code
- **Handler:** `RebuyCartCallbacks`
- **Behavior:** Apply proportional discounts, show original vs. discounted prices, update totals based on quantity changes

Scenario 3: URL-Based Discount Codes

- **Detection:** Discount codes applied via URL parameters, detected in checkout-level applications
- **Handler:** `RebuyCartCallbacks`
- **Behavior:** Same as manual codes but applied automatically via URL, can combine with other discount types

Scenario 4: Free Gift Implementation

- **Types:**
 - **0% discount:** `discountData.value === 0` + valid gift code
 - **With % discount:** `discountData.value > 0` + valid gift code + threshold met
- **Detection:** Relevant object found in `window.discountGiftSettings` for frontend / cart validation
- **Sources:** Manual code input OR URL-based
- **Handler:** `FreeGiftHandler`
- **Behavior:** Add free gift to cart if threshold met, maintain quantity = 1
- **Critical Issue:** 0% codes can be misinterpreted as large fixed amounts (e.g., 1690 instead of 0%), causing negative subtotals

Scenario 5: 3-for-2 Promotions (Manual)

- **Detection:** Code found in `window.active3for2codes.buyXgetYcodes` for frontend validation
- **Sources:** Manual code input OR URL-based
- **Handler:** `ThreeForTwoHandler`
- **Behavior:** Shows discounted prices, adds "GRATIS" tags, calculates cheapest items as free

Scenario 6: 3-for-2 Promotions (Automatic Shopify)

- **Detection:** Items with `discounted_price = 0` and `price > 0`
- **Handler:** `ThreeForTwoHandler` + `RebuyCartCallbacks`
- **Behavior:** Items are already free in Shopify, just add visual indicators

Scenario 7: COMBINATION SCENARIOS - The Complex Cases

Supported Combinations:

- Auto 3-for-2 + Manual percentage/fixed discount codes
- Auto 3-for-2 + URL-based percentage/fixed discount codes
- Auto 3-for-2 + Manual free gift codes (0% or with discount)
- Auto 3-for-2 + URL-based free gift codes (0% or with discount)
- **Special case:** Auto 3-for-2 + URL free gift (0%) + Manual percentage (triple combination)
- **Detection:** `hasAuto3For2 && hasManualCode && !isZeroPercentGiftCode`
- **Handler:** Both `ThreeForTwoHandler` and `RebuyCartCallbacks`
- **Behavior:**
 - Free items stay free (managed by Shopify)
 - Additional manual discount applies only to non-free items
 - Custom subtotal calculation required

Critical Functions

`shouldUpdateSubtotal(cart)`

Purpose: Detects when manual subtotal calculation is needed

Logic:

```
javascript

// Only update subtotal when:
// 1. Auto discount present (items with discounted_price = 0)
// 2. Manual discount code active (not 3-for-2, not 0% gift code)
// 3. All items have quantity = 1
return hasAutoDiscount && hasValidManualCode && allQuantityOne;
```

Why needed: Shopify's subtotal doesn't account for cosmetic DOM price changes from manual codes.

`allocateDiscountAcrossItems()` VS `allocateFixedDiscountsAcrossItems()`

Percentage discounts: Use proportional allocation across all items **Fixed discounts:** Distribute fixed amount proportionally by item value

State Tracking with Signatures

Purpose: Prevent unnecessary DOM updates

javascript

```
const discountSignature = `${currentDiscount.value}:${currentDiscount.type}`;  
const itemsSignature = cart.items.map(item => `${item.id}:${item.quantity}:${item.discounted_price || 'non'}
```

Debugging Guide

Debug Mode Activation

javascript

```
// Enable debug logging for all modules, call these in the inspector and ... do stuff in the cart for thorough l  
window.RebuyCartCallbacks.debug(true);  
window.ThreeForTwoHandler.debug(true);  
window.FreeGiftHandler.debug(true);
```

Common Issues & Solutions

Issue: Prices not updating

Check:

1. `shouldProcessCart()` returning `false`? Check signatures.
2. `getActiveDiscount()` returning expected discount type?
3. Are DOM selectors finding the right elements?

Issue: Subtotal incorrect

Check:

1. `shouldUpdateSubtotal()` logic - are all conditions met?
2. Is `updateCustomSubtotal()` finding all price elements correctly?
3. Check for 3-for-2 free items being included in subtotal calculation

Issue: Combination discounts not working

Check:

1. `isZeroPercentGiftCode` logic - might be excluding valid codes
2. `nonFreeItems` filter - are the right items being selected?
3. Order of operations - is `ThreeForTwoHandler` running before `RebuyCartCallbacks`?

Issue: 0% Gift Codes Causing Negative Subtotals

Problem: 0% gift codes were being interpreted as massive fixed-amount discounts (e.g., 1690 instead of 0%), causing crazy negative numbers in UI.

Root Cause: System was detecting the raw discount amount instead of recognizing it as a 0% percentage discount.

Solution: Enhanced detection logic excludes 0% gift codes from discount processing:

javascript

```
const isZeroPercentGiftCode = manualCode &&  
  window.FreeGiftHandler &&  
  window.FreeGiftHandler.isValidGiftCode(manualCode) &&  
  window.RebuyUtils.getActiveDiscount().value === 0;
```

Check:

1. Verify `isValidGiftCode()` is working for the problematic code
2. Check that `getActiveDiscount().value` returns `0` not the raw amount
3. Ensure the code is properly excluded from `shouldUpdateSubtotal()` logic

Issue: 0% Gift Codes Breaking Other Calculations

Problem: The fix for 0% gift codes creating negative numbers can interfere with legitimate discount calculations.

Symptoms:

- Valid percentage discounts not applying
- Combination scenarios not working
- Subtotal not updating when it should

Check:

1. Is `isZeroPercentGiftCode` incorrectly flagging valid discount codes?
2. Are valid codes being excluded from `shouldUpdateSubtotal()` logic?
3. Check the order of detection - gift code validation vs. discount code validation

Issue: Allocation calculations incorrect with auto discounts

Problem: During auto discount scenarios, discount allocations don't match expected behavior, especially after quantity changes.

Root Cause: Shopify's native behavior with auto discounts - when users increment quantities via line item selectors, Shopify may:

1. Split single line items into multiple entries (one free + one paid)
2. Merge them back unpredictably as quantities change
3. This affects which items are marked as "free" vs "paid" in our detection logic

Symptoms:

- Same product appears multiple times in cart with different prices
- Free item detection (`discounted_price === 0`) becomes unreliable
- Allocation calculations distribute discounts incorrectly

Debugging Steps:

1. Check `cart.items` array for duplicate product IDs with different prices
2. Verify which items have `discounted_price` set vs not set
3. Test the scenario: start with auto discount → increment quantity → check cart structure
4. Note: This is native Shopify behavior, occurs even without our custom code

Workaround: The system attempts to handle this via signature tracking, but edge cases may persist. Consider this when troubleshooting "inconsistent" behavior during auto discount scenarios.

Diagnostic Commands

```
javascript

// Check current discount state
log('Active discount:', window.RebuyUtils.getActiveDiscount());

// Check cart state
window.RebuyUtils.fetchCart().then(cart => log('Cart:', cart));

// Check 3-for-2 handler state
log('3-for-2 state:', window.ThreeForTwoHandler.getState());

// Check what signatures are being generated
// (Add this temporarily to shouldProcessCart function)
log('Discount signature:', discountSignature);
log('Items signature:', itemsSignature);
```

Configuration

Admin Settings (Liquid)

3-for-2 Codes

liquid

```
window.active3for2codes = {  
  buyXgetYcodes: {{ settings.active_3_for_2_codes | split: ',' | json }}  
};
```

Free Gift Settings

liquid

```
window.discountGiftSettings = {  
  enableFreeGift: {{ settings.new_enable_free_gift | json }},  
  freeGifts: [  
    {  
      minSubtotal: {{ settings.new_free_gift_min_subtotal_1 | times: 100 | json }},  
      discountCodes: {{ settings.new_free_gift_codes_1 | split: ',' | json }},  
      product: {{ settings.new_free_gift_product_1.variants.first.id | json }}  
    }  
    // ... up to 5 gifts supported  
  ]  
};
```

Known Limitations & Edge Cases

1. **Single Item Carts:** Special handling required for exact subtotal matching
2. **Race Conditions:** 300ms timeouts used to coordinate with Rebuy's DOM updates
3. **Cookie Dependencies:** Free gift codes rely on browser cookies
4. **DOM Selector Fragility:** Updates may break if Rebuy changes their HTML structure
5. **0% Gift Code Complexity:** 0% discount codes can be misinterpreted as large fixed amounts, requiring special detection logic that can interfere with other discount calculations
6. **Combination Scenario Fragility:** The more discount types that can combine, the more edge cases emerge - this system handles the known scenarios but new combinations may require additional logic
7. **Shopify Auto-Discount Line Item Splitting:** When auto discounts are active and users increment quantities via line item selectors, Shopify sometimes splits single line items into multiple entries (e.g., one free item + one paid item), then merges them back unpredictably. This is native Shopify behavior but affects our allocation calculations.

Emergency Procedures

If the system breaks completely:

1. **Disable via Admin:** Set `new_enable_free_gift: false` and clear 3-for-2 codes

2. **JavaScript Errors:** Check browser console for specific error messages
3. **Rollback:** Previous working version should be in version control

Contact Information:

- **Primary Developer:** Ian Fleming (i.fleming@bears-with-benefits.com)
- **Stores Affected:** 8 live stores using this system -> DE, IT, FR, UK, CH, NL, ES, US (although in a simpler format due to subscription / lack of required functionality)
- **Critical Path:** Any changes should be tested in staging first

Testing Checklist

Before any changes, test all supported scenarios:

Individual Discount Types

- ☐ Manual 3-for-2 codes
- ☐ Automatic 3-for-2 (Shopify Buy-X-Get-Y)
- ☐ Manual percentage discount codes
- ☐ URL-based percentage discount codes
- ☐ Manual fixed amount discount codes
- ☐ URL-based fixed amount discount codes
- ☐ Manual free gift codes (0% discount)
- ☐ URL-based free gift codes (0% discount)
- ☐ Manual free gift codes (with % discount)
- ☐ URL-based free gift codes (with % discount)

Combination Scenarios (Auto 3-for-2 + Order-level discounts)

- ☐ Manual discount codes + Auto 3-for-2
- ☐ URL discount codes + Auto 3-for-2
- ☐ URL-based free gift codes (with discount) + Auto 3-for-2
- ☐ URL-based free gift codes (0% discount) + Auto 3-for-2
- ☐ Manual free gift codes (with discount) + Auto 3-for-2
- ☐ Manual free gift codes (0% discount) + Auto 3-for-2

Special Complex Scenarios

- ☐ URL-based free gift codes (0% discount) + Manual percentage discount + Auto 3-for-2 (triple combination)

General Function Tests

- ☐ Cart quantity changes during active discounts
- ☐ Discount removal

- ☐ Subtotal calculations
- ☐ Cross-browser compatibility

Last Updated: 28.07.2025 System Version: July 2025 Enhanced