Guide to implementation of ACAP Version 1.1 Communication with Crawlers

A component of the ACAP Technical Framework

Issue 1, 9 October 2009

Document history

Version	Release date
Version 1.0 Issue 1	2008-03-18
Version 1.1 Issue 1	2009-10-09

Table of contents

Acknowledgements

Many of the examples provided in the Implementation Guide have been derived from the robots.txt files and META tags created and used by publishers to test ACAP Version 1.0. We are grateful to all members of the ACAP Technology Working Group for their assistance with reviewing this Implementation Guide prior to publication, and especially to the following for allowing us to use their test results as source material in preparing the examples:

- Gert François, Manager New Media, De Persgroep (www.persgroep.be)
- David Sommer, Commercial Director, MPS Technologies (www.mpstechnologies.com)
- Daniel Neethling, General Manager, Southern Newspapers, Media 24 (www.media24.com)
- Edgar Schouten, Reed Business Information – Netherlands (www.reedbusiness.nl)
- Paul Mostert, Elsevier Science (www.elsevier.com)
- Sébastien Richard, Director, Web Development, Exalead S.A. (www.exalead.com).

1 Introduction

ACAP enables publishers to communicate their policies for access to and use of their content in machine-readable forms to search engines and other content aggregators.

ACAP Version 1.0 was published in November 2007. It has been adopted by more than 1250 online publishers worldwide. ACAP Version 1.0 continues to provide the basis for a quick and simple implementation of ACAP, where the intention of the publisher is to indicate support for ACAP without making changes to their existing policies as expressed in existing robots.txt files and in web pages using Robots META Tags.

ACAP Version 1.1 clarifies and extends the functionality of ACAP for existing use cases, as well as addressing the needs of some additional use cases. We recommend that, for any implementation of ACAP beyond the quick and simple implementation described on the ACAP website, ACAP Version 1.1 should be implemented in preference to ACAP Version 1.0.

This Implementation Guide aims to help you understand and implement ACAP Version 1.1. For guidance and a tool to support a quick and simple way of implementing ACAP Version 1.0 go to the ACAP website and click on 'Implement ACAP'.

1.1 Why implement ACAP Version 1.1

ACAP Version 1.1 is designed to meet the needs of a specific set of use cases that focus on the activity of search engines and other "web harvesters" in relation to general web content; ACAP is being extended to support a variety of other business models for the delivery and use of online content. These use cases are not covered by ACAP Version 1.1 and are therefore not dealt with in this version of the Implementation Guide. For further information on how ACAP is being extended for application for business models other than search and web archiving, please visit the ACAP website.

ACAP Version 1.1 represents a small but significant upgrade of ACAP Version 1.0, clarifying existing features and including a small number of new features. There is no absolute need for existing implementations of ACAP Version 1.0 to upgrade – ACAP Version 1.1 is backwards-compatible with ACAP Version 1.0 – but new implementers should use the new version for anything other than a "quick and simple implementation" and existing implementers are advised to review the additional features in the new version before deciding whether or not to upgrade.

Search engines typically use software systems – often referred to as crawlers (also known as ‘robots’, 'bots' or ‘spiders’) – to crawl web content, follow links within it to other web content, index it, preserve (copy) it and present it in different ways to end-users. ACAP Version 1.1 enables publishers to express which of these usages are

permitted or prohibited for which online content and, in the case of permitted usages, express some restrictions on those usages. ACAP Version 1.1 also enables publishers to express that any usage other than the five basic usages (crawl, follow, index, preserve and present) are prohibited.

If you wish to communicate clear and precise policies as to how crawler-based aggregation services, including search engines, should use your website content, you probably need to implement ACAP Version 1.1.

From this point forward, when referring to "ACAP" without any explicit reference to version, we mean "ACAP Version 1.1".

1.2 Who need to read this Implementation Guide

This Guide is aimed principally at webmasters and website administrators, but is not a full technical specification. The full technical specifications of ACAP can be downloaded from the ACAP website. References to these specifications can be found at the end of this Guide.

1.3 ACAP and the Robots Exclusion Protocol

ACAP makes use of an existing communication protocol that is the most widely used means of communicating with search engine crawlers: the Robots Exclusion Protocol (REP). ACAP makes use of both the major components of REP:

- the "robots.txt" file that is placed on a web server specifically for communication with crawlers,. and
- the Robots META tags which may be embedded in the headers of individual HTML web pages.

In both cases, ACAP uses REP to communicate expressions of what uses of the specified content the publisher wishes to permit. By defining a set of extensions to REP ACAP enables publishers to express more precisely nuanced permissions than is possible using the conventional REP formats.

It is important to understand that REP takes a permissive approach to crawler communication. If a robots.txt file does not explicitly prohibit a usage, a crawler may well assume it to be permitted. For publishers used to the underlying assumptions of copyright, this may seem counter-intuitive, but it is the way that REP works in practice and the proper expression of policies using ACAP will take this into account. One consequence of this is that ACAP defines a usage verb "other", in addition to the five basic usage verbs already mentioned, to enable usages, other than those explicitly permitted, to be prohibited.

1.4 How this Guide works

This Guide aims to guide you through the process of working out how to use ACAP to express your content access and use policies. It does so in a series of steps, and at each step you will be presented with a series of options to choose from, each of which is illustrated by examples to assist you in selecting the options that most closely match your own use case. If you find that it is not clear which options to select or how to adapt the examples to your own case, please contact us for further advice.

If you find that none of the options seems to apply to your use case, it is possible that ACAP does not yet provide a solution for you, in which case we would be pleased to hear from you, so that we can see how ACAP might be extended to meet your needs.

However, it is also possible that ACAP already meets your needs, but we have simply not made that clear in the way that the options and examples are presented, in which case we would be delighted to put this right.

At various points you have the option of contacting us to discuss your precise requirements. Please follow this option, if that seems appropriate.

NOTE – All examples in the main part of this Implementation Guide are intended to be realistic but fictitious. Any similarity in names of crawler, domains or resources between the examples in this Guide and real-world examples are unintended and entirely coincidental. By contrast, the examples in the Annex are real-world examples provided by publishers who have tested ACAP Version 1.0 in real business use cases.

1.4.1 How to read the examples

The examples all show how ACAP policies can be expressed either within a robots.txt file or in META tags embedded in HTML pages. In the robots.txt examples the hash symbol # indicates that the remainder of the line is a comment and therefore would be ignored by a crawler.

In the META tag examples the tags are shown in the form specified by the HTML v4.01 specification. Only META tags and any associated comments are shown. These must be placed inside the HEAD element of an HTML page, in accordance with the HTML 4.01 DTD and specification.

1.5 Preparation for implementing ACAP

This Guide assumes that you will be using an existing robots.txt file to express most of the policies that apply to your online content, reserving the use of META tags for special cases in which the precise policy varies (at least in certain respects) from one HTML page to another. Other approaches may be appropriate in specific circumstances but are not covered by this Guide.

Is there already a robots.txt file on your website? If not, you will need to study guides to REP, including webmaster guidance provided by the major search engines, before creating a robots.txt file that communicates your web content access and use policies at least so far as is possible with conventional REP. For most practical purposes you cannot implement ACAP without first having a robots.txt file on your website.

For the foreseeable future it must be assumed that many crawlers will not be able to interpret ACAP policies, so we strongly recommend that you *add* ACAP policies to your robots.txt file *without removing its current contents*. Crawlers that understand your ACAP policies can be instructed to ignore the pre-existing content of your robots.txt file, as shown in this User Guide. For the same reason we also recommend that ACAP permissions be embedded in HTML pages by adding new META tags without replacing existing tags.

If your website content is managed using a web content management system (CMS), we advise you to talk to your CMS supplier before starting to implement ACAP Version 1.1. Your CMS may be able to assist you in managing your content access and usage policies, in which case the advice given in this Guide may not be the best way of meeting your requirements.

For what kinds of content do you need to express access and usage policies? ACAP enables expression of policies for general web content such as HTML pages. ACAP also provides support for expression of policies that depend upon license metadata embedded in photographic images using the PLUS Coalition's License Definition Format. ACAP does not yet enable the expression of complex permissions depending upon properties of audio-visual and other non-textual content.

It's easy to start implementing ACAP now! We strongly recommend that you convert your existing robots.txt file to use ACAP forms of expression, using the tool available for this purpose on the ACAP website. Using this tool your robots.txt file will be extended to include *exactly the same information that it already contains*, but expressed in ACAP terms. This will not only serve notice to crawlers that you are implementing ACAP, but will also reduce the amount of effort that may be involved in manually editing your robots.txt file to include ACAP expressions of your access and usage policies. If you wish to modify your robots.txt file to use the richer forms of expression provided by ACAP, we

1.6 What will and won't happen when you implement ACAP

ACAP is not a technical protection mechanism. For ACAP to function as intended, crawler operators must voluntarily re-program their crawlers to interpret and act upon ACAP forms of expression in robots.txt and embedded in web content. How a crawler interprets ACAP will be dependent upon how they are re-programmed to do so.

Without re-programming, a crawler will simply ignore ACAP expressions in both robots.txt and embedded in web content. This is fortunate, because it means that

implementing ACAP will not disrupt the way in which your website is crawled and indexed currently by search engines and other aggregators.

2 Step-by-step implementation guide

2.1 Step 1: Define the crawlers to which to address your access and usage policies

First you need to decide to which crawlers you need to address your access and usage policies. ACAP enables you to define which crawlers may or may not have access to your content.

Crawlers can be addressed in two ways:

- as distinct groups of one or more crawlers addressed by name
- as a single group of all crawlers.

Several crawlers to which the same policies apply can be addressed as a group. Crawlers are expected to check all policies expressed using ACAP to determine which policies apply to them.

ACAP enables crawlers to be addressed in these ways by replicating the functionality in conventional REP that is provided by the `User-agent` field in robots.txt, but using a syntax that enable crawlers to distinguish conventional REP from the ACAP extensions to REP.

ACAP extends the functionality of conventional REP to enable a single crawler to be addressed by name in a META tag within an HTML page. Multiple named crawlers may be addressed by META tags, but only one named crawler per META tag.

ACAP policies always override conventional REP policies that apply in the same circumstances, See Option 1C for details of how to communicate specifically and differently with all crawlers that can interpret ACAP policies.

2.1.1 Option 1A: Address all crawlers

Any policies that are addressed to all crawlers define ‘default’ policies that apply to all crawlers unless those policies are overridden by policies addressed to specific crawlers by name.

Example 1.1: Prohibit all crawlers from crawling your content

The most basic policy is one that prohibits crawlers in general from crawling your content. Because of the essentially permissive nature of REP, it is necessary to be explicit about prohibiting crawlers from crawling your content, but it is not necessary to explicitly *permit* crawlers to crawl your content – silence is taken as permission.

The following example shows how to express that you wish to prohibit all crawlers from crawling your content. The example uses both conventional REP and ACAP expressions in the same robots.txt file.

```
##ACAP version=1.1
# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
ACAP-disallow-crawl: /
```

Example 1.2: Restrict all crawlers to crawl specified content

In the following example, all crawlers are permitted to crawl only a specific page `/index.html`. This is achieved by prohibiting access generally, but permitting access to the one file.

```
##ACAP version=1.1
# Conventional policies here...
User-agent: *
Disallow: /
Allow: /$   # A crawler may give up if not allowed to crawl the root
Allow: /index.html$
# The 'Allow' field is a widely-recognised extension to conventional REP

# ACAP policies here...
ACAP-crawler: *
ACAP-disallow-crawl: /
ACAP-allow-crawl: /$
ACAP-allow-crawl: /index.html$
```

2.1.2 Option 1B: Address a group of named crawlers

A group of crawlers may be addressed by name. In most cases the name of a crawler is advertised every time it crawls a website by being included in each request for a resource from the site.

Restricting which search engine crawlers may have access to content can be done in one of two ways:

1. Explicitly state which named crawlers are permitted access and prohibit access to all others. In example 1.3 permission is given to crawlers named 'SearchBot1' and ‘SearchBot2’ to crawl all content, while all others are prohibited.

2. Explicitly state which crawlers are prohibited access and permit access to all others, regardless of whether or not they are able to interpret ACAP policies. In example 1.4 ‘SearchBot1' and ‘SearchBot2’ are prohibited access to all content.

Example 1.3: Permit only named crawlers to crawl content

```
##ACAP version=1.1

# Conventional policies here..
User-agent: *
Disallow: /

User-agent: SearchBot1
User-agent: SearchBot2
Allow: /

# ACAP policies here...
ACAP-crawler: *
ACAP-disallow-crawl: /

ACAP-crawler: SearchBot1
ACAP-crawler: SearchBot2
ACAP-allow-crawl: /
```

Example 1.4: Prohibit named crawlers from crawling content

```
##ACAP version=1.1

# Conventional policies here...
User-agent: SearchBot1
User-agent: SearchBot2
Disallow: /

User-agent: *
Allow: /

# ACAP policies here...
ACAP-crawler: SearchBot1
ACAP-crawler: SearchBot2
ACAP-disallow-crawl: /

ACAP-crawler: *
ACAP-allow-crawl: /
```

2.1.3 Option 1C: Address all crawlers that can interpret ACAP expressions differently from those that can only interpret conventional REP expressions

A crawler that has not been programmed to interpret ACAP expressions will not be able to act upon your policies in full. If your policies cannot be fully expressed in conventional REP, you will almost certainly wish to communicate different policies to those crawlers that can interpret ACAP.

In the first of the following two examples, crawlers that cannot interpret ACAP policies are prohibited from crawling the site.

Example 1.5: Permit only ACAP-enabled crawlers to crawl all content

In this example all crawlers that can only interpret conventional REP are prohibited from crawling content, while all crawlers that can interpreted ACAP expressions are permitted to crawl all content.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
```

```
ACAP-allow-crawl: /
...
```

Example 1.6: Permit ACAP-enabled crawlers to crawl all content and advise them to ignore conventional REP policies

In this example, the policies expressed in conventional REP are different to those expressed using ACAP. While ACAP policies should always override conventional policies, crawlers that can interpret ACAP policies can be expressly asked to ignore conventional REP policies, as in this example.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /
#...other conventional policies here

# ACAP policies here...
ACAP-ignore-conventional-records

ACAP-crawler: *
ACAP-allow-crawl: /
#... other ACAP policies here
```

2.2 Step 2: Define the purposes for which a crawler may use your content

In most cases crawlers crawl content for the purpose of using it in a specific service, such as in a general search service or in a news, sports or other themed service. However, in some cases the content accessed by a single crawler may be used for more than one distinct purpose: it may be used in more than one service, or a single service may be delivered in a variety of ways (e.g. through several different web addresses or portals). You may not wish all your content to be used for all the purposes for which a crawler is crawling your content.

ACAP makes it possible to communicate to a crawler that it may only crawl content for certain specified usage purposes only. Rather like a specific crawler name, a usage purpose is specified using a label that is recognised by the crawler. ACAP also makes it possible to communicate that a crawler may *not* crawl content for certain specified purposes. Permissions and prohibitions that relate to a specified set of services always override permissions and prohibitions for which no purpose is specified.

There are no standard labels for search engine services. If the crawler operator does not specify a particular service label to use, it is suggested that the web address (URI) for the service in question be used, as this usually is sufficient to identify uniquely the service in question. It is expected that individual search engine

operators will publish names or labels that their crawlers will recognise and associate with the various services that they deliver.

If there is no concern about which usages permitted resources are used for, then there is no need to specify an ACAP usage purpose.

Example 2.1: Prohibit specified crawler from crawling content for use in specified services

The following example denies crawler 'SearchBot1' access to a site for the purpose of using content in services identified respectively by use of the text patterns "`news.*`" and "`images.*`". Note that this restriction does not apply to any other crawler. These text patterns are examples of a simple form of regular expression in which the asterisk * is a 'wild card' character representing any sequence of characters at the end of the pattern, so it is assumed that the crawler 'SearchBot1' is capable of correctly interpreting such patterns.

```
##ACAP version=1.1

# Conventional policies here...

# ACAP policies here...
ACAP-crawler: *
ACAP-allow-crawl: /

ACAP-crawler: SearchBot1
ACAP-usage-purpose: news.*
ACAP-usage-purpose: images.*
ACAP-disallow-crawl: /
```

Example 2.2: Permit crawlers to crawl content only for use in a named service

In this example crawlers are only permitted to crawl content for use in services identified by the label "sports".

```
##ACAP version=1.1

# Conventional policies here...

# ACAP policies here...
ACAP-crawler: *
ACAP-disallow-crawl: /
ACAP-usage-purpose: sports
ACAP-allow-crawl: /
```

2.3 Step 3: Define the content for which usage is either permitted or prohibited

ACAP policies in robots.txt either permit or prohibit the crawling and use of specified content resources. In all of the examples shown in Steps 1 and 2, except example 1.2, the policies being expressed apply to all content on a website.

A strict use of conventional REP only permits access to be prohibited for the whole website or for named directories or named resources. ACAP enables usages to be explicitly permitted. This feature is based upon a widely-recognised extension to conventional REP that also enables a wider prohibition to be overridden for specific resources, using the 'Allow:' construct.

The resources to which a permission or prohibition policy applies are specified by what is known as a 'resource path pattern'. This is a string against which a crawler will match the resource path part of each content item's web address, i.e. the part of the web address that follows the host name (and occasionally the port number), beginning with the slash character '`/`'. A slash on its own represents all content on the website. A slash followed by any other characters represents some subset of resources on the website whose resource paths match the pattern of characters. Such patterns of characters are a simple form of regular expression in which asterisks and dollar signs have special meaning. For more information on resource path patterns see the full technical specification of ACAP Version 1.1.

The longer the resource path pattern, the more specific the set of resources that match the pattern will generally be. If one pattern represents a subset of the resources represented by another pattern, any policy using the former (narrower) pattern will override a contradictory policy using the latter (broader) pattern.

For example:

```
ACAP-disallow-crawl: /
```

is overridden by

```
ACAP-allow-crawl: /public/
```

for content items that match the resource path pattern '`/public/`' on that web server. This is interpreted to mean that a crawler is permitted to crawl a resource located in the directory `/public`, or in a sub-directory below this directory, but is prohibited to crawl resources that are located elsewhere.

The above example shows how a permission policy can override a prohibition policy. A prohibition policy can override a permission policy in the same way.

NOTE – It is advised that directory names in resource path patterns should be terminated by a slash, to avoid inconsistencies in interpretation by crawlers.

Example 3.1: Permit ACAP-enabled crawlers to crawl all content except if located in a specified directory

In this example the robots.txt file has been set to express that only crawlers that can interpret ACAP policies are permitted to crawl the website, while the ACAP policy states that any crawler is explicitly permitted to crawl and index the site as a whole, with the exception of the directory `/nocrawl`.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# explicitly allow access to the whole site...
ACAP-allow-crawl: /
# ...except for the content of directory /nocrawl
ACAP-disallow-crawl: /nocrawl/
```

Example 3.2: Permit all crawlers only to crawl content in specified directories

An alternative approach is to specify exactly which directories a crawler may crawl. In this example any crawler may crawl the website home page `/index.html` and specified directories, but otherwise is not permitted to crawl the site. This is expressed using both conventional REP extended by 'Allow:', as well as using the equivalent ACAP forms of expression.

```
##ACAP version=1.1

# conventional policies here...
User-agent: *   # addressed to all crawlers
# prohibit access to all directories...
Disallow: /     # but...
# permit access to the home page
Allow: /$   # A crawler may give up if not allowed to crawl the root
Allow: /index.html$
# permit access to the public directory and sub-directories
Allow: /public/
# permit access to the promotion directory and sub-directories
Allow: /promotion/
# permit access to the news directory and sub-directories
Allow: /news/
```

```
# ACAP policies here...
ACAP-crawler: *   # addressed to all crawlers
# prohibit access to all directories
ACAP-disallow-crawl: /   # but...
# permit access to the index page
ACAP-allow-crawl: /$
ACAP-allow-crawl: /index.html$
# permit access to the public directory and sub-directories
ACAP-allow-crawl: /public/
# permit access to the promotion directory and sub-directories
ACAP-allow-crawl: /promotion/
# permit access to the news directory and sub-directories
ACAP-allow-crawl: /news/
```

The following is a re-statement of example 3.2 but the ACAP expression shows how a 'resource set' can be defined in order to express a policy more concisely. The resource set '`permitted`' is defined to include all resources that match any of the resource path patterns '`/index.html`', '`/public/`', '`/promotion/`' or '`/news/`'. For more information on resource sets see the full technical specification of ACAP Version 1.1.

```
##ACAP version=1.1

# conventional policies here...
User-agent: *   # addressed to all crawlers
# prohibit access to all directories...
Disallow: /     # but...
# permit access to the home page
Allow: /$   # A crawler may give up if not allowed to crawl the root
Allow: /index.html$
# permit access to the public directory and sub-directories
Allow: /public/
# permit access to the promotion directory and sub-directories
Allow: /promotion/
# permit access to the news directory and sub-directories
Allow: /news/

# ACAP policies here...

ACAP-resource-set: permitted /$ /index.html /public/ /promotion/ /news/

ACAP-crawler: *   # addressed to all crawlers
# prohibit access to all directories
ACAP-disallow-crawl: /   # but...
# permit access to any resources in resource set 'permitted'
ACAP-allow-crawl: the-acap:resource-set:permitted
```

Example 3.3: Permit crawlers to crawl specific files and file types

Many search engine crawlers recognise an extension of the capability of conventional REP to use wildcard characters to create 'globs' or 'patterns' against which many resources sharing some common pattern of characters in their path or filename can be matched. The ACAP extensions to REP also enable such pattern matching. In this example the use of such resource path patterns is shown. The asterisk * represents zero or more characters and the dollar sign $ indicates that the preceding character must be the final character in the path.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *   # addressed to any crawler
# prohibit access to content generally...
Disallow: /
# but allow access to the home page /index.html
Allow: /$   # A crawler may give up if not allowed to crawl the root
Allow: /index.html$
# and to HTML files in the directory /public
Allow: /public/*.htm$
Allow: /public/*.html$

# ACAP policies here...
ACAP-crawler: *   # addressed to any crawler
# prohibit access to content generally...
ACAP-disallow-crawl: /
# but allow access to the home page /index.html
ACAP-allow-crawl: /$
ACAP-allow-crawl: /index.html$
# and to HTML files in the directory /public
ACAP-allow-crawl: /public/*.htm$
ACAP-allow-crawl: /public/*.html$
```

2.4 Step 4: Define policies for specific usages

Apart from the basic usage 'crawl', ACAP can be used to express policies that involve a number of other usages that are involved in the way that most search engines (and similar aggregators) deliver their services to end-users. This enables publishers to define more precisely how their content may be used. Permitting a crawler to crawl a resource enables a crawler to have access to a resource for subsequent use of that resource. If crawlers are permitted to crawl, but the policies are silent about other usages, crawlers will interpret silence as permission.

ACAP supports policies that express permission or prohibition of the following usages:

- **crawl** a content resource

- **follow** links from a resource to other resources
- **index** a resource
- **preserve** a persistent copy of a resource
- **present** a resource in some form to an end-user
- **other** usages not explicitly permitted or prohibited

The preceding steps have dealt with the crawl usage. All other usages depend upon being permitted to crawl, so it is not necessary to prohibit other usages if crawling is prohibited.

2.4.1 Option 4A: follow
Express whether or not crawlers may following links

Permission to **follow** a resource means that web links that it contains may be followed by the crawler in order to find other resources to crawl. However, it may not be necessary, desirable or perhaps even safe for a crawler to follow all the links on a website (some links are not safe for crawlers to follow, such as links that are queries on very large databases containing mapping or similar data). The ACAP follow usage provides a means for specifying permission and prohibition policies for the following of links.

Conventional REP supports the communication of permissions or prohibitions to follow links using META tags embedded in HTML pages, but does not support communication of such policies in robots.txt. ACAP supports the communication of such policies in both robots.txt and in META tags.

Example 4.1: Prohibit crawlers from following links found in specified content

In this example, crawlers are permitted to crawl the content of the directory `/public`, but are prohibited from following links in content within `/public/catalog`.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
# prohibit non-ACAP-enabled crawlers from crawling anything
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /

# allow access to /public/ via the root /
ACAP-allow-crawl: /$
ACAP-allow-crawl: /public/
ACAP-disallow-follow: /public/catalog/
```

Example 4.2: Use an embedded META tag to prohibit crawlers from following links found in a resource

In this example, crawlers are not permitted to follow links in the HTML page in which these META tags are embedded.

```
<META NAME="robots" CONTENT="nofollow"> <!-- conventional policy -->
<META NAME="robots" CONTENT="ACAP disallow-follow"> <!-- ACAP policy -->
```

2.4.2 Option 4B: index
Express whether or not crawled content may be indexed

Permission to **index** enables the derivation and storage of index entries from the crawled content (according to whatever indexing methods are employed by the crawler operator).

At this Step we show how policies for indexing entire resources may be expressed. Step 5 will show how indexing of a resource may be limited to fragments of text within a resource.

Example 4.3: Permit indexing of specified resources

In the simplest case, a publisher may wish to restrict indexing (and possibly other defined usages) to specific resources such as the pages created for public consumption. Assuming that these are to be found in the directory `/public` (and its sub-directories) then Example 4.3 shows how ACAP permissions are used to permit indexing of resources in this directory only. Although in this case the policy is silent about indexing resources outside this directory, the prohibition to crawl other resources means that only resources in `/public` can be indexed. Other usages are explicitly prohibited in this case.

```
##ACAP version=1.1

# Conventional policies
User-agent: *
Disallow: /

# ACAP policies
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /
# allow access to /public/ via the root /
ACAP-allow-crawl: /$
ACAP-allow-crawl: /public/
ACAP-allow-index: /public/
ACAP-disallow-other: /public/
```

Example 4.4: Prohibit indexing of image files

In this example, the crawler SearchBot1 is permitted to index content in the '/TEXTS' directory but is prohibited from indexing any image files, which are identified by the patterns of characters used in image filename extensions.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /

# Policies for SearchBot1
ACAP-crawler: SearchBot1
ACAP-allow-crawl: /$   # always allow crawlers access to the root /
ACAP-allow-crawl: /TEXTS/
ACAP-allow-index: /TEXTS/
ACAP-disallow-index: /TEXTS/*.gif$
ACAP-disallow-index: /TEXTS/*.png$
ACAP-disallow-index: /TEXTS/*.jpg$
```

The following is a restatement of example 4.4 using a defined resource set.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /

ACAP-resource-set: images /TEXTS/*.gif$ /TEXTS/*.png$ /TEXTS/*.jpg$

# Policies for SearchBot1
ACAP-crawler: SearchBot1
ACAP-allow-crawl: /$   # always allow crawlers access to the root /
ACAP-allow-crawl: /TEXTS/
ACAP-allow-index: /TEXTS/
ACAP-disallow-index: the-acap:resource-set:images
```

Example 4.5: Permit indexing of HTML content only

In this example, the crawler SearchBot1 is only permitted to index HTML pages in the '/TEXTS' directory but is prohibited from indexing any other content.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /

# Policies for SearchBot1
ACAP-crawler: SearchBot1
ACAP-allow-crawl: /$
ACAP-allow-crawl: /TEXTS/
ACAP-disallow-index: /TEXTS/
ACAP-allow-index: /TEXTS/*.html
```

Example 4.6: Use an embedded META tag to prohibit crawlers from indexing a resource

In this example, crawlers are not permitted to index the HTML page in which these META tags are embedded.

```
<META NAME="robots" CONTENT="noindex"> <!-- conventional policy -->
<META NAME="robots" CONTENT="ACAP disallow-index"> <!-- ACAP policy -->
```

2.4.3 Option 4C: preserve
Express whether or not copies of crawled content may be preserved

Permission to **preserve** enables the persistent storage of crawled content. Terms such as "store" and "cache" have been avoided in ACAP terminology, due to the risks that these terms would be misinterpreted.

The term "preserve" should not be understood to imply 'permanent preservation' in an archival sense, but does imply storage beyond the immediate requirements to process the crawled content for indexing or other immediate purposes. The meaning of the search engine term "cache", implying 'storage until re-crawled' is encompassed by the meaning of the ACAP term "preserve".

When a search engine stores crawled HTML content in a 'cache', it may be necessary to make changes to the way that the content is encoded in HTML to ensure that, when the cached copy is presented in an end-user's browser, it resembles as faithfully as possible the page that was originally crawled. The term

"preserve" therefore also conveys the sense that a copy of the crawled content is stored in a form that, when presented, closely resembles the original on the publisher's website, implying that some modification of the crawled content prior to storage may have been involved.

ACAP policies can express whether or not a search engine is permitted to preserve a copy of a crawled resource in its 'cache'. Step 5 will show how permission to preserve may be qualified to limit the time for which a copy of a crawled resource may be stored.

It should be noted that search engines usually generate snippets from cached copies of resources, so if they are prohibited from preserving a resource, they are usually prevented from presenting a snippet for that resource.

Example 4.7: Permit crawlers to index but not to preserve copies of resources

In this example, the crawler SearchBot1 is permitted to crawl and index web content in directory '`/public`', but not to preserve (i.e. cache) copies of the resources found there.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /

# Permissions for SearchBot1
ACAP-crawler: SearchBot1
ACAP-allow-crawl: /$
ACAP-allow-crawl: /public/
ACAP-allow-index: /public/
ACAP-disallow-preserve: /public/
```

Example 4.8: Permit crawlers to preserve copies of specified resources

In this example, the crawler SearchBot1 is permitted to crawl and index web content in directory '`/public`', and may cache HTML pages, but may not cache other content from that directory (e.g. images).

```
##ACAP version=1.1
```

```
# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /

# Permissions for SearchBot1
ACAP-crawler: SearchBot1
ACAP-allow-crawl: /$
ACAP-allow-crawl: /public/
ACAP-allow-index: /public/
ACAP-disallow-preserve: /public/
ACAP-allow-preserve: /public/*.htm$
ACAP-allow-preserve: /public/*.html$
```

Example 4.9: Use an embedded META tag to prohibit crawlers from preserving a resource

In this example, crawlers are not permitted to preserve the HTML page in which these META tags are embedded.

```
<!-- There is no conventional REP equivalent -->
<META NAME="robots" CONTENT="ACAP disallow-preserve"> <!-- ACAP policy -->
```

2.4.4 Option 4D: present
Express whether or not a representation of a crawled resource may be delivered for presentation in an end-user's browser

Permission to **present** enables a search engine or other aggregator to deliver representations of a crawled resource to an end-user's browser.

Resources can be represented in a variety of ways. ACAP supports the expression of policies for the presentation of the following representations of crawled resources:

- links to the original resource on the content owner's website
- snippets, usually generated by the search engine
- thumbnail images, also usually generated by the search engine
- preserved copies from the search engine cache, if necessary distinguishing between copies of the most recently-crawled version of a resource and an older version of the same resource
- the original resource retrieved by the search engine in real time from the content owner's website.

Example 4.10: Permit presentation of some but not all representations of specified resources

In this example, a publisher permits the specified crawler SearchBot1 to create and present snippets for HTML pages in the '/news' directory, and may also present thumbnails and links, but is not permitted to present other representations (e.g. cached or original copies) of the same content.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /

# Policies for SearchBot1
ACAP-crawler: SearchBot1
ACAP-allow-crawl: /$
ACAP-allow-crawl: /news/
ACAP-allow-index: /news/
ACAP-allow-preserve: /news/
ACAP-disallow-present: /news/
ACAP-allow-present-snippet: /news/*.htm$
ACAP-allow-present-thumbnail: /news/*.htm$
ACAP-allow-present-links: /news/*.htm$
ACAP-allow-present-snippet: /news/*.html$
ACAP-allow-present-thumbnail: /news/*.html$
ACAP-allow-present-links: /news/*.html$
```

Example 4.11 Prohibit thumbnails images from being presented

In this example all forms of presentation are permitted except for thumbnails.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally
ACAP-disallow-crawl: /

# Policies for SearchBot1
ACAP-crawler: SearchBot1
ACAP-allow-crawl: /$
```

```
ACAP-allow-crawl: /news/
ACAP-allow-index: /news/
ACAP-allow-preserve: /news/
ACAP-allow-present: /news/
ACAP-disallow-present-thumbnail: /news/
```

Example 4.12: Use an embedded META tag to prohibit crawlers to present a resource snippet

In this example, crawlers are not permitted to present a snippet for the HTML page in which these META tags are embedded.

```
<!-- There is no conventional REP equivalent -->
<!-- ACAP policy -->
<META NAME="robots" CONTENT="ACAP disallow-present-snippet">
```

Example 4.13: Prohibit presentation of cached copies of a resource, but permit presentation of the original resource (retrieved in real time from the publisher's website), provided it is an HTML page

In this example, ACAP is used to instruct all crawlers to present the original version only of all resources in the directory '/news'.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# prohibit access to content generally...
ACAP-disallow-crawl: /
# ...but allow the content of /news to be crawled
ACAP-allow-crawl: /$
ACAP-allow-crawl: /news/
ACAP-allow-index: /news/
ACAP-allow-preserve: /news/
ACAP-allow-present: /news/
ACAP-disallow-present-currentcopy: /news/
ACAP-disallow-present-oldcopy: /news/
ACAP-disallow-present-original: /news/
ACAP-allow-present-original: /news/*.htm$
ACAP-allow-present-original: /news/*.html$
```

Example 4.14 Permit a preserved copy to be displayed, provided it is the most recently-crawled version

In this example, ACAP is used to prohibit access to the directory '`/archive`' for all crawlers with the exception of crawler called SearchBot1, which is given permission to preserve copies of the resources in '`/archive`' and to present these preserved copies.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Allow: /   # implied, but included for clarity
Disallow: /archive/

# ACAP policies here...
ACAP-crawler: SearchBot1
# permit access all content, including /archive
# ACAP-allow-crawl: /   # implied, but shown for clarity
ACAP-allow-crawl: /archive/   # overrides conventional policy
# ACAP-allow-index: /archive/   # implied, but shown for clarity
# ACAP-allow-preserve: /archive/   # implied, but shown for clarity
ACAP-allow-present: /archive/
ACAP-disallow-present-oldcopy: /archive/
# ACAP-allow-present-currentcopy: /archive/  #implied
```

2.5 Step 5: Define restrictions upon basic usages

ACAP policies may be refined to place restrictions upon basic usages. ACAP supports the current types of restrictions:

- time limit for indexing, preserving or presentation
- which resource to use for indexing
- maximum length for presentation of a snippet
- modification prior to presentation
- presentation context.

2.5.1 Option 5A: Time-limit restriction of permissions for basic usages

There are three ways of expressing a time limit in an ACAP policy, all of which may be applied to the basic usages index, preserve and present. These are:

1. until the resource is re-crawled

2. until a specified date after which the permitted usage ceases

3. for a specified number of days after the resource is first indexed, after which the permitted usage ceases.

Example 5.1 Permit content to be indexed, preserved and presented until recrawled

Although this is often the default behaviour of a search engine, it may be beneficial to make explicit the restriction that a resource may only be indexed and preserved until it is next re-crawled, implying that older versions of the resource may not remain in the index or be preserved. This example uses ACAP to permit the crawler SearchBot1 to preserve the index page '`index.html`' until it is re-crawled.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: SearchBot1
ACAP-disallow-crawl: /
# allow to crawl /index.html
ACAP-allow-crawl: /$
ACAP-allow-crawl: /index.html$
ACAP-allow-index: /index.html time-limit=until-recrawled
ACAP-allow-preserve: /index.html time-limit=until-recrawled
```

Example 5.2: Permit a copy of a resource to be indexed and preserved until a specified date

In this example, ACAP is used to tell all crawlers that content in the '`/serial`' directory may only be indexed and preserved until the specified date (after which it is to be moved into an archive).

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# allow /serial to be crawled
ACAP-allow-crawl: /$
ACAP-allow-crawl: /serial/
ACAP-allow-index: /serial/ time-limit=until-2008-03-01
ACAP-allow-preserve: /serial/ time-limit=until-2008-03-01
```

Example 5.3: Use an embedded META tag to permit crawlers to index and preserve a resource until a specified date

If varying policies are to be applied to different resources, it may be easier to present this in the Robots META tags format, the date being set as appropriate for each case.

In this example the crawler SearchBot1 is permitted to index and preserve the HTML page in which these META tags are embedded until the specified date.

```
<!-- There is no conventional REP equivalent -->
<!-- ACAP policy -->
<META NAME="SearchBot1"
      CONTENT="ACAP allow-preserve time-limit=until-2008-03-01">
```

Example 5.4: Permit a resource to be indexed and preserved for a specified number of days

This example shows a time limit of 14 days for indexing and preserving content in the directory '`/current`' for crawler SpecialBot1. Such a limit might be set by prior arrangement with a specialist search engine, which only crawls infrequently and accepts a variety of limits on how long it may hold content in its index, but this is unlikely to be appropriate when communicating with search engines in general.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: SpecialBot1
ACAP-disallow-crawl: /
# allow to crawl /current
ACAP-allow-crawl: /$
ACAP-allow-crawl: /current/
ACAP-allow-index: /current/ time-limit=14-days
ACAP-allow-preserve: /current/ time-limit=14-days
```

2.5.2 Option 5B: Permit indexing of content, but specify resource to be used for indexing

ACAP can be used to specify a particular resource to be used for indexing. This may be specified in a number of ways.

Warning! Some of the ways in which this feature may be used could be treated by search engines as 'cloaking' – an attempt to deceive a search engine as to the true nature of the content of a page – which could cause a website to be blacklisted, so

this feature should be used with caution. In some cases, a special arrangement may need to be made with search engines to have these features accepted by their crawlers.

If the crawled resource contains text, and is in a format that the indexing processes can handle (e.g. plain text, HTML, XML or PDF), the whole of the crawled content will normally be used for indexing purposes. For the publisher this may be inappropriate.

If, on the other hand, the crawled resource is an object that does not contain text, but is an image or some other non-text object, or is text in an unreadable format, the search engine will not be able to index it directly. If the object is embedded in an HTML page, or the crawler has followed a link to it from an HTML page, the indexing process will use text in the referring page, close to the embedded or linked object, to index it. From the publisher's point of view this, too, may be inappropriate.

In the case of an HTML page, the indexing of that page can be specified to be any of the following:

- an extract from the page comprising a single HTML element within the body of the page, specified by the value of its 'class' attribute
- an extract from the page comprising a single HTML element within the body of the page, specified by the value of its 'id' attribute
- the value of the 'content' attribute on a META tag within the page header, specified by its 'name' attribute.
- a separate resource specified by its URI

In the case of any other type of content, the indexing of that resource can be specified to be a separate resource specified by its URI.

It is inadvisable to specify a separate resource to be used to index a crawled resource without checking first that this is acceptable to the search engines concerned, because this is viewed by many search engines as a form of cloaking.

A future version of ACAP is expected to support the embedding of policies in non-text content, in which case it will become possible to specify within the non-text content how it should be indexed.

Example 5.5: Permit HTML content to be indexed using only text specified by 'class' attribute value

In the following example the crawler SearchBot1 may crawl and index pages in the directory `/articles`, but only using an element (assumed to exist in each page) whose 'class' attribute has the value 'abstract'.

```
##ACAP version=1.1
```

```
# Conventional policies
User-agent: *
Disallow: /

# ACAP policies
ACAP-crawler: SearchBot1
# prohibit access to content generally
ACAP-disallow-crawl: /
# allow access to /articles/
ACAP-allow-crawl: /$
ACAP-allow-crawl: /articles/
ACAP-allow-index: /articles/ must-use-resource=the-acap:extract:class:abstract
```

Example 5.6: Use an embedded META tag to permit crawlers to index a resource using only text specified by 'id' attribute value

The use of 'id' attribute values for specifying text to be used for indexing purposes is most likely to be practical when the policy is embedded in a META tag.

```
<!-- There is no conventional REP equivalent -->
<!-- ACAP policy -->
<META NAME="SearchBot1"
CONTENT="ACAP allow-index must-use-resource=the-acap:extract:id:i012345">
```

Example 5.7: Permit HTML content to be indexed using only text in the 'content' value of a specified META tag

In the following example the crawler SearchBot1 may crawl and index pages in the directory `/articles`, but only using the 'content' value of a META tag with a specified value of its 'name' attribute (assumed to exist in each page header).

```
##ACAP version=1.1

# Conventional policies
User-agent: *
Disallow: /

# ACAP policies
ACAP-crawler: SearchBot1
# prohibit access to content generally
ACAP-disallow-crawl: /
# allow access to /articles/
ACAP-allow-crawl: /$
ACAP-allow-crawl: /articles/
ACAP-allow-index: /articles/ must-use-resource=the-acap:extract:meta:index-text
```

2.5.3 Option 5C: Permit presentation of snippets with maximum snippet length restriction

ACAP allows the expression of permission to present snippets to be qualified through the specification of a maximum snippet length. The length can be expressed either as a number of characters or words.

Example 5.8 Permit presentation of snippets with a restriction on maximum length, expressed in robots.txt

In this example, all crawlers are permitted to index and present snippets for crawled content in the directory '`/news`' but snippets are restricted to 20 words in length.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# permit to crawl /news
ACAP-allow-crawl: /$
ACAP-allow-crawl: /news/
# ACAP-allow-index: /news/   # implied, but included for clarity
# ACAP-allow-preserve: /news/   # implied, but included for clarity
# ACAP-allow-present: /news/   # implied, but included for clarity
ACAP-allow-present-snippet /news/ max-length=20-words
```

Example 5.9: Use an embedded META tag to permit crawlers to present a resource snippet with restricted maximum length

If varying policies are to be applied to different resources, it may be easier to present this in the Robots META tags format, the maximum length of snippet being set as appropriate for each case.

```
<!-- There is no conventional REP equivalent -->
<!-- ACAP policy -->
<META name="robots"
      content="ACAP allow-present-snippet max-length=20-words">
```

2.5.4 Option 5D: Permit presentation of snippets, but specify text to be used

ACAP can be used to specify a particular resource to be used as a snippet in search results. This may be specified in a number of ways, but the simplest way of doing so is shown here.

Warning! Some of the ways in which this feature may be used could be treated by search engines as 'cloaking' – an attempt to deceive a search engine as to the true nature of the content of a page – which could cause a website to be blacklisted, so this feature should be used with caution. In some cases, a special arrangement may need to be made with search engines to have these features accepted by their crawlers.

Example 5.10: Use an embedded META tag to permit crawlers to present a resource snippet containing specified text

```
<!-- There is no conventional REP equivalent -->
<!-- ACAP policy -->
<META name="snippet" content="This is the snippet text to be presented">
<META name="robots"
      content="ACAP allow-present-snippet
            must-use-resource=the-acap:extract:meta:snippet">
```

2.5.5 Option 5E: Permit presentation of content but prohibit specified modifications

There can be many reasons why publishers do not want the form, layout or context of their content changed or manipulated. ACAP enables expression of policies that permit presentation of content but without modification to:

- data format
- typographic style / layout
- text translation
- text annotation, for example with end-user ratings or 'tags'
- any modification, including all the above.

Example 5.11 Permit presentation but prohibit changes to format (PDF to HTML)

ACAP enables publishers to specify that no modification is permitted to the data format of the content to be presented to the end-user. A common example of this is where PDF content is converted to HTML, disrupting layout and in the worst case, rendering the transformed content meaningless. Another example would be to prohibit image format transformations.

In the example, permission is given for the preserved copy of resources to be presented but no changes to the format are permitted.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /
```

```
# ACAP policies here...
ACAP-crawler: *
# permit to crawl content generally
ACAP-allow-crawl: /
# ACAP-allow-index: /   # implied, but included for clarity
# ACAP-allow-preserve: /   # implied, but included for clarity
# ACAP-allow-present: /   # implied, but included for clarity
ACAP-allow-present-currentcopy: / prohibited-modification=format
```

Example 5.11 Use an embedded META tag to permit crawlers to present a resource but prohibit transformation of the format (e.g. to PDF or plain text)

In this example the prohibition of format change is embedded in the HTML page.

```
<!—ACAP policies -->
<META name=”robots”
content=”ACAP allow-present-currentcopy prohibited-modification=format”>
```

Example 5.12: Permit presentation but prohibit change of layout

Layout and presentation of content is generally integral to a publisher’s brand. ACAP enables publishers to prohibit any changes to typographic layout when content is presented to the end-user.

Publishers should bear in mind that it is unreasonable to expect a search engine or other aggregator to maintain the layout of content if it is difficult or impossible for this to be done using a copy of the content stored in the aggregator's cache. Publishers are advised to make it as easy as possible for a requested restriction on layout changes to be complied with.

In this example, all search engines have permission to index and to present HTML content on the website (by filename pattern) but if presenting a cached copy, this should be presented without any modification to typographic style or layout.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
ACAP-disallow-crawl: /
# permit to crawl HTML content only
ACAP-allow-crawl: /$
ACAP-allow-crawl: *.htm$
ACAP-allow-crawl: *.html$
```

```
ACAP-allow-present-currentcopy: *.htm$ prohibited-modification=style
ACAP-allow-present-currentcopy: *.html$ prohibited-modification=style
```

Example 5.13: Permit presentation but prohibit translation of resources

In this example, presentation of translated versions of content are not permitted.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# permit to crawl content generally
ACAP-allow-crawl: /
# ACAP-allow-index: /   # implied, but included for clarity
# ACAP-allow-preserve: /   # implied, but included for clarity
ACAP-allow-present: / prohibited-modification=translation
```

Example 5.14: Permit presentation but only without any end-user annotations

In certain situations, publishers may wish to prevent their content from being presented by an aggregator with end-user ratings or tags, for example peer-reviewed scientific, technical or medical content. In the example, aggregators are prohibited from presenting content with any associated end user ratings or tags.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# permit to crawl content generally
ACAP-allow-crawl: /
# ACAP-allow-index: /   # implied, but included for clarity
# ACAP-allow-preserve: /   # implied, but included for clarity
ACAP-allow-present: / prohibited-modification=annotation
```

2.5.6 Option 5F: Permit presentation of content but require or prohibit specified presentation 'contexts'

The context in which content is presented is important: what surrounds the content or is beside it when viewed by the end-user in their browser. Search engines define the

context in which search results are presented and in many cases they are either unable or unwilling to allow the publisher to restrict how they do this.

There are some limited circumstances in which a publisher might reasonably request a restriction on the context of presentation, and these all relate to the presentation of the content in full.

ACAP enables a publisher to express a policy that restricts the context of presentation in the following limited cases:

- when presenting a preserved copy of the crawled content, it may not be presented in a frame added by the aggregator
- when presenting the original crawled content, it must be presented in the frame as intended by the content owner.

Example 5.15: Permit presentation of cached copies but prohibit their presentation in a frame

In this example cached copies of any web pages may be presented, but must not be presented in a frame added by the aggregator.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /

# ACAP policies here...
ACAP-crawler: *
# permit to crawl content generally
ACAP-allow-crawl: /
# ACAP-allow-index: /   # implied, but included for clarity
# ACAP-allow-preserve: /   # implied, but included for clarity
# ACAP-allow-present: /   # implied, but included for clarity
ACAP-allow-present-currentcopy: / prohibited-context=within-user-frame
```

Example 5.16: Permit presentation of the original content but require presentation in the original frame

In this example the original web pages may be presented, but must be presented in a frame in which they would be presented by the publisher.

```
##ACAP version=1.1

# Conventional policies here...
User-agent: *
Disallow: /
```

```
# ACAP policies here...
ACAP-crawler: *
# permit to crawl content generally
ACAP-allow-crawl: /
# ACAP-allow-index: /   # implied, but included for clarity
# ACAP-allow-preserve: /   # implied, but included for clarity
# ACAP-allow-present: /   # implied, but included for clarity
ACAP-allow-present-original: / required-context=within-original-frame
```

3 Glossary

The following glossary is based upon the ACAP Dictionary.

Allow	An indicator in ACAP expressions that a particular usage is permitted. The conventional Robots Exclusion Protocol does not include this term, but it is a widely-recognised term in established extensions to REP and its use is extended further by ACAP.
Content item	A piece of content on a publisher's website, at any level of granularity which it is possible to reference in an ACAP policy statement. In robots.txt only whole resources may be associated with a policy, but in META tags fragments of resources may be associated with a policy.
Content item index entries	The index entries (see below) required to make a content item searchable in accordance with a service's indexing practices.
Content item link	A link to a content item on the publisher's website.
Content item retrieved copy	A copy of a content item that has been retrieved by a crawler for processing. NB a retrieved copy may or may not be stored after it has been processed.
Content item snippet	A short indication of the content of a content item and, where applicable, its relevance to an end user search, generated by a service from content item index entries or from a content item preserved copy using proprietary algorithms.
Content item extract	An extract taken *verbatim* from the preserved copy of a content item. The order of content in an extract may not be modified.
Content item thumbnail	A thumbnail image of a content item webpage, either generated by a service from content item preserved copy or supplied by the publisher.
Content property	A property of a content item, encoded as part of the item or in associated metadata.
Crawl (usage)	The action of a crawler, to retrieve content items in a systematic way. A basic ACAP usage verb.

Crawler	Software controlled by a service operator that systematically retrieves content items from the web in order to take copies of some or all the resources on that server for indexing and other purposes.
Disallow	An indicator in ACAP expressions that a particular usage is not permitted. The conventional Robots Exclusion Protocol uses this term, but its use is extended by ACAP.
Follow (usage)	The action of a crawler, which uses links contained within a content item that it has crawled to request further resources from the same server or from other servers. A basic ACAP usage verb.
Index (usage)	The action of a crawler, or software that uses content retrieved by a crawler, to create index entries for a content item, for use in a search service or for other purposes. A basic ACAP usage verb.
Index entry	An entry in an index that comprises (a) a look-up string derived from a content item retrieved copy, (b) a link to the content item, and (c) optionally, positional information to be used only for the purpose of increasing the effectiveness of searching and relevance assessment, eg by enabling the proximity of words in text to be taken into account.
Other (usage)	The action of software controlled by a search engine, or other aggregator, to use a content item in ways other than those covered by explicitly permitted usages.
Present (usage)	The action of software controlled by a search engine, or other aggregator, that uses content retrieved by a crawler to deliver a content item, however it might be represented (e.g. by snippet, thumbnail, link, cached copy), for presentation by a web browser. A basic ACAP usage verb.
Preserve (usage)	The action of a crawler, or software that uses content retrieved by a crawler, to store a persistent copy of a content item, preserving as far as possible to arrangement and style of presentation of the original. A persistent copy is a copy that is stored for purposes other than those (essentially transient purposes) that are associated with crawling and indexing the content item, and is referred to as a 'preserved copy'. A 'cached' copy is a preserved copy, even if it is overwritten or removed when the same content item is crawled on a subsequent occasion. A basic ACAP usage verb.
Process	Any process that is operated by or for the operator of a crawler.
Publisher	A person or organisation that places content on the web.

Publisher content item indexing resource	A resource from which content item index entries must be generated, either based upon a content item extract or supplied by its publisher.
Publisher content item summary	A descriptive summary of a content item supplied by its publisher.
Publisher snippet	A resource specified by the publisher that must be used for presentation instead of a snippet that would be generated by a process controlled by the crawler operator.
Publisher thumbnail	A resource specified by the publisher that must be used for presentation instead of a thumbnail that would be generated by a process controlled by the crawler operator.
Service	A service, such as a search service, that provides organised access to content that has been harvested from the web. A service is normally delivered from one or more specific web addresses, or portals.

4 References

4. 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.

5. ACAP Technical Framework – Communicating access and usage policies to crawlers using extensions to the Robots Exclusion Protocol – Part 2: 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-Part2-V1.1.pdf.

6. ACAP Dictionary of Access and Usage Terminology. Current version available at http://www.the-acap.org/download.php?ACAP-TF-Dictionary-V1.1.pdf.

7. 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 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.

8. HTML 4.01 Specification – W3C Recommendation 24 December 1999. Current version available at http://www.w3.org/TR/html401/.

Annex – Sample content from robots.txt and HTML files used in testing

A.1 Introduction

The following are examples of robots.txt file content and HTML META tags created by one of the publishers that participated in test implementations of ACAP Version 1.0 during 2007. They are shown here to provide a more realistic impression of the way that ACAP might be used in practice.

A.2 De Persgroep samples

De Persgroep (www.persgroep.be) are a major Flemish news group. They tested ACAP for communicating access and usage policies to crawlers visiting their public news website.

A.2.1 Content of the robots.txt file on test website acap.persgroep.be

```
##ACAP version=0.2  #pilot version
# for http://acap.persgroep.be
# author: gert francois - de persgroep
# with minor modifications for use in the ACAP Implementation Guide

# current REP instructions
User-agent: *
Disallow: /

ACAP-ignore-conventional-records

# ACAP instructions for all crawlers
```

```
ACAP-crawler: *
ACAP-allow-crawl: /
ACAP-allow-index: /
ACAP-disallow-crawl: /site/        # not allowed to crawl the non-content directories
ACAP-allow preserve: /  time-limit=100-days  # not allowed to store/preserve content longer than 100 days
ACAP-disallow-present-currentcopy: /   # not allowed to display preserved (cached) copies

# ACAP instructions for ExaBotAcap12 - the test crawler implemented by Exalead.com
ACAP-crawler: ExaBotAcap12
ACAP-usage-purpose: http://acap2.exalead.com  # only allowed to use content in service http://acap2.exalead.com
ACAP-disallow-index: /nieuws/         # not allowed to index the news section
ACAP-allow-index: /sport/             # allowed to index the sports section
ACAP-allow present-snippet: /sport/ max-length=150-chars  # limit the snippet of sport to 150 chars
```

A.2.2 META tags used in HTML test content on website acap.persgroep.be

The following sample shows only those META tags that contain ACAP policy information. They are extracted from the header of an HTML page on De Persgroep's test website, and have been modified slightly for inclusion in this Implementation Guide.

```
<meta name="snippet-exalead-acap" content="Op de vandaag gepubliceerde nieuwe ATP-rankings vallen er slechts hier en daar
veranderingen te bespeuren" />

<meta name="ExaBotAcap12" content="ACAP allow-present-snippet must-use-resource=the-acap:extract:meta:snippet-exalead-
acap"/>
```