

**Communicating access and usage policies to
crawlers using extensions to the
Robots Exclusion Protocol
Part 2: Extension of the Robots META
Tags format and other techniques for
embedding permissions in HTML
content**

A component of the ACAP Technical Framework

Implementation Version 1.1, 25 September 2009

Document history

Version	Release date
Pilot Version 1	2007-09-15
Pilot Version 2	2007-10-15
Pilot Version 3	2007-11-19
Implementation Version 1.0	2007-11-26
Implementation Version 1.0 with corrections	2007-11-30
Implementation Version 1.1 DRAFT 1	2009-05-29
Implementation Version 1.1 DRAFT 2	2009-07-06
Implementation Version 1.1 DRAFT 3	2009-07-09
Implementation Version 1.1 DRAFT 4	2009-08-19
Implementation Version 1.1 DRAFT 5	2009-09-02
Implementation Version 1.1	2009-09-25

Changes relative to previous version

1. General changes

- Removal of inconsistencies in vocabulary and presentation.
- Removal of text relating to features previously marked as not ready for implementation.
- All pages numbered.

2. Specific changes

Table of contents extended to include an additional subsection level.

Introduction of new features (indicated by [V1.1] in the text):

- Clarification of how crawlers are by default expected to interpret an element-specific permission or prohibition (expressed in a `class` attribute) that they don't fully "understand" (see Section 2.2.6).
- New ACAP Version META Tag added to allow the version of ACAP used in preparing the META Tags to be communicated (see Section 2.2.7).

Table of contents

Document history.....	ii
Changes relative to previous version	iii
Table of contents	iv
1 Introduction.....	1
1.1 Why publish ACAP Version 1.1.....	1
2 Description of ACAP extensions to the Robots META Tags format.....	2
2.1 Overview	2
2.2 Detailed description of the extended META tag format.....	3
2.2.1 Addressing permissions and prohibitions to crawlers.....	3
2.2.2 Usage purposes	4
2.2.3 Expressing permissions in META tags.....	4
2.2.3.1 Unqualified permissions.....	4
2.2.3.2 Qualified permissions	5
2.2.4 Expressing prohibitions in META tags.....	5
2.2.5 Expressing element-specific permissions and prohibitions as values of class attributes	5
2.2.6 [V1.1] Fallback interpretation of element-specific permissions and prohibitions	6
2.2.7 [V1.1] ACAP Version META Tag.....	6
2.2.8 Conflict resolution	7
2.2.9 Usage types and usage qualification.....	7
2.2.9.1 Permission qualified by location	7
3 Formal specification of the syntax of the proposed extensions to the Robots META Tags format.....	8
4 References.....	9

1 Introduction

ACAP (Automated Content Access Protocol) is being developed as an open industry standard to enable the providers of all types of content (including, but not limited to, publishers) to communicate permissions information (relating to access to and use of that content) in a form that can be readily recognized and interpreted by a search engine (or any other intermediary or aggregation service), so that the operator of the service is enabled systematically to comply with the individual publisher's policies. ACAP will provide a technical framework that will allow publishers worldwide to express access and use policies in a language that machines can read and understand.

The Robots Exclusion Protocol (REP) is the formal name for what is currently the most widely-used method of communicating permissions to web crawlers (also frequently referred to as 'robots' or 'spiders') [1]. This method of communication is in two parts: a format for a file called 'robots.txt' that contains machine-readable statements of which sets of resources on a website a crawler is permitted (allowed) or prohibited (disallowed) to crawl; and a format for embedding permissions in HTML page headers, called Robots META Tags.

This document is Part 2 of a two-part specification of a method of communication based upon proposed extensions to the Robots Exclusions Protocol. This part describes proposed extensions to the robots META tags format to meet the requirements of a series of use cases tested in the ACAP pilot project. The format proposed by this document has been tested against several important use cases and is considered to be ready for implementation for most use cases that involve communication to search engine crawlers of access and usage policies relating to publicly-accessible online content. The format has also been tested for use cases that involve similar communication of access and usage policies relating to online content that is *not* publicly-accessible, but it is recognised that further clarification and extension of the format may be needed in this area.

A companion document forms Part 1 of the specification[2], which contains proposed extensions to the robots.txt format and specifies a large part of the ACAP syntax for expressing access and usage policies that is also used in this document.

Both this document and its companion use access and usage terminology that is defined in the ACAP Dictionary of Access and Usage Terminology[3].

1.1 Why publish ACAP Version 1.1

ACAP Version 1.0 was published in November 2007. It has been adopted by more than 700 online publishers worldwide. It has not yet been implemented by any major crawler operators, and publisher implementations have in the main been strictly limited to mimic their existing use of the Robots Exclusion Protocol (REP).

The main barriers to implementation of ACAP are political, legal and commercial. The purpose of publishing ACAP Version 1.1 is to ensure that there are no technical barriers to implementation, by clarifying and extending the functionality of ACAP to meet known use cases for the communication of access and usage policies to web crawlers. Features that in ACAP Version 1.0 were marked as not ready for implementation have been reviewed and their intended uses have been clarified. Some new features have been added, and are indicated in the text by “[V1.1]”.

The proposals for extension of REP contained within this two-part specification will remain proposals until they are adopted and substantially implemented by major crawler operators as well as by online publishers.

DISCLAIMER. All ACAP features may change or be withdrawn without notice. All implementations of these proposals are entirely at the implementer’s own risk.

2 Description of ACAP extensions to the Robots META Tags format

2.1 Overview

This document proposes extensions to the Robots META Tags format to express a content owner’s policy for allowing or denying crawlers access to and use of their online content. These extensions do not replace the existing Robots META Tags format, but enable unambiguous expression of permissions, both unqualified and qualified by a range of restrictions¹, and outright prohibitions as to what a crawler and associated automated follow-on processes may or may not do with the resource in which the expression of these policies is embedded.

The ACAP extensions to the Robots META Tags format are designed to be used alongside META tags using the existing format. It will take time for crawlers to be programmed to recognise and use the proposed ACAP extensions.

A typical HTML page that uses these extensions will contain a sequence of META tags in the page header, containing ACAP permissions and prohibitions. The following example shows what an HTML page containing such META tags might look like in outline.

```
<HTML>  
<HEAD>
```

¹ Future revisions of this document are expected to include a method for positively expressing the absence of a restriction on a permission. This document only defines qualifiers that are used to communicate restrictions on permissions.

```
<TITLE>Page title displayed above browser window</TITLE>
<!-- Conventional REP: indexing permitted, following links prohibited. -->
<META name="robots" content="index nofollow">
<!-- The same expressed in ACAP syntax -->
<META name="robots" content="ACAP allow-index">
<META name="robots" content="ACAP disallow-follow">
</HEAD>
<BODY>
<!-- Content of page here... -->
</BODY>
</HTML>
```

NOTE – Throughout this two-part specification examples are presented in monospaced text on a pale green background.

2.2 Detailed description of the extended META tag format

The HTML element `META`, as defined in the W3C HTML Specification[4], contains several attributes of which only one, `content`, is required. The Robots META Tags format uses this attribute and also the `name` attribute.

ACAP extends the Robots META Tags format by using the `class` attribute to carry permission and prohibition data on elements within the body of an HTML page.

The order in which META tags are placed within the page header is insignificant.

In cases where fields within a record have contradictory or overlapping interpretations, a mechanism for resolving such conflicts is proposed below – see Section 2.2.6.

2.2.1 Addressing permissions and prohibitions to crawlers

A permission or prohibition in a META tag may either be addressed to a single named crawler or to “any crawler”. A permissions and prohibition contained in a META tag addressed to a named crawler overrides a permission or prohibition with the same usage purposes (if any) and usage types and matching the same component(s) contained in a META tag addressed to “any crawler”.

The Robots META Tags format as defined on the Robots Exclusion Protocol website recognises only one value of the `name` attribute of a META tag, which is “robots”. Some search engines recognise the name of a specific crawler in place of “robots”. ACAP makes use of this format extension.

In order to extend the Robots META Tags format without interfering in the interpretation of the existing format, all ACAP permissions, prohibitions and

definitions are distinguished by an initial token `ACAP` at the start of the value of the `content` attribute, for example

```
<META name="robots" content="ACAP allow-index">
```

or, for a named crawler:

```
<META name="crawler-name" content="ACAP allow-index">
```

The order in which permissions and prohibitions are expressed in META Tags is not significant.

2.2.2 Usage purposes

A usage purpose is a specific service or process served by one or more crawlers to which a permission or prohibition is addressed.

A META tag that addresses a permission or prohibition associated with a set of usage purposes may contain a usage purpose pattern following the initial `ACAP` token and preceding the remainder of the permission or prohibition string, for example:

```
<META name="crawler-identification" content="ACAP news allow-index">
```

2.2.3 Expressing permissions in META tags

2.2.3.1 Unqualified permissions

The basic syntax for expression of a permission as the value of the `content` attribute is as follows:

```
ACAP allow-usage
```

where *usage* is a standard usage type as proposed in Part 1 of this specification, e.g.

```
<META name="robots" content="ACAP allow-index">
```

If it is desired to associate a permission with a set of usage purposes, this can be expressed as follows:

```
ACAP usage-purpose-pattern allow-usage
```

where the syntax of *usage-purpose-pattern* is as proposed in Part 1 of this specification, e.g.

```
<META name="searchbot" content="ACAP news.search.com allow-index">
```

2.2.3.2 Qualified permissions

If it is desired to qualify a permission in some way, this can be expressed as follows:

```
ACAP allow-usage qualifiers
```

where the syntax of *qualifiers* is as proposed in Part 1 of this specification, e.g.

```
<META name="searchbot" content="ACAP allow-present-snippet  
max-length=100-chars">
```

The same mechanisms are available for qualifying permissions in META tags as in robots.txt, as specified in Part 1. For example, a permission may specify a particular component within the page to be used for indexing purposes, such as an element whose `class` attribute has value `abstract`, as in:

```
<META name="searchbot" content="ACAP allow-index  
must-use-resource=the-acap:extract:class:abstract">
```

2.2.4 Expressing prohibitions in META tags

The basic syntax for expression of an ACAP prohibition as the value of the content attribute is as follows:

```
ACAP disallow-usage
```

If it is desired to associate a prohibition with a set of usage purposes, this can be expressed as follows:

```
ACAP usage-purpose-pattern disallow-usage
```

As in the ACAP extensions to the robots.txt file format specified in Part 1, prohibitions may not be qualified.

2.2.5 Expressing element-specific permissions and prohibitions as values of class attributes

A basic permission or prohibition may be associated with an element in the body of an HTML page by assigning a special value to its `class` attribute.

A permission may be associated with an element in the body of an HTML page by assigning the following value to its `class` attribute:

`the-acap:allow-usage`

For example, a permission to index a DIV element may be expressed thus:

```
<DIV class="the-acap:allow-index">...Content of element...</DIV>
```

A prohibition may similarly be associated with an element by assigning the following value to its `class` attribute:

`the-acap:disallow-usage`

2.2.6 [V1.1] Fallback interpretation of element-specific permissions and prohibitions

In the event that a crawler is unable to interpret an element-specific permission or prohibition expressed in a `class` attribute because it is unable to process permissions or prohibitions at the element level, the specified usage is to be interpreted as prohibited for the HTML page as a whole.

If the crawler is able to process permissions or prohibitions at the element level, but is unable to interpret an element-specific permission or prohibition because it is unable to interpret the specified usage type, qualifiers or fallbacks involved, the method specified in Part 1 should be applied to replace the element-specific permission or prohibition with a fallback permission or prohibition (specific to the same elements) that the crawler is able to interpret.

2.2.7 [V1.1] ACAP Version META Tag

An ACAP Version META Tag communicates to all crawlers that the ACAP features used in this HTML page are those defined by the specified ACAP Version. It has the following proposed syntax:

```
<META name="robots" content="ACAP Version version-number">
```

where *version-number* is a string comprising integers separated by dots and is the ACAP version number, e.g. 1.1.

The ACAP Version META Tag is optional, but it is essential that this field be used in all cases where features of ACAP Version 1.1 or later are employed.

If an ACAP Version field is not included, a crawler is expected to ignore any ACAP field that contains features of all versions of ACAP other than Version 1.0.

If an ACAP Version META Tag is included and specifies Version *n.m* (for some integers *n* and *m*, where *n* indicates a major, non-backwards-compatible, ACAP Version and *m* indicates a minor, backwards-compatible, ACAP Version), a crawler is

expected either to interpret correctly, or to interpret using the fallback mechanism, all features of Versions *n.0* through *n.m* inclusive – i.e. all features of the specified and any preceding minor ACAP Versions back to and included the preceding major ACAP Version (*n.0*) – and ignore any features of ACAP Versions later than *n.m*.

If an ACAP Version META Tag is not included, a crawler is expected to ignore any ACAP field that contains features of all versions of ACAP other than Version 1.0.

2.2.8 Conflict resolution

If two META Tags contains conflicting permission and prohibition, both addressed to the same named crawler or both addressed to all crawlers, both applying to the same usage type and both applying to the same part of the HTML page, the usage shall be interpreted as prohibited for that part of the HTML page for which the conflict exists.

If a META Tag conflicts with a permission or prohibition field in a robots.txt file, the META Tag overrides the field in the robots.txt file.

NOTE – Permissions and prohibitions embedded in a resource, whether qualified or not, are semantically equivalent to permission and prohibition fields in a robots.txt file with a resource specification that specifies the single resource in which these permissions and prohibitions are embedded. If a crawler is capable of interpreting permissions and prohibitions embedded in resources, these always override permission and prohibition fields in a robots.txt file that are addressed to the same crawler and applying to the same resource (see Part 1 of this specification).

2.2.9 Usage types and usage qualification

The same usage types may be specified in permissions and prohibitions embedded in HTML pages as in robots.txt.

One qualifier may be used in permissions expressed in META tags that is not used in robots.txt. The `location` qualifier may be used to qualify permissions for any standard usage type.

2.2.9.1 Permission qualified by `location`

A permission may be restricted to be dependent upon the location of the crawled resource. The permission only applies if the URI specified by the `location` qualifier is the same as the URI where the crawled resource is actually located. A permission of this kind is expressed in the following way:

```
ACAP allow-usage location=URI
```

where *usage* is any of the standard usage types; and *URI* must match the URI for the resource as a whole, otherwise the usage is prohibited. For example:

```
<META name="robots" content="ACAP allow-crawl
location=http://myserver.com/legitimate-copy.htm">
```

NOTE – The use of the usage type `crawl` in this example may appear paradoxical, since the page must have been crawled for the qualified permission to be read. It should be interpreted to mean “if this page is not located where expected, it should be treated as if it were not permitted to crawl it”.

3 Formal specification of the syntax of the proposed extensions to the Robots META Tags format

A formal definition of the syntax of the ACAP extensions to the Robots META Tags format is given here, using the ABNF notation defined in IETF RFC 2234[5]. “URI” and “relative-part” are defined in IETF RFC 3986[6].

The token `<ACAP-meta-tag-name>` defines a syntax extension for the value of the `name` attribute on a META element in the header of an HTML (or XHTML) resource. The token `<ACAP-meta-tag-content>` defines a syntax extension for the value of the `content` attribute on a META element.

The syntax tokens `<crawler-name>`, `<usage-purpose-pattern>`, `<ACAP-usage-name>`, `<used-resource-type-name>`, `<resource-specification>`, `<qualifier-specification>` and `<name>` are defined in Part 1 of this specification. The token `<WSP>` is defined in IETF RFC 2234.

<code>ACAP-meta-tag-name</code>	= "robots" / crawler-name
<code>ACAP-meta-tag-content</code>	= "ACAP" 1*WSP (version / permission / prohibition)
<code>version</code>	= "version" 1*WSP "1.1"
<code>permission</code>	= basic-permission / qualified-permission
<code>basic-permission</code>	= [usage-purpose-pattern 1*WSP] "allow-" ACAP-usage-name ["-" used-resource-type-name]
<code>qualified-permission</code>	= [usage-purpose-pattern 1*WSP] "allow-" ACAP-usage-name ["-" used-resource-type-name] 1*(1*WSP qualifier-specification)
<code>prohibition</code>	= [usage-purpose-pattern 1*WSP] "disallow-" ACAP-usage-name ["-" used-resource-type-name]
<code>element-set</code>	= "the-acap:" ("idlist:" / "classlist:" / "taglist:") id-or-name *(", " id-or-name)

```
id-or-name = name-start-char *name-char
```

The tokens `<class-attribute-permission>` and `<class-attribute-prohibition>` define the syntax for embedding permissions and prohibitions respectively in values of class attributes in elements within the body of an HTML page.

```
class-attribute-permission = "the-acap:allow-"
                             (ACAP-usage-name / local-usage-name)

class-attribute-prohibition = "the-acap:disallow-" ACAP-usage-name
```

4 References

1. Robots Exclusion Protocol: An informal specification, based upon an original June 1994 “consensus” of robot authors and others, can be found on the web at <http://www.robotstxt.org/>, including guidance on both the robots.txt file format and the use of Robots META Tags. A number of extensions have been proposed by major search engine operators and others, and some of these extensions are in widespread use.
2. ACAP Technical Framework – Communicating access and usage policies to crawlers using extensions to the Robots Exclusion Protocol – Part 1: Extension of the Robots META Tag format and other techniques for embedding permissions in HTML content. Current version available at <http://www.the-acap.org/download.php?ACAP-TF-CrawlerCommunication-Part1-V1.1.pdf>.
3. ACAP Dictionary of Access and Usage Terminology. Current version available at <http://www.the-acap.org/download.php?ACAP-TF-Dictionary-V1.1.pdf>.
4. HTML 4.01 Specification – W3C Recommendation 24 December 1999. Current version available at <http://www.w3.org/TR/html401/>.
5. Augmented BNF for Syntax Specifications: ABNF (Internet RFC 2234), Internet Engineering Task Force, November 1997 – see <http://www.ietf.org/rfc/rfc2234.txt> for more details.
6. Uniform Resource Identifier (URI): Generic Syntax (Internet RFC 3986) , Internet Engineering Task Force, January 2005 – see <http://www.ietf.org/rfc/rfc3986.txt> for more details.