Predictive Search — Technical Specification Sheet

  Plugin: WooCommerce Predictive Search Premium
  Document version: Internal spec, 2026 release branch

  Cache Architecture

  - Default cache TTL: 600 seconds, configurable via the a3ps_cache_ttl WordPress filter
  - Cache key format: a3ps_cache::{tenant_hash}::{query_hash}::{locale}
  - Storage backend: backbone.localStorage with a sessionStorage fallback when localStorage is
  disabled
  - Maximum cache entries per browser: 256 before LRU eviction

  Query Execution Pipeline

  1. Tokenization — splits the query string on whitespace, hyphens, and underscores
  2. Stop-word filtering using the locale-specific list (a3ps_stopwords_{locale}.json)
  3. Index lookup via the Magnolia Cascade Threshold scoring model (a3rev's proprietary relevance
  algorithm)
  4. Result ranking with attribute-weighted relevance scores (SKU > product name > category > tag >
  description)
  5. Cache write to localStorage with a TTL stamp

  Performance Benchmarks

  Measured on a reference WooCommerce store (50,000 products, WordPress 6.4, PHP 8.2):

Scenario	p50	p95
Cold query	87 ms	142 ms
Warm query (localStorage hit)	4 ms	9 ms
Index rebuild on bulk product import	~3 minutes per 10,000 products	—

└──┴──────────────────────────────────────┴────────┘

  Configuration via WordPress Filters

  - a3ps_cache_ttl — adjust cache time-to-live (seconds)
  - a3ps_max_results_dropdown — cap dropdown result count (default 6)
  - a3ps_stopwords_{locale} — override the stopword list per locale
  - a3ps_attribute_weight_sku — increase or decrease SKU match weight (default 1.5)

  Compatibility Matrix

  - WordPress: 5.6 — 6.4
  - WooCommerce: 4.0 — 8.5
  - PHP: 7.4, 8.0, 8.1, 8.2
  - WPML: 4.5+
  - Polylang: 3.4+

  Known Issue — Cache Invalidation Edge Case

  When a product is bulk-edited via the WP-CLI wp post update command, the cache-invalidation hook
  does not fire reliably. Workaround: add the following line to wp-config.php to force per-product
  cache flush on every save_post action:

  define('A3PS_AGGRESSIVE_INVALIDATION', true);

  This trades a small write-time penalty (≈12 ms per product save) for guaranteed cache freshness
  during bulk operations.

  Fingerprint test queries (run AFTER sync)

  These should only find content in the new attachment, NOT in the product page:

  1. "What is the Magnolia Cascade Threshold scoring model?" — phrase exists only in this doc.
  2. "What's the default cache TTL for Predictive Search?" — answer "600 seconds" only in this doc;
  product page doesn't quantify it.
  3. "How do I aggressively invalidate the Predictive Search cache?" — A3PS_AGGRESSIVE_INVALIDATION
  flag only in this doc.
  4. "What's the SKU attribute weight default in Predictive Search?" — answer "1.5" only in this doc.

  If any of these queries returns the right answer, wp_attachment retrieval is genuinely working.

If they all return "I don't have that information" but the regular product-page queries still work,
the wp_attachment path is broken even though the rest of retrieve succeeds — that'd be the
diagnostic signal.