AMI LLMS File — Technical Reference

Version 1.9.1 · July 2026 · amiweb.co.uk

Requirements

Component	Requirement
WordPress	5.0 or higher; tested up to 7.0
PHP	7.4 or higher
Database	MySQL/MariaDB as supported by the WordPress install
External services	None — the plugin makes no external API calls

Endpoints and routing

Two rewrite rules are registered at the site root:

```
^llms\.txt$   →   index.php?ami_llms_txt=1
^llms-full\.txt$   →   index.php?ami_llms_full_txt=1
```

Rules are flushed on activation and automatically after plugin updates (a stored version number is compared on init). Responses are served as text/plain; charset=utf-8 with X-Robots-Tag: noindex and no-cache headers (Cache-Control: no-cache, must-revalidate), and the DONOTCACHEPAGE constant is defined so page-cache plugins skip these URLs.

File generation

llms.txt structure

- Header: "# Site Name" and a blockquote intro (custom setting → homepage meta description → site tagline)
- Sections in order: Company, one section per parent page with children (named after the parent), Resources (posts), Contact, Policies
- Pages are classified by hierarchy and slug/title keywords (privacy, terms, cookie, legal, gdpr → Policies; contact → Contact); the homepage is listed first under Company; ordering follows menu order, then title
- Entry format: "- [Label](URL): description" (with an optional Updated date) — the label is the page's on-page H1 (fallback: WordPress title); the URL is the canonical URL
- Taxonomy sections (optional): each selected public taxonomy — e.g. WooCommerce product_cat and brand taxonomies, EDD download_category, blog category — becomes its own section listing up to 100 non-empty terms with archive links and term descriptions

Content extraction (llms-full.txt)

Each page's own URL is fetched server-side and reduced to readable text: scripts, styles, comments, nav, header, footer, form, aside and heading (h1–h6) elements are removed; the <main> or <article> element is

preferred when present, otherwise the body is used. This makes extraction reliable on page builders and block themes, where the raw editor field is often empty. If the fetch fails, the plugin falls back to rendering the editor content (blocks and shortcodes).

Extracted text and the rendered meta description are cached per post in a transient (ami_llms_ptext_{post_id}) keyed to the post's modified time, for up to 7 days. A rebuild therefore refetches only pages that actually changed.

Description source order

- 1. Per-page override (_ami_llms_blurb post meta)
- 2. Yoast SEO surface API (YoastSEO()->meta->for_post()) — resolves templates and %%variables%%
- 3. Raw per-page meta of Yoast (_yoast_wpseo_metadesc), Rank Math (rank_math_description), AIOSEO (_aioseo_description), SEOPress (_seopress_titles_desc), with template-variable resolution where available
- 4. The meta description / og:description tag of the rendered page (SEO-plugin agnostic)
- 5. Auto-generated from the rendered page text, max 160 characters; otherwise blank (entry shows name and URL only). The raw editor field and excerpt are deliberately not used.

Every candidate is validated: placeholder copy (Lorem ipsum) is rejected from any source, and truncation is sentence-aware — text over the limit ends at the last complete sentence that fits, falling back to a word-boundary cut with an ellipsis only when no usable sentence boundary exists.

Caching and invalidation

- Built files are cached in transients (ami_llms_txt_cache, ami_llms_full_txt_cache) for one hour
- The cache is cleared immediately on: save_post, delete_post, transition_post_status, wp_trash_post, settings save, taxonomy term changes (created/edited/deleted), plugin updates, and on changes to watched post-meta keys (the SEO title/description fields of the four supported SEO plugins, plus the plugin's own per-page keys) — covering bulk editors, quick edit, and imports
- The per-post extraction cache is version-stamped, so every plugin update re-extracts with the new logic on the first rebuild
- Excluded pages are filtered with a meta_query on _ami_llms_exclude; queries cap at 200 posts per type with no_found_rows for performance

Database footprint

Item	Purpose
Option: ami_llms_settings	All plugin settings (one row)
Option: ami_llms_version	Version marker for update routines
Option: ami_llms_history	Last 10 Manual Override snapshots
Table: {prefix}ami_llms_hits	Crawler analytics: hit_time, file, bot, user_agent. Indexed on hit_time and bot. Rows older than 90 days are purged automatically.

Post meta: _ami_llms_exclude, _ami_llms_blurb	Per-page exclusion flag and description override
Transients: file caches + per-post page text	Regenerated on demand

Analytics requests store the time, requested file, matched crawler name, and user-agent string only. No IP addresses, cookies, or visitor identifiers are recorded.

robots.txt integration

The plugin appends reference lines to WordPress's virtual robots.txt via the robots_txt filter (priority 99):

```
# AI / LLM content index (https://llmstxt.org)
# llms.txt: https://yoursite.com/llms.txt
# llms-full.txt: https://yoursite.com/llms-full.txt
```

A physical robots.txt file on the server bypasses WordPress entirely; in that case the settings page displays the lines for manual addition.

Crawler identification

Requests are matched against known user-agent tokens, most specific first. Recognised crawlers include: GPTBot, OAI-SearchBot, ChatGPT-User (OpenAI); ClaudeBot, Claude-User (Anthropic); PerplexityBot, Perplexity-User; Google-Extended, GoogleOther, Googlebot; Bingbot; CCBot (Common Crawl); Bytespider (ByteDance); Amazonbot; Applebot and Applebot-Extended; Meta-ExternalAgent; DuckAssistBot; Mistral AI; Cohere; YouBot. Unmatched user agents are grouped as "Other / browser".

Uninstall behaviour

Deleting the plugin through the WordPress admin removes everything it created: all options (including legacy keys from earlier versions), the analytics table, both per-page post-meta keys across all posts, and all transients. No orphaned data remains.

Compatibility notes

- SEO plugins: Yoast SEO, Rank Math, All in One SEO, SEOPress (direct); all others via rendered-page detection
- Page builders / themes: Elementor, Divi, Gutenberg, block/FSE themes — content is read from the rendered page
- Caching: compatible with page caches and CDNs; the llms URLs opt out of caching via headers and DONOTCACHEPAGE
- Hosting: works on hosts without file-write permissions — files are served dynamically, never written to disk