Supported Checkouts

Facebook & Meta Marketing Suite for Magento 2

Overview of all checkout solutions compatible with the Facebook & Meta Marketing Suite module for event tracking.

Overview

The module automatically detects and supports various checkout implementations. Events like `InitiateCheckout`, `AddPaymentInfo`, and `Purchase` are tracked across all supported checkouts.

Checkout	Type	Support
Magento Luma Onepage	Native	Full
Hyvä Checkout	Hyvä	Full
Aheadworks One Page Checkout	Third-party	Full
IWD One Step Checkout	Third-party	Full
Mageplaza One Step Checkout	Third-party	Full
Swissup FireCheckout	Third-party	Full
Klarna Checkout (KCO)	Payment Provider	Full
Klarna Checkout (Legacy)	Payment Provider	Full

Native Magento Checkout

Magento Luma Onepage Checkout

The default Magento checkout. No additional configuration required.

Tracked Events:

- `InitiateCheckout` - When customer enters checkout
- `AddPaymentInfo` - When payment method is selected
- `Purchase` - When order is placed

Hyvä Checkout

Hyvä Checkout

Full support for Hyvä's native checkout solution.

Tracked Events:

- `InitiateCheckout` - When customer enters checkout
- `AddPaymentInfo` - When payment method is selected
- `Purchase` - When order is placed

Note: Works with both Pixel (client-side) and Conversions API (server-side) tracking.

Third-Party Checkouts

Aheadworks One Page Checkout

Single-page checkout by Aheadworks.

Tracked Events:

- `InitiateCheckout` - When checkout page loads
- `AddPaymentInfo` - When payment method is selected
- `Purchase` - When order is placed

IWD One Step Checkout

One-step checkout solution by IWD Agency.

Tracked Events:

- `InitiateCheckout` - When checkout page loads
- `AddPaymentInfo` - When payment method is selected
- `Purchase` - When order is placed

Mageplaza One Step Checkout (OPC)

Popular one-step checkout by Mageplaza.

Tracked Events:

- `InitiateCheckout` - When checkout page loads
- `AddPaymentInfo` - When payment method is selected
- `Purchase` - When order is placed

Swissup FireCheckout

Flexible checkout solution by Swissup.

Tracked Events:

- `InitiateCheckout` - When checkout page loads
- `AddPaymentInfo` - When payment method is selected
- `Purchase` - When order is placed

Payment Provider Checkouts

Klarna Checkout (KCO)

Full Klarna Checkout integration where Klarna handles the entire checkout process.

Tracked Events:

- `InitiateCheckout` - When Klarna checkout loads
- `Purchase` - When order is confirmed by Klarna

Note: `AddPaymentInfo` event may not fire as payment is handled within Klarna's iframe.

Klarna Checkout (Legacy)

Alternative/legacy Klarna Checkout integration for older implementations.

Tracked Events:

- `InitiateCheckout` - When checkout page loads
- `Purchase` - When order is confirmed

How Detection Works

The module uses route detection and layout handles to identify the active checkout:

1. **Route-based detection** - Checks the current URL/route
2. **Layout handle detection** - Identifies checkout-specific layout handles
3. **Observer-based tracking** - Uses Magento events for server-side tracking

This automatic detection means no manual configuration is needed for supported checkouts.

Purchase Event

All supported checkouts use Magento's default checkout success page for the `Purchase` event. This ensures reliable tracking regardless of which checkout solution is used, as long as the order completes through Magento's standard success page flow.

Unsupported Checkouts

If your checkout is not listed above, events may not track correctly. Common issues:

- Custom checkout routes not detected
- Non-standard order placement flow
- AJAX-based checkouts without proper events

Solution: Contact support for custom integration assistance.

Custom Checkout Support

Using a checkout that's not listed? We're happy to add support for additional checkout solutions. Contact us with details about your checkout implementation and we'll evaluate adding native support.

Request custom checkout support: Contact Us

Verifying Checkout Tracking

Using Facebook Pixel Helper

1. Install the Facebook Pixel Helper browser extension
2. Navigate through your checkout process
3. Verify events fire at each step:
    - Checkout page → `InitiateCheckout`
    - Payment selection → `AddPaymentInfo`
    - Order success → `Purchase`

Using Events Manager

1. Go to Facebook Events Manager
2. Select your Pixel
3. Click "Test Events"
4. Complete a test order
5. Verify all events appear in real-time

Using Processing Queue (CAPI)

1. Go to **Marketing → Magmodules Meta → Processing Queue**
2. Complete a test order
3. Verify `Purchase` event appears in queue
4. Wait for cron (or process manually)
5. Confirm status changes to "success"

Troubleshooting

Events Not Firing

1. **Check module is enabled** - Verify Pixel/CAPI is enabled in configuration
2. **Check event selection** - Ensure checkout events are selected
3. **Clear cache** - Full page cache may affect JavaScript tracking
4. **Check for JS errors** - Browser console may show conflicts

Purchase Event Missing

1. **Verify success page** - Some checkouts redirect externally
2. **Check CAPI** - Server-side tracking may capture what Pixel misses
3. **Review order flow** - Non-standard flows may need custom integration

Need More Help?

Documentation:

- All Help Articles - Complete documentation overview

Support:

- Contact Support - Get help from our team

For a complete overview of features and configuration options, see the Facebook & Meta Marketing Suite extension on magmodules.eu

All articles for Facebook & Meta Marketing Suite

Installation

1	Installation using Composer (recommended)
2	Installation using the Adobe Marketplace
3	Install through FTP and SSH

Configuration

1	Quick Start Guide
2	Feed Configuration
3	Pixel & Conversions API Configuration

Troubleshooting

1	Troubleshooting

Grids

1	Data Feed Logs
2	Processing Queue

Background

1	CLI Commands
2	**Supported Checkouts (current)**