

PRESTASHOP MODULE

WebP Optimizer & Google Commerce Tools

Installation, configuration and feature guide

alteralgo

Version 0.6.6 · October 2026

Contents

- 1. Start here
- 2. Installation and upgrades
- 3. Google Merchant
- 4. Product snippets
- 5. XML Sitemap
- 6. WebP
- 7. Google Search Console Fixer
- 8. Google Tag Manager and pixels
- 9. Security, translation and multistore
- 10. Maintenance and support

1. Start here

This guide covers the complete **weboptimizer** module: Google Merchant, Product snippets, XML Sitemap, WebP, Google Search Console Fixer and Google Tag Manager. Open its main Back Office sidebar entry, or **Modules** → **Module Manager** → **Configure**. These six tabs share one installation but have separate settings and enable switches.

Designed for PrestaShop **1.7.8 through 9.2**, using the PHP version required by your PrestaShop release. Local fixtures and browser tests are not certification of every shop, theme, hosting stack or minor version. Test updates in staging before production.

English, French, Spanish, Italian, German, Polish and Portuguese interface catalogs are included. Back Office uses the employee language; storefront text uses the shop language. This PDF is the English guide. Vendor-supplied API errors may retain the vendor's language.

Tab	Purpose	Dependency
Google Merchant	Filtered product feed, preview and protected URL	Merchant Center for scheduled imports; optional OAuth/API for remote diagnostics
Product snippets	Product/ProductGroup JSON-LD and catalog validation	Google decides rich-result appearance
XML Sitemap	Active product and optional category URLs	Submit the URL in Search Console
WebP	Generate and serve WebP companions to JPEG/PNG	GD/Imagick WebP support and web-server delivery configuration
Google Search Console Fixer	Import reports, inspect URLs, manage repairs and robots.txt	Search Console CSV export; access to your storefront
Google Tag Manager	Consent-aware ecommerce analytics and advertising	Your GA4, Ads, Meta or published GTM destination

Feeds, sitemaps and JSON-LD are independent outputs. Enabling one does not enable the others. No feature guarantees Google approval, indexing, rankings or rich-result display.

2. Installation and upgrades

Requirements

- A supported PrestaShop shop, writable module/cache folders and sufficient disk space for original images plus WebP companions.
- GD with WebP support or Imagick with the WebP format for conversion. Other tabs do not require image conversion.
- XML/DOM support for XML and page inspection; HTTPS, cURL and OpenSSL for Google authentication and remote checks.
- Appropriate employee permissions. Tracking and sensitive Google/robots actions require a SuperAdmin and one selected shop.
- Backups of the database, module, robots.txt and .htaccess before installation or SEO/server changes.

Install or update

1. Back up the shop and test the package in staging.
2. Upload the ZIP through **Module Manager → Upload a module**. It contains one top-level `webpoptimizer/` directory.
3. For manual deployment, copy that directory into `modules/`, then run PrestaShop's install/upgrade action. Do not rename it.
4. Confirm version **0.6.6** in Configure and select the intended shop.
5. Clear **Advanced Parameters → Performance → Cache**, hard-refresh the browser and purge CDN asset caches if applicable.
6. Recheck enabled features on the storefront. Upgrade the existing installation; uninstalling to update files can delete module data.

The package includes its runtime loader; Composer is not required on the server. Installation does not run a full catalog image conversion.

Changes in this release

Version 0.6.6 corrects Marketplace PHP file-comment spacing and adds explicit **Direct** GA4, Google Ads and Meta delivery. Older saved tracking configurations remain in **GTM** mode to avoid silently duplicating published tags. Select Direct and save if that is the desired setup. Tracking is disabled by default on a new installation. The Product snippets preview uses an eligible real product, not a demonstration item. This guide now covers all six tabs.

3. Google Merchant

Step 1 — Connect or skip

The numbered steps appear on the left; the current form appears on the right. Connect with Google for account/API access, or choose **Continue without Google** to create a scheduled XML feed. OAuth is not required to generate the feed or submit its URL in Merchant Center.

For OAuth, use a Google user with access to the Merchant account, authorize access, then select/save the intended account. Use HTTPS. OAuth consent alone is not sufficient for Merchant API: Google's project enablement and developer registration are also required. A project-not-registered error must be resolved with Google; OAuth cannot bypass it. Allow the delay stated in Google's error before retrying.

If accounts are missing, check the Google user's permissions, selected shop and OAuth project configuration. Do not put secrets or private tokens in support screenshots.

Step 2 — Filter your catalog

Choose feed language, currency, country and Google product-category mapping. Review the available filters:

- Included categories and product IDs, using the selection helpers.
- Excluded categories and the descendant-category option. A product assigned to an excluded category can be omitted even if it also belongs to an included one.
- Brand suggestions or comma-separated catalog manufacturer names.
- Minimum/maximum price, out-of-stock inclusion and identifier-only filtering.
- One item per combination, with related items linked by `item_group_id`.

Click **Next** to save each step and wait for confirmation. Completed steps keep their numbers. There is no separate general Save merchant feed settings button.

Step 3 — Shipping

Set the shipping destination and rate when including shipping in the feed. Enter zero only for genuinely free delivery. Ensure currency and amounts match the storefront and Merchant Center rules. A configured flat rate does not simulate every carrier, postcode and customer group. Feed shipping, Product snippets shipping and PrestaShop carrier settings are separate; keep them consistent.

Step 4 — Preview and publish

The final step has the feed switch and protected URL. The banner also displays enabled/disabled status. **Preview and validate feed** enables the feed and checks the first 20 eligible products; it does not certify the entire catalog or rewrite product records.

Resolve warnings, copy the complete URL including the token, and create a scheduled data source in Merchant Center. Google must be able to fetch it over HTTPS. Maintenance mode, HTTP authentication, firewalls and bot protection can block imports. Keep the token confidential; after rotating it, update the scheduled source.

Identifiers and diagnostics

GTINs come from catalog EAN/UPC values and are validated/normalised. The module must not invent identifiers for products without legitimate GTINs. Correct product/combination codes at source; use brand/MPN and identifier handling only when appropriate for the actual product.

Feed items include product URLs/images, title/description, price/currency, condition, stock and supported identifier/variant fields. Local preview warnings differ from Google's remote diagnostics. Remote diagnostics require working OAuth/API access and processed products in the selected account. Review product-level issues and paginate through the results. An empty page of issues is not proof that every item is approved. Google's processing and policy decisions can lag behind feed updates.

4. Product snippets

This tab generates **JSON-LD structured data**, not a visual rich-text editor. Eligible pages receive Product or ProductGroup data, offers and configured policies. Values must reflect the actual product and visible customer information.

Setup and readiness

Review the fixed mappings: name, short description with description fallback, product images, manufacturer/brand, reference/SKU, EAN/UPC and MPN. Products with combinations can use ProductGroup and individual variants. Configure exclusions and output options, then save JSON-LD settings.

The checklist tracks product setup, offers/policies, reviews and validation. Saving sections updates their completion state. Finishing product and offers alone does not complete validation. Decide explicitly whether reviews apply.

Offers, shipping costs and returns

Offers reflect shop/language/currency context and actual price/availability. Enable shipping details and supply country, **shipping cost**, handling and transit times to emit **OfferShippingDetails**. Do not enter a zero placeholder for paid delivery. Configure return-policy country, window and other supported settings only when consistent with your published policy. These fields do not configure checkout carriers or customer-service rules.

Use supported approved product-review data when available. Products without eligible reviews should have no artificial rating output. Test integration with your installed review module.

Validate a real product

1. Save product, offer/shipping and review settings.
2. Enter an active product URL from the current shop in Validation. Genuine encoded product URLs are supported; external/private-network targets and inactive/excluded products are rejected.
3. Inspect JSON-LD and warnings. **Google product result** displays the validated product, or an eligible catalog product when available. With no eligible product it shows an empty state, not a dummy item.

4. Check name, price, currency, availability and any rating against the product page.
5. Run **Scan catalog** for paginated checks with progress. Correct missing images, identifiers and prices, then enable output.
6. Inspect live page source and Google's Rich Results Test. Avoid duplicate/conflicting product schema from the theme or another module.

The result preview is illustrative; Google controls its search display. Local validation can inspect generated data before publication, while a live-page check may still show cached or third-party markup. Clear relevant caches after changes.

5. XML Sitemap

The sitemap has its own tab, separate from WebP and Merchant feeds.

1. Turn on the sitemap switch. Settings save automatically; there is no Save sitemap button.
2. Select whether to include categories alongside active product URLs. Wait for the save confirmation.
3. Open the generated URL and confirm an XML response. Preserve its token and shop/language parameters.
4. Submit it to the corresponding Search Console property. Google needs network access to the complete URL without a Back Office session.

Disabled sitemaps and invalid tokens are denied. For errors, check the switch, exact URL/token, shop domain/SSL settings, rewrites, maintenance/firewall restrictions and server logs. This output lists catalog URLs, not WebP conversions or Merchant items. Avoid conflicting sitemap sources. Submission helps discovery but does not force indexing.

6. WebP

Conversion and supported images

WebP companions are generated for supported shop-owned JPEG/PNG files; originals are preserved. Temporary outputs are validated before replacement. Product, category, manufacturer and supplier image types come from PrestaShop. Bounded scanning also covers supported banner, CMS and active-theme locations, not arbitrary folders or every other module's image directory.

Review GD/Imagick support, engine, quality, batch size, retries, logging and image families. Quality defaults to 80. Test visual quality and file size before regenerating a large catalog.

Action	Effect
Scan image coverage	Discover supported sources and report existing, missing or stale coverage
Generate missing WebP	Queue missing/stale companions and process bounded batches
Regenerate all WebP	Rebuild companions using current settings; uses more CPU/disk
Retry failed	Retry queue items after correcting their cause
Clean orphaned WebP	Remove recognised companions whose sources no longer exist
Delete generated WebP	Remove generated companions, not original JPEG/PNG files

The queue is database-backed and resumable. Use progress/statistics and recent results to monitor it. Large catalogs benefit from cron. Product-edit tools can generate/test a single product. Image/thumbnail hooks support automatic maintenance; scan after imports that bypass those hooks.

Cleanup/deletion is a real filesystem operation. Back up generated files if you need immediate rollback. Shared physical multistore images may affect several shops.

Background jobs

Enable authenticated cron and copy its exact secret URL. Example schedule with placeholders:

```
* / 5 * * * * curl -fsS 'https://example.com/module/webpoptimizer/cron?token=YOUR_TOKEN' >/dev/null
```

Invalid tokens or disabled cron are rejected. Repeated calls resume bounded queue work. Keep tokens private and rotate exposed values.

On supported PrestaShop 8/9 console installations, run from the shop root:

```
php bin/console webp:generate --missing
php bin/console webp:regenerate
php bin/console webp:cleanup
php bin/console webp:status
```

Delivery and fallback

Generating a file is separate from serving it. Use supported Apache/LiteSpeed rewrite integration or have your host apply the module's Nginx snippet. Back up .htaccess first. Proxy/CDN caching must distinguish WebP-capable requests with appropriate Accept/Vary handling.

Run the delivery test and inspect an image request in browser developer tools: expect **Content-Type: image/webp** for a capable request and original-format fallback where appropriate. A **.jpg** URL can return WebP through content negotiation. A **?t=...** cache suffix alone does not prove delivery. Test the public domain after clearing image/CDN caches.

Conversion failures: check engine support, source validity, file permissions, memory/disk and queue logs. Existing WebP files served as JPEG: investigate rewrites, CDN and server configuration instead of repeatedly regenerating.

7. Google Search Console Fixer

This native workspace imports reports and investigates technical SEO. It does not automatically connect to Search Console or control Google's indexing decisions. Select one shop and start with a supported Search Console CSV export.

Import and inspect

Import URL/issues CSV data, choosing a default issue where needed. Imports are bounded to 10,000 rows; split larger exports. Review issue groups, URL details, rules and activity. Scan to inspect status codes, redirects, canonicals, robots/noindex restrictions and reachable destinations within permitted shop origins.

Interpret findings first. Redirect sources, alternate canonicals and private-account pages may be correctly excluded. "Crawled, currently not indexed" may call for content/internal-link improvements rather than a redirect.

Repairs and verification

Review suggested destinations and apply supported redirect/canonical repairs only for genuine page relationships. Unsafe destinations are rejected. If enabling automatic repair, review the activity and verify storefront results; not every issue can safely be auto-fixed.

Redirects can be synchronised to supported Apache rules with backups. Nginx/proxy/CDN behaviour may require host configuration. Canonical rules adjust supported HTML output but do not force Google's selected canonical. Re-scan repaired URLs, purge full-page caches and disable/undo obsolete rules.

Robots.txt

Robots editing requires elevated permissions. Review the file, preserve private paths and verify the public response after saving. Supported edits create backups. Subdirectory shops may need the domain owner's help because the effective robots.txt is at the root; other shops or the CDN may share/serve it.

The module does not remove access protection, repair hosting failures or bypass Google policy. Request Search Console validation only after verifying a real repair. Google must recrawl and may still choose not to index.

8. Google Tag Manager and pixels

Choose delivery

This tab has five panels and one **Save tracking settings** button. Select a single shop as SuperAdmin. Enable tracking, choose delivery, enter valid IDs, choose event groups and save.

Mode	Setup	Loaded after consent
Direct	GA4 G ..., Ads AW ... and/or numeric Meta Pixel ID; Ads conversion labels for conversions	Google tag and/or Meta SDK, without the module's GTM container
Google Tag Manager	GTM ... ID plus reviewed, published tags/triggers	GTM with the module's dataLayer events

If an existing pixel receives no data: select **Direct: GA4, Google Ads and Meta Pixel**, enter IDs, save and clear cache. Previously, destination IDs alone did not publish tags in your GTM account. Alternatively, keep GTM mode and configure/publish its tags. Avoid the same destinations in another module or a separate theme container.

The simplified interface has no Microsoft UET, refund-delivery setup, export or dashboard buttons. Older hidden preferences are retained where applicable, but that does not configure new integrations. Direct mode does not deliver Microsoft UET.

Consent

All four Consent Mode v2 signals default to **denied**. Direct GA4 requires analytics consent. Direct Ads/Meta require advertising storage, user-data and personalisation consent. No direct SDK loads before its required consent.

Use the translated banner, or connect your CMP before disabling it. The CMP must call the bridge after it exists, both for the initial saved choice and later changes. Analytics-only example:

```
window.WebpGtm.consent({
  analytics_storage: 'granted',
  ad_storage: 'denied',
  ad_user_data: 'denied',
  ad_personalization: 'denied'
});
```

Use **denied** for all values on withdrawal. Do not infer consent from visiting or logging in. With no banner and no CMP bridge, tracking remains denied. Downloaded scripts cannot be unloaded; withdrawal stops module event forwarding and updates vendor consent. Audit independently installed tags too.

Events and values

Supported groups cover product/list views and selection, category/search, confirmed cart quantity changes, cart view, checkout, shipping/payment selection, purchase, login/sign-up/profile update, sharing, wishlist success and contact success. Enable both the group and desired events. Classic selectors are provided; custom checkouts/wishlist/contact modules may require success-event adapters. Shipping/payment events indicate selection, not successful charging.

GA4 receives supported named events with allowlisted ecommerce fields. Direct Ads sends configured cart, checkout and purchase conversions to their labels; an account ID alone is insufficient. Direct Meta maps supported events to PageView, ViewContent, Search, AddToCart, AddToWishlist, InitiateCheckout, AddPaymentInfo, Purchase, CompleteRegistration and Lead. Unsupported mappings are not invented as standard events.

GA4 merchandise value is separate from tax/shipping. Purchases use shop/order IDs and browser deduplication per destination. A blocked SDK does not consume the module's purchase marker. Browser deduplication cannot cover all devices/storage scenarios or replace vendor transaction handling.

Customer name, email, address and payment credentials are not explicit payload fields. Google page-location/referrer values omit query strings. Vendor scripts can independently collect page/device data: review automatic collection/enhanced measurement, published tags and your privacy policy. This module is not a legal certification.

Verify tracking

1. Test with valid IDs in staging, without blocking extensions. Enable debug temporarily.
2. Reject optional cookies: no direct vendor SDK should load. Grant analytics only: GA4 should initialise, but Ads/Meta should not.
3. Grant advertising consent and test a product, cart, checkout and customer-owned test order confirmation.
4. Inspect `window.WebpGtm.getStatus()`: mode, consent, script state and dispatched counts. Dispatch means a browser tag call, **not** confirmed receipt/reporting by Google or Meta.
5. Check GA4 DebugView/Realtime, Tag Assistant or Meta Test Events in your own account. Verify IDs, values and duplicates. GTM mode also requires Preview/Publish.
6. Test withdrawal and order-page reload; turn debug off after testing.

If the bridge is absent, verify enabled settings, footer hook, module scripts, cache and CSP. An `idle` status usually means no relevant consent/ID; `error` means a script failed or was blocked. Fix blocking and reload. Loaded scripts with empty dashboards require checking outgoing requests, destination IDs, toggles, GTM publication, account filters and processing delay. Do not bypass consent to make counters increase.

Legacy refunds

Retained refund code has no setup/send controls in this tab and adds no automatic sender. Older configured refunds require credit slips, analytics consent, GA client attribution and an encrypted Measurement Protocol secret. Enabling purchases does not enable refund delivery. Queue activity prunes records older than 90 days; inactive stores need retention maintenance. Consult the developer maintaining any existing integration.

9. Security, translation and multistore

Select the intended shop for feeds, tracking and sensitive settings. Verify every public domain/language separately. Back Office actions require employee permissions and request tokens. For 403/token errors, reopen from the authenticated module menu, check employee access and HTTPS/admin domain, and clear stale cache. Never disable token validation.

Feed, sitemap and cron URLs contain private tokens: rotate exposed tokens and update external schedules. URL inspection is restricted to allowed origins/destinations, not a general proxy or file reader. UI text, XML and structured data are encoded for their contexts; do not add arbitrary HTML or tracking code to the module.

Keep credential backups private and revoke unused Google grants. After language updates, clear translation/cache layers and check employee locale and shop overrides. Do not duplicate XLIFF IDs when editing catalogs. Local tests do not replace Marketplace scanning, staging installation or assessment of other modules and published tags.

10. Maintenance and support

After theme, CDN, server or PrestaShop changes, check WebP queue/delivery, feed/sitemap accessibility, schema output and consent/tracking. Revalidate snippets after pricing, review or carrier changes. Back up rules and files before SEO edits and remove obsolete rules.

Disable an affected feature while investigating incorrect output. Disabling the module does not remove independent tags, other modules' schema, exported vendor data or every external server/CDN rule. Verify each layer separately.

Before uninstalling, back up configuration, module tables, .htaccess, robots.txt and generated images. Uninstall removes module settings/storage, including SEO/tracking history; it is not an upgrade procedure. Verify rewrite cleanup and image fallback afterward. Manage data already exported to Google/Meta in those accounts.

For support, provide module/PrestaShop/PHP/server versions, shop/language, the exact action and timestamp, console/network errors and relevant server logs. Redact credentials, private tokens and customer details. Distinguish local generation, public delivery and remote processing/reporting.

Copyright 2026 alteralgo. Module source: MIT License. Product names belong to their respective owners.