

discover.js Re-Index + LightRAG Fallback

discover.js — Re-Index + LightRAG Fallback

Owner: AgentForge **Implemented:** 2026-04-17 (Hive Activation Phase 4 T9 + T10) **Tool:**

`~/system/tools/discover.js` **Index:** `~/system/tools/.alai/discover-index.json` **Post-sync wrapper:**
`~/system/tools/library-sync-wrapper.sh`

Purpose

Before T9, `discover.js "drop"` returned 3 product hits and 0 hits across tools/skills/agents/MCP/BookStack. Index was slow and shallow.

After T9+T10: persistent inverted index (521 entries from 6 sources), sub-50ms queries, and a semantic LightRAG fallback when the local index is thin.

Sources indexed

Source	Count (first build)	Origin file
tools	206	<code>~/system/tools/manifest-index.md</code> + <code>manifest.md</code>
skills	64	<code>~/system/databases/skill-registry.db</code>
agents	22	<code>~/system/agents/specialist-mapping.json</code> + <code>~/ .claude/agents/*.md</code>
mcp	7	<code>~/ .claude.json</code> <code>.mcpServers</code>
bookstack	182	<code>~/system/config/bookstack-sync-map.json</code>
products	40	<code>~/system/data/product-index.json</code>

Rebuild

```
# Manual
node ~/system/tools/discover.js --rebuild-index

# Automatic – happens every 5 min via library-sync-wrapper.sh,
# which is what com.alai.library-sync plist invokes after library sync
```

Atomic writes: index is written to `.tmp`, then renamed. No partial state visible to readers.

Query behavior

```
node ~/system/tools/discover.js "<query>"
# → tokens match inverted index → grouped by source

node ~/system/tools/discover.js "<query>" --no-lightrag
# → suppress LightRAG fallback entirely
```

If total hits across (tools + skills + agents + mcp + bookstack) < 3, the script queries LightRAG with a 5-second timeout. Results are prefixed `LIGHTRAG (fallback – semantic)` so you can tell them apart from keyword matches.

If LightRAG is slow or unavailable, the fallback silently times out and returns whatever the local index had. No hang.

Known issues

- First rebuild takes ~2s on current corpus. If total entries grow > 10k, consider a stemmed index or move to SQLite FTS5.
- `products` source expects `~/system/data/product-index.json`; if missing, that category prints 0 hits. Fine — not a crash.
- LightRAG fallback depends on the drain from System Evolution T6 finishing (65k pending). While the drain is running, most semantic queries will time out and silently return nothing — that's the designed behavior.

Related

- System Evolution T4 made LightRAG default-on. T10 here adds the conditional fallback.

- Library auto-push (runbook `library-auto-push.md`) runs the wrapper that rebuilds this index.
-

Revision #3

Created 2026-04-16 23:03:43 UTC by John

Updated 2026-06-21 20:03:05 UTC by John