

The GEO templates pack

The GEO templates pack is every file our page checks ask for, written out and ready to paste: robots.txt rules per crawler, an llms.txt skeleton, the structured-data blocks, a page skeleton and a fact sheet. The crawler tokens come from the same registry our access checker grades against, so they do not drift apart.

Fill every bracket before shipping — a placeholder that reaches production is a fact an engine will happily repeat back to your customers.

robots.txt

Crawler access

Two decisions, not one. **Search and answer agents** fetch pages to build the answer a user is reading right now and usually cite the source — blocking them removes you from AI answers, which is a visibility decision. **Bulk training crawlers** collect corpora for the next model — declining them is a licensing decision, and either answer is defensible. Conflating the two is the expensive mistake.

```
# --- Search and answer agents -----
# These build the answer a person is reading, and usually cite you.
# Blocking them removes you from AI answers.
User-agent: OAI-SearchBot      # OpenAI – ChatGPT search index
Allow: /
User-agent: ChatGPT-User      # OpenAI – user-initiated fetch
Allow: /
User-agent: Claude-SearchBot  # Anthropic – Claude search
Allow: /
User-agent: Claude-User      # Anthropic – user-initiated fetch
Allow: /
User-agent: PerplexityBot     # Perplexity – index
Allow: /
User-agent: Perplexity-User   # Perplexity – user-initiated fetch
Allow: /
User-agent: Applebot          # Apple – search and Siri
Allow: /
User-agent: DuckAssistBot     # DuckDuckGo – DuckAssist
Allow: /
User-agent: YouBot            # You.com
Allow: /

# --- Bulk training crawlers -----
# A separate, deliberate decision. Change Allow to Disallow to appear in
# AI answers without feeding the next training run.
User-agent: GPTBot            # OpenAI
Allow: /
User-agent: ClaudeBot         # Anthropic
Allow: /
User-agent: Google-Extended   # Google – Gemini training and grounding
Allow: /
User-agent: Applebot-Extended # Apple
Allow: /
User-agent: Meta-ExternalAgent # Meta
Allow: /
User-agent: CCBot             # Common Crawl – feeds many models
Allow: /
User-agent: Bytespider        # ByteDance
Allow: /
User-agent: cohere-ai         # Cohere
Allow: /

# Google-Extended does not affect Search indexing. Blocking Googlebot or
```

```
# Bingbot removes you from the indexes that AI search runs on top of.
Sitemap: https://www.example.com/sitemap.xml
```

robots.txt is a request honoured by reputable crawlers, not an enforcement mechanism. If you need actual prevention, that is authentication or edge rules — and a rule here will not tell you whether a firewall is blocking the agent anyway. The **access checker** (mentionbeat.com/ai-access) probes each agent live and catches exactly that.

/llms.txt

The llms.txt index

A Markdown file at the site root pointing at your most citable pages. Cheap insurance rather than a lever: Google says it is not needed for Search, some assistants read it, and adoption may standardise or fade. Ship it if it costs an hour.

```
# [Company]

> [One sentence: what the company does, for whom.]

## Start here
- [What [Company] does](https://www.example.com/): [Canonical one-liner.]
- [Pricing](https://www.example.com/pricing): [What it costs and what's included.]

## Products
- [[Product] hub](https://www.example.com/products/[product]): [Canonical definition.]
- [[Product] specifications](https://www.example.com/products/[product]/specs)

## Reference
- [Glossary](https://www.example.com/glossary)
- [[Company] on Wikipedia](https://en.wikipedia.org/wiki/[Company])
- [[Company] on Wikidata](https://www.wikidata.org/wiki/[Q-id])

## Company
- [About](https://www.example.com/about)
- [Contact](https://www.example.com/contact)
```

Generate one automatically from your sitemap with the **llms.txt generator** (mentionbeat.com/tools/llms-txt) rather than hand-keeping this file — a stale index is worse than none.

Organization + sameAs

Grounding the brand entity

This is the block most sites are missing, and it is the one that lets an engine tell your brand apart from a similarly named one. `sameAs` is the literal machine instruction "these all refer to the same thing".

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Organization",
  "@id": "https://www.example.com/#org",
  "name": "[Company]",
  "alternateName": ["[Short name]", "[Legacy name before the rebrand]"],
  "url": "https://www.example.com/",
  "logo": "https://www.example.com/[real-logo].png",
  "description": "[Canonical one-sentence definition, reused verbatim everywhere]",
  "foundingDate": "[YYYY]",
  "sameAs": [
    "https://en.wikipedia.org/wiki/[Company]",
    "https://www.wikidata.org/wiki/[Q-id]",
    "https://www.linkedin.com/company/[handle]",
    "https://github.com/[handle]"
  ]
}
</script>
```

Product

The product entity

Every value here has to match what a reader sees on the page and what your datasheet says. Structured data that contradicts the visible text is worse than none — a divergent number is the leading cause of a model hedging about you.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Product",
  "name": "[Full product name]",
  "alternateName": ["[Short name]", "[Legacy / alias name]", "[Model number]"],
  "category": "[Category / entity class]",
  "description": "[Canonical one-sentence definition – reused verbatim everywhere]",
  "url": "https://www.example.com/products/[product]",
  "image": "https://www.example.com/[descriptive-filename].jpg",
  "brand": {"@type": "Organization", "name": "[Manufacturer]"},
  "manufacturer": {"@id": "https://www.example.com/#org"},
  "sameAs": ["https://www.wikidata.org/wiki/[product Q-id]"],
  "additionalProperty": [
    {"@type": "PropertyValue", "name": "[Spec name]",
      "value": "[value + unit + standard or qualifier]"},
    {"@type": "PropertyValue", "name": "[Spec name]",
      "value": "[value + unit + standard or qualifier]"}
  ]
}
</script>
```

Units and qualifiers belong inside the value — "up to 200 m, IEC 61400-12-1 classified" rather than "long range". Models reproduce the qualifier when it sits beside the number.

FAQPage

Questions and answers

Mirror the visible block exactly. The visible questions are what get extracted; the markup only makes them unambiguous. Marking up answers that do not appear on the page is the one way this block can hurt you.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "FAQPage",
  "mainEntity": [
    {
      "@type": "Question",
      "name": "[Real buyer question, verbatim from the page]",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "[One-sentence direct answer, then the supporting detail]"
      }
    }
  ]
}
</script>
```

Article

Articles and guides

For anything dated: posts, guides, research notes. The author should be a real person entity with `sameAs` links, and `dateModified` should follow real edits rather than a scheduler.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Article",
  "headline": "[Title, matching the H1]",
  "description": "[The standfirst, answer-first]",
  "datePublished": "[YYYY-MM-DD]",
  "dateModified": "[YYYY-MM-DD]",
  "inLanguage": "en",
  "author": {
    "@type": "Person",
```

```

    "name": "[Real person]",
    "sameAs": ["https://www.linkedin.com/in/[handle]"]
  },
  "publisher": {"@id": "https://www.example.com/#org"}
}
</script>

```

Page skeleton

The page, top to bottom

A default for a product or solution page. Only two positions are fixed — the answer-first opening and the key facts — because the top of a page earns a disproportionate share of citations and is what survives truncation. Reorder the rest to suit the reader.

H1 [Brand] [Product] – [category]

[50-120 words. First sentence: "[Brand] [Product] is a [category] made by [Maker] that [does what] for [who]." Then the direct answer to the core question. No narrative above this block.]

H2 What is [Brand] [Product]?	<- first sentence is the definition
H2 Key facts	<- compact, self-contained, liftable
H2 How it works	<- real text steps, not a picture
H2 Why it matters	<- with a comparison table
H2 Applications	<- the true breadth, as crawlable text
H2 Specifications	<- a real <table>, units inline
H2 Proof	<- numbers, dates, standards, names
H2 What it is not designed for	<- honest scoping; models reward it
H2 Frequently asked questions	<- 6-10, visible, mirrored in JSON-LD
H2 Related	<- 2+ links, descriptive anchors, pointing back to the hub

Last updated: [date] <- visible, and matching dateModified

Fact sheet

One approved set of facts

The gate underneath everything else. If your own surfaces contradict each other, every downstream effort amplifies the noise — and contradiction is the primary cause of a model hedging about you. Keep one file, diff every surface against it, and give it an owner.

```
subject:           [Canonical name, exact casing]
aliases:           [Short name] · [Legacy name] · [Model numbers] · [Misspellings]
category:          [Entity class – what kind of thing this is]
maker:             [Legal entity name]
definition:        [One sentence, reused verbatim on the site, the datasheet,
                    the knowledge graph and every partner listing]
canonical_url:     https://www.example.com/products/[product]
specs:
  - name: [Spec]    value: [number + unit]    qualifier: [standard/condition]
  - name: [Spec]    value: [number + unit]    qualifier: [standard]
disambiguation:    [What this is NOT – the same-named things it gets merged with]
knowledge_graph:
  wikidata:        [Q-id]
  wikipedia:        [URL, or "none – notability not yet established"]
corroboration:     [Independent sources stating the same category and numbers]
not_for:           [Honest scoping – the uses this is wrong for]
owner:             [Name]
review_date:       [YYYY-MM-DD]
open_questions:    [Facts still awaiting sign-off – never publish these]
```