

AI Preferences
Internet-Draft
Intended status: Standards Track
Expires: 30 October 2026

P. Keller
Open Future
M. Thomson, Ed.
Mozilla
28 April 2026

A Vocabulary For Expressing AI Usage Preferences
draft-ietf-aipref-vocab-06

Abstract

This document defines a vocabulary for expressing preferences regarding how digital assets are used by automated processing systems. This vocabulary allows for the declaration of restrictions or permissions for use of digital assets by such systems.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://ietf-wg-aipref.github.io/drafts/draft-ietf-aipref-vocab.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-aipref-vocab/>.

Discussion of this document takes place on the AI Preferences Working Group mailing list (<mailto:ai-control@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/ai-control/>. Subscribe at <https://www.ietf.org/mailman/listinfo/ai-control/>.

Source for this draft and an issue tracker can be found at <https://github.com/ietf-wg-aipref/drafts>.

Note to Readers

As detailed below, this is a working document. Its contents DO NOT REFLECT CONSENSUS of the Working Group either in whole or part. Presence or absence of any particular text does not indicate consensus, and this document is published solely as a basis of further discussion.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 30 October 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Conventions and Definitions	3
3. Statements of Preference	4
3.1. Understanding Preferences	5
3.2. Applying Preferences	5
4. Vocabulary Definition	6
4.1. AI Model Training	6
4.2. Search	6
4.3. Vocabulary Extensions	7
5. Applying Statements of Preference	7
5.1. Combining Preferences	8
5.2. More Specific Instructions	8
6. Exemplary Serialization Format	9
6.1. Usage Category Labels	9
6.2. Preference Labels	9
6.3. Text Encoding	10
6.4. Syntax Extensions	10
6.5. Processing Algorithm	10
6.6. Alternative Formats	12
7. Security Considerations	12

8. IANA Considerations	12
9. References	12
9.1. Normative References	12
9.2. Informative References	13
Acknowledgments	13
Authors' Addresses	13

1. Introduction

This document defines a vocabulary of preferences regarding how automated systems process digital assets -- in particular, the training and use of AI models. This vocabulary can be used to describe the types of uses that a declaring party may wish to explicitly restrict or allow.

The vocabulary is intended to be used in jurisdictions where expressing preferences results in legal obligations, as well as where there are no associated legal obligations. In either case, expressing preferences is without prejudice to applicable laws, including the applicability of exceptions and limitations to copyright.

Section 3 defines the data model for AI Preferences. Section 4 defines the terms of the vocabulary. Section 5 explains how to use AI Preferences in a data processing application, and Section 6 describes a way to serialize preferences into a string. Section 5 describes a process for determining the preference for a category of use.

[ATTACH] defines mechanisms to associate preferences with assets. Other means of association might be defined separately in the future.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the following terms:

Asset:

A digital file or stream of data, usually with associated metadata.

Declaring party:

The entity that expresses a preference with regards to an Asset.

3. Statements of Preference

NOTE: This section does not yet have consensus; see "Note to Readers" above.

The vocabulary is a set of categories, each of which is defined to cover a class of usage for assets. Section 4 defines the core set of usage categories in detail.

A statement of preference -- or usage preference -- is made about an asset. A statement of preference follows a simple data model where a preference is assigned to each of the categories of use in the vocabulary. A preference is either to allow or disallow the usage associated with the category.

A statement of preference can indicate preferences about some, all, or none of the categories from the vocabulary. This can mean that the preference is unknown for some usage categories.

Some categories describe a proper subset of the usages of other categories. A preference that is stated for the more general category applies if the preference is unknown for the more specific category.

For example, a more general category might be assigned a preference that allows the associated usage. In the absence of any statement of preference regarding categories that are more specific subsets of that usage category, usage within those categories would also be allowed. An explicit preference regarding the more specific usage category can be used to disallow the more specific usage, while indicating that other usage within the more general category is permissible.

After processing a statement of preferences the recipient associates each category of use one of three preference values: "allowed", "disallowed", or "unknown". In the absence of a statement of preference, all usage categories are assigned a preference value of "unknown".

The process for consulting a statement of preference is defined in Section 5.

Different declaring parties might each make their own statement of preference regarding a particular asset. The process for managing multiple statements of preference is defined in Section 5.1.

An exemplary syntax for statements of preference is defined in Section 6.

3.1. Understanding Preferences

This document and [ATTACH] describe how statements of preference are associated with assets.

The goal of these specifications is to ensure that the recipient of an asset knows what preferences have been associated with the asset. What a recipient then does with that information depends on many factors; see Section 3.2.

There are also some caveats that need to be considered as it relates to understanding what the preferences for a given asset are (as opposed to what actions might then follow).

A recipient can only understand preferences expressed through mechanisms it has implemented. Those methods might be limited to those in [ATTACH] or it could also include other methods (see Section 1.3 of [ATTACH]). If a preference is associated with an asset using a method the recipient does not understand or recognize, the recipient will remain ignorant of that preference.

Depending on the way in which preferences are expressed, a recipient might be unable to tell the source of the preference. Unless the source is explicitly identified, no assumptions can be made about where a preference originates. For example, preferences in robots.txt (see Section 3 of [ATTACH]) only implies that a server is the source of those preferences.

A method of associating preferences with assets could explicitly define the source of the preferences, which might involve authentication. Otherwise, no assumptions can be made about the origin of preferences. The apparent source of preferences could be representing their own preferences, the preferences of others, or the synthesis of multiple preferences from different sources.

3.2. Applying Preferences

This specification provides a set of definitions for different categories of use, plus a system for associating simple preferences to each (allow, disallow, or unknown; see Section 3).

The categories of use in the vocabulary (see Section 4) describe concrete, observable outcomes that depend on the use of assets, they seek to avoid describing internal details of implementations or their architecture.

This specification only seeks to ensure that preferences can be understood, not provide a means of ensuring that preferences are respected. There may be considerations that take precedence over any stated preferences, especially when taking the public interest into account.

Enforcement is not provided by this specification. Preferences do not themselves create rights, obligations, or prohibitions. Other mechanisms -- technical, legal, contractual, or otherwise -- might enforce adherence or non-adherence to these preferences and thereby determine the consequences of not respecting a stated preference.

An entity that receives usage preferences has a choice about whether to respect the preferences it has discovered. It makes this choice according to its understanding of how the asset is used, how that usage corresponds to the usage categories where preferences have been stated, and the applicable legal context.

4. Vocabulary Definition

NOTE: This section does not yet have consensus; see "Note to Readers" above.

This section defines the categories of use in the vocabulary.

4.1. AI Model Training

The act of using an asset in the production or refinement of an AI model that can generate content in one or more modalities (text, image, audio, etc...).

4.2. Search

Use of an asset in an application where the primary purpose of the application is to select assets and direct users to the location of those assets.

This category of use only applies under the following conditions:

- * Where the presentation of an asset in search output -- if selected for presentation -- includes a direct reference or link to the original location from which the asset was retrieved.
- * When excerpts from the asset are displayed they serve to assist users in evaluating the relevance of the result.

This category does not include the use of assets to generate summaries.

Non-substantive changes to the presentation of titles or excerpts from assets are included for the purposes of accessibility. Translation, transcription, or text-to-speech are examples of non-substantive changes that could help users understand what is being presented. Where existing controls restrict presentation of these items, such as limitations on snippet size, those apply before any changes.

A preference to allow this category of use includes allowing any processing internal to the application that is performed on assets. Allowing this use is conditional on the outputs of any processing being exclusively used by the search application according to the other restrictions in this section. That includes the training of AI models using the assets and the use of those models provided that the resulting models and their outputs are used exclusively in ways that meet the above conditions.

4.3. Vocabulary Extensions

Extensions to this vocabulary need to be defined in an RFC that updates this document.

Any future extensions to this vocabulary MUST NOT introduce additional categories that include existing categories defined in the vocabulary. That is, new categories of use can be defined as a subset of an existing category, but not a superset.

Systems that use this vocabulary might define their own extensions as part of a larger data model. Section 6.6 describes how concepts from an alternative format might be mapped to this vocabulary.

5. Applying Statements of Preference

After acquiring a statement of preference, which might use the process in Section 6.5, an application can determine the status of a specific usage category as follows:

1. If the statement of preference contains an explicit preference regarding that category of use -- either to allow or disallow -- that is the result.
2. Otherwise, if the usage category is a proper subset of another usage category, recursively apply this process to that category and use the result of that process.
3. Otherwise, the preference for that category is unknown.

This process results in one of three potential answers: allow, disallow, and unknown. Applications can use the answer to guide their behavior.

One approach for dealing with an "unknown" outcome is to assign a default value. This document takes no position on what default might be assigned.

5.1. Combining Preferences

an application might receive multiple statements of preference, obtained using different methods or from different declaring parties. This might result in conflicting preferences.

Absent some other means of resolving conflicts, the following process applies to each usage category:

- * If any statement of preference indicates that the usage is disallowed, the result is that the usage is disallowed.
- * Otherwise, if any statement of preference allows the usage, the result is that the usage is allowed.
- * Otherwise, the preference for that category is unknown.

This process ensures that the most restrictive preference applies.

5.2. More Specific Instructions

A recipient of a statement of preferences that follows the model in Section 3 might receive more specific instructions in two ways:

- * Extensions to the vocabulary might define more specific categories of usage. Preferences about more specific categories override those of any more general category.
- * Contractual agreements or other specific arrangements might override statements of preference.

For instance, a statement of preferences might indicate a preference to disallow a category of use for an asset. If arrangements, such as legal agreements, exist that explicitly permit the use of that asset, those arrangements likely apply despite the existence of machine-readable statements of preference, unless the terms of the arrangement explicitly say otherwise.

6. Exemplary Serialization Format

This section defines an exemplary serialization format for preferences. The format describes how the abstract model could be turned into Unicode text or sequence of bytes.

The format relies on the Dictionary type defined in Section 3.2 of [FIELDS]. The dictionary keys correspond to usage categories and the dictionary values correspond to explicit preferences, which can be either y or n; see Section 6.2.

For example, the following states a preference to allow model training (Section 4.1), disallow search (Section 4.2), and preference for other categories other than subsets of these categories are unknown:

```
train-ai=y, search=n
```

6.1. Usage Category Labels

Each usage category in the vocabulary (Section 4) is mapped to a short textual label. Table 1 tabulates this mapping.

Category	Label	Reference
AI Model Training	train-ai	Section 4.1
Search	search	Section 4.2

Table 1: Mappings for Categories

These tokens are case sensitive.

Tokens defined for a new usage category can only use lowercase latin characters (a-z), digits (0-9), "_", "-", ".", or "*". These are encoded using the mappings in [ASCII].

6.2. Preference Labels

The data model in Section 3 used has two options for preferences associated with each category: allow and disallow. These are mapped to single byte Tokens (Section 3.3.4 of [FIELDS]) of y and n, respectively.

6.3. Text Encoding

Structured Fields [FIELDS] describes a byte-level encoding of information, not a text encoding. This makes this format suitable for inclusion in any protocol or format that carries bytes.

Some formats are defined in terms of strings rather than bytes. These formats might need to decode the bytes of this format to obtain a string. As the syntax is limited to ASCII [ASCII], an ASCII decoder or UTF-8 decoder [UTF8] can be used. This results in the strings that this document uses.

Processing (see Section 6.5) requires a sequence of bytes, so any format that uses strings needs to encode strings first. Again, this process can use ASCII or UTF-8.

6.4. Syntax Extensions

There are two ways by which this syntax might be extended: the addition of new labels and the addition of parameters.

New labels might be defined to correspond to new usage categories. Section 4.3 addresses the considerations for defining new categories. New labels might also be defined for other types of extension that do not assign a preference to a usage category. In either case, when processing a parsed Dictionary to obtain preferences, any unknown labels **MUST** be ignored.

The Dictionary syntax (Section 3.2 of [FIELDS]) can associate parameters with each key-value pair. This document does not define any semantics for any parameters that might be included. When processing a parsed Dictionary to obtain preferences, any unknown parameters **MUST** be ignored.

In either case, new extensions need to be defined in an RFC that updates this document.

6.5. Processing Algorithm

To process a series of bytes to recover the stated preferences, those bytes are parsed into a Dictionary (Section 4.2.2 of [FIELDS]), then preferences are assigned to each usage category in the vocabulary.

This algorithm produces a keyed collection of values, where each key has at most one value and optional parameters.

To obtain preferences, iterate through the defined categories in the vocabulary. For the label that corresponds to that category (see Table 1), obtain the corresponding value from the collection, disregarding any parameters. A preference is assigned as follows:

- * If the value is a Token with a value of *y*, the associated preference is to allow that category of use.
- * If the value is a Token with a value of *n*, the associated preference is to disallow that category of use.
- * Otherwise, the preference for that category is unknown.

Note that this last alternative includes the key being absent from the collection, values that are not Tokens, and Token values that are other than *y* or *n*. All of these are not errors, they only result in the corresponding preference being unknown.

It is important to note that if the same key appears multiple times, only the last value is taken. This means that duplicating a key could result in unexpected outcomes. For example, the following means that preferences are unknown, because the type of the parameter values (a boolean and a string respectively) are not tokens:

```
train-ai=y, train-ai, search=n, search="n"
```

If the parsing of the Dictionary fails, preferences are unknown. This includes where keys include uppercase characters, as this format is case sensitive (more correctly, it operates on bytes, not strings).

This document does not define a use for parameters. Where parameters are used, only those parameters associated with the value that is selected according to Section 4.2.2 of [FIELDS]. Parameters can therefore be carried for any preference value, including where preferences are unknown.

For example, the following train-ai preference has parameters even though the preference is unknown:

```
train-ai;has;parameters="?";
```

This process produces an abstract data model that assigns a preference to each usage category as described in Section 3.

6.6. Alternative Formats

This format is only an exemplary way to represent preferences. The data model described in Section 3 can be used without this serialization.

Any alternative format needs to define the mapping both from that format to the model used in this document and from the model to the alternative format. This includes any potential for extensions (Section 6.4).

The mapping between the data model and the alternative format does not need to be complete, it only needs to be clear and unambiguous.

For example, an alternative format might only provide the ability to convey preferences for a subset of the categories of use. A mapping might then define that an unknown preference is associated with other categories.

7. Security Considerations

Preferences are not a security mechanism. Section 3.2 addresses what it means to express a preference.

Processing a concrete instantiation of the exemplary format described in Section 6 is subject to the security considerations in Section 6 of [FIELDS].

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

- [ASCII] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <<https://www.rfc-editor.org/rfc/rfc20>>.
- [FIELDS] Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", RFC 9651, DOI 10.17487/RFC9651, September 2024, <<https://www.rfc-editor.org/rfc/rfc9651>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

9.2. Informative References

[ATTACH] Illyes, G. and M. Thomson, Ed., "A Vocabulary For Expressing AI Usage Preferences", Work in Progress, Internet-Draft, draft-ietf-aipref-attach-06, 28 April 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-aipref-attach-06>>.

[UTF8] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/rfc/rfc3629>>.

Acknowledgments

The following individuals made significant contributions to this document:

- * Cullen Miller
- * Erin Simon
- * Felix Reda
- * Krishna Madhavan
- * Laurent Le Meur
- * Leonard Rosenthol
- * Lila Bailey
- * Nate Hake
- * Sebastian Posth
- * Timid Robot Zehta

Authors' Addresses

Paul Keller
Open Future
Email: paul@openfuture.eu

Martin Thomson (editor)
Mozilla
Email: mt@lowentropy.net