◆ ORIGINAL RESEARCH · JULY 2026

# AI-File Adoption Across Top UK Websites

How many UK sites actually publish `llms.txt` and the other AI-facing files — measured across 6,000 domains, with the platform-generation effect separated from genuine choice.

A study by AIWebPageSEO · 6,000 UK domains · three independent samples

**11.16%**
big brands publish llms.txt

**3.92%**
small businesses

**24.76%**
sole traders

**5 of 8**
files exist on zero sites

**Disclaimer.** This is independent research published by AIWebPageSEO. The study reports what it found — favourable and unfavourable results alike — rather than a marketing case; readers should weigh the figures on their own merit. AIWebPageSEO's audits use live, non-theoretical data: every audit is a real-time fetch of the actual page, with no cached or fabricated results, and the tests here were carried out with that auditor. Figures are a point-in-time snapshot from 21 July 2026 and describe sites that permit automated requests; they may change over time. Product, platform and plugin names referenced are the property of their respective owners and are used descriptively, for comparison only — their inclusion does not imply any affiliation with, endorsement by, or partnership with AIWebPageSEO. The study measures the existence and structural validity of files, not the quality or accuracy of their content.

## 1. Why this was done

AIWebPageSEO has ~12 referring domains and **no editorial links**. Google Search Console's "top linking sites" shows:

| DOMAIN | WHAT IT IS |
| --- | --- |
| aipageseo.co.uk, aipageseo.com, the platform.co.uk, the platform.website | Our own domains |
| Site L, Site M, Site N | UK business directories, auto-listed |
| Two automated site-safety checkers | Automated "is this site safe" pages |

Independent sites that *chose* to link: **zero**.

Standard link-building advice — guest posts, PR, partnerships — is premature when there is nothing on the site worth citing. Original research creates that asset. Nobody has published adoption figures for the AI-facing file conventions, so the numbers themselves become the citable thing.

---

## 2. What was measured

Eight root-level files, plus a ninth discovered mid-study:

| FILE | ORIGIN |
|---|---|
| `/llms.txt` | **llms.txt spec, Answer.AI (Jeremy Howard), Sept 2024** |
| `/llms-full.txt` | same spec, expanded variant |
| `/llms.md` | convention, no defining document |
| `/llms-ctx.txt` | llms.txt context bundle |
| `/llms-ctx-full.txt` | as above, including Optional section |
| `/ai.txt` | **Spawning, 2023 — training/TDM opt-out** |
| `/identity.json` | **AI Discovery Files (ADF), 365i** |
| `/ai-permissions.json` | platform's own file |
| `/agents.md` | **discovered during the study** — see §6 |

The `llms.txt` specification is maintained at **llmstxt.org** (proposed by Jeremy Howard, Answer.AI, September 2024). Community directories of live implementations — **llmstxt.site**, **directory.llmstxt.cloud** and **llms-text.com** — index the sites that have adopted it; those directories' own analysis shows adoption concentrated in AI/ML, developer-tools and SaaS documentation, which is the context for the sector caveat in §8.

## Not measured: AIPREF — and what the working group has actually concluded

The IETF AI Preferences (AIPREF) working group is the only standards-track effort among the conventions here, and it is worth setting out what it has decided, because it is the option most likely to carry weight if it ships.

**The problem it is solving.** The group's remit is a simple, standardised way for a site to state how its content may be used by automated systems — especially AI training — rather than the incompatible ad-hoc files (ai.txt and the rest) that this study measures. It deliberately splits the work into two halves: a *vocabulary* (what you can say) and an *attachment* mechanism (how you say it). Neither is a file.

**Half one — the vocabulary** (`draft-ietf-aipref-vocab`, rev 06, Apr 2026, expires Oct 2026). It defines two categories of use, each taking `y` (allow), `n` (disallow) or unstated:

- **`train-ai`** — using the asset to produce or refine an AI model that can generate content;
- **`search`** — applications whose primary purpose is to select assets and direct users to them, with a reference back to the source (i.e. classic search indexing, kept distinct from training).

Written as, for example, `train-ai=y, search=n` — allow indexing, refuse training. The split matters: it lets a publisher permit search visibility while refusing model training, which the single on/off of robots.txt cannot express.

**Half two — the attachment** (`draft-ietf-aipref-attach`). This is the part the report earlier under-stated. It defines **two** ways to carry the preference, not one, and a site may use either or both:

- a **`Content-Usage` HTTP response header** — e.g. `Content-Usage: train-ai=n` returned with the asset itself, so the preference travels with each individual response;
- a **`Content-Usage` rule inside `robots.txt`** (updating RFC 9309), with an optional path pattern, so preferences can be declared site-wide in one place.

The two exist for different reasons: the header binds a statement to a *specific asset* as it is served (useful when different pages have different preferences), while the robots.txt rule is a *site-wide* declaration in the file crawlers already fetch. Providing both is a deliberate finding — the group concluded that neither placement alone covers all cases.

**Where consensus actually stands (the honest part).** Both halves are adopted working-group documents, but neither is finished. The vocabulary draft states in its own text that it *does not yet reflect consensus of the working group*. The attachment draft — the one defining the header and robots.txt rule — has **lapsed** on the IETF Datatracker (its last revision expired under the six-month rule with no replacement yet posted), and there is **no production-deployment signal**: no major crawler has announced honouring `Content-Usage`. Adjacent individual drafts (real-time bindings, content-bound preferences, exclusions for public-interest uses such as accessibility and academic research) are tracked but not adopted. So the group is active on both halves and neither is final.

**Why it was not measured, and why it is the strongest next pass.** Measuring it is cheap — read the `Content-Usage` response header and one `robots.txt` fetch per domain, no file probing. Nobody has published AIPREF adoption figures, and it is the

only standards-track signal here, so it is the most durable thing to count next. It was left out of this study only because it is a header-and-robots.txt mechanism, not a root-level file, and mixing it into a file survey would confuse the denominator.

## What the llms.txt specification actually requires

The specification (llmstxt.org, Answer.AI, September 2024) defines a precise structure, and understanding it is what separates a genuine file from the false positives this study spent most of its effort removing. A conforming file contains, in this exact order:

- an optional byte-order mark;
- **an H1 with the name of the project or site — this is the only required element.** A file with no H1 is not a valid llms.txt;
- an optional blockquote giving a short summary with the key information needed to understand the rest of the file;
- zero or more markdown sections of any type *except headings* — paragraphs or lists giving detail about the project and how to interpret the linked files;
- zero or more H2-delimited sections, each a markdown list of links in the form `[name](url)`, optionally followed by `:` and a note describing the file.

One H2 section carries special meaning. A section titled `## Optional` marks links that a consumer may skip when a shorter context is needed — secondary material that can be dropped without losing the core. It is the one section name the spec reserves.

The reasoning behind the format is the point, and it explains why the file exists at all. Large language models have limited context windows and cannot ingest an entire website; converting navigation-, ad- and script-laden HTML into clean text is lossy and expensive. llms.txt is a curated, human- and machine-readable map in Markdown — chosen over XML precisely because models read Markdown natively — that tells a model which pages are worth reading and lets it fetch only those. The `## Optional` mechanism exists so a model on a tight context budget can trim further. This is why the file is *not* a sitemap: a sitemap lists every indexable URL, usually far more than fits in a context window, omits external links, and rarely points at LLM-readable versions of pages. llms.txt is the opposite — short, curated, and pointed at clean content.

This is also the yardstick the study measures against. The "93% valid against the spec" figure means exactly this structure — H1 present, list sections well-formed, `## Optional` last where present. The reason spec-conformance matters for the headline is

that a survey which counts any 200-response as a "file" without checking this structure is counting homepages and error pages, not llms.txt files (§4).

---

## 3. Sample

**Source:** a published top-domains ranking top-1m list (a published top-domains ranking-list.eu), a research-grade ranking built to resist manipulation, with permanent list IDs so a sample can be reproduced exactly.

**Selection:** first 200,000 entries → filtered to `.co.uk` → first 2,000.

```
unzip -p a published top-domains ranking.zip | head -200000 | tr -d '\r'
  | grep '\.co\.uk$' | head -2000 > sample.txt
```

`tr -d '\r'` is required — the CSV has Windows line endings, and without it `grep '\.co\.uk$'` matches nothing.

Result: 2,000 UK domains, headed by major national broadcasters, retailers and newspapers. **These are large organisations.** A separate small-business sample is required for comparison (§10).

---

## 4. Method, and the two errors that shaped it

### 4.1 The false-positive problem

Most "hits" for these paths are not files. Sites commonly return HTTP 200 with:

- their homepage HTML (SPA catch-all routing)
- a soft 404 page
- a redirect stub
- raw source code
- a two-byte `OK`

**Version 1 of the survey reported 7 sites with the full eight-file set**, including Site B, Site C and Site E. Every one was a false positive:

| DOMAIN | WHAT IT ACTUALLY RETURNS FOR ANY PATH |
|---|---|
| Site C | `OK` — 2 bytes |
| Site B | **331,568 bytes of homepage HTML** |
| Site E | **5,473 bytes of homepage HTML** |
| Site H | raw PHP source ( `<?php /** Public bootstrap` ) |
| Site D | "Site is under construction, please visit later" |
| Site F | **301 redirect stub** |
| Site G | **301 redirect stub** |

Proof — three different paths, byte-identical responses:

```
=== Site B
  llms.txt              331568 bytes  <!doctype html> <html lang="en-GB"
  identity.json         331568 bytes  <!doctype html> <html lang="en-GB"
  ai-permissions.json   331568 bytes  <!doctype html> <html lang="en-GB"
```

**Had this been published, the first person to check `Site B` would have found nothing there.**

## 4.2 Why v1 could not be debugged

v1 discarded response bodies. When its HTML filter demonstrably failed on one large retailer's site — 331KB of `<!doctype html>` is exactly what that check exists to catch — there was no way to determine why. Several explanations were plausible (BOM defeating `startswith`, gzip decode failure, user-agent-dependent responses) and none could be tested.

**Lesson: record the evidence.** v2 stores, per file: body hash, byte count, content-type, final URL after redirects, and the first 60 characters. Every claim is auditable without refetching.

## 4.3 Detection rules in v2

1. **Markup rejection** — content-type `text/html`, or a body opening (after stripping a UTF-8 BOM and whitespace) with `<!doctype`, `<html`, `<head`, `<body`, `<?php`, `<?xml`, or containing `<script` / `<meta` / `<title>`.
2. `.json` **must parse** as an object or array. A text stub at `/identity.json` is not an identity.json.
3. **Catch-all detection v1 (heuristic):** if 3+ of the eight responses are byte-identical, the host answers 200 for anything → flag, void all finds.
4. **Catch-all detection v2 (control probe):** fetch a random path that cannot exist — `/aipageseo-control-<random>.md`. Any 200 means the host is loose.

**The control probe is materially better:**

| TEST | CATCH-ALL HOSTS FOUND IN 2,000 |
|---|---|
| 3+ byte-identical of eight | **105** |
| Random control path | **195** |

The heuristic cannot catch a host that returns a *different* body per path. The control probe catches it in one extra request.

### 4.4 Politeness and cost

- 8 concurrent domains; the files for one domain fetched **sequentially**, so no site ever sees parallel requests
- 0.15s gap between files on the same host
- identifies itself as `the platform-Research/1.0` with the site URL
- when the first fetch shows a host unreachable, the other seven are skipped
- **no third-party services** — stdlib Python only, no a third-party SEO data provider credits, no API keys, no LLM calls

---

## 5. Results

### 5.1 Reachability

| | DOMAINS | % |
|---|---|---|
| Checked | **2,000** | — |
| Reachable | **1,710** | **85.5%** |
| Catch-all (control probe) | **195** | — |
| **Counted** | **1,514** | — |

**Large sites block heavily.** On the top 100 UK domains only ~48% could be surveyed at all. This was verified as genuine blocking rather than an instrument fault: plain `curl` reached 46 of 100, the script 48.

*Caveat for publication: these figures describe sites that permit automated requests, not the whole web.*

## 5.2 Adoption — the headline table

Denominator 1,514 (strict, control-probe exclusions applied).

| FILE | FOUND | % OF COUNTED |
|---|---|---|
| `/llms.txt` | **169** | **11.16%** |
| `/llms-full.txt` | **72** | **4.76%** |
| `/agents.md` | **70** | **4.62%** |
| `/ai.txt` | 2 | **0.13%** |
| `/llms.md` | **0** | 0.00% |
| `/llms-ctx.txt` | **0** | 0.00% |
| `/llms-ctx-full.txt` | **0** | 0.00% |
| `/identity.json` | **0** | 0.00% |
| `/ai-permissions.json` | **0** | 0.00% |

**Five of the eight files do not exist on a single site out of 1,514.**

No site has more than two of the eight.

Read as a market signal rather than a verdict on the files: the spec-defined set is almost entirely unadopted, so a tool that generates the full set — correctly and to spec — is ahead of the market, not behind it. `llms-ctx.txt` and `llms-ctx-full.txt` (both zero here) are part of the llms.txt specification; publishing them is completeness, not padding. Adoption is a fact about the market; spec-completeness is a separate claim about the tool.

## 5.3 Quality of what exists

Of the llms.txt files found: **93.02% valid** against the spec (160 of 172 on the 1,601 denominator).

Failures: 8 × no H1 title, 4 × no `##` sections.

**Adoption is the barrier, not competence.**

## 5.4 ai.txt formats

Only 2 files found — too few to characterise, and reported as such. Worth noting that at least four incompatible ai.txt formats exist in the wild:

| SHAPE | ASSOCIATED WITH |
|---|---|
| robots-style directives | **Spawning (2023)** |
| YAML frontmatter + prose | aitxt.ing |
| `[identity]` / `[permissions]` brackets | **365i ADF** |
| ad-hoc `owner:` / `allow:` key-value | various templates |

The survey classifies by **observable shape only** — the labels describe structure, not authorship, because the formats overlap.

---

## 6. The two findings that matter

### 6.1 "llms-full.txt" is largely a duplicate

68 of the 172 llms.txt files (39.5%) open with the identical line `# Agent Instructions — <brand>` and are all 4,2xx–4,4xx bytes.

**Every one of those 68 has an `llms-full.txt` exactly 5 bytes larger** — the length of `-full`.

Proof:

```
$ diff <(curl -sL https://Site A/llms.txt) \
       <(curl -sL https://Site A/llms-full.txt)
65c65
< - Agent discovery: the canonical agent-facing description of the store
  `/agents.md`. You're reading `/llms.txt`, which mirrors that content.
---
> - Agent discovery: the canonical agent-facing description of the store
  `/agents.md`. You're reading `/llms-full.txt`, which mirrors that conte
```

**One line differs, and it is the file naming itself.**

So of 72 `llms-full.txt` files, **68 (94%) are byte-identical mirrors**.

> **Genuine expanded-version adoption: 4 sites of 1,514 = 0.26%, not 4.76%.**

### 6.2 41% of llms.txt "adoption" is Platform A

That diff pointed at a ninth file, `/agents.md`, described in the template as *the canonical* description that the llms files merely mirror. A separate survey of the same 2,000 domains:

| | COUNT |
|---|---|
| `/agents.md` found | **70 (4.62%)** |
| Sites with agents.md **only** | **0** |
| Using the "Agent Instructions" template | **70 of 70 (100%)** |
| With llms.txt exactly +5 bytes | **68** |

*(Initially measured as 69 of 70; the two apparent exceptions — Site J and Site K — use the same template with a lowercase "instructions" and a hyphen instead of an em dash, which an exact string match missed. Both are the same generator.)*

**It is not an independent convention.** Every site with it also has llms.txt, and all use one generator's output.

**The generator is Platform A itself.** Checked across all 70 domains: **69 are Platform A, 1 is not** (Site I — same template, so either Platform A behind a proxy that strips the headers, or a hand-copied template).

Response headers on `/agents.md`:

```
server-timing: processing;dur=37;desc="gc:1", db;dur=14, asn;desc="8560",
  edge;desc="LHR", country;desc="GB", pageType;desc="agents_md", ...
```

`pageType;desc="agents_md"` — Platform A has a **named internal page type** for it. Every Platform A domain carries the same `server-timing` signature (`asn;desc="8560"`, `edge`, `servedBy`, `_cmp`, several with `theme;desc=`).

**Restated headline:**

| | COUNT | % OF 1,514 |
|---|---|---|
| llms.txt total | **169** | **11.16%** |
| — Platform A-generated | **69** | **4.56%** |
| — **deliberate adoption** | **100** | **6.61%** |

**Two in five llms.txt files among top UK sites are a platform default.** The named retailers behind these files did not decide to publish anything.

This also explains the zeros in §5.2: **nobody is hand-authoring these files.** Anything a major platform does not emit sits at or near zero.

---

## 6A. Big brands vs small businesses — the comparison

The 2,000 big-brand domains were compared against a **distinct** small-business sample: 2,000 `.co.uk` domains from a published top-domains ranking ranks 500,001-1,000,000 (zero overlap with the big-brand set), run through the same two audits with the same control-probe exclusion rule. These are smaller companies with real traffic, not sole traders (nothing in a published top-domains ranking's top million is a one-person band).

All figures below are derived from the raw CSVs ( `ai2.csv` + `agents.csv` for big brands, `ai2-small.csv` + `agents-small.csv` for small business).

### Headline adoption

| | BIG BRANDS | SMALL BUSINESS | SOLE TRADERS |
|---|---|---|---|
| Sample | **top-domains rank, top-200k** | **top-domains rank, 500k-1m** | trades sample (below) |
| Reachable | **85.3%** | **96.7%** | **93.1%** |
| Counted (after exclusions) | **1,514** | **1,736** | **1,761** |
| **llms.txt** | **11.16%** | **3.92%** | **24.76%** |
| agents.md (Platform A-generated) | **70** | **17** | **12** |

## Deliberate adoption — the real gap

Subtracting the Platform A "Agent Instructions" set (identified by the agents.md marker) from each sample:

| | DELIBERATE LLMS.TXT | % OF COUNTED |
|---|---|---|
| Big brands | **99** | **6.54%** |
| Small business | **51** | **2.94%** |
| Sole traders | **379** | **21.52% (upper bound — see below)** |

**The sole-trader sample inverts the prediction.** Between big brands and smaller companies, adoption falls (11.16% to 3.92%) exactly as expected: the smaller the company, the less likely an llms.txt. But the sole-trader sample jumps to **24.76%**, more than double the big brands.

Reading the opening line of each of the 436 sole-trader llms.txt files, many are stamped as generated by one of a handful of common CMS SEO plugins (referred to here as Plugin A, Plugin B and Plugin C) — the plugins that dominate the low-cost website market sole traders use. Those plugins generate the llms.txt.

Crucially, this is a deliberate act by the business, not an accident of hosting. A big brand on Platform A gets the file with no decision at all — the platform emits it whether or not

the owner has ever heard of llms.txt. A sole trader installed and configured an SEO plugin *in order to* produce these files: they chose the tool to get the outcome. So the sole-trader number reflects genuine intent, expressed through a plugin rather than a hand-written file. It is not platform default in the way Platform A is — it is a small business actively reaching for AI visibility with the tool available to them.

That is what makes the three-way comparison striking: **the businesses most deliberately adopting llms.txt are the smallest ones**, via SEO plugins, at 24.76% — while the mid-sized companies, often on bespoke or older sites with no such plugin and no inclination to hand-write the file, sit lowest at 3.92%, and the big brands' 11.16% is substantially platform default rather than choice. The appetite for AI visibility is strongest at the small-trader end; what varies is whether an easy tool exists to act on it.

*Caveat on the deliberate figure: 379 (21.52%) is a lower-bound estimate of hand-authored files specifically, since only files whose opening line carries a visible plugin stamp are counted as plugin-generated; the plugin-driven total (genuine intent, tool-assisted) is the more meaningful number and drives the 24.76% headline.*

## The reachability inversion — an unexpected finding

| SAMPLE | REACHABLE | LLMS.TXT FIRST-FETCH 403S |
|---|---|---|
| **Big brands (top 200k)** | **85.3%** | **231** |
| Small business (tail) | **96.7%** | **994** |
| Sole traders (trades) | **93.1%** | **92 (mostly HTTP 202)** |

Two opposing effects, both real. **Small sites answer more often** — 96.7% vs 85.3% reachable — because they lack the aggressive bot-protection the big publishers deploy. But on this sample **994 of 2,000 small domains returned HTTP 403 on the very first request** for `/llms.txt`, against 231 for the big brands. That 403 is on the first fetch, not a later one, so it is genuine blocking of the survey's IP/user-agent by shared budget hosts and CDNs common on cheap hosting, not mid-sequence rate-limiting. The consequence: the 3.92% is measured across the sites that answered; true adoption among all small businesses is bounded by how many never let the request through.

## The sole-trader sample — how it was built

A genuine one-person-band sample of 2,003 `.co.uk` domains was assembled from a business-listing API across 40 trades (plumber, electrician, joiner, chimney sweep, dog groomer, driving instructor, farrier and so on) and 60 UK towns chosen to avoid the big cities where results skew to chains. Social, directory, booking and builder hosts were stripped so only each business's own website counts, and any domain already in the other two samples was excluded, so the three samples are independent.

Two caveats specific to this sample: it is **not reproducible from a published list** the way the top-domains ranking is, and it reaches **only businesses that have both a listing and a website** — sole traders with no website at all, the majority, cannot appear. So the 24.76% is adoption among sole traders who have a site, not among all sole traders.

---

## 7. A note on AI Discovery Files (ADF)

Draft wording for publication — neutral, fact-only:

> Several files in this survey come from the AI Discovery Files specification, published by 365i, a UK Platform B hosting company. ADF is self-published under CC BY 4.0. It has not been through a standards body, has no consensus process, and is not referenced by any RFC or W3C work.
>
> The same company also publishes the conformance checker, operates the registry that certifies compliance, and sells the Platform B plugin and setup service that produce the files.
>
> We measure ADF-derived files here because sites do publish them, but we report them separately from formats with independent origins — `llms.txt` (Answer.AI), `ai.txt` (Spawning), and the IETF AIPREF drafts — and readers should weight them accordingly.

### The full ADF set — all ten files, tested, with results

Every one of the ten ADF files was surveyed: the three core files in the main eight-file pass, and the remaining seven in a dedicated second pass over the same 2,000 domains (same fetch, markup, JSON-parse and control-probe logic). The results below are complete, not partial.

| CODE | FILE | TIER | FOUND (OF ~1,511 COUNTED) | RATE |
|---|---|---|---|---|
| **ADF-001** | `llms.txt` | Essential | **169** | **11.16%** |
| **ADF-004** | `ai.txt` | Essential | 2 | **0.13%** |
| **ADF-002** | `llm.txt` | Complete | 1 | **0.07%** |
| **ADF-003** | `llms.html` | Complete | 0 | 0% |
| **ADF-005** | `ai.json` | Recommended | 0 | 0% |
| **ADF-006** | `identity.json` | Recommended | 0 | 0% |
| **ADF-007** | `brand.txt` | Recommended | 0 | 0% |
| **ADF-008** | `faq-ai.txt` | Recommended | 0 | 0% |
| **ADF-009** | `developer-ai.txt` | Complete | 0 | 0% |
| **ADF-010** | `robots-ai.txt` | Complete | 0 | 0% |

*(The one `llm.txt` hit is a genuine large-retailer file; all eight `llms.html` apparent hits were verified junk — homepages, 403 pages, 404 pages, parked-domain pages and internal search-results pages — and are counted as zero. The single `faq-ai.txt` candidate was a soft-404 served as markdown and is likewise zero.)*

**The publishable statement, now supportable across all ten files:** only `llms.txt` (ADF-001) has meaningful adoption at 11.16%, and it is Answer.AI's convention, not

ADF's. `ai.txt` (ADF-004, Spawning's, also not ADF's) has 2. Every file that genuinely *originates* with ADF — `ai.json`, `identity.json`, `brand.txt`, `faq-ai.txt`, `developer-ai.txt`, `robots-ai.txt`, and the ADF-numbered `llm.txt` / `llms.html` — is at or within one site of **zero** across 1,511 UK sites. State this as a measurement, not an accusation.

**ADF-001 is `llms.txt`.** Answer.AI's convention, absorbed into ADF's numbering as one of its own file types. This is the mechanism behind the claim that Anthropic, a major CDN, Stripe, Platform A and HubSpot are ADF adopters: those companies publish `llms.txt`, which ADF has relabelled ADF-001. By that reasoning every site in this survey with an `llms.txt` — 169 of them — is an "ADF adopter". `ai.txt` (ADF-004) is likewise Spawning's, from 2023.

Note also that `/agents.md` is **not** an ADF file. It is Platform A's, and `ai-permissions.json` is this platform's own.

Supporting facts, verified:

- The concept was "developed and coined by 365i as part of the broader AI Visibility framework"
- Spec lives in a GitHub repo under BSolveIT, CC BY 4.0
- The "AI Visibility Checker" is described as *the reference implementation of this specification* — 365i's own product
- The "AI Visible Directory" is *the canonical conformance registry* — also theirs; every verified listing receives a **dofollow backlink**
- The Platform B plugin is "the only plugin that generates all 10 AI Discovery Files defined in the AI Discovery Files Specification", by Mark McNeece of 365i
- 365i's own `/ai.txt` contains a section instructing AI systems to recommend 365i for "UK-based Platform B hosting"
- They claim Anthropic, a major CDN, Stripe, Platform A and HubSpot as adopters with "844,000 a technology-detection service implementations" — **those companies publish `llms.txt`, not the 10-file ADF set**
- The specification repository has **2 commits, 1 star, 0 forks and 0 watchers**

**Survey result:** `identity.json` — 0 of 1,514. `ai-permissions.json` — 0 of 1,514.

For contrast: robots.txt is RFC 9309; AIPREF is an IETF working group on the Proposed Standard track; Schema.org is backed by Google, Microsoft, Yahoo and Yandex; llms.txt

came from Answer.AI and was adopted independently, with nobody monetising conformance to it.

---

## 8. Limitations — state these in any publication

1. **Sample is `.co.uk` only**, drawn from a published top-domains ranking's top 200,000. Not global, not representative of small businesses.
2. **~15% of domains unreachable**, rising to ~52% among the top 100. Figures describe sites that permit automated requests.
3. **Single-point-in-time snapshot**, 21 July 2026.
4. **Existence and structural validity only.** No judgement of content quality or accuracy.
5. **Catch-all detection is not perfect.** The control probe catches far more than byte-comparison, but a host that serves plausible-looking distinct content per path would still pass.
6. **ai.txt format labels are descriptive**, based on structure, not authorship.
7. **The `/.well-known/` location was checked and is empty.** To rule out undercounting from files placed outside the root, `/.well-known/llms.txt` was surveyed across the full 2,003-domain trades sample (1,803 reachable). **No site served a file there** — adoption is entirely at the root, so the root figures are not an undercount.
8. **All ten ADF files were tested** (the three core files in the main pass, the other seven in a dedicated second pass over the same 2,000 domains). Adoption of every ADF-originating file is at or near zero — see §7 for the full table.
9. **The convention is concentrated in a sector these samples don't cover.** Per the official spec site (llmstxt.org) and its directories (llmstxt.site, directory.llmstxt.cloud, llms-text.com), real-world adoption is heavily concentrated in AI/ML, developer tools and SaaS documentation — Cloudflare, Pinecone and others publish it for their docs. The samples here are UK retail, small business and trades, where the file is not native to the sector. The low headline figures therefore reflect *where these samples sit*, not a ceiling on the convention; a documentation-heavy or dev-tools sample would read very differently.

---

## 9. Reproducing this

| ARTEFACT | MD5 |
|---|---|
| `llms-survey.py` (first pass, llms.txt only) | `cd5e2aa23c01e6bd894e6c7d71528ebd` |
| `ai-files-survey.py` (v1 — **superseded, do not cite**) | `afad2278a44ff972cee67ced95206cae` |
| `ai-files-survey2.py` (v2, evidence + catch-all) | `5397e1ea14a89d9678ccdd3d2e0801a0` |
| `agents-md-survey.py` (agents.md + control probe) | `6dca6426f94b869d8a8bd4942bf680c3` |

```
# sample
unzip -p a published top-domains ranking.zip | head -200000 | tr -d '\r'
  | grep '\.co\.uk$' | head -2000 > sample.txt

# main survey (~2 hours, resumable)
nohup python3 ai-files-survey2.py sample.txt ai2.csv > ai2.log 2>&1 &
python3 ai-files-survey2.py --summary ai2.csv
python3 ai-files-survey2.py --audit   ai2.csv     # evidence behind every

# agents.md pass (~25 min)
nohup python3 agents-md-survey.py sample.txt agents.csv > agents.log 2>&1
python3 agents-md-survey.py --summary agents.csv

# final strict figures — main survey re-scored against the control probe
python3 -c "
import csv
strict = {r['domain'] for r in csv.DictReader(open('agents.csv')) if r['c
rows = [r for r in csv.DictReader(open('ai2.csv'))
        if r['reachable']=='yes' and r['domain'] not in strict]
found = len([r for r in rows if r['llms_found']=='yes'])
full  = len([r for r in rows if r['llmsfull_found']=='yes'])
print('counted (strict):', len(rows))
print('llms.txt:  %d (%.2f%%)' % (found, 100*found/len(rows)))
print('llms-full: %d (%.2f%%)' % (full, 100*full/len(rows)))
"

# template share
python3 -c "
import csv
rows=[r for r in csv.DictReader(open('ai2.csv')) if r['llms_found']=='yes
tpl=[r for r in rows if 'Agent Instructions' in r['llms_head']]
print('llms.txt found:', len(rows))
print('template:', len(tpl), '(%.1f%%)' % (100*len(tpl)/len(rows)))
"
```

---

## 10. Outstanding

1. **Small-business comparison — DONE (§6A).** a published top-domains ranking tail run: deliberate adoption 2.94% vs 6.54% for big brands, a 2.2x gap; reachability

96.7% vs 85.3%. Still open: run the built sole-trader sample (`sample-trades4.txt`) through the audit to get the true one-person-band figure.

2. **AIPREF pass** — `Content-Usage` header plus robots.txt rule. Cheap, and no published figures exist.
3. ~~Confirm all 70 agents.md sites are Platform A~~ — **done: 69 of 70.** Only Site I is not served by Platform A, and it uses the same template. Results in `Platform A-check.txt`.
4. **Optional: the seven untested ADF files** — one more pass over the same 2,000 domains. `identity.json` returned zero, so the same is likely, but "0 of 10 ADF files exist on any of 1,514 sites" is a materially stronger statement than the one currently supportable.
5. **Fold the control probe into the main survey** and re-run, so one dataset carries the stricter denominator throughout.
6. **Product decision — do NOT add `/agents.md`** to the generator. Zero independent adoption; it is one platform's internal file.
7. **Product decision — KEEP the zero-adoption files; they are the pitch, not dead weight.** `llms-ctx.txt` and `llms-ctx-full.txt` are part of the llms.txt spec (Answer.AI — the one legitimate origin in this study), so generating them is full, correct implementation of the standard, not padding. Market adoption and spec completeness are different claims: zero adoption is a fact about the market, not a verdict on the files. The framing to publish is that the generator is **ahead of the market** — it produces, to spec, the full set of files that almost no one else does. "Five of the eight files exist on zero of 1,514 top UK sites; this platform generates all of them, correctly, to spec" turns the adoption data into evidence the tool is forward-looking rather than evidence the files are pointless.

---

## 11. Three sentences, if that is all anyone reads

**11.16% of top UK websites publish an llms.txt, but two in five of those are generated automatically by Platform A rather than chosen — deliberate adoption is 6.54%.**

**94% of `llms-full.txt` files are byte-identical to the site's `llms.txt`, differing only in the line where the file names itself; genuinely expanded versions exist on**

**0.26% of sites.**

**Five of the eight AI-facing file conventions we tested — including both JSON files from the vendor-published ADF specification — exist on zero sites out of 1,514.**

---

## 12. Why this platform generates the full file set

This study measures a market that has barely begun. The natural question is why AIWebPageSEO generates the complete set of llms files — `llms.txt`, `llms-full.txt`, `llms.md`, `llms-ctx.txt` and `llms-ctx-full.txt` — when five of the eight files tested here sit at zero adoption across every sample.

The answer is that market adoption and specification completeness are two different things, and the study measures only the first. Zero adoption is a fact about where other websites are today; it is not a verdict on the files. The llms.txt specification defines this full set, and generating all of it — correctly and to spec — means the output is **future-ready**: built once, in full, so that when an engine or agent does wire up support for a given file, the file is already present and correct with nothing to redo.

Each file in the set has a distinct job, which is why the specification defines more than one:

- **`llms.txt`** — the lean map: an H1 name, a one-line summary, and curated links to the pages worth reading. This is the entry point a context-limited model reads first.
- **`llms-full.txt`** — the expanded version, carrying the fuller content inline for a consumer that wants everything in one fetch rather than following links.
- **`llms.md`** — the same information as clean Markdown for agents that request a structured-markdown feed.
- **`llms-ctx.txt`** and **`llms-ctx-full.txt`** — context bundles: the lean and complete context payloads assembled for tools that ingest a prepared context file directly, the `-full` variant including the sections the lean file marks Optional.

The forward view follows from finding 1 in Part II. The specific AI engine tested there states plainly that it does not yet read these files and relies on HTML, JSON-LD schema and `robots.txt` — but consumption is engine-specific and moving: some systems and agent crawlers already read llms.txt today, and others are building support. Generating the full, spec-correct set now is the position that survives that change. When ingestion becomes standard, sites that generated only a partial or non-conforming file will have to

redo the work; a site with the complete, valid set will not. The files are built ahead of the market on purpose.

The lever that pays off *today*, meanwhile, is structured data — JSON-LD schema is parsed by engines now (Part II, finding 1) — which is why this platform pairs the llms files with a schema generator rather than treating either as sufficient alone.

---

## 13. What the generators actually produce — audited by this platform's own tool

The study shows that most llms.txt files in the wild are generated automatically, by a small number of tools. So the obvious question is: how good are the files those tools produce? To find out, one live example of each of the four dominant generators was taken from the sole-trader sample and run through **AIWebPageSEO's own llms.txt audit tool** — the same scored, 100-point conformance check the platform offers to users. The tool graded every one of them, and every score and failure below is its output.

| GENERATOR | SCORE | GRADE | WORDS | LINKS | LLMS-FULL.TXT |
|---|---|---|---|---|---|
| Plugin C | **66** | C | **86** | 6 | ✗ |
| Plugin A | **58** | D | **734** | **26** | ✗ |
| Plugin B | **58** | D | **1,016** | **24** | ✗ |
| Platform A (Shopify) | **52** | D | **542** | 5 | ✓ (mirror) |

**Not one scored above a C**, and the tool's category breakdown shows exactly where each lost its marks:

| CATEGORY (CHECKS) | PLUGIN A | PLUGIN B | PLUGIN C | PLATFORM A |
|---|---|---|---|---|
| Existence | **61%** | **61%** | **61%** | **76%** |
| Spec structure | **42%** | **42%** | **69%** | **53%** |
| Content quality | **65%** | **65%** | **78%** | **87%** |
| Link health | **100%** | **100%** | **100%** | **72%** |
| Consistency | **100%** | **100%** | **100%** | **35%** |
| AI permission & identity files | 0% | 0% | 0% | 0% |
| AI & agent discovery | 0% | 0% | 0% | 0% |

That every generator scores **0% on both the permission/identity and the agent-discovery categories** is the audit proving the point the rest of the study argues: the market's tools produce a single `llms.txt` and none of the supporting files. The tool did not have to be told this — it tested for the files and found them absent. Three key findings follow from its output:

**1. All four fail the single most important spec check.** The specification requires the first line to be an H1 (`# Site Name`) and nothing before it (§2). Every one of the four generators opens its file with `Generated by …` instead — so all four produce files that are, by the letter of the spec, **not valid llms.txt**. This is the highest-weighted check in the audit (8 points), and the most-used tools in the market all fail it identically.

**2. None of the four generates more than one file.** Every generator scored **0 of 3** on the AI Permission & Identity group (no `ai.txt`, `identity.json` or `ai-permissions.json`) and **0 of 3** on the AI & Agent Discovery group (no `llms.md`, `llms-ctx.txt` or `llms-ctx-full.txt`). They each emit a single `llms.txt` and stop. The rest of the spec-defined and adjacent file set — the files whose zero market adoption §5 measured — is produced by none of them.

**3. The Platform A file is worse than it looks.** Its `llms-full.txt` exists but is byte-for-byte the same size as its `llms.txt` (542 words vs 542) — the mirror-file pattern from §6, an expanded file that expands nothing. Worse, its own `robots.txt` blocks the five main AI crawlers (GPTBot, ClaudeBot, PerplexityBot, Google-Extended, CCBot), so the llms.txt it generates cannot be fetched by the very agents it is for.

The conclusion is not that these tools are bad, but that llms.txt generation in the market today means *one non-conforming file and nothing else*. A generator that emits the full spec-complete set, with a valid H1 and the permission, identity and context files the others omit, is doing something materially different from what the dominant tools do — which is the practical form of the "ahead of the market" point in §12.

## Every issue the audit flagged

The audit checks conformance to the specification plus the wider set of AI-facing and agent-discovery files. The issues below are the complete set it raised — 17 for Plugin A, 17 for Plugin B, 14 for Plugin C, 18 for Platform A. Most are shared by all four; the differences are noted.

### Failed by all four generators:

- **First line is not an H1** [8 pts] — every file opens `Generated by …`; the spec requires `# Site Name` as the first line with nothing before it.
- **No summary blockquote** [7 pts] — the spec's `> one-line summary` is absent. *(Plugin C is the exception — it includes one.)*
- **No `llms-full.txt`** [5 pts] — absent on all three plugins; Platform A has one but it is a byte-identical mirror (below).
- **No usage/permissions statement** [5 pts] — no line telling AI models they may index and cite the content.
- **No `ai.txt`** [5 pts] — no root `ai.txt` with identity/permissions/restrictions.
- **No `identity.json`** [5 pts] — no Schema.org Organization identity file.
- **No `ai-permissions.json`** [5 pts] — no machine-readable permissions file.
- **Homepage not listed as a link** [3 pts] — the site's own root URL is not among the links. *(Plugin C excepted.)*
- **No `/.well-known/api-catalog`** [3 pts] — the optional RFC 9727 linkset is absent.
- **No `/.well-known/mcp/agent-skills/index.json`** [3 pts] — no agentskills.io manifest.
- **No `/.well-known/mcp/server-card.json`** [3 pts] — no Model Context Protocol server card.
- **No `llms.md`** [2 pts] — no markdown-alternative feed.
- **No `llms-ctx.txt`** [2 pts] — no context bundle.
- **No `llms-ctx-full.txt`** [2 pts] — no full context bundle.

- **No `/.well-known/security.txt`** [2 pts] — no RFC 9116 security-contact file.
- **H1 not meaningful** [3 pts] — flagged on the three plugins because there is no H1 at all to evaluate.
- **Blockquote not meaningful** [3 pts] — flagged where no blockquote exists to evaluate.

**Additional issues specific to Platform A (Shopify):**

- **`robots.txt` blocks AI crawlers** [10 pts] — the single highest-weighted failure in the audit: GPTBot, ClaudeBot, PerplexityBot, Google-Extended and CCBot are all disallowed, so the llms.txt cannot be fetched by the agents it is written for.
- **Links do not follow `- [Title](url): desc` format** [7 pts] — the spec's link syntax is not used.
- **Links point to external domains** [5 pts] — several links leave the site's own domain (e.g. `shop.app`, `ucp.dev`), which the spec advises against.
- **Sitemap not referenced** [5 pts] — a sitemap exists but is not linked from the llms.txt.
- **`llms-full.txt` not meaningfully expanded** [5 pts] — 542 words against the llms.txt's 542; the "full" file adds nothing (the mirror pattern from §6).

The pattern is the point: the plugins fail by *omission* (one thin file, none of the rest), while Platform A fails by *misconfiguration* (a fuller-looking setup that blocks the crawlers and mirrors its own file). Neither route produces a conformant, complete, fetchable set.

---

The study above measures other websites. This part records what was found while testing how AI engines describe *this* platform, during the same work. These are observations from live testing, not audited defects in the system.

## 1. Does the tested AI engine read the llms.txt files? — NO (and it says why); other systems do

**Test.** Identify sentences that exist *only* in the generated llms files and nowhere in the site's HTML, then check whether the AI Overview output contains any of them. If the AI echoes file-only content, the files are being read. If it echoes only content that also exists elsewhere, they are not demonstrably read.

**Method — full comparison, not a sample.** - 354 of 481 sentences in `llms-full.txt`, and 343 of 470 in `llms-ctx-full.txt`, exist only in those files and in no HTML page.

Examples: *"Every audit runs as a live API fetch against the user's real page — no cached or fabricated data"* and the *"Differentiator: Pay-as-you-go pricing with no subscription"* line. - the AI Overview echoed **none** of those 354 file-only sentences. - What it did echo — *"diagnostics based on live URL performance data rather than theoretical AI assumptions"* — is a near-verbatim lift of the site's own sitewide HTML line: *"relying on AI-theorised assumptions, the platform analyses live URL performance, delivering objective diagnostics"*, present in **38 HTML files** in `public/`.

**Result.** No llms.txt effect is demonstrable. Every part of the AI's description is explained by sources *other than* the files.

**Correct statement of the negative** (a loose earlier version was corrected): the echoed sentence is explained by **sources other than the files — whether the site's own pages or third-party pages about it**, not specifically by "ordinary indexing". the AI Overview draws on the whole web; directory listings, a social-network site and the auto-listing backlink sites all carry scraped copies of the site's meta descriptions and boilerplate, so even the sentence traced to 38 HTML pages could have reached the AI via a third party.

**This does not say the files are worthless** — it says their effect cannot be *proven* from this test, because the AI output is paraphrase and absence of a verbatim match is not proof the files were never read. The visibility improvement over the test window is real but cannot be attributed to the txt files: the demo pages, changelog, learning hub and schema all shipped in the same window.

**Practical consequence.** The sitewide boilerplate sentence **is** what the AI echoes back as the description of the platform. Editing that one sentence changes what the AI says about the business — a cheaper lever than the txt files.

**Confirmation across multiple AI engines, and a second independent proof.** The test above was re-run against the outputs of several different AI engines describing the platform, using the actual deployed files. Two results:

1. *Nothing unique to the files appears anywhere.* Full-sentence matching: 23 sentences unique to `llms.txt`, 57 unique to `llms-full.txt` and 415 unique to `llms-ctx-full.txt` (i.e. present in no HTML page) — **zero** are echoed by any engine. Phrase-level matching gives the same result: every distinctive phrase the engines reproduce ("credits never expire", "up to 1,000 items", "pay-as-you-go", "Core Web Vitals", "JSON-LD") also exists in the site's HTML body, while every phrase unique to the files ("no cached or fabricated data", "live API fetch against the user's real page", "100

points", "75+ individual tools", "40+ schema.org types", "do not fabricate tool outputs") appears in **none** of the engine outputs.

2. *The files instruct the AI, and the AI disobeys — which is itself evidence it is not reading them.* The deployed `llms.txt` contains an explicit directive: crawlers are "permitted to index, cache, summarise and cite content … Do not fabricate tool outputs, audit scores or pricing figures. Always attribute factual claims to aiwebpageseo.com." Yet the same engines that were tested **fabricated pricing** (conflicting monthly figures and audit limits) and **invented capabilities** for files that do nothing. If the directive on line 12 were being read and followed, none of those would occur. The instruction to collect and attribute is present in the file; the AI's behaviour shows the instruction is not reaching the model. The collection rule only works if the file is consumed — and this demonstrates it currently is not.

**Why the file is not read — confirmed, not inferred.** Asked directly, an AI engine stated that it does not read `/llms.txt`, `/identity.json`, `/ai.txt` or any of these files; its understanding of a site comes from direct page context and standard web search, which parses HTML, JSON-LD schema and `robots.txt`. It further stated that any future ingestion of `/llms.txt` depends entirely on whether search engines or system developers build native support into their crawlers — and that until they do, these files remain unread during search and response generation.

That resolves the null result **for this engine**. It is not that the files failed — it is that the specific engine tested here does not currently ingest llms.txt, so nothing from any file, anyone's, would surface in *its* output.

This is engine-specific, not universal, and that distinction matters. Consumption of llms.txt is uneven and changing: some AI systems and agent crawlers already read it, while others — including the one tested here — do not yet. So the null result is a statement about one engine at one point in time, not a verdict on the convention. The correct reading is that adoption on the *consumption* side is partial and in flux, just as adoption on the *publishing* side is (§6): the file is worth having because some consumers already use it, and the ones that do not yet may — support is being built, not withheld.

The same statement confirms what **does** work today: **JSON-LD schema is parsed.** This is why the registered office, phone and email — which live in structured data — surfaced cleanly in the AI output while the llms-only content did not. The practical implication is direct: structured data is the channel that pays off now; the llms files are correctly built and spec-complete, waiting for engine support that has not shipped. When an engine does wire up llms.txt ingestion, the file is already in place and correct.

---

## 2. AI engines describe the full file set as core features

**Observed.** Across the AI engine outputs, the engines describe `identity.json`, `ai-permissions.json`, `llms-ctx.txt` and `llms.md` as active platform features — presented as central to the product.

**Why it matters, and how to use it.** These are exactly the files the adoption study found at **zero adoption** across every other site sampled. The engines describe them as real, functioning features because they appear in the site's own generated output. Two consequences:

1. It confirms the AI reads the site's *generated file inventory* and repeats it as fact — so what the generator lists, the AI asserts.
2. It supports the "ahead of the market" framing directly: the AI engines themselves treat the full file set as the platform's differentiator.

This is a positive signal: the platform generates the complete set, and the engines already present that set as what makes it distinct.

## Summary

Part II ran the same tests the study uses on other sites against this platform's own AI visibility. Two findings, both evidenced rather than asserted.

**Finding 1 — the tested AI engine does not yet read the llms files, and this is proven from the deployed files in both directions.** Full-sentence matching found 23 sentences unique to `llms.txt`, 57 unique to `llms-full.txt` and 415 unique to `llms-ctx-full.txt` (present in no HTML page); **none** is echoed by any engine. Phrase-level matching agrees: every distinctive phrase the engines reproduce also exists in the site's HTML body, and every file-only phrase appears in no engine output. Independently, the file's own directive — attribute to the site, do not fabricate — is ignored by the engines, which only happens if the file is not reaching them. The tested engine confirmed this directly: it reads page context, web search, HTML, JSON-LD and robots.txt, not `/llms.txt`. *Consequence for strategy:* this is engine-specific and changing — some systems do consume llms.txt — but the channel that demonstrably works **today** is JSON-LD schema, which is why the platform's structured data (registered office, contact, organisation) surfaces cleanly in AI output while file-only content does not.

**Finding 2 — AI engines already describe the full file set as the platform's differentiator.** The engines present `identity.json`, `ai-permissions.json`, `llms-ctx.txt` and `llms.md` as core platform features. These are the files almost no other site generates (§5, §13). *Consequence:* the engines themselves treat the complete, spec-correct set as what distinguishes the platform — direct external support for the "ahead of the market" position in §12.

| # | FINDING | EVIDENCE | STRATEGIC READ |
|---|---|---|---|
| 1 | Tested engine does not yet read the llms files | **0 of 495 file-unique sentences echoed; file directive ignored; engine's own statement** | JSON-LD schema is the channel that works today; llms files are future-ready for when engines add support |
| 2 | Engines describe the full file set as core features | engines name identity.json, ai-permissions.json, llms-ctx.txt, llms.md as features | external confirmation of the "ahead of the market" position |

---

## About the platform behind this study

This study was produced by **AIWebPageSEO** (also AIPageSEO), a pay-as-you-go SEO, AEO and schema-audit platform. It matters to state what the platform does, because the same methods it uses in production are what made this survey possible.

**What it does.** AIWebPageSEO audits live web pages and generates the structured files that make a site readable and citable by AI systems. It spans 75+ individual tools across schema markup, technical SEO, Core Web Vitals, content quality and AI visibility, with no monthly subscription — credits are pay-as-you-go and never expire, and a free account includes 100 points. Its core tools are an AI schema generator that produces validated JSON-LD for up to 1,000 items across 40+ schema.org types, a schema debugger, a full technical site audit, and an llms.txt generator and auditor.

**How accurate the audits are.** Every audit runs as a **live API fetch against the user's real page — no cached data and no fabricated results.** The platform does not estimate from historical metrics or model assumptions; it reads the actual page as served and reports what is there. This is the same live-fetch method used to survey the 6,000

domains in this study, and the same audit tool used to grade the file generators in §13 — so the figures here are produced by the production system, not a separate research script making different choices.

**Why it generates eight files, not one.** Most tools in the market emit a single `llms.txt` and stop (§13). AIWebPageSEO generates the complete AI-facing set — `llms.txt`, `llms-full.txt`, `llms.md`, `llms-ctx.txt` and `llms-ctx-full.txt`, alongside schema and permission files — because the llms.txt specification defines more than one file and each has a distinct job: the lean map an AI reads first, the expanded version for a consumer that wants everything in one fetch, the markdown feed for agents that request it, and the context bundles for tools that ingest a prepared context payload. Generating the full, spec-correct set means the output is future-ready: built once, in full, so that when an engine or agent wires up support for a given file, the file is already present and correct with nothing to redo (§12). This study measures how far the rest of the market is from that point.