

OPEN STANDARD / RELEASE 1.0.0

Ad Space Manifest 1

A publisher-hosted discovery contract for public advertising inventory.

CANONICAL RESOURCE

/adspace.json

JSON is the deterministic contract. AI agents can help publishers implement it, but no consumer needs an LLM to parse it.

AD SPACE ATLAS / 30 AUGUST 2026

Contents

The normative contract, compatibility rules, producer workflow, deterministic consumer behavior, trust boundaries, and governance for Manifest 1.

Abstract	4
1. Resource and authority	4
2. Representation and HTTP behavior	4
3. Compatibility model	5
4. Common rules	5
4.1 Unknown and absent data	5
4.2 Open vocabularies	6
4.3 IDs and references	6
4.4 Time	6
4.5 Money	6
5. Top-level object	6
6. Publisher	7
7. Collection	7
8. Listing	7
9. Location	8
10. Placement and availability	8
11. Offer and commercial observations	9
12. Links	9
13. Producer process	10
14. Consumer process	10

15. Trust and verification	11
16. Interoperability	11
17. Security and privacy	11
18. Release and change policy	12
References	12

Release: 1.0.0

Status: stable

Published by: Ad Space Atlas

Schema: <https://adspaceatlas.com/standards/adspace/1.0.0/schema.json>

Abstract

Ad Space Manifest 1 is a small, publisher-hosted JSON format for publicly describing advertising opportunities, their concrete placements, current availability, commercial offers, and related resources.

The manifest is a discovery declaration. It does not authorize a seller, prove legal identity, conduct an auction, reserve inventory, accept creative, process payment, or replace ads.txt, sellers.json, OpenRTB, AdCOM, or OpenDirect.

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL are to be interpreted as described by BCP 14 when they appear in all capitals.

1. Resource and authority

The conventional resource is:

TEXT

`https://{publisher-origin}/adspace.json`

The publisher origin is the scheme, host, and port of the publisher's canonical public URL. HTTPS is REQUIRED. The path is exactly `/adspace.json` and remains stable across specification versions.

The resource MUST be public and MUST NOT require credentials, cookies, or a session. A consumer MUST NOT assume that an apex domain, [www](#) host, sibling subdomain, different port, or different scheme has the same administrator.

Serving the file demonstrates control of content at that origin. It does not by itself prove legal identity, inventory ownership, authorization to sell, or the truth of a commercial claim.

2. Representation and HTTP behavior

A successful producer response MUST:

- return HTTP 200;
- use `Content-Type: application/json` with optional `charset=utf-8`;
- encode one top-level JSON object as UTF-8;
- contain no duplicate object member names;
- contain no JSON `null` values;
- remain at or below 1,000,000 decompressed bytes; and
- satisfy a Manifest 1 envelope.

A producer SHOULD include ETag, Last-Modified, and a reasonable Cache-Control policy, support conditional GET, return 304 when appropriate, avoid redirects, and generate stable bytes while the underlying facts remain

unchanged.

A consumer MUST bound response bytes, time, redirects, and destination safety. The reference limits are a 20-second timeout, three redirects, and 1,000,000 decompressed bytes. Redirects MUST remain on the original HTTPS host in Manifest 1.

3. Compatibility model

Manifest 1 separates wire compatibility from authoring release metadata.

- `specVersion` selects the parser. Its only Manifest 1 value is the string "1".
- `$schema` identifies an immutable authoring release such as `1.0.0`.

An older Manifest 1 consumer MUST continue processing the known core of a newer same-major document. It MUST ignore unknown properties, preserve exact source bytes when acting as an archive, and map an unknown open-vocabulary value to local `unknown` behavior without invalidating unaffected facts.

A consumer that does not support a future `specVersion` MUST report `unsupported_version` and MUST NOT interpret that document as Manifest 1.

The following changes are compatible within Manifest 1:

- adding an optional property with documented absence behavior;
- adding an optional object or array;
- documenting another value for an open vocabulary;
- defining a link relation or namespaced extension; and
- adding examples or clarifying prose without changing accepted data or meaning.

The following changes require a new major specification version:

- adding a required property;
- removing or renaming a property;
- changing a property's type, units, scope, identity, or meaning;
- reusing an ID for a different entity;
- changing money or currency semantics;
- changing origin or redirect trust semantics; or
- adding behavior that cannot be safely ignored.

4. Common rules

4.1 Unknown and absent data

Every Manifest 1 object permits additional properties. Consumers MUST ignore unknown properties when interpreting the known core and MUST NOT present them as verified facts.

Producers SHOULD put non-standard data under `extensions`. Extension keys MUST be absolute HTTPS URIs. An extension MUST NOT change the meaning of a core property.

An unknown optional fact **MUST** be omitted. Producers **MUST NOT** guess, emit `null`, or use an empty string as a substitute. Zero is a known numeric value. An empty array is a known empty collection.

4.2 Open vocabularies

Open-vocabulary values use lowercase `snake_case` tokens. Experimental values **SHOULD** use absolute URIs. Consumers **MUST** preserve unknown values in source evidence, map the local dimension to `unknown`, and continue processing unaffected data.

4.3 IDs and references

IDs are opaque, stable, URL-safe strings. Collection and listing IDs are unique within a manifest. Placement and offer IDs are unique within their listing. An ID **MUST NOT** be reassigned to another real-world entity.

Reordering arrays does not change identity. A `collectionId` or `placementIds` reference **MUST** resolve within the same manifest scope.

4.4 Time

All date-time values are RFC 3339 instants with an explicit UTC offset. UTC `Z` form is **RECOMMENDED**. `generatedAt` describes when the complete snapshot was generated, not the time of every request.

4.5 Money

Money uses:

- `minorUnits`: a non-negative interoperable safe integer;
- `currency`: an uppercase ISO 4217 alphabetic code; and
- `basis`: an open vocabulary describing what the amount buys.

A current public price also includes `observedAt`. Commercial observations carry `observedAt` on the observation record.

Core basis values include `flat`, `placement`, `day`, `week`, `month`, `impression`, `thousand_impressions`, `click`, `action`, `other`, and `unknown`.

Different currencies **MUST** be represented separately. A consumer **MUST NOT** compare, total, convert, or sort across currencies without an explicit external conversion source and observation time.

5. Top-level object

Property	Requirement	Meaning
<code>\$schema</code>	Recommended	Immutable URL of the exact authoring release.
<code>specVersion</code>	Required	Major wire version. Manifest 1 uses <code>"1"</code> .
<code>generatedAt</code>	Required	Time this complete snapshot was generated.
<code>language</code>	Optional	Default BCP 47 language tag for human-readable text.
<code>publisher</code>	Required	Entity making the declaration.
<code>collections</code>	Optional	Publisher-defined groups of listings.
<code>listings</code>	Required	Complete set of listings currently declared. MAY be empty.
<code>links</code>	Optional	Typed links to related resources.
<code>extensions</code>	Optional	Namespaced extension data.

An empty `listings` array explicitly means the publisher currently declares no listings.

6. Publisher

Property	Requirement	Meaning
<code>name</code>	Required	Public publisher or media-owner name.
<code>url</code>	Required	Canonical HTTPS public URL. Its origin SHOULD match the manifest origin.
<code>description</code>	Optional	Plain-text public description.
<code>contactUrl</code>	Recommended	HTTPS advertising inquiry page or form.
<code>identifiers</code>	Optional	External identifiers with absolute-URI <code>scheme</code> and string <code>value</code> .
<code>extensions</code>	Optional	Namespaced extension data.

Descriptive strings are untrusted text. Consumers MUST NOT render them as HTML.

7. Collection

A collection is an optional grouping such as a publication, campaign, channel, event, creator project, or media package.

Property	Requirement	Meaning
<code>id</code>	Required	Stable manifest-local collection ID.
<code>name</code>	Required	Human-readable name.
<code>url</code>	Optional	Canonical HTTPS page.
<code>description</code>	Optional	Plain-text description.
<code>lifecycleStatus</code>	Optional	Open vocabulary. Core values: <code>active</code> , <code>ended</code> , <code>historical</code> , <code>unknown</code> .
<code>extensions</code>	Optional	Namespaced extension data.

8. Listing

A listing is one independently understandable advertising opportunity.

Property	Requirement	Meaning
<code>id</code>	Required	Stable manifest-local listing ID.
<code>collectionId</code>	Optional	Reference to a collection in this manifest.
<code>name</code>	Required	Public listing name.
<code>description</code>	Recommended	Plain-text explanation.
<code>url</code>	Recommended	Canonical HTTPS human-readable page.
<code>medium</code>	Required	Open vocabulary. Core: <code>physical</code> , <code>digital</code> , <code>hybrid</code> , <code>other</code> , <code>unknown</code> .
<code>lifecycleStatus</code>	Required	Open vocabulary. Core: <code>active</code> , <code>paused</code> , <code>ended</code> , <code>historical</code> , <code>unknown</code> .

Property	Requirement	Meaning
<code>modifiedAt</code>	Optional	Time the publisher last changed this listing declaration.
<code>location</code>	Optional	Coarse public location.
<code>placements</code>	Required	One or more concrete surfaces, positions, slots, or units.
<code>offers</code>	Optional	Current ways to acquire placements. MAY be empty or absent.
<code>links</code>	Optional	Related public resources.
<code>extensions</code>	Optional	Namespaced extension data.

Lifecycle, publication, and availability are distinct. Consumers MUST NOT collapse them into one boolean.

9. Location

Property	Requirement	Meaning
<code>label</code>	Optional	Human-readable public location.
<code>countryCode</code>	Optional	ISO 3166-1 alpha-2 uppercase code.
<code>region</code>	Optional	Region or administrative area.
<code>locality</code>	Optional	City or locality.
<code>extensions</code>	Optional	Namespaced geometry or venue data.

Publishers SHOULD disclose only location precision appropriate for a public advertising catalog.

10. Placement and availability

A placement describes what can be bought or sponsored. Its ID is scoped to its listing.

Property	Requirement	Meaning
<code>id</code>	Required	Stable placement ID.
<code>name</code>	Required	Human-readable placement name.
<code>description</code>	Optional	Plain-text details.
<code>surfaceCategory</code>	Optional	Open vocabulary for broad classification.
<code>availability</code>	Optional	Current capacity and observation time.
<code>creativeRequirements</code>	Optional	Public creative constraints.
<code>extensions</code>	Optional	Namespaced data, including AdCOM mappings.

Core surface categories include `website`, `app`, `clothing`, `vehicle`, `building`, `device`, `audio`, `video`, `out_of_home`, `other`, and `unknown`.

Availability contains:

Property	Requirement	Meaning
<code>status</code>	Required	Open vocabulary. Core: <code>available</code> , <code>limited</code> , <code>sold_out</code> , <code>unavailable</code> , <code>unknown</code> .
<code>availableUnits</code>	Optional	Non-negative safe integer.

Property	Requirement	Meaning
<code>totalUnits</code>	Optional	Positive safe integer.
<code>observedAt</code>	Required	Time this state was asserted or observed.
<code>extensions</code>	Optional	Namespaced extension data.

If both counts are present, `availableUnits` MUST NOT exceed `totalUnits`. Missing counts mean unknown quantity, not zero.

Creative requirements may contain a plain-text `description`, an `acceptedMediaTypes` array, and `extensions`. The core intentionally does not reproduce AdCOM, VAST, or every physical-production model.

11. Offer and commercial observations

An offer describes one current way to acquire one or more placements. Its ID is scoped to its listing.

Property	Requirement	Meaning
<code>id</code>	Required	Stable offer ID.
<code>placementIds</code>	Required	Non-empty references to placements in the same listing.
<code>pricingModel</code>	Required	Open vocabulary.
<code>price</code>	Conditional	Current public asking or fixed price.
<code>salesUrl</code>	Recommended	HTTPS inquiry, reservation, auction, checkout, or product URL.
<code>validFrom</code>	Optional	Start time for these terms.
<code>validThrough</code>	Optional	End time for these terms.
<code>observations</code>	Optional	Latest current commercial signals.
<code>extensions</code>	Optional	Namespaced extension data.

Core pricing models include `fixed_price`, `auction`, `sponsorship`, `pay_to_rank`, `rate_card`, `negotiated`, `other`, and `unknown`.

A fixed advertised amount requires `price`. A genuinely negotiated or unknown amount omits it. `validThrough` MUST NOT precede `validFrom`.

A commercial observation contains `kind`, `money`, `observedAt`, and optional `extensions`. Core kinds include `current_bid`, `highest_bid`, `asking_price`, and `reported_revenue`. Producers SHOULD publish at most one current observation for each kind in an offer. Consumers construct history from immutable manifest snapshots.

Reported revenue remains a publisher assertion and MUST NOT be presented as audited merely because it appears in an origin-controlled manifest.

12. Links

A link follows the RFC 8288 relationship model:

Property	Requirement	Meaning
<code>rel</code>	Required	Registered relation token or absolute URI.
<code>href</code>	Required	Absolute HTTPS target.
<code>type</code>	Optional	Expected media type.

Property	Requirement	Meaning
<code>title</code>	Optional	Human-readable label.

Consumers MUST NOT automatically crawl every link. Every relation needs an explicit fetch and trust policy.

The relation <https://adspaceatlas.com/standards/adspace/rerelations/authorized-sellers> may link to an existing ads.txt resource. A manifest does not recreate ads.txt records.

13. Producer process

- 1 Inventory public facts already supported by the publisher's system.
- 2 Create a manifest with stable IDs, publisher, generated time, and the complete current listing set.
- 3 Omit unknown facts and use `listings: []` when no inventory is currently declared.
- 4 Validate against the immutable schema and semantic rules.
- 5 Review every public claim, especially identity, money, basis, availability, capacity, dates, and links.
- 6 Publish at `/adspace.json` with the required HTTP behavior.
- 7 Give a consumer the exact URL only when the publisher wants that consumer to read it.

An AI coding agent MAY assist with implementation. The publisher remains responsible for reviewing the public response. The normative implementation prompt is published with this release.

When facts change, publish a complete current snapshot, retain stable IDs, update observation times only for facts actually re-observed, and change `generatedAt`. Producers SHOULD NOT embed unbounded history.

To withdraw inventory while keeping the integration, publish `listings: []`. To withdraw the integration, return 404 or 410. Consumers retain earlier evidence under their retention and legal policies.

14. Consumer process

- 1 Begin only from an exact URL offered to the consumer.
- 2 Normalize and verify HTTPS, exact path, credentials, query, fragment, host, and public destination.
- 3 Fetch manually with bounded redirects, timeout, and decompressed bytes.
- 4 Validate content type and UTF-8 bytes.
- 5 Reject duplicate JSON member names and malformed JSON.
- 6 Read `specVersion` and dispatch to the matching parser.
- 7 Run structural and semantic validation.
- 8 Preserve exact changed bytes and fetch metadata before mapping.
- 9 Map known facts as publisher reported, with unknown vocabulary fallback.
- 10 Append observations and reviewable versions without overwriting history.
- 11 Use ETag or Last-Modified on later reads and avoid duplicate snapshots for unchanged content.

An HTTP 404 or 410 means the current manifest is withdrawn, not that prior evidence never existed. Transient failures do not erase last-known-good data. Consumers SHOULD label freshness and failures explicitly.

15. Trust and verification

The following states are independent:

- exact bytes were observed at a time and URL;
- the serving origin was controlled;
- a publisher identity was corroborated;
- an advertising opportunity exists;
- the publisher may sell it;
- a commercial claim was independently verified; and
- Atlas approved a public catalog projection.

Consumers **MUST NOT** promote one state into another silently. A consumer-specific challenge belongs under a namespaced extension, not as a neutral core property.

16. Interoperability

- ads.txt declares advertising systems authorized to sell inventory. This manifest describes the inventory and current public offer.
- sellers.json identifies sellers and intermediaries. This manifest may link to it but does not reproduce it.
- OpenRTB and AdCOM model programmatic transactions and advertising objects. A placement extension may map to them.
- OpenDirect models authenticated direct-buying workflows. A manifest may link to an OpenDirect service but remains public discovery.
- Schema.org and JSON-LD may support web-page SEO. They are not the canonical manifest wire format.

17. Security and privacy

- Producers **MUST** publish only intentionally public data.
- Descriptions and labels **MUST** be treated as untrusted plain text.
- Consumers **MUST** enforce redirect-aware SSRF protection at every hop.
- Consumers **MUST** bound CPU, time, decompressed bytes, object depth, arrays, and strings.
- Links **MUST NOT** be fetched without a relation-specific policy.
- Public validator bodies **MUST** be bounded and **SHOULD NOT** be logged or retained.
- A manifest **MUST NOT** contain secrets, tokens, private addresses, unpublished personal contact data, or internal identifiers not intended for public disclosure.

18. Release and change policy

Every release publishes immutable schema, prose, examples, prompt, conformance fixtures, and a changelog entry. A change proposal includes real producer examples, compatibility classification, old/new reader behavior, fixtures, security and provenance impact, Atlas mapping impact, and rollout policy.

At least one producer and one independent consumer SHOULD exercise a new additive feature before it becomes recommended. Atlas-specific fields remain namespaced extensions until multi-consumer evidence supports a neutral core field.

References

- IAB Tech Lab, ads.txt 1.1: <https://iabtechlab.com/wp-content/uploads/2022/04/Ads.txt-1.1.pdf>
- IAB Tech Lab, OpenDirect: <https://iabtechlab.com/standards/opendirect/>
- IAB Tech Lab, AdCOM: <https://iabtechlab.com/standards/adcom-advertising-common-object-model/>
- RFC 7493, I-JSON: <https://www.rfc-editor.org/rfc/rfc7493.html>
- RFC 8288, Web Linking: <https://www.rfc-editor.org/rfc/rfc8288.html>
- RFC 9110, HTTP Semantics: <https://www.rfc-editor.org/rfc/rfc9110.html>
- RFC 9413, Maintaining Robust Protocols: <https://www.rfc-editor.org/rfc/rfc9413.html>
- JSON Schema 2020-12: <https://json-schema.org/draft/2020-12>