

Microbite CAPI setup guide

Connect WooCommerce Purchase events to Meta, add browser tracking through Google Tag Manager, and optionally measure ecommerce in GA4. Instructions cover the v1.8.6 interface and the v1.8.7 wording cleanup. Fresh Credentials and Setup Assistant images are included; captions identify remaining earlier demo captures. A connection test confirms PageView acceptance; an order test verifies Purchase separately.

Before you begin

You need WordPress administrator access, an active WooCommerce store, your Microbite CAPI ZIP and licence, access to the intended Meta dataset, and a Google Tag Manager web container. Use a separate staging store and Meta test dataset for demonstrations.

Install and activate

In WordPress, open Plugins > Add Plugin > Upload Plugin. Choose the customer ZIP, install it and activate Microbite CAPI. Open WooCommerce > Manage License and activate your licence. Use the customer ZIP for v1.8.6. Fresh Credentials and Setup Assistant captures show v1.8.6; other captions identify earlier demo interfaces.

Use Setup Assistant

Open WooCommerce > Microbite CAPI > Setup Assistant. Follow the seven stages, their actions and expected results. Your selected stage and question resume within the browser session. Review the expandable message before copying it to your own ChatGPT account; this is a copy-and-paste handoff, not a live website connection.

Identifier	Purpose	Example format
Pixel or dataset ID	Same Meta destination in CAPI and browser pixel	Digits
CAPI access token	Saved privately in Credentials	Secret value
Test Event Code	Routes server tests to Meta Test Events	TEST...
GTM container ID	Loads the website tag container	GTM-...
GA4 Measurement ID	Optional Analytics web stream	G-...

Use your own values

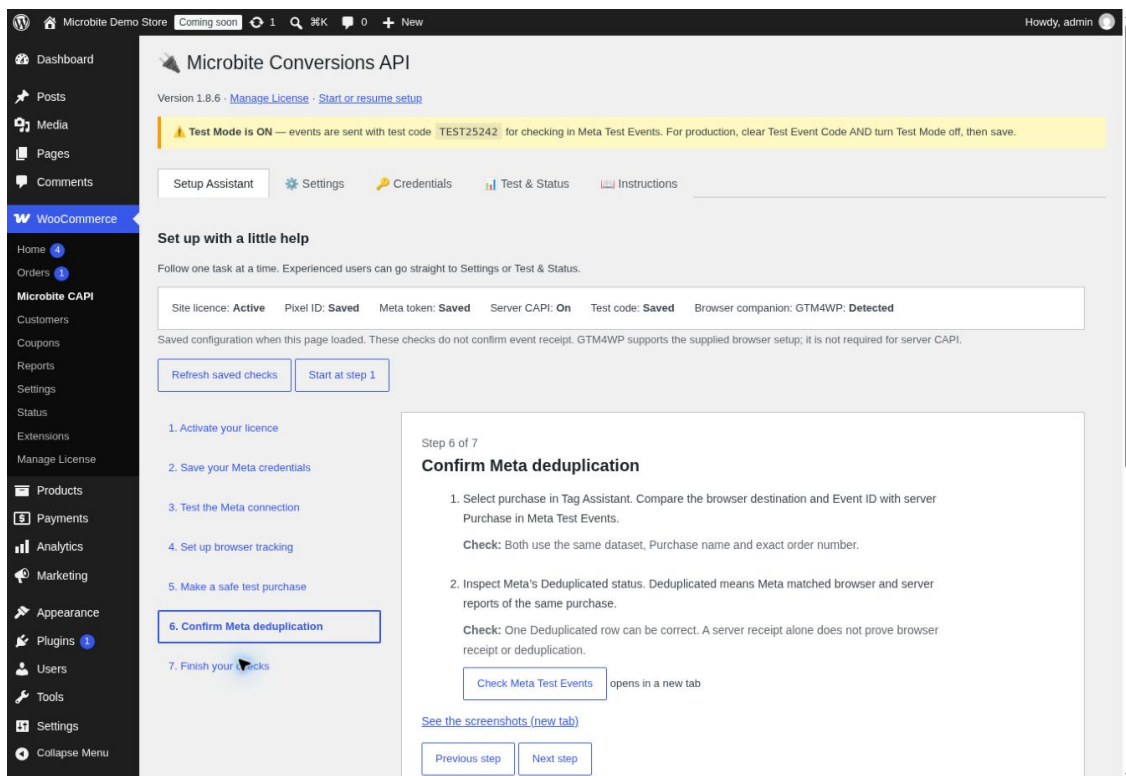
Screenshots show the isolated demo.microbite.com.au setup. Its IDs, test code and order numbers are examples, not values to copy into a customer installation. All order details shown are synthetic; no real payment or dispatch was involved.

Guide contents

2 Get a Meta token · 3 Save credentials · 4 Test the connection · 5 Configure GTM4WP · 6 Import and publish GTM · 7 Place a test order · 8 Verify Meta Purchase · 9 Verify GA4 · 10 Troubleshoot and go live.

AI-guided setup with your preferred assistant

Use ChatGPT, Claude or another AI assistant to work through setup one task at a time. Complete prompts are available on the Microbite setup guide; no file download is required.



Microbite CAPI v1.8.6 Setup Assistant, captured 4 October 2026. Seven built-in stages guide setup. Saved checks describe local configuration; check event receipt separately.

1. Choose a website prompt. Open the illustrated setup guide and expand the task you need help with, such as saving credentials, understanding Valid, publishing GTM or checking deduplication.

2. Copy and paste. Review and copy the complete prompt into ChatGPT, Claude or your preferred assistant. Each prompt contains the Microbite CAPI context and works in a new chat.

3. Check one result at a time. Answer the questions about your version and store environment, carry out the next action yourself and report what you see. Confirm diagnostic PageView, server Purchase, browser Purchase, deduplication and optional GA4 receipt separately.

Optional configuration summary. Inside Setup Assistant, choose a help question, review the full message and click Copy setup message. Paste it into your preferred assistant. The plugin's Open ChatGPT button opens ChatGPT.

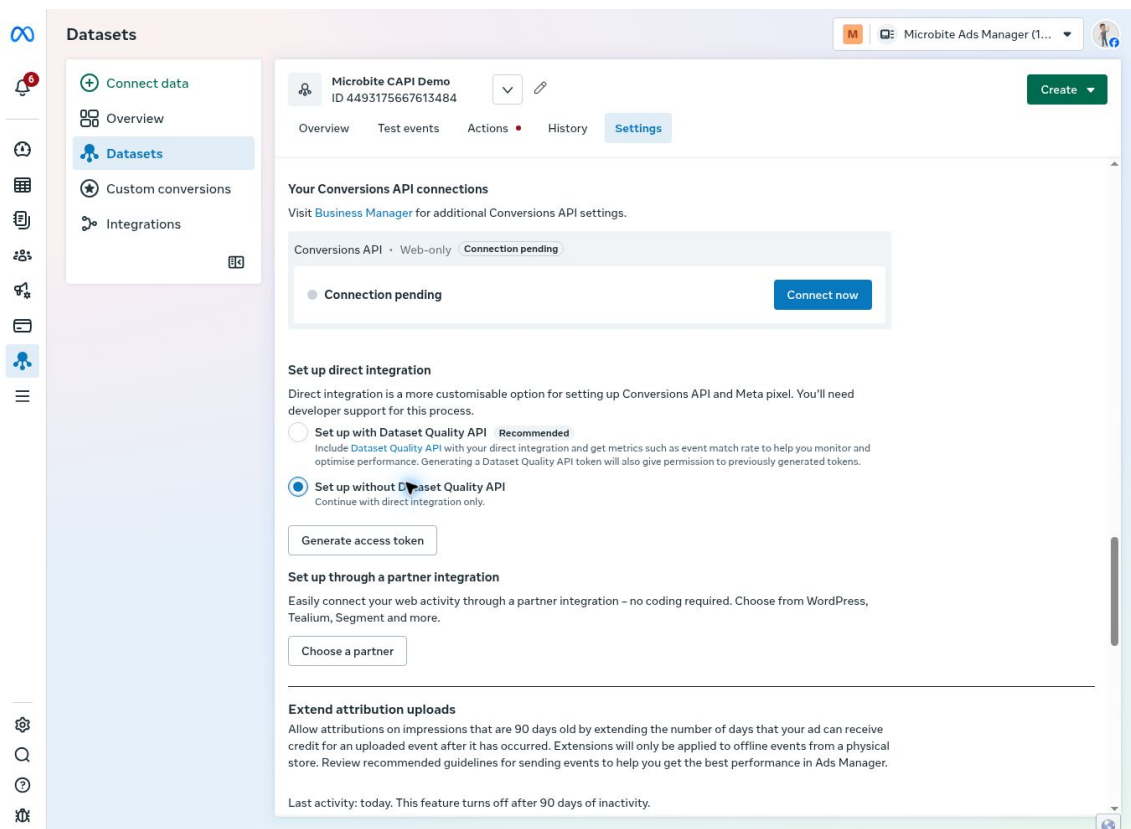
Under your control. Nothing is sent automatically, and copying a prompt does not give an assistant access to your site. No Microbite AI service or AI API key is used. Keep secrets and customer information out of added messages or screenshots.

Optional downloads. A customer Markdown guide and text version provide offline copies. The website prompts are the main way to start.

Browse the setup prompts:
shop.microbite.com.au/microbite-capi-setup-guide/

Get your Meta access token

- 1. Select the correct dataset.** Open Meta Events Manager, sign in and select the business portfolio and the dataset for the website. Copy its Pixel or dataset ID. This must match the Meta Pixel ID used by the browser tags.
- 2. Open Settings.** Find Conversions API and Set up direct integration. For a direct event submission setup, the captured interface provides Set up without Dataset Quality API, then Generate access token.
- 3. Generate and store the token privately.** Complete Meta's prompts and keep the token in a password manager. Paste it directly into the plugin Credentials screen. Never include it in a screenshot, public page or GTM variable.



Meta Settings before token generation, captured 2 October 2026. No token is displayed. Controls may vary by business account.

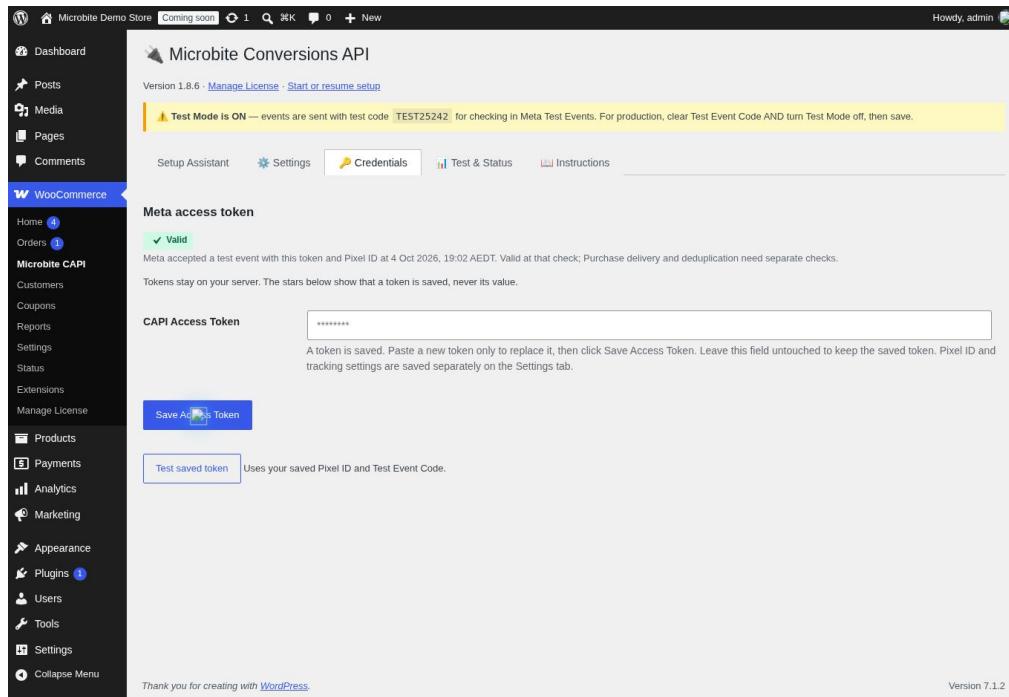
If you use a system user

Open Meta Business Settings, select your business and look for Users > System users. An existing system user token can also be used when its app and assigned dataset access permit event submission. Check the app, dataset and token permissions together. A returning setup screen does not prove that a token failed, and a permissions error alone does not prove that Full control is required.

Save the token in Credentials

1. Save the destination. Open WooCommerce > Microbite CAPI > Settings. Enter your own Pixel ID and click Save Settings. General settings and token saving use separate tabs.

2. Save the secret. Open Credentials. Paste the Meta token into CAPI Access Token and click Save Access Token.



Microbite CAPI v1.8.6 Credentials, captured 4 October 2026. The saved-token field shows eight fixed stars and a Valid badge dated 19:02 AEDT after an accepted diagnostic PageView. No token value is displayed.

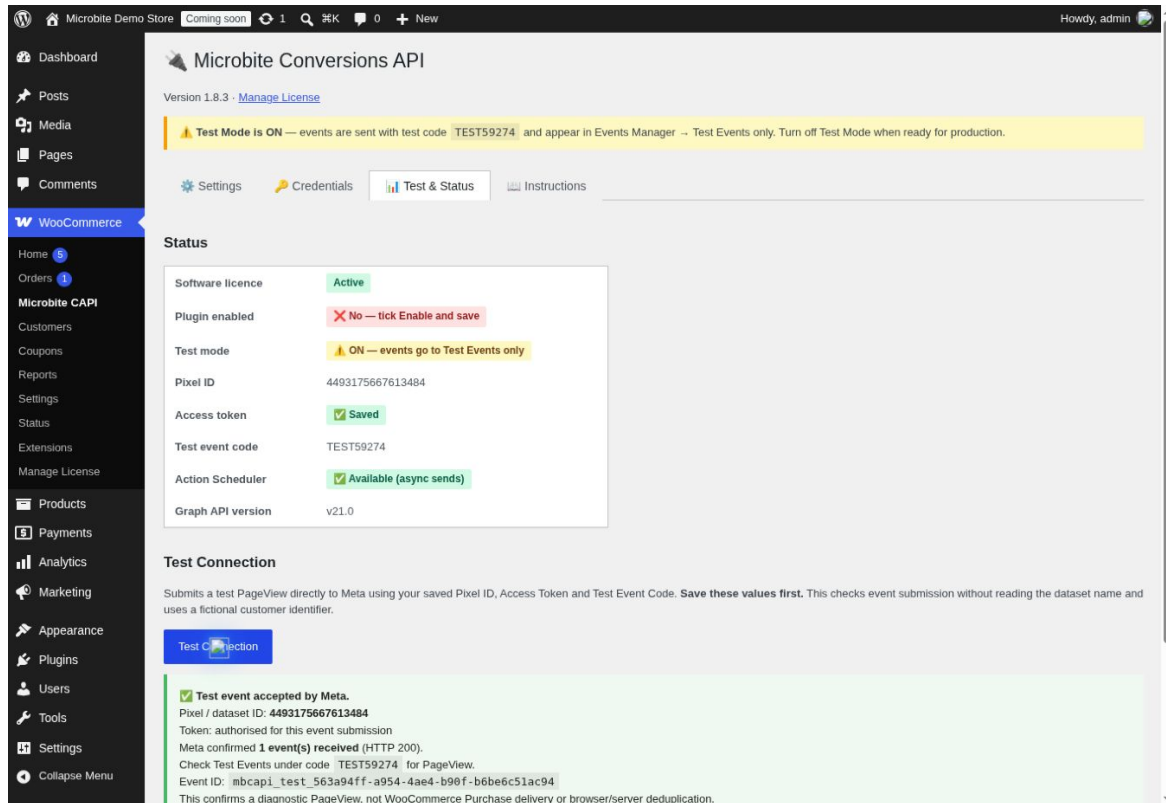
Read the mask and status. In v1.8.6, a saved token displays *****. These are placeholder stars; its value and length are never revealed. Leave the field untouched to keep the token. Paste a new token only to replace it. Saved - verification needed means storage is confirmed, but Meta acceptance needs a fresh check.

Check the saved token. Click Test saved token to open Test & Status. A successful diagnostic adds a dated Valid badge. This evidence expires after 24 hours and does not apply if the token or Pixel ID changes. Check failed can mean transport or permission problems; inspect the diagnostic result before replacing the token.

Test the connection before orders

1. Configure a test. In Meta Events Manager > Test events, copy the code for your selected dataset. In the plugin Settings, enable Test Mode, enter that Test Event Code, enable CAPI for the later order test, and save. Enable Debug Log temporarily to inspect test sends.

2. Run Test Connection. Open Test & Status, confirm the licence is active and the Pixel ID and token are saved, then click Test Connection. Test Connection submits a diagnostic PageView directly to Meta. Saving or viewing the token does not run an automatic test.



Earlier v1.8.3 demo: Meta accepted a diagnostic PageView with HTTP 200, before Enable CAPI was switched on for order tests. v1.8.6 also refreshes the dated token badge.

What success proves

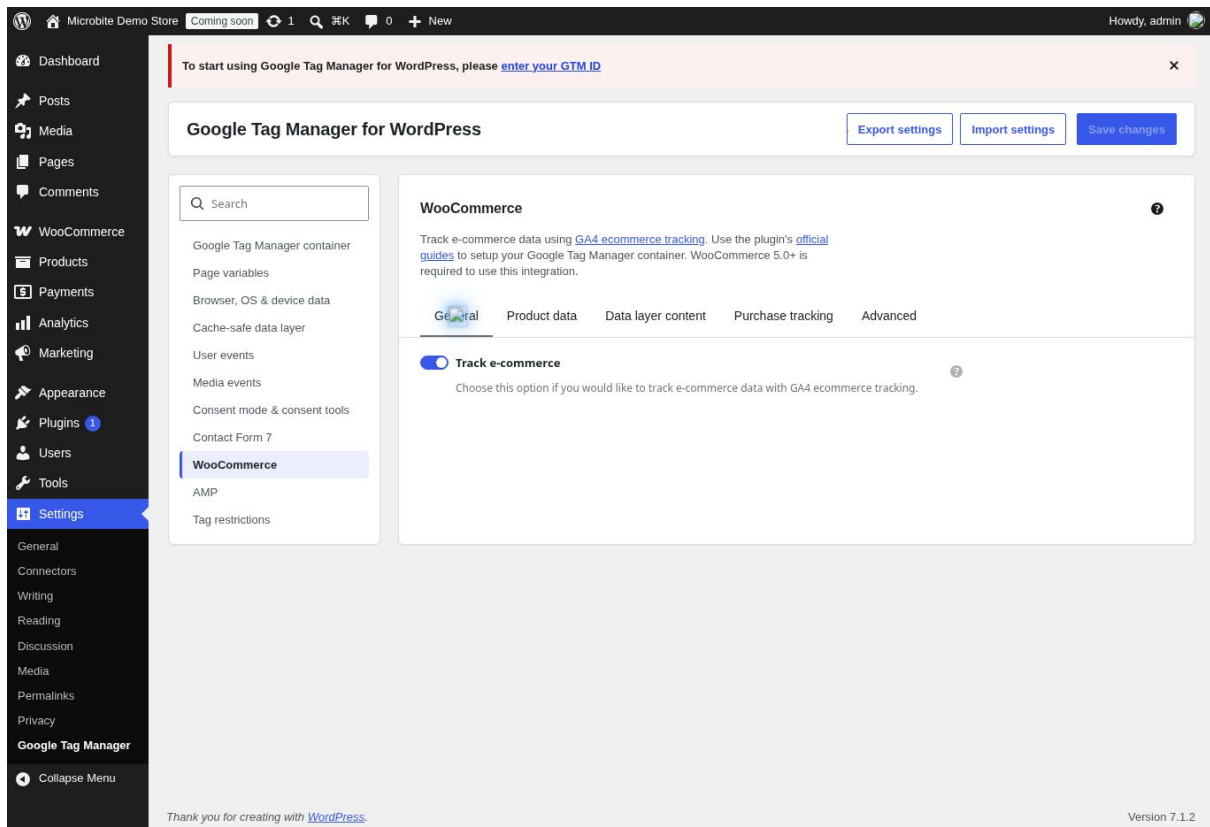
HTTP 200 with events_received greater than zero confirms acceptance of this event submission. Check the matching PageView in Meta Test Events. It does not prove WooCommerce Purchase delivery, browser Pixel tracking or browser/server deduplication.

Read Valid in context. The badge means Meta accepted a test submission with the current token and Pixel ID at the shown time. It is not a continuing validity guarantee or proof of Purchase delivery. Continue with browser setup and an order test.

Configure the WooCommerce data layer

1. Install GTM4WP. Activate Google Tag Manager for WordPress from the WordPress plugin directory. In Settings > Google Tag Manager, add your site's GTM container ID and save. Avoid installing the same container twice through another theme, plugin or snippet.

2. Enable ecommerce. In the GTM4WP WooCommerce section, switch Track e-commerce on. This supplies the WooCommerce ecommerce data layer for the provided browser workflow. Server-side CAPI sending works independently of GTM4WP.



GTM4WP 2.0.5 WooCommerce integration with Track e-commerce enabled. The setup banner in this captured view is not the final verification of the container ID.

3. Check Purchase tracking. Leave the transaction ID prefix blank for this configuration. Use the order statuses needed by your checkout; the demo used Processing, On hold and Completed. Compare the actual browser transaction ID with the CAPI order number, especially if you use custom order numbering.

Verify in GTM Preview. Browse a product, add it to the cart, open checkout and complete a test order. Look for view_item, add_to_cart, begin_checkout and purchase, including the transaction ID, value, currency and items. Keep optional customer and order details out of the data layer unless your measurement setup requires them.

Import and publish Meta with GA4

1. Export a backup. In GTM, export the existing container before importing. Download the supplied JSON from the plugin Instructions tab or use the reviewed package prepared for your site. The Universal bundle has more features than the demo's seven-tag setup.

2. Merge into an existing container. In Admin > Import Container, choose the file and workspace, then Merge. Overwrite conflicting tags, triggers and variables affects matching resources; Overwrite the entire container replaces the container. Check every conflict and the preview for unintended deletions. Use a whole-container overwrite only for an intentionally empty container.

3. Set the correct IDs and event ID. Update the Meta Pixel ID and optional GA4 ID constants. In the Meta Purchase tag, map Event ID to the data-layer variable for ecommerce.transaction_id. The variable name can differ between bundles. It must equal the CAPI order number for that purchase. Pause unused Google Ads or other destination tags.

Version ID ↓	Status	Name	Created	Published	Published By
4	Live, Latest	Demo GA4 and Meta tracking together	10/03/2026	10/03/2026	info@solarstreetlightsaustralia.com.au
3		Replace Meta Pixel tracking 10/03/2026 with GA4 tracking	10/03/2026	10/03/2026	info@solarstreetlightsaustralia.com.au
2		Demo Meta tracking	10/03/2026	10/03/2026	info@solarstreetlightsaustralia.com.au
1		Empty Container	10/03/2026	10/03/2026	info@solarstreetlightsaustralia.com.au

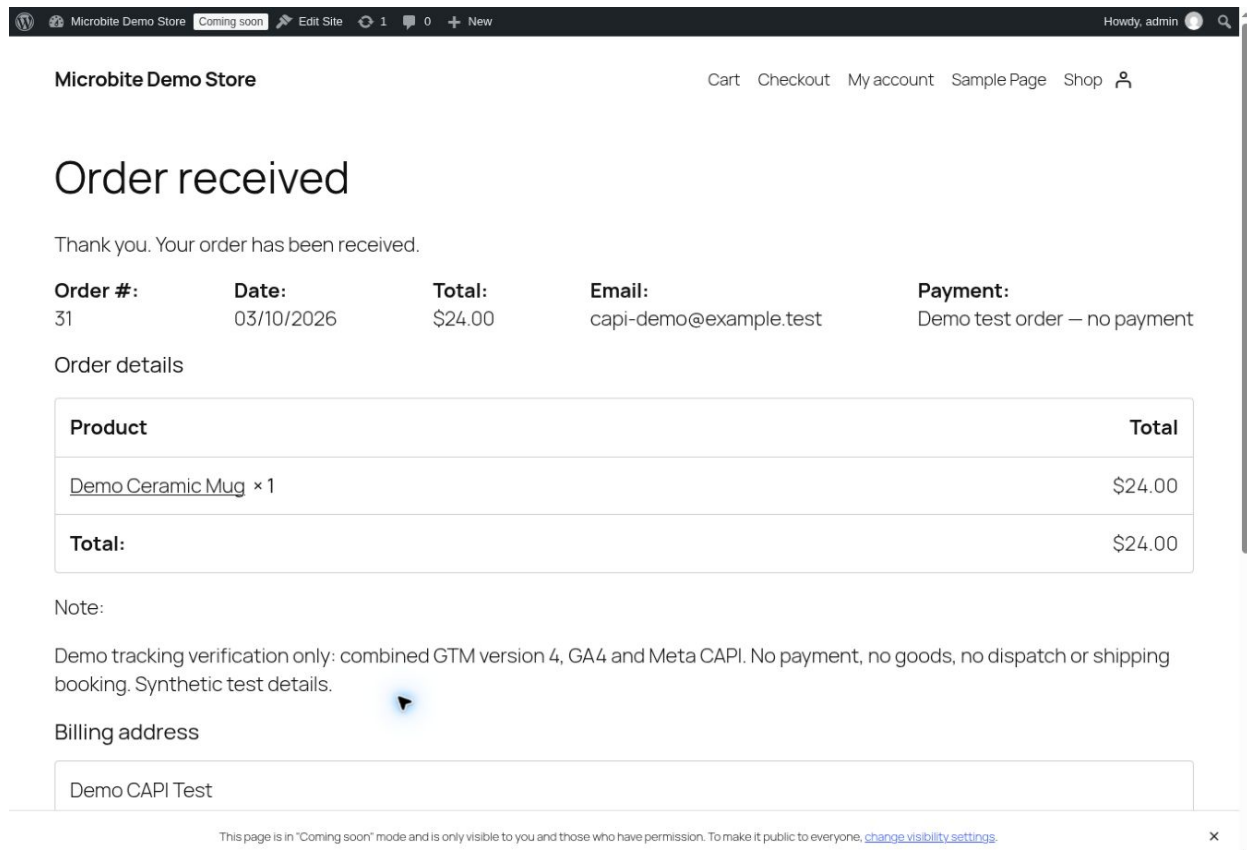
Published demo version 4 contains Meta and GA4 together: seven tags, seven triggers and five custom variables. This is a demo package, not the full Universal container.

4. Preview, then publish. Check the funnel, destination IDs and tag firing before Submit > Publish. Confirm the intended version is Live. An imported or saved workspace alone does not change the tags served to normal visitors.

Place a safe test order

1. Keep the test views open. Leave Meta Test Events open for the selected dataset. Start GTM Preview on the same store to inspect the browser Purchase tag. Confirm the plugin has Enable CAPI and Test Mode on, with the current code saved.

2. Complete a fresh checkout. Use a staging store, synthetic details and a test payment method. Do not rely on refreshing an old order confirmation page; Purchase duplicate guards may stop a resend. Record the order number, total and currency.



Demo order 31 contains one mug for A\$24 using a no-payment demonstration method. This screenshot contains synthetic contact details only.

3. Check the server receipt. Open WooCommerce > Status > Logs, choose the mbcapi source for the test date, and locate the order. Sends use Action Scheduler, so they can arrive after checkout. The demo receipt was:

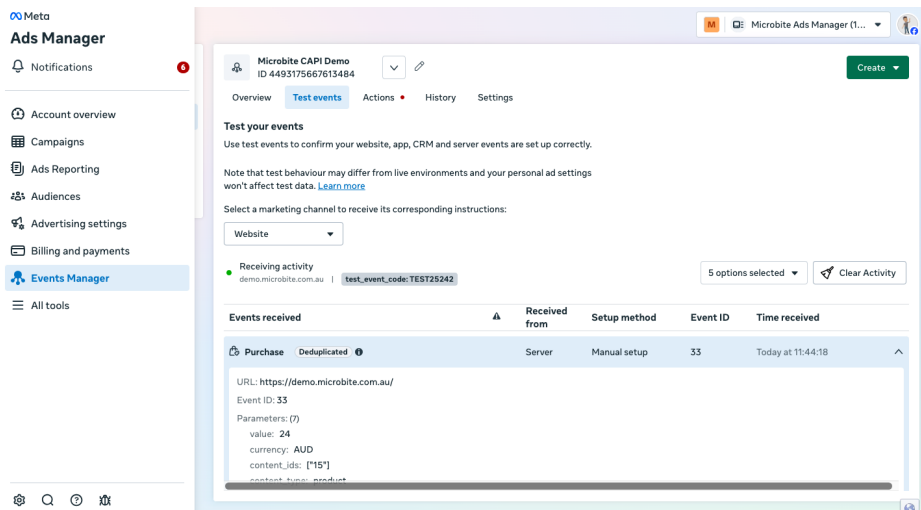
Order 31 Purchase event_id=31 -> HTTP 200; events_received: 1

What this proves

The order was created and Meta accepted its server Purchase. The checkout page and server log alone do not prove that a browser Purchase arrived or that Meta deduplicated the two events.

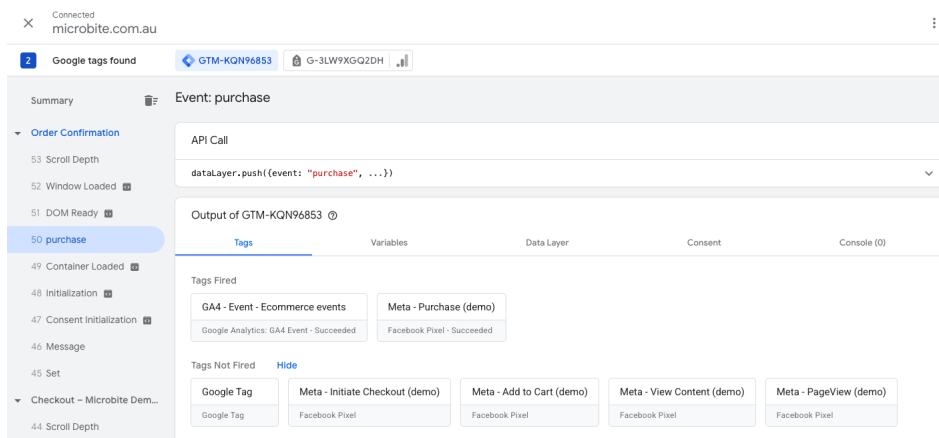
Verify Meta Purchase and deduplication

1. Confirm Server Purchase. Select the correct dataset in Meta Test Events. Check the Purchase event ID, value and currency. The completed demo test is order 33 for A\$24 AUD.



Meta Test Events marks Purchase 33 Deduplicated, with value 24 and currency AUD. Original demo screenshot supplied 4 October 2026.

2. Check the browser tag. Select purchase in GTM Preview. Confirm Meta Purchase succeeds, uses the same Pixel ID and sends an Event ID identical to the CAPI order number. For this demo, the resolved Pixel ID was 4493175667613484 and the Event ID was 33.



The selected purchase event shows both Meta Purchase and GA4 ecommerce tags Succeeded. Tag firing is checked separately from destination receipt.

3. Read the deduplication result. Matching settings alone do not prove receipt. In this test, Meta explicitly reported Deduplicated for order 33. One visible entry does not mean failure, and two separately Processed rows are not required. A repeat purchase by the same buyer should have a new event ID; orders 30 and 33 are different purchases.

Demo result

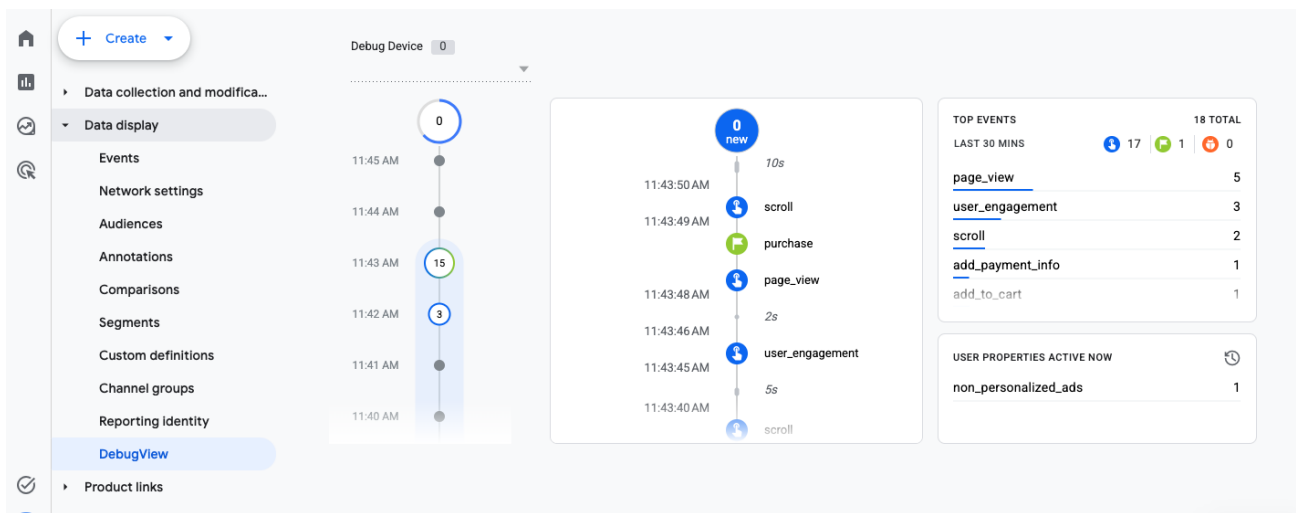
Browser Purchase tag succeeded with the matching ID; Meta reported Purchase 33 Deduplicated. Use your own IDs when repeating these checks on another store.

Add and verify optional GA4 tracking

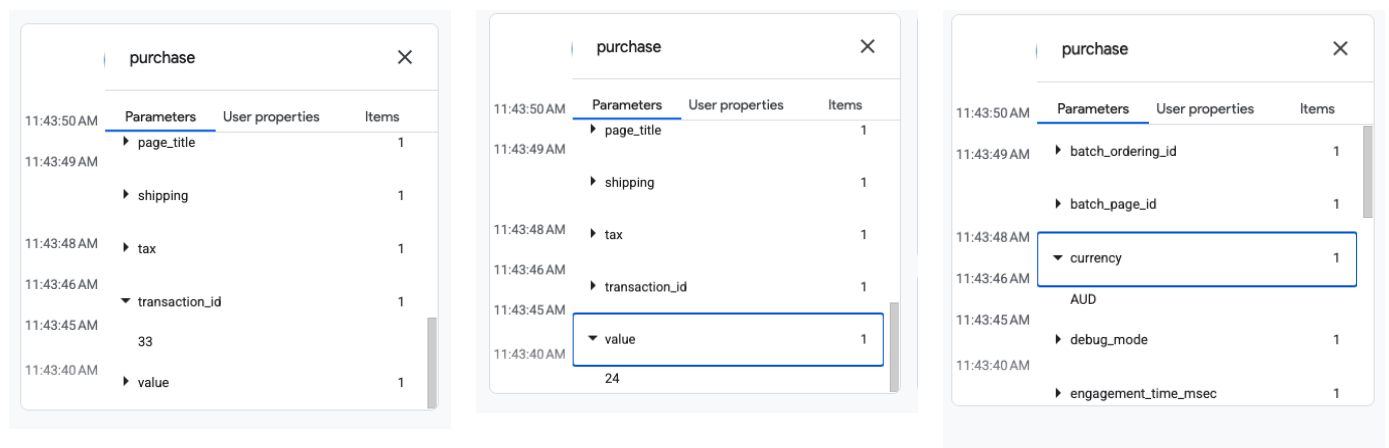
1. Use your web stream. In GA4 Admin > Data streams, copy the G-... Measurement ID. Configure one Google tag for that destination in GTM and review existing direct or plugin installs to avoid duplicate collection.

2. Send ecommerce events. Configure the GA4 Event tag with Event Name {{Event}}, Send Ecommerce data enabled and Data Layer as its source. Include purchase with transaction_id, value, currency and items. Merge alongside Meta, preview and publish.

3. Check GA4 receipt. Open Admin > Data display > DebugView with GTM Preview connected. Select purchase in the timeline and expand its parameters. GTM tag success and GA4 receipt are separate checks.



GA4 DebugView shows the received purchase event in the demo timeline.



transaction_id: 33

value: 24

currency: AUD

Demo result

GA4 received purchase with transaction_id 33, value 24 and currency AUD. The demo Measurement ID is G-3LW9XGQ2DH. These screenshots confirm the three expanded parameters; check items separately in your own test. GA4 transaction deduplication is separate from Meta browser/server deduplication.

Troubleshoot and prepare production

Symptom	Check
Token field looks blank	v1.8.6 shows stars when saved; older UI clears entry. Check plugin version and saved status.
Missing Permission	Correct dataset, token app and assigned assets; validate event submission before widening access.
PageView succeeds but no Purchase	Enable CAPI, licence, order status, mbcapi log and Scheduled Actions.
Only Server Purchase appears	GTM live version, Pixel ID, browser trigger, consent state, blockers and data layer.
Two Purchase signals do not match	Same Purchase name, destination and exact Event ID; custom order-number prefixes.
GA4 has page views only	purchase data-layer event, GA4 Event tag and received ecommerce parameters.

Before going live

Confirm a fresh WooCommerce server Purchase, a matching browser Purchase and Meta deduplication. If GA4 is installed, verify its received Purchase separately. In the plugin, clear Test Event Code **and** switch Test Mode off, then save and confirm the persisted values. A non-empty test code is still included by the Purchase sender even if the Test Mode checkbox is off. Clear the code for production; keep a dedicated demo in test mode.

End temporary debugging after verification. Review privacy disclosures and your consent implementation; the plugin does not automatically provide consent management. Customer identifiers are hashed for matching, while fields such as IP address and user agent may be sent without hashing. Server-side CAPI Purchase sending is for WooCommerce orders; SureCart requires its own integration.

Demo check at preparation	Evidence
Saved token	Saved with fixed mask; diagnostic Valid checked 4 Oct 2026, 19:02 AEDT
Diagnostic PageView	Meta accepted one event, HTTP 200
Server WooCommerce Purchase	Orders 29, 30, 31 and 33 received
Combined Meta and GA4 container	GTM version 4 Live
Browser Purchase and Meta deduplication	Tag succeeded; Meta order 33 Deduplicated
GA4 Purchase receipt	transaction_id 33; value 24; currency AUD

Further reference

Google Tag Manager container export and import
 Google Analytics ecommerce with Tag Manager
 Meta Events Manager Conversions API setup help