ICRA labeling

System Specification

Target readership
This document is intended to be the definitive source of advice on labeling content using the ICRA system. Whilst every attempt has been made to present the information clearly, a large amount of detail is included, along with references to a variety of web technologies.

ICRA publishes shorter and simplified explanations for users with little or no experience of website creation, or those who don't need the level of detail provided here, on its support pages[1].

Version 1.0.3
January 2006

Contents

1 Introduction

ICRA exists to help users to find what they want, to trust what they find and to avoid content that they regard as inappropriate for themselves or their children. A vocabulary is provided that can be used to describe any and all digital content in a manner that reflects a broad range of parental concerns around the world[2]. The underlying system can, however, carry any kind of metadata for any purpose.

The descriptions are machine-understandable and may be used by a variety of agents such as filters, search engines and helper applications that display extra information for users.

ICRA labels are encoded in RDF[3], one of the key technologies behind the Semantic Web[4]. This document does not set out the many advantages to content providers afforded by the semantic web except to note that features such as RSS, shared bookmarks, blogs and wikis are among its contributory elements.

Note: ICRA also offers a simplified PICS version of the label along with the Link tag, in order to support legacy systems, notably Internet Explorer's Content Advisor. Information on PICS labelling can be found at www.icra.org/pics/faq/professional/.

1.1 Namespaces and documentation

The namespace of the RDF schema that provides the framework for ICRA labels is `http://www.w3.org/2004/12/q/contentlabel#` and the recommended QName is `label`. The relevant documentation is at `http://www.w3.org/2004/12/q/doc/content-labels-schema.htm`.

The namespace for the ICRA vocabulary is `http://www.icra.org/rdfs/vocabularyv03#` and the recommended QName is `icra`. The plain text version of the ICRA vocabulary and its supplementary definitions is at `http://www.icra.org/vocabulary/`.

2 The key concepts

A Content Label is a description, i.e. a set of metadata, that can be applied to multiple resources. One or more labels are placed in a file and resources link to it either using an (X)HTML Link tag or an HTTP Response Header.

The file containing the labels is an RDF instance and is usually called labels.rdf. This is the name of the file created by the ICRA label generator (see section 2.3), although it is not significant and can be changed to anything.

Resources may link to a specific label or may link to a data set that allows clients to match the resource's URI against a series of rules that resolve to give the correct label.

Content providers can thus choose whether the association of a resource with its label is undertaken client side or server side.

2.1 Linking to specific labels (server-side processing)

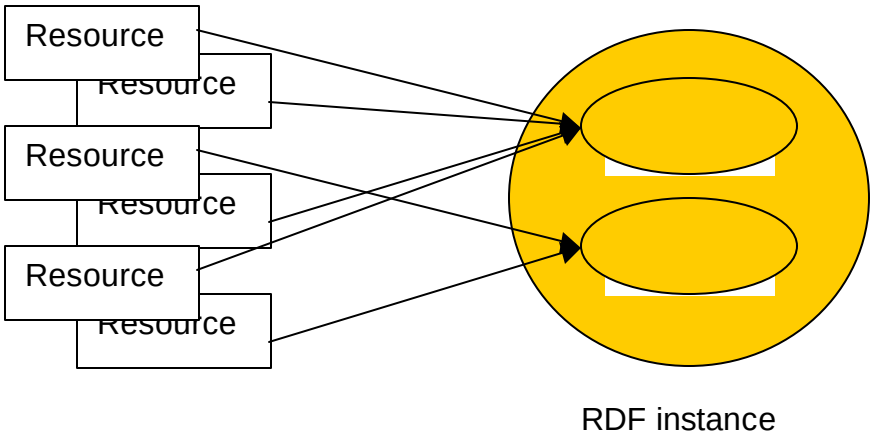

Figure 1 Server-side association of content with labels

Figure 1 shows each resource linked to a specific label. If the RDF instance is called labels.rdf and is located in the root of the website, and if a resource should be linked to a label with the ID "label_1", then the Link tag is as shown in Example 1:

```
<link rel=”meta” href=”/labels.rdf#label_1”
type=”application/rdf+xml” title=”ICRA labels” />
```

Example 1 A typical link tag that associates a specific label with a resource

The equivalent HTTP Response Header is:

```
Link: </labels.rdf#label_1>; /="/"; rel="meta"
type="application/rdf+xml"; title=”ICRA labels”;
```

Example 2 The HTTP Response Header equivalent of Example 1

Note that the specific label within the RDF instance is identified by the URL fragment given in the `href` attribute. The `title` attribute is optional but is recommended for clarity. The location of the labels.rdf file is unimportant. It can be at any location on any server, but, of course, its location must be given in the `href` attribute.

2.2 Linking to a set of rules (client-side processing)

Figure 2 shows the alternative approach. All resources are linked to the RDF instance, but the link does not identify the label. Instead, the RDF instance defines a default label and may then also define a sequence of rules, based on Perl 5 regular expressions that can override that default. The first rule in the sequence to be satisfied identifies the correct label.

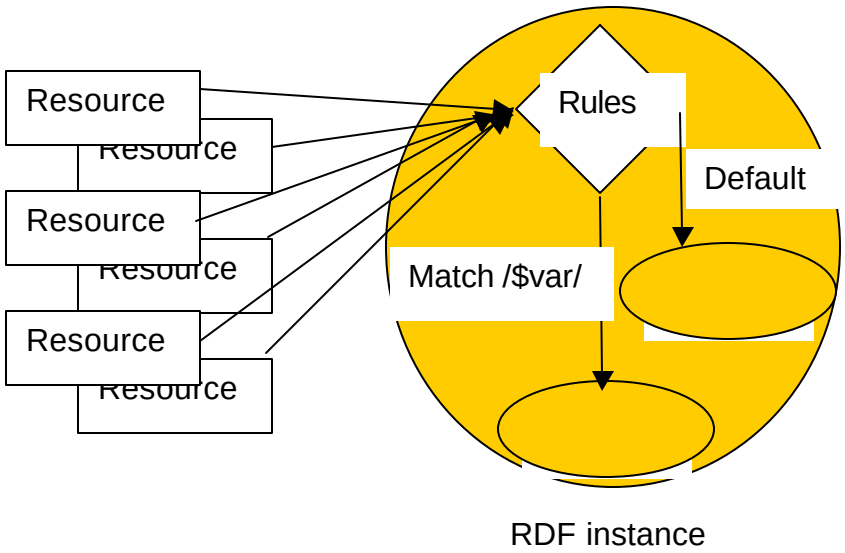

Figure 2 Client-side association of resources and labels

If the RDF instance is called labels.rdf and is located in the root of the website then the Link tag is shown in Example 3:

```
<link rel=”meta” href=”/labels.rdf”
type=”application/rdf+xml” title=”ICRA labels” />
```

Example 3 Typical tag linking a resource with an RDF instance that contains rules that identify the correct label.

The equivalent HTTP Response Header is:

```
Link: </labels.rdf>; /="/"; rel="meta"
type="application/rdf+xml"; title=”ICRA labels”;
```

Example 4 The HTTP Response Header equivalent of Example 3.

As with Example 1, the location and name of the RDF instance are not significant.

2.3 Creating the RDF instance

ICRA provides a tool on its website for creating the RDF instance and the necessary tags, known as the label generator[5]. It is designed to be used by those with little or no knowledge of web authoring techniques as well as more advanced users. The label generator builds the RDF instance based on the client-side processing model described above (section 2.2), although it is equally valid for the server-side model.

3 The content of the RDF instance

The RDF instance must define 1 or more labels. More specifically, it must define at least one instance of the RDF class Content Label as defined by `http://www.w3.org/2004/12/q/contentlabel#ContentLabel`.

N.B. RDF Content Labels may contain statements from any RDF schema; however this document is concerned solely with ICRA's implementation.

The RDF instance can further define zero or more of the following:

1. The host(s) for which the label(s) are applicable. Sub-domains are in scope.
2. An additional string that must match the resource's URI for any labels in the RDF instance to be applicable.
3. The default label.
4. An ordered sequence of rules that should be matched against a resource's URI. If a rule is satisfied it must provide a label that overrides any default.
5. A description of the RDF instance itself that identifies where additional information about the label can be found, including how its veracity can be assessed.

These elements are explained in detail with reference to Example 5. Like all examples in this document and others produced by ICRA, the RDF is serialized in XML. However, this is not a requirement; other serializations, such as N3[6], are equally valid.

1	`<?xml version="1.0"?>`
	`<rdf:RDF`
	`  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
	`  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"`
	`  xmlns:label="http://www.w3.org/2004/12/q/contentlabel#"`
	`  xmlns:dc="http://purl.org/dc/elements/1.1/"`
	`  xmlns:icra="http://www.icra.org/rdfs/vocabularyv03#">`

2	`  <rdf:Description rdf:about="">`
	`  <dc:creator rdf:resource="http://www.icra.org" />`
	`  <label:authorityFor>http://www.icra.org/rdfs/vocabularyv03#`
	`  </label:authorityFor>`
	`  </rdf:Description>`

3	`  <label:Ruleset>`
	`  <label:hasHostRestrictions>`
	`  <label:Hosts>`
	`  <label:hostRestriction>example.org</label:hostRestriction>`
	`  <label:hostRestriction>example.com</label:hostRestriction>`
	`  </label:Hosts>`
	`  </label:hasHostRestrictions>`
	`  <label:hasDefaultLabel rdf:resource="#label_1" />`

```
4     <label:rules rdf:parseType="Collection">
        <rdf:Description>
          <label:hasURI>photography</label:hasURI>
          <label:hasLabel rdf:resource="#label_2"/>
        </rdf:Description>
          <label:UnionOf>
         <label:hasURI>guestbook</label:hasURI>
          <label:hasURI>messages</label:hasURI>
          <label:hasLabel rdf:resource="#label_3" />
        </label:UnionOf>
      </label:rules>
    </label:Ruleset>

5   <label:ContentLabel rdf:ID="label_1">
      <rdfs:comment>Label for all/most of website</rdfs:comment>
      <rdfs:label>No nudity, no sexual content, no violence, no
       potentially offensive language, no potentially harmful
       activities, no user-generated content</rdfs:label>
      <icra:nz>1</icra:nz>
      <icra:sz>1</icra:sz>
      <icra:vz>1</icra:vz>
      <icra:lz>1</icra:lz>
      <icra:oz>1</icra:oz>
      <icra:cz>1</icra:cz>
    </label:ContentLabel>

    <label:ContentLabel rdf:ID="label_2">
      <rdfs:comment>Label for photography section</rdfs:comment>
      <rdfs:label>Exposed breasts, Bare buttocks, No sexual
      content, no violence, no potentially offensive language,
      no potentially harmful activities, no user-generated content
      This material appears in an artistic context</rdfs:label>
      <icra:na>1</icra:na>
      <icra:nb>1</icra:nb>
      <icra:sz>1</icra:sz>
      <icra:vz>1</icra:vz>
      <icra:lz>1</icra:lz>
      <icra:oz>1</icra:oz>
      <icra:cz>1</icra:cz>
      <label:hasModifier><icra:xa /></label:hasModifier>
    </label:ContentLabel>

    <label:ContentLabel rdf:ID="label_3">
      <rdfs:comment>Label for guestbook/message board</rdfs:comment>
      <rdfs:label>No nudity, no sexual content, no violence, no
      potentially offensive language, no potentially harmful
      activities, user-generated content (moderated)</rdfs:label>
      <icra:nz>1</icra:nz>
      <icra:sz>1</icra:sz>
      <icra:vz>1</icra:vz>
      <icra:lz>1</icra:lz>
      <icra:oz>1</icra:oz>
      <icra:ca>1</icra:ca>
    </label:ContentLabel>
  </rdf:RDF>
```

Example 5 An example RDF instance containing ICRA labels

3.1.1 Section 1

The namespaces are declared. The QNames `label` and `icra` are recommended for their respective namespaces.

3.1.2 Section 2

This short section declares that the labels were created by ICRA and that further information is available at www.icra.org. Since it is possible to include descriptions based on other schemas, this section specifies that www.icra.org only has information about the ICRA namespace.

3.1.3 Section 3

This section declares the hosts for which the data is valid. In this instance, we have declared that the labels can be applied to both example.org and example.com. Subdomains are covered, for example, www.example.org, sub.example.com etc.

This section also declares that the default Content Label for material on those hosts is "label_1" (see 3.1.5).

If labels are to be restricted to a particular area of the example.org and example.com hosts, this would be included thus:

```
<label:hasURI>foo</label:hasURI>
```

Labels in this RDF instance would then only be in scope for resources with URIs on the example.org or example.com hosts that *also* contain ‘foo.’ This feature is included primarily for ISPs who offer personal web space with URLs like www.example.org/username. If more than one `hasURI` property is included, a URI is in scope if any one of them matches.

3.1.4 Section 4

The rules that determine where the default label should be overridden by another label are declared next. In this example, everything in the photography section of both example.com and example.org will be associated with "label_2," everything with *either* the word guestbook *or* messages in the URL will be associated with “label_3”. Otherwise, the default applies.

Matching is done using Perl 5 regular expressions[7] so that if a rule should apply to "all URLs ending in .jpg" then this would appear as `\.jpg$`.

The use of `rdf:parseType="Collection"` ensures that rules are processed in order. The first rule to be satisfied is the one that is used, and processing stops at that point.

3.1.5 Section 5

Finally the labels themselves are defined. In the example, “label_2” declares that there are exposed breasts, bare buttocks, and that the material appears in an artistic context. “Label_3” declares that there is moderated user-generated content and “label_1” states "none of the above" in all categories of the ICRA vocabulary.

4 Setting the MIME type

The correct MIME type for RDF instances is application/rdf+xml[8]. Your server may not support this by default[9]. If this is the case you’ll need to do one of two things:

1. Ideally, add the MIME type application/rdf+xml, usually associated with file extension .rdf

2. If you are unable to do this, try changing the name of the RDF instance to labels.xml. The XML MIME type (application/xml) is an acceptable alternative and is more widely included in default server configuration.

3. Some servers may offer text/xml as a MIME type for files with the .xml extension. This is unlikely to cause problems for clients looking for ICRA labels but should not be used if you’re including ICRA labels in a more sophisticated data set such as a database, or if the character set is not iso-8859-1 (Latin-1).

If none of these options is followed, your server may use a default MIME type such as text/plain. In this situation a client may or may not recognize the data as RDF and therefore may or may not process it correctly.

If you run IIS servers and are unsure how to add new MIME types, please see Section 5.3 below.

If your server is protected by a firewall, you may to need to configure this accordingly too.

5 Methods for inserting the tags

Having created the RDF instance, the next step is to insert the links to it. For a website to be considered fully labeled, links must be included on every (X)HTML page and ideally should be included in all resources.

The ability to shift label processing to the client rather than the server offers one crucial advantage: an identical link can be inserted on all resources. This is true whether the labels cover one small website or a global network of internet properties.

The most efficient way to do it is to configure the server(s) to include the link in the HTTP Response Headers. This also avoids accidentally deleting the tag (or omitting it) when pages are redesigned. Control of the labels is then firmly in the hands of the person (or department) responsible for managing the RDF instance. This may or may not be the same as those responsible for content creation. Alternatively, an (X)HTML Link tag (similar to Example 1 or Example 3 as appropriate) can simply be included in a document template or any other method you may use to include the same data in every page's <head> section.

5.1 Is it important to include the tag on every page?

Yes. When a user visits your site for the first time, their client will only detect the labels if a link is in place. If the link is only included in, say, the homepage, then users who enter the site via other routes will not benefit.

5.2 Apache server configuration

There is more than one way to control Apache's HTTP Response Headers. If you already set headers for other reasons, continue to use the same method. If not, the method given below is robust and will work.

5.2.1 Install Mod_Headers

Mod_Headers is not generally included in the default configuration but will almost certainly be included in your Apache installation and just needs to be "switched on" by removing the comment symbol before two lines in the httpd.conf file.

There are many different "flavors" of Apache, but what follows is likely to be at least close to what is required.

In the DSO section of the httpd.conf file look for

```
LoadModule headers_module      modules/mod_headers.so
```

In some builds, that's enough; others will also require the command below:

```
AddModule mod_headers.c
```

The comments in your config file and the presence (or absence) of similar commands for other modules will give you a good clue as to what to do.

5.2.2 Setting the same Response Header for all resources

Assuming that the RDF instance is called labels.rdf and is in the web server's document root, the following command, inserted in the httpd.conf file, will achieve the desired result.

```
Header set Link '</labels.rdf>; /="/"; rel="meta"
type="application/rdf+xml"; title=”ICRA labels”;'
```

NB. This command should appear all on one line

5.2.3 Linking to specific labels with HTTP Response Headers

Like other Apache configuration options, HTTP Response Headers can be set within block directives. Example 6 sets the link to “label_2” for all resources in /var/www/images/.

```
<Directory /var/www/images/>
  Header add Link '</labels.rdf#label_2>; /="/";
  rel="meta" type="application/rdf+xml";
  title=”ICRA labels”;'
</Directory>
```

Example 6 A simple block directive setting a header for all resources in the images directory

As above, the `Header add Link` command should appear on a single line.

Block directives also offer very fine control over the HTTP Response Headers where required*. Example 7 sets a header pointing to “label_1” for all resources in the /var/www/ directory (and its subdirectories), but where the filename ends with .gif, .jpg, .jpeg or .png, the header linking to “label_2” is invoked.

```
<Directory /var/www/>
  Header add Link '</labels.rdf#label_1>; /="/";
  rel="meta" type="application/rdf+xml";'
  <Files ~ "\.(gif|jpe?g|png)$">
    Header unset Link
    Header add Link '</labels.rdf#label_2>; /="/";
    rel="meta" type="application/rdf+xml";'
  </Files>
</Directory>
```

Example 7 A nested block directive setting a different header for image files than for other files in the same block.

Notice in Example 7 that the link is `unset` within the file block directive. This is because where a resource is linked to a specific label, that label is given the highest priority and can't be overridden (see section 7). It is therefore an error to include more than one link to specific labels, and the expected behavior of clients is not defined in these circumstances[10].

* Some versions of Apache may not allow headers to be set in a Virtual Host block directive.

5.3 Microsoft IIS server configuration

Microsoft has made configuring its servers to include Link tags very easy. The header information is set in the Website properties dialogue using the Custom HTTP Headers function. IIS uses a hierarchical architecture, with the HTTP Headers property page being configurable at the following levels:

- Web server
- Home directory / Web site (IIS 4 and later support multiple websites)
- Virtual directory
- Folder
- Page

To set the HTTP Header properties, select the required level from the IIS Control Panel, right click and select properties, then select the HTTP Headers property page. The screen shot below shows the HTTP Headers property page for the default website.

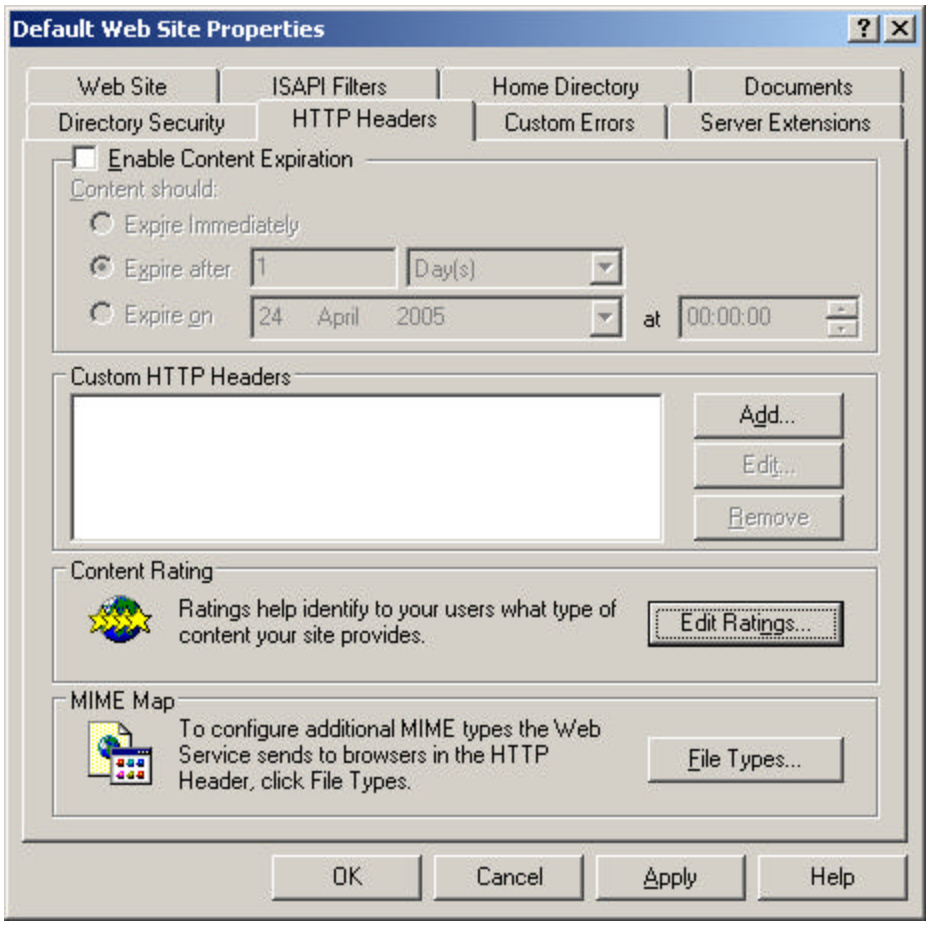

Figure 3 The properties dialogue box in IIS

Click the Add button.

As shown in Figure 4, enter `Link` In the Custom Header Name field and in the Custom Header Value field enter the following:

```
</labels.rdf>; /="/"; rel="meta" type="application/rdf+xml";
title=”ICRA labels”;
```

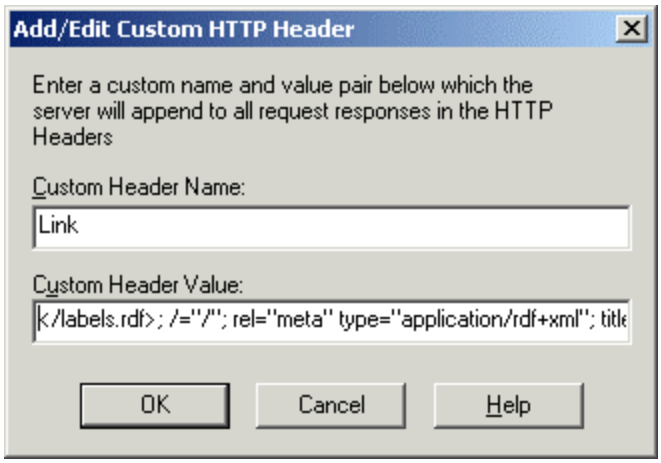

Figure 4 The custom header name and value fields in IIS

Click OK to return to the web properties dialogue box. If you haven't done so already, add the RDF MIME type now! In the MIME Map section (see Figure 3), click File Types and enter the information as shown in Figure 5.

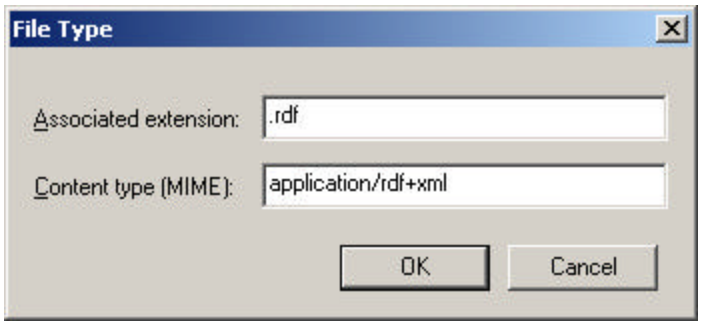

Figure 5 Adding the RDF MIME type in IIS

NB. Please ignore the Content Rating options (this uses an obsolete system).

6 The Ruleset in Detail

Many content providers will need only a single label, or at most, a handful of labels for their site. The Ruleset, however, offers a great deal of flexibility and fine control over which label is associated with which resources. Three basic types of rule are available:

`<rdf:Description>`
A simple rule that declares a single regular expression in a `hasURI` element that, if matched, identifies the correct label.

`<label:UnionOf>`
A rule that includes two or more regular expressions in `hasURI` elements that, if *any* of them match, identifies a correct label

`<label:IntersectionOf>`
A rule that includes two or more regular expressions in `hasURI` elements that, if *all* of them match, identifies a correct label.

In Example 5, two rules are declared:

```
<rdf:Description>
  <label:hasURI>photography</label:hasURI>
  <label:hasLabel rdf:resource=”#label_2” />
</rdf:Description>
```

Any resource whose URI includes the string “photography” (and is on one of the declared hosts) will be described by “label_2”

```
<label:UnionOf>
  <label:hasURI>guestbook</label:hasURI>
  <label:lasURI>messages</label:hasURI>
  <label:hasLabel rdf:resource="#label_3" />
</label:UnionOf>
```

If a URI does not match the first rule, a client will attempt to match it against “guestbook” and “messages.” If a match is found (for either), then “label_3” applies.

It is possible to nest rules as shown in Example 8. “Label_2” would be applied if the URL contained both “color” and “image” or both “monochrome” and “image.” Note that `hasLabel` is a property of the “outer” rule.

```
<label:UnionOf>
  <label:rules>
    <label:IntersectionOf>
      <label:hasURI>color</label:hasURI>
      <label:hasURI>image</label:hasURI>
    </label:IntersectionOf>
    <label:IntersectionOf>
      <label:hasURI>monochrome</label:hasURI>
      <label:hasURI>image</label:hasURI>
    </label:IntersectionOf>
  </label:rules>
  <label:hasLabel rdf:resource=”#label_2”” />
</label:UnionOf>
```

Example 8 A nested rule

7 Global defaults, local overrides

It is possible to set a global default label using a Ruleset that is then overridden by a label linked directly from the resource. This is because if a resource links directly to a specific label, as described in section 2.1, this is treated as more authoritative than the output of the Ruleset.

In a typical example, a content provider might configure servers to include an HTTP Response header that linked to a file similar to Example 5. A resource at www.example.org/page.html will be described by the default label ("label_1"), as none of the rules match.

However, if page.html were to include a link to "label_2", using a tag like this:

```
<link rel="meta" href="/labels.rdf#label_2"
type="application/rdf+xml" title="ICRA label" />
```

this would override the default and associate it with "label_2".

An important point
You cannot override a link to a specific label. Therefore, if you configure your system to include a link to labels.rdf#label_1 (whether through a Link tag or an HTTP Response Header) you **cannot** then override it with a link to a different label. You must use a label:`Ruleset` to identify the default label and then override it with a specific link.

8 Processing ICRA labels

For a given resource (a page, an image, etc.) there are three possible sources of a label:

- It may be possible to deduce a label by processing data already held in the filter's cache (memory). Such labels are referred to below as Type 1.
- A resource may include a link to data that contains rules that can be followed to identify a label. Labels identified in data sources linked from the resource itself are referred to below as labels of Type 2.
- A resource may include a direct link to a label. Labels identified by a direct link from a resource are referred to below as labels of Type 3.

As detailed below, filters SHOULD assign increasing priority to each of these sources.

8.1 Before retrieving the resource

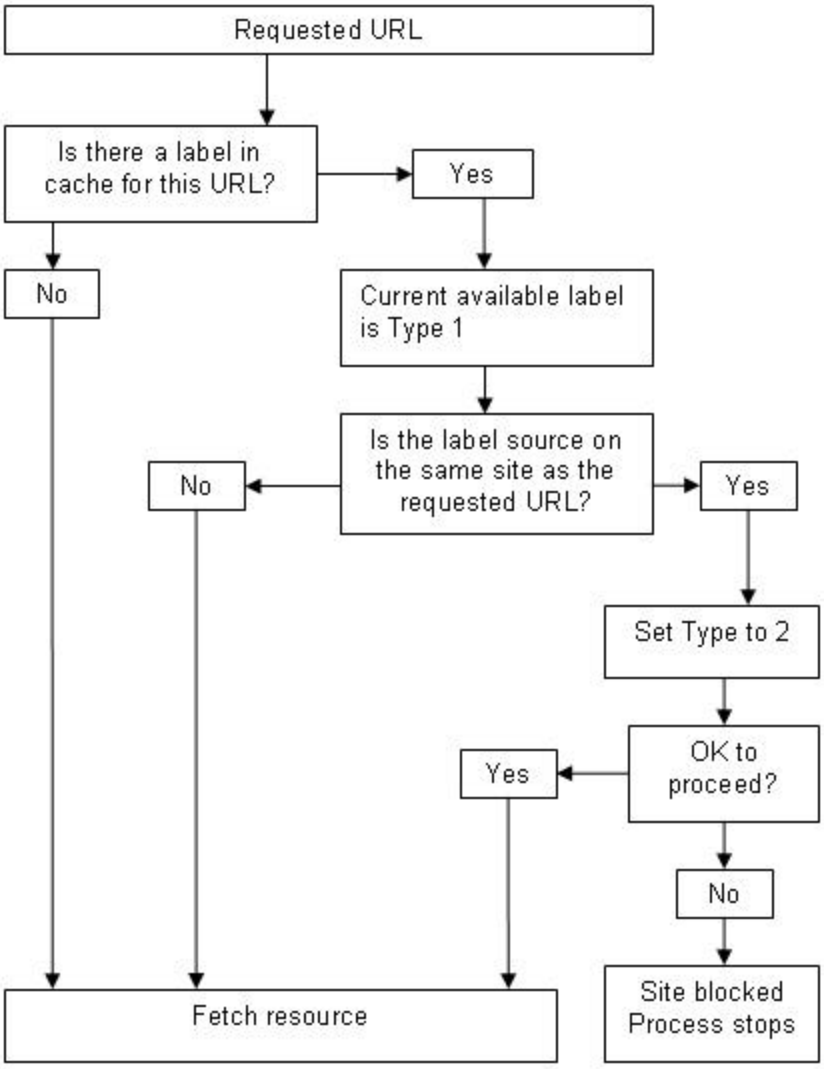

Figure 6 The processing to be carried out prior to any request to the internet.

If the filter already has a label for the requested URL in its cache then, immediately, a Type 1 label is available. If that label was initially retrieved from the same site as the URL we're interested in now, the label SHOULD be considered as a Type 2.

If the label data in cache was retrieved from a different website, it remains a Type 1 and the resource should be fetched and checked for further data.

For clarity:

- A Type 1 label MUST NOT be used to block access to a URL before it has been fetched.

- A Type 2 label MAY be used to block access to a URL before it has been fetched.

There are several reasons for this, but they boil down to the idea that labels linked from a resource are "closer" to the content provider than labels that may have been published by someone with little or no connection with the described content. This is extended in the next section, when further priority is given to labels that are linked from the resource itself.

Type 1 labels should not be confused with third-party labeling. If a filter is configured to request labels from a third-party source, such as an online database or a content analyzer, the filter will handle that data separately. Type 2 labels take precedence over Type 1 labels purely in the context of self-labeling.

If there is no data in the filter's cache, the resource at the URL MUST be fetched.

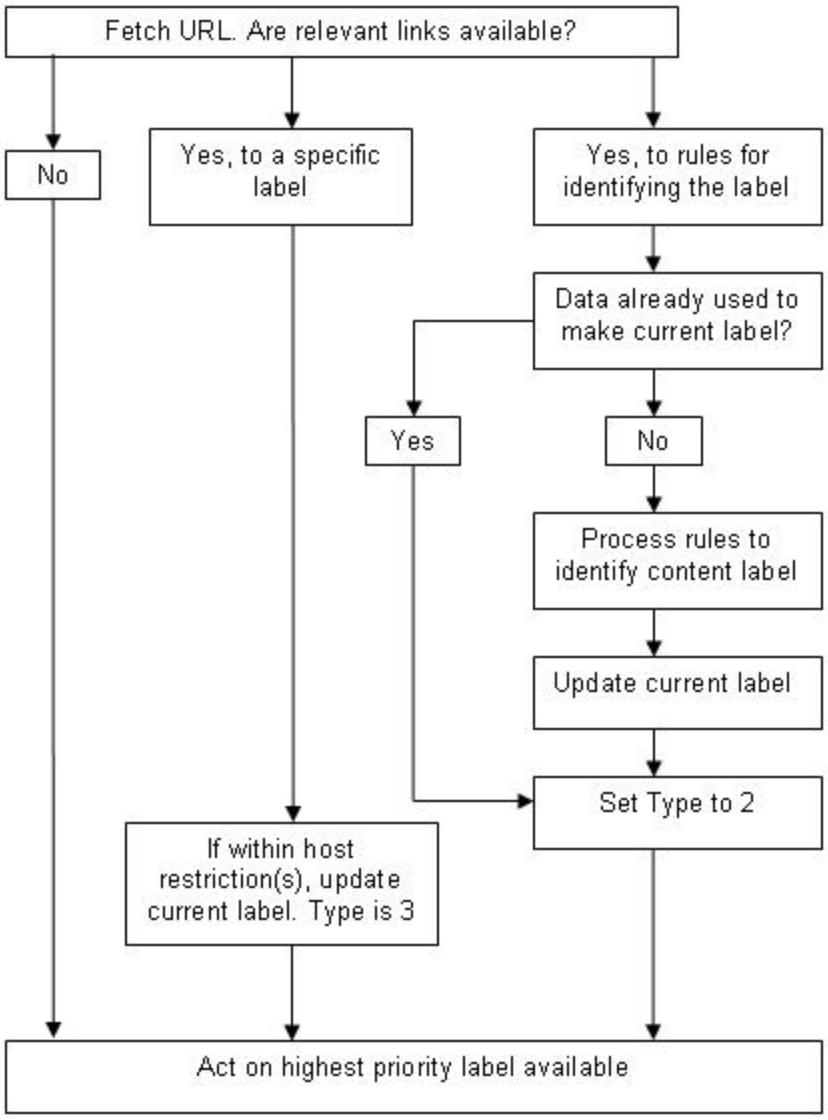

Figure 7 The processing rules after the resource has been fetched.

8.2 Identifying the correct label

If the resource includes links to label data, it may be necessary to fetch and process it. (Remember that labels are always held separately, never actually within the resource itself.)

If the resource includes a link to a specific label, this is classed as a Type 3. Since this is the highest priority in the hierarchy, once a Type 3 label is available, no further processing is necessary to identify the correct label to use for this resource.

However, clients SHOULD check any host restrictions. Clearly a label should only be recognized as valid if the resource pointing to it is from the declared host(s). If no host restrictions are declared, a client MAY accept the label.

The priority given to Type 3 labels is the crucial step that allows a content provider to work with the notion of a default label with local overrides.

If the resource carries a link to the same resource that had already been processed to identify a Type 2 label, clearly no further processing is necessary; the correct label has already been identified.

However, if a link points to a different data source than had already been used to derive a Type 2 label, the new data SHOULD be processed. This is because it is possible to include any number of data files on a site and it can be assumed that the one linked to from a given resource is the one the content provider intended to be used.

If no links are present in the resource then clearly the only information available is that which was available before the resource was fetched.

If multiple labels of the same Type are available, this is an error on the part of the content provider. The filter MAY use any of them but, purely for reasons of efficiency, will normally simply use the first one found of a given type.

8.3 Unlabelled resources

ICRA recommends that filtering clients offer users two options: to block/allow access to unlabelled web pages i.e. (X)HTML, and to block/allow access to all unlabelled resources. The latter option is designed to help to avoid websites being blocked because a script file or stylesheet is fetched from the network before a suitable label is retrieved.

9 Showing users that a site is ICRA labeled

Labeled sites may carry either a graphic or a text linked to http://www.icra.org/sitelabel. Users following that link will be able to see a dynamically generated interpretation of your label and find out more about ICRA.

Text links should say “Labeled with ICRA”. Phrases such as “Approved by ICRA” and “ICRA certified” are strictly forbidden. (ICRA does not approve or certify sites – content providers label their content using the ICRA system.)

Many sites have some sort of "small print" links at the bottom of one or more pages, perhaps to a privacy statement or contact point. You could add a "Labeled with ICRA" text link in similar fashion.

Alternatively, you can add any of the graphics available from icra.org[11]. These are available in a number of languages and in both American and British English (i.e. labeled and labeled respectively).

10 Working with other RDF schemas

An ICRA label and the Ruleset are simply RDF Classes whose types are defined by the relevant schema. These elements can all be included in any RDF instance and, conversely, any other RDF metadata can be included in “an ICRA labels file.”

Indeed, content providers are encouraged to make use of other schemas.

10.1 Management information

Perhaps the best-known metadata schema that is regularly encoded in RDF is Dublin Core[12]. Along with Creative Commons[13] licenses and similar schemas, this can be included directly in content labels but can also be put in a special class for management information.

In exactly the same way as Content Labels, the label schema defines the properties `hasDefaultManagementInfo` and `hasManagementInfo`. These link to RDF Content Labels that can include data such as title, author and publication date. Management Info labels exist independently of Content Labels linked by `hasDefaultLabel` and `hasLabel` properties labels, so it’s possible to have a rule that overrides the default management information label without overriding the ICRA label.

10.2 Classification

Content Labels, whether they carry ICRA descriptors or any other metadata, are designed to include detailed descriptors. More formally, they are classes with properties that describe the resource. A third type of description is also defined in the label schema – a classification. Again there are analogous properties: `hasDefaultClassification` and `hasClassification`, but unlike a Content Label, a classification is designed to be a description in itself.

For example, the classification may be an age-based film rating or identify an article as being about “politics” as opposed to “fashion.” Whatever the classification is, clients are not obliged to process any properties of a class linked by a `hasClassification` or `hasDefaultClassification` property.

Again, classifications exist independently, so that overriding a default classification has no effect on either the management information or label.

11 Frequency modifiers

Movies, TV programs, games and other content that "has a duration" may need more than one label. It may be appropriate to provide a label that describes a particular scene in a movie or a content type that occurs occasionally throughout a game. To support this, the Content Label schema supports a set of frequency modifiers:

- has frequent scenes
- has several scenes
- has occasional scenes
- has a single scene

A standard RDF description might appear as shown in Example 9. This can stand alone in the same way as any other RDF description, or form part of a sequence of rules in a `label:Ruleset`.

```
<rdf:Description rdf:about=”http://example.org/movie.mov>
  <label:hasLabel rdf:resource= “#label_1” />
  <label:hasOccasionalScenes rdf:resource=”#label_2” />
  <label:hasSingleScene rdf:resource=”#label_3” />
</rdf:Description>
```

Example 9 Description of a movie using frequency modifiers

Frequency modifiers have a range of `label:ContentLabel`. That is, they MUST link to a class of that type.

12 Some tips

12.1 It's just RDF

Content Labels, host restrictions, rules – these are all just RDF fragments. They do not need to all be in a single file called labels.rdf. If you're familiar with RDF, think of ICRA labels simply as part of your metadata.

12.2 Managing labels for multiple websites

If you create lots of websites that should have the same ICRA label, create a file with the label in it and make the Link tag that points to it part of your regular template. Remember that the labels do not have to be on the same server, they can be anywhere.

You do not need to include a host restriction at all – if a resource points to a label and there's no host restriction included in the RDF instance, the label is valid. On the downside, it means *anyone* can point to your label, which may put extra load on your server.

If you do want to include a host restriction it can be in a separate file all on its own. Example 10 shows how this can be done. The two fragments of RDF can be in the same file (as shown here) or in separate files on different servers. In this case you'd need to include a full URI (including the fragment identifier) as the `rdf:resource`.

```
<label:Ruleset>
  <label:hasHostRestrictions rdf:resource=”#hosts” />
   …
</label:Ruleset>

<label:Hosts rdf:ID=”hosts”>
  <label:hostRestriction>example.com</label:hostRestriction>
  <label:hostRestriction>example.org</label:hostRestriction>
</label:Hosts>
```

Example 10 A Ruleset that links to an "external" list of host restrictions

This allows you to set up a stable file for the labels and then generate the host restriction list dynamically if desired.

12.3 A specific label for a specific page

Labels that apply to a single resource can be put in a separate file. You might set up a default labels file (with a Ruleset) and link everything to that and then create a completely separate label file for a particular page with a specific link to the label.

In short, work out what's best for you. It'll probably work in practice.

13 Testing the labels

The ICRA website includes an online tool that will identify the correct label for a given URL[14].

14 Change log

Version 1.0.1 Added section on linking to icra.org/sitelabel (section 9). Subsequent sections renumbered.

Version 1.0.2 Amended documentation of hostRestriction to include hasHostRestrictions property and Hosts class.

Version 1.0.3 Added section linking referencing PICS labelling document.

---

[1] http://www.icra.org/support
[2] http://www.icra.org/vocabulary
[3] http://www.w3.org/RDF
[4] http://www.w3.org/2001/sw/
[5] http://www.icra.org/label
[6] http://infomesh.net/2002/notation3/
[7] See, for example, http://www.perl.com/doc/FMTEYEWTK/regexps.html
[8] http://www.faqs.org/rfcs/rfc3870.html
[9] Apache's default configuration has supported this since summer 2003. At the time of writing (May 2005) IIS does not.
[10] More information on the use of Apache's block directives to control HTTP Response Headers is available at http://www.icra.org/archive/labellingWG/mod_headers/
[11] http://www.icra.org/buttons
[12] http://www.dublincore.org/
[13] http://creativecommons.org/
[14] http://www.icra.org/label/tester/