



Professional website labelling

Choice not censorship

Version 2.2
Phil Archer
Internet Content Rating Association
5th March 2004

Contents

1	Introduction.....	5
2	Elements of a PICS label	7
3	General comments on server configuration.....	9
4	Apache configuration	10
4.1	Controlling labels using Apache's block directives	10
4.2	Using Wildcards and Regular Expressions	12
4.3	Using a .htaccess file	14
4.3.1	Just a suggestion	14
5	Configuring Microsoft servers.....	16
6	Testing/Viewing HTTP Response Headers	18
7	Labelling sites using HTML meta tags	19
7.1	Different labels for different parts of the site	21
7.2	Labels for resources pulled from other domains	22
7.3	Summary for HTML meta tags	23
8	Scripting techniques, SSIs etc.	24
8.1	The same SSI for each page.....	24
8.2	Multiple domains pointing at a single site	24
8.3	Using a script to write the label in an HTTP Response Header	24
8.4	Keeping the bandwidth down	25
9	Testing the labels	26
9.1	How the label tester works.....	26
10	HTTP and HTML together.....	27
11	Label generation.....	28
11.1	Z 0 - the don't know option	28
12	References and links	30
13	Document history	31
13.1	Changes from version 1.1	31
13.2	Changes from version 2.0	31
13.3	Changes from version 2.1	31
14	Index.....	32

1 Introduction

The key concept behind labelling websites is pretty straightforward - content is delivered to the client with a set of encoded descriptors which filtering software can block or allow, depending on parental settings. Sounds like a censor's dream? No - and here's why:

1. The ICRA descriptors are designed to be as objective as possible. A feature is either present or absent on the site. There is little room for personal judgement (although we freely admit that, despite our best efforts to be wholly neutral, there is some).
2. You - not ICRA - rate the content on your site .
3. The parent - not ICRA - decides what their children can and cannot see.

The platform currently used is the Platform for Internet Content Selection (PICS) which is a W3C Recommendation. ICRA expects to publish its vocabulary as an RDF schema for use in XML documents during 2004. There are other rating services that use the PICS system but, to a greater or lesser extent, they all carry their own cultural values. The ICRA system is the only one designed to be fully international and cross cultural, enjoying the backing of many of the largest names on the internet.

Rating labels can be applied at all levels, from every file served from a given server, irrespective of domain, down to individual files.

In order for a PICS-based filter to decide whether any particular file downloaded from a website - be it an HTML document, image or anything else - should be allowed through on the basis of its content rating label, one of two conditions must be true:

1. The file arrives with a label included in its header information
2. The filter already has a label in cache which can be applied to the incoming content.

This translates into two basic approaches to labelling:

1. Configuring the server to include PICS labels in the HTTP headers of each file served. This is the efficient "do it once and forget it" approach. This puts it under the control of the server engineers.
2. Including one or more meta tags in the HTML header section of each page. This can be achieved alongside other common elements such as links to style sheets. This approach puts labelling under the control of the webmasters.

This document gives details of the various elements in a PICS label and then discusses how they can be delivered according to the two methods outlined above.

2 Elements of a PICS label

A basic PICS label takes the following form:

```
'(pics-1.1 "RATING SERVICE URL" l r (RATING))'
```

The elements here are:

pics-1.1 Defines which version of PICS we're using

RATING SERVICE URL A quoted URL that is *always* in double quotes (which plays merry havoc with the web authoring tools but never mind). As it is a URL, it serves as a unique identifier for the rating service as well as being a location from which information about the service can be obtained. In ICRA's case, the rating service URL is <http://www.icra.org/ratingsv02.html>.

l This is a lower case "L" and is short for labels (optionally you can write the word labels in full). This declares the beginning of the label or list of labels that follow, all of which use the defined rating service

r Short for ratings (which optionally you can write in full). This is the actual rating according to the rating service.

The syntax defines that the label be enclosed within single quotes and parentheses as shown.

Which leads us to our first example complete ICRA label:

Example 1: A basic ICRA label

```
'(pics-1.1 "http://www.icra.org/ratingsv02.html" l r
(cz 1 lz 1 nz 1 oz 1 vz 1))'
```

The ratings shown in this example are ICRA-code for "none of the above" in all categories. So this label is making a positive statement that the file that carries it contains:

- No chat facilities or message boards (cz 1)
- No potentially offensive language (lz 1)
- No images, descriptions or portrayals of nudity or sexual activity (nz 1)
- None of the descriptors in the "Other" category (oz 1)
- No images, descriptions or portrayals of violence of any kind (vz 1)

The codes used in the ratings are explained on the ICRA website (see section 12 page 30).

As mentioned earlier, if labels are only to be sent with some files and then applied to content which doesn't carry a label of its own, additional information is added to control how filtering applications should cache and apply those labels. This is achieved by means of a statement like this:

```
gen true for "http://www.example.org/"
```

gen Short for generic. This flag can be set to true or false. If true, then *any* URL that begins with the string quoted in the **for** statement is covered by the label. Such gen true labels will be cached by filters for subsequent use. If the gen flag is set to false, then the label can *only* be applied to the specific URL quoted. Gen false labels therefore quote a specific page rather than a domain name, thus:

```
gen false for "http://www.example.org/page.html"
```

Example 2: A full ICRA label for a whole domain

A full ICRA label declaring "none of the above" in all categories for the ever popular example.org domain would therefore be:

```
'(pics-1.1 "http://www.icra.org/ratingsv02.html" 1  
gen true for "http://www.example.org/" r (cz 1 lz 1  
nz 1 oz 1 vz 1))'
```


3 General comments on server configuration

Here are some quick statements to get us moving along quickly here:

1. If you include a label with every file served, you don't need to include any information about what the label refers to, it refers to the file carrying the label.
2. Configuring Apache or Microsoft servers to include a label with every file served is easy.

The following two sections describe how to set up Apache and Microsoft servers to include PICS labels. In these sections, the assumption is made that you will be able to configure your server(s) to include labels with every file served, whether it be an HTML page, images, video clips or anything else. There is no need to identify to which resource these labels apply since each file arrives at the PICS aware client carrying its own label.

As noted in the previous section, it is also possible to send labels that carry extra information so that they will be cached and applied to other resources, thereby reducing the total number of labels served. This means including statements like

```
gen true for "http://www.example.org"
```

You *can* include these in labels written as HTTP response headers and there are situations where this is what you want to do. However, 'gen true' really comes into its own when delivering labels as HTML meta tags. Therefore, the full discussion of this aspect is saved for the section on HTML.

4 Apache configuration

The following explanation assumes you have at least a grounding in Apache configuration.

NB.

To include PICS labels in HTTP Response Headers you need to use the `mod_headers` module. This may not be available on your system without being compiled/loaded before proceeding.

Example 3: Setting a default label for all content served from a single machine

```
Header set pics-label '(pics-1.1
"http://www.icra.org/ratingsv02.html" l r (cz 1 lz 1
nz 1 oz 1 vz 1))'
```

Put this in your *config* file outside any block directive and the job's done. Every file served will include this label in its HTTP header.

The elements in this are as follows:

Header set pics-label Fairly self explanatory - this tells Apache to set the value of `pics-label` header to the following value. By using `set`, as I recommend in all cases rather than `append` or `add`, any previously set label is overwritten.

'(pics-1.1 "http://www ...)' The label itself, or as far as Apache is concerned, the value of the `pics-label` header. Notice that it is enclosed in single quotes. You must use single and double quotes as shown here. Unusually for coding, PICS does not permit you to swap their usage.

4.1 Controlling labels using Apache's block directives

HTTP Response Headers can be set within the following block directives:

- <none> i.e. act as a default
- <VirtualHost>
- <Directory> and <DirectoryMatch>
- <Files> and <FilesMatch>
- <Location> and <LocationMatch>

These block directives support wildcards - that is, "?" to match a single character and "*" to match any number of characters; as well as Regular Expressions for detailed pattern matching. Only <Files> and <FilesMatch> can be set within a `.htaccess` file. We'll return to these issues shortly.

NB. A earlier version of this document stated that HTTP Response Headers cannot be set in a <VirtualHost> block directive. Experience has proved this to be inaccurate, certainly for v 1.xx. If you do use a <VirtualHost> directive, do so with caution.

The order of the above list is important. <Directory> is overridden by <Files> is overridden by <Location>.

For full details of block directives, please consult the official Apache documentation, in particular <http://httpd.apache.org/docs/sections.html>

The key thing about all this of course is that you can apply different labels to different sections of your content. As some documentation suggests that the <VirtualHost> directive does not support HTTP Response Headers, the recommended way to label a given website on a server is to apply a <Location> or <Directory> block directive thus:

Example 4: Setting headers within a Directory block directive

```
<Directory dir>
Header set pics-label '(pics-1.1
"http://www.icra.org/ratingsv02.html" l
r (cz 1 lz 1 nz 1 oz 1 vz 1))'
</Directory>
```

To label a whole website, *dir* should be the absolute path to the website's root directory on the server.

The same block directive can be used to label a particular section of a website if all its files are stored in a given directory - just set up another <Directory> block directive with *dir* set as appropriate. As a facetious example, you might want to label www.animals.com/birds/ differently from www.animals.com/insects/.

Apache processes <Directory> block directives in increasing order of the number of elements. So that <Directory "D:/root/website1"> is processed before <Directory "D:/root/website1/section">. Therefore, the label you intend to apply to the section directory will overwrite the previous one correctly. See section 10 for more on this.

<Files> and <Location> block directives are processed in the order in which they appear in the *config* file.

Example 5: Setting headers for a specific file

For our purposes, this is just a logical extension of the <Directory> block directive. As an example imagine you had a site which should carry rating A,

but that your index page, uniquely, should carry rating B. This would take care of it:

```
<Files index.html>
Header set pics-label '(pics-1.1
"http://www.icra.org/ratingsv02.html" l
r (cz 1 lz 1 nz 1 oz 1 vz 1))'
</Files>
```

Notice that the <Files> block directive takes a relative path (to DocumentRoot) not an absolute one.

Example 6: Using the <Location> block directive

Depending on your situation, this is perhaps the most easy to use block directive since it takes a URL as its argument rather than filenames and paths on your server. Labelling www.example.org becomes:

```
<Location www.example.org/>
Header set pics-label '(pics-1.1
"http://www.icra.org/ratingsv02.html" l
r (cz 1 lz 1 nz 1 oz 1 vz 1))'
</Location>
```

4.2 Using Wildcards and Regular Expressions

The examples so far have all been very specific. Apache block directives, however, are far more flexible than we have hitherto discussed. This works very much to our advantage in terms of labelling.

For example, the ICRA labelling matrix includes a section on chat. ca 1 codes for unmoderated chat (or message boards), cb 1 codes for moderated chat and cz 1 declares that there are no chat facilities or message boards. So you might have a default label for most of your site that declares cz 1, but you might also have a full-blown chat facility and the chances are that all the relevant URLs have the word chat in there somewhere. So use a wildcard like this:

Example 7: Using wildcards to label a type of content

```
<Location *chat*>
Header set pics-label '(pics-1.1
"http://www.icra.org/ratingsv02.html" l r (ca 1 lz 1
nz 1 oz 1 vz 1))'
</Location>
```

With that in place, no matter how many times the pages are updated, improved and added to by the webmaster team, the chat areas will carry this label.

The danger here, of course, is that *any* URL that includes chat as four consecutive characters will carry this label. Bad news for a site about the Chatanooga Choo Choo.

This is where some interplay between different people in your organisation becomes important! If the block directive in Example 7: were amended simply to include a forward slash after the word chat thus: `<Location *chat/*>` then only content whose URLs included a path which at some point had chat immediately before a forward slash would carry this label.

Example 8: Using Regular Expressions

The subject of Regular Expressions is has filled many books and I'm not about to attempt give a full lesson on it here! However, they are an extremely powerful tool. Imagine your server has 4 websites:

- cats.com
- dogs.com
- warthogs.com
- zebras.com

You can label all the content in the cats, dogs and any other site beginning with "a" through "m" with a block directive like this:

```
<DirectoryMatch /[a-m].*>
```

Meanwhile the warthogs, zebras and other latter end of the alphabet wildlife would be taken care of by this block directive:

```
<DirectoryMatch /[n-z].*>
```

(You've seen enough PICS labels now, these are just the opening tags for the block directive!)

Example 9: Setting up your own classification scheme

Using wildcards or regular expressions, it is possible to establish your own easy-rating system by simply naming files in a pre-defined way. For example, you might want to divide the content on your site into age-based categories. You may decide, for example, that some content on your site should carry a "PG" rating or a "12" rating. OK - set up these two `<Files>` directives:

```
<Files *-pg.*>
```

and

```
<Files *-12.*>
```

Now *any* file on your site which has -pg. immediately before the file extension will carry your PG rating, any file with -12. immediately before the file extension will have a 12 rating. Any file with neither string immediately before the file extension would carry the default label (if you set one).

4.3 Using a .htaccess file

It is possible to add/delete/amend PICS labels to web content without stopping/restarting the server by including HTTP Header Responses in a .htaccess file.

NB. Only the <Files> and <FileMatch> block directives can be used in .htaccess files, not <Directory>, <Location> or <VirtualHost>.

The pros and cons of using a .htaccess file are well understood (flexibility vs. server load). For our purposes here it is probably most applicable as a mechanism for labelling ephemeral content. However, the suggestion outlined below may be of interest to geographically diverse organisations and networks.

4.3.1 Just a suggestion

You might consider setting up a secondary .htaccess file specifically to handle the labels. Apache supports multiple .htaccess files so one option might be to include a configuration like this:

```
AccessFileName .htaccess, .filename
```

The .htaccess file would contain whatever you put in your .htaccess file now with the separate *.filename* file just used for labelling.

In tests, I used the following <Files> directives:

```
<Files *-pg.*>
Header set pics-label '(pics-1.1
"http://www.icra.org/ratingsv02.html" l r (cz 1 lb 1
nz 1 oz 1 vz 0))'
</Files>
```

```
<Files *-12.*>  
Header set pics-label '(pics-1.1  
"http://www.icra.org/ratingsv02.html" l r (cb 1 lb 1  
lc 1 nz 0 oz 1 vz 0))'  
</Files>
```

Initially, these were tested with one block directive in each of 2 separate files: .htaccess and another file that I called .piclables (the name is not significant) - and it failed. Only the block directive in whichever file was declared second in the AccessFileName declaration in the *config* file worked. However, putting both block directives in a single file worked perfectly, whether this was declared first or second in the AccessFileName list.

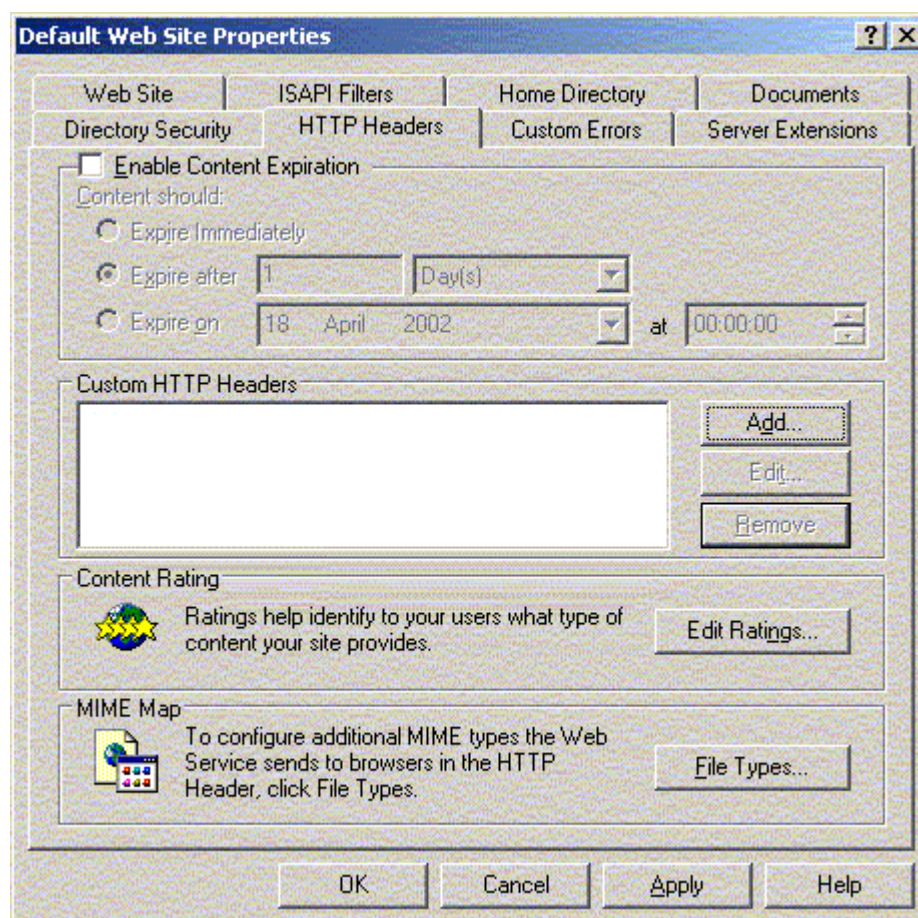
The policy/organizational implication here being that this method makes it possible for a member of staff to maintain a labels file as a separate entity. Give that member of staff FTP access to the relevant directory on your server and s/he can take care of the whole job by remote.

5 Configuring Microsoft servers

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

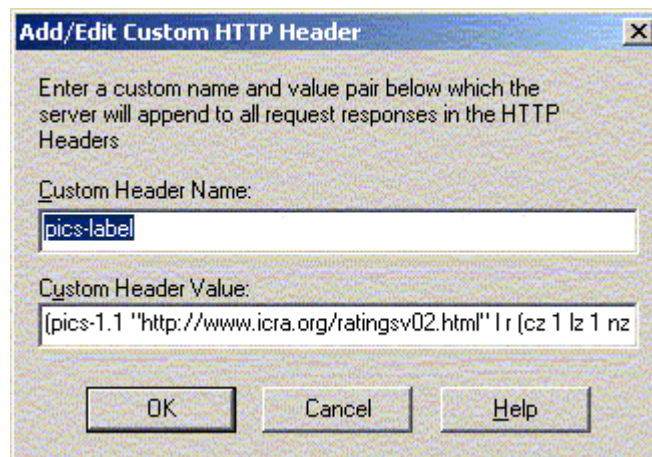
- Web server
- Home directory / Web site (IIS 4 and later support multiple web sites)
- Virtual directory
- Folder
- Page

To set the HTTP Header properties, select the required level, 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. As shown, an e-mail address and content expiry date can also be sent within the HTTP Header (these are unrelated to PICS labels).



Please **do not** use the [Edit Ratings] function. If you add the ICRA .rat file (the file that defines the ICRA rating system within the PICS standard) to the System32 folder, then you can see the ICRA ratings in the relevant dialogue. **But** Microsoft makes a mess of things by using the old RSACi identifier and writing in a whole jumble of a label which, not surprisingly, the filters can't make sense of. So please, just stick to the custom headers.

Click the Add button, enter pics-label in the Custom Header Name field and the label itself in the Custom Header Value field to give you something like this:



And that's it. If you have a dedicated server for your site and you can legitimately apply the same rating to every page and you use a Microsoft server - this one addition will label the whole site - without a meta tag in sight.

One limitation: Windows Server 2003 seems to set a limit of 200 characters on a header which might be restrictive if you have a long rating and want to include several gen true for "URL" statements. The way round it would simply to have several Pics-Label headers. You don't have to put all your labels in a single header.

You can apply labels to directories and specific pages by going through the same process as required (just right click on the relevant directory or file). However, some of the "nice touches" that Apache offers - such as maintaining and storing the labels in a separate file are not available with Microsoft servers.

6 Testing/Viewing HTTP Response Headers

The ICRA website has a tool that will visit your site and test the labels. It also offers the option of showing you the retrieved headers and content.

To see the labels in your HTTP response headers directly you could telnet your site, but there are a number of tools on the web for showing them more easily.

See section 12, page 30

7 Labelling sites using HTML meta tags

As an alternative to HTTP Response Headers, PICS labels can be delivered as meta data within the HEAD section of HTML pages.

Example 10: A full ICRA label for www.example.org in a meta tag

```
<meta http-equiv="pics-label" content='(pics-1.1
"http://www.icra.org/ratingsv02.html" 1
gen true for "http://www.example.org/"
r (cz 1 lz 1 nz 1 oz 1 vz 1))'>
```

The elements in this label are exactly as described in section 2 but the label is delivered as an http-equiv meta tag. If you're using this method, the gen - for elements are crucial. Recall that in order for a filter to apply a rating label to a given web resource, either the label must be delivered with that resource, or - importantly for us here - the filter must already hold a label in cache which can be applied to it.

Furthermore, HTTP is a stateless protocol - every call to an external file is a completely separate transaction between client and server.

Example 11: A simple HTML fragment (unlabelled):

```
1) <HTML>
2) <HEAD>
3) <TITLE>A title</TITLE>
4) <SCRIPT SRC="/scripts/script1.js"></SCRIPT>
5) </HEAD>
6) <BODY>
7) <H1>That title again</H1>
8) <IMG SRC="/images/image.gif" ...
```

To load this page requires not one but three separate client-server requests - the HTML document, the external JavaScript file and the image. An exchange between the server and a PICS-aware client would go something like this:

- GET the HTML document, notice that there is no label in the HTTP headers. Is there a label in cache for this file? No.
- Is there a label present? Not yet, but might be one later as we haven't reached the end of the <HEAD> section
- GET the JavaScript file, notice that there is no label in the HTTP headers. Not an HTML document. Is there a label in cache? No - resource is unrated
- </HEAD> tag reached, stop looking for labels. Page is unrated
- GET the image, notice that there is no label in the HTTP headers. Not an HTML file. Is there a label in cache? No. resource is unrated.

In other words, this simple fragment of HTML could generate **three** block messages in a filter that was set to block access to unrated sites.

In order to label this page using an HTML meta tag, the tag must be placed around line 3 - after the <HEAD> tag and before the first call to an external file as in the following example:

Example 12: A simple HTML fragment (labelled):

```
1) <HTML>
2) <HEAD>
3) <TITLE>A title</TITLE>
4) <meta http-equiv="pics-label" content='(pics-1.1
    "http://www.icra.org/ratingsv02.html" 1
    gen true for "http://www.example.org/"
    r (cz 1 lz 1 nz 1 oz 1 vz 1))'>
5) <SCRIPT SRC="/scripts/script1.js"></SCRIPT>
6) </HEAD>
7) <BODY>
8) <H1>That title again</H1>
9) <IMG SRC="/images/image.gif" ...
```

The three exchanges with the server would now be something like:

- GET the HTML document, notice that there is no label in the HTTP headers. Is there a label in cache? No.
- Is there a label present? Yes.
- Does the label include a gen true statement? Yes - add to cache
- GET the JavaScript file, notice that there is no label in the HTTP headers. Not an HTML file. Is there a label in cache that can be applied to this? Yes
- GET the image, notice that there is no label in the HTTP headers. Not an HTML file. Is there a label in cache that can be applied to this? Yes.

If the label were placed on a line after the external script had been called, the fourth test - whether there was a label in cache that could be applied to the

script file - would fail, even though there is a label later in the <HEAD> section.

So the key point from this is that the position of the label matters - it must come **before any external file is called** if you are relying on gen-true-for elements to label those files.

7.1 Different labels for different parts of the site

Suppose you want to label a section or an individual page differently from the rest of the site? PICS allows you to do that. Recall the basic principle of the gen-true-for elements. If a label includes a gen true flag, then the label is cached and can be applied to any URL that begins with the string quoted in the for statement.

Example 13: Two labels for different parts of the same site

```
<meta http-equiv="pics-label" content='(pics-1.1
"http://www.icra.org/ratingsv02.html" 1

gen true for "http://www.example.org"
r (cz 1 lz 1 nz 1 oz 1 vz 1)

gen true for "http://www.example.org/branch/"
r (ca 1 lz 1 nz 1 oz 1 vz 1))'>
```

NB: Example 16: below gives more details of how to put multiple labels in a single meta tag.

The meta tag in Example 13: effectively labels all URLs that begin with "http://www.example.org/branch/" with the ca 1 ICRA descriptor which codes for unmoderated chat. The remainder of the site is labelled with "none of the above" in all sections of the questionnaire.

Here's the crucial bit:

When deciding which rating to apply to a given resource, the filter will use the one with the longest matching string in the *for* statement. Therefore http://www.example.org/branch/page.html is labelled as containing chat whilst http://www.example.org/another/page.html is labelled as having none.

You can use a gen false tag to specifically label an HTML document:

Example 14: A specific (gen false) label

```
<meta http-equiv="pics-label" content='(pics-1.1
"http://www.icra.org/ratingsv02.html" 1
gen false for "http://www.example.org/page.htm"
r (cz 1 lb 1 nz 1 oz 1 vz 1))'>
```

Here, the specific page (page.html) carries a label that declares crude words or profanity.

- Q. What label would be applied to any images on this page?
- A. Any generic label held in cache for which there was a matching gen-true-for statement, NOT this one which ONLY applies to the HTML document.
- Q. Would the label be cached?
- A. No. Only gen true labels are cached.

7.2 Labels for resources pulled from other domains

See if you can spot the problem with the next example - there's just one change from Example 12:

Example 15: HTML fragment

```
1) <HTML>
2) <HEAD>
3) <TITLE>A title</TITLE>
4) <meta http-equiv="pics-label" content='(pics-1.1
"http://www.icra.org/ratingsv02.html" 1
gen true for "http://www.example.org"
r (cz 1 lz 1 nz 1 oz 1 vz 1))'>
5) <SCRIPT
SRC="http://script.com/scripts/script1.js">
</SCRIPT>
6) </HEAD>
7) <BODY>
8) <H1>That title again</H1>
9) <IMG SRC="/images/image.gif" ...
```

The HTML document at example.org is labelled, as is the image, but the script isn't. That's because it's pulled from another domain, in this case "script.com." The label on the document, which the filter will cache, only covers URLs

beginning with `http://www.example.org`. Therefore, to label the script, we need to include more than one label in our meta tag as shown in the next example.

Example 16: Multiple domains in a single meta tag

```
<meta http-equiv="pics-label" content='(pics-1.1
"http://www.icra.org/ratingsv02.html" l
gen true for "http://www.example.org/"
r (cz 1 lz 1 nz 1 oz 1 vz 1)

gen true for "http://script.com/"
r (cz 1 lz 1 nz 1 oz 1 vz 1))'>
```

The opening of the meta tag, rating service identifier and the lower case `l` are not repeated, just the `gen-true-for` statements and the ratings parentheses.

This is how you can include a label in your site that labels material pulled from sites you don't control. This is particularly useful if your site carries **banner advertising!**

7.3 Summary for HTML meta tags

The key thing to remember at all times is that any file, whether it be an HTML page or objects pulled into it, must either arrive at the filter with a label, or the filter must already have a label in cache with the relevant `gen-true-for` elements which mean the label can be applied to the incoming content.

If *all* your visitors *always* enter your site through the homepage, and if *all* the content on your site should have the same rating, *then* a single meta tag in the index file of your root directory can effectively label the whole of your site. But you don't need me to point out how unlikely this scenario is!

If you're labelling your site using HTML meta tags, then every page on your site should carry a label which covers not just the page itself, but all the elements pulled into that page.

8 Scripting techniques, SSIs etc.

There are any number of ways you can use scripts and SSIs to add labels to websites. The way your particular site is organized will determine what's best for you so I can't give a definitive set of answers here, only some pointers.

8.1 *The same SSI for each page*

If every page on your site uses the same SSI, or one of a small number of SSIs to write the <HEAD> section then it's easy enough to add in the ICRA meta tag!

8.2 *Multiple domains pointing at a single site*

If you have multiple domains pointing to your site - and in this context domain names with and without the www prefix count as two - then an SSI that calls a system variable can save a lot of space.

Example 17: Writing in the domain name with an SSI

```
<meta http-equiv="pics-label" content='(pics-1.1
"http://www.icra.org/ratingsv02.html" 1
gen true for "http://<!--#echo var="HTTP_HOST" -->"
r (cz 1 lz 1 nz 1 oz 1 vz 1))'>
```

You can still add in further labels to cover your banner ads etc as described in Example 16: but these would need to be hard coded.

8.3 *Using a script to write the label in an HTTP Response Header*

CGI scripts can include labels in either the HTTP Response headers or in the HTML header, whichever you prefer. If you want to put the label in an HTML meta tag, include it in the usual way with the rest of your HTML. But you can do it the smart way and include the label in the HTTP Response Headers

Example 18: Using CGI to write a PICS label into the HTTP Response Header

```
print "pics-label: (pics-1.1
\"http://www.icra.org/ratingsv02.html\" 1 gen true
for \"http://www.example.org\" r (cz 1 lz 1 nz 1 oz
1 vz 1))\n";
```

Anywhere before the usual

```
print "content-type: text/html\n\n";
```


Or, if applicable, the server redirect line

```
print "Location: http://somewhereelse.com/\n\n";
```

NB. Note the escaped double quotation marks around the quoted URLs.

8.4 Keeping the bandwidth down

Configuring your server(s) to include a label with every file means that the label does not need to include the `gen-true-for` elements and is therefore short. The effect on bandwidth is going to be minimal. If, however, you want or need to write labels that include one or more `gen-true-for` elements, they can become large and you won't want to serve labels to a filter that's already got what it needs held in cache. So how's this:

Example 19: Sending labels first time only

```
if (index($ENV{HTTP_REFERER}, $ENV{HTTP_HOST})==-1){
  print "pics-label: (pics-1.1
  \"http://www.icra.org/ratingsv02.html\" 1
  gen true for \"http://www.example.org\"
  r (cz 1 lz 1 nz 1 oz 1 vz 1))\n";
}
```

In other words, if the visitor to the page is coming from another one on your site, they'll already have received the label which should be held in the filter's cache. If they've come from somewhere else (i.e. your host name is not a substring of the referring URL), then they won't have the label so you'd send it to them.

9 Testing the labels

ICRA provides an online tool for testing labels. This returns one of three possible results:

1. Green light – as far as the system can tell, everything the tester can find at the URL you entered is labelled.
2. Amber light – the page itself is labelled but it contains some elements that are not labelled (see below).
3. Red light – no valid labels found at the URL given.

9.1 *How the label tester works*

The label tester is not a sophisticated tool. It simply visits the URL given using its own agent (a Perl module). ICRA labels found on the page are recognized and their gen true | false flag(s) noted. The tester then attempts to parse the document looking for external element sources such as images, JavaScript files etc. It does this by simply looking for “src = “ (or syntactic equivalents). If the external source is not covered by any gen true labels already found, the agent visits those sources and again looks for a valid label. Any resource that isn't labelled is presented to the user in a table and the overall output is set to “Amber light.”

If the concept of elements within a labelled page being unlabelled isn't clear, please revisit section 7.2.

10 HTTP and HTML together

A common question is: if we configure the server to include a default label in the HTTP Response Headers, can we then override that at document level with an HTML meta tag?

Yes, you can, but only if you include a gen true for “URL” sequence in the server-fed label. Here’s why:

As discussed in section 7.1 , if a filter has two labels, either of which can be applied to the same URL then the closer match within the “for” statement is applied. So, if a filter has a rating for `http://www.example.org/` and a different rating for `http://www.example.org/branch/`, then the latter will be applied to all URLs in the `/branch/` directory in preference to the one for the whole `example.org` domain.

More specific overrides more generic.

But what about labels that don’t include a for statement? The PICS standard states that these labels are *the most specific* of all and will therefore be given precedence over any label that includes a gen true or even a gen false statement.

11 Label generation

The information in this document is enough (just) for you to simply write an ICRA rating label - and ICRA doesn't mind a bit if you do that. However, there are easier ways and, from our point of view, preferable ways.

As well as the main label generator on the ICRA website, there is a variety of label generators and modules available for download and use on or off line. The aim is to make label generation easy. See section 12 page 30.

Use of ICRA labels, whether generated automatically or written by hand, is subject to the organisation's terms and conditions which are available on the ICRA website.

11.1 Z 0 - the don't know option

When labelling a large network of properties it is not always possible to determine what rating should be applied to some content. Your servers may host content over which you have no direct control for instance. How should you label such areas?

One option open to you is to make explicit use of a "z 0" descriptor.

For example, if we were to write out in full a label that declared a resource as containing mild expletives it would be

```
la 0 lb 0 lc 1 lz 0
```

That is, explicit sexual language is absent (la 0), crude words or profanity is absent (lb 0), mild expletives are present (lc 1), none of the above is therefore not applicable (lz 0).

For the sake of brevity, an ICRA label doesn't actually include all these terms, we just write in the lc 1 as the zero terms are implied by their absence. Setting all the language descriptors to zero therefore has a subtle meaning:

la 0 Explicit sexual language is absent
lb 0 Crude words or profanity is absent
lc 0 Mild expletives are absent
lz 0 None of the above does not apply

In theory, we could leave out all the language descriptors but in practice we always include at least one descriptor for each category. And if we actually want to declare 0 for all language descriptors we do it by explicitly stating lz 0 - "none of the above does not apply." In the absence of any declared descriptors, this translates into natural language as "no declaration of the language content is being made." In simpler terms, we don't know.

The parent's view of this, when setting a filter is to be able to *require* lz 1 to be declared, or accept lz 0 in the absence of any other descriptor being set to 1. The screen shot below shows the relevant section of the ICRA^{plus} control panel.

Language	Explicit sexual	<input type="radio"/>	<input type="radio"/>
	Crude/profanity	<input type="radio"/>	<input type="radio"/>
	Mild expletives	<input type="radio"/>	<input type="radio"/>
	Only allow if none of the above	<input checked="" type="checkbox"/>	

By selecting "Only allow if none of the above" the other options are greyed out - the filter requires lz 1 to be present. Deselecting the "Only allow if none of the above" option makes the others available but leaves them set to block as shown below:

Language	Explicit sexual	<input type="radio"/>	<input type="radio"/>
	Crude/profanity	<input type="radio"/>	<input type="radio"/>
	Mild expletives	<input type="radio"/>	<input type="radio"/>
	Only allow if none of the above	<input type="checkbox"/>	

In the situation shown here, any of la 1, lb 1 or lc 1 would be blocked, but lz 1 and lz 0 would both be allowed.

Similar logic can be applied to the other categories. In the violence category and the nudity and sexual material category it's possible to write a label that says "there may or may not be any of these things but if present, they're presented in a medical, educational or artistic context and are suitable for young children."

12 References and links

Documentation

The W3C PICS Recommendation document is at <http://www.w3.org/TR/REC-PICS-labels>

ICRA's FAQs may answer your specific questions quickly. See <http://www.icra.org/faq/>

The ICRA rating codes are described at <http://www.icra.org/faq/decode/>

Label generation tools

The primary ICRA label generator is at <http://www.icra.org/label/generator/>

Alternative label generation tools are available at <http://www.icra.org/label/tools/>

Label testing tools

The ICRA label tester is available at <http://www.icra.org/label/tester/>

An example of an HTTP header viewer can be found on DJ Delorie's site at <http://www.delorie.com/web/headers.html>.

13 Document history

This is version 2.2 and includes input from Michael Radwin, the guys at Waldo Kitty who checked out the <VirtualHost> issue and Colin at Insight Eye UK to all of whom I am most grateful. Any further feedback, positive, negative or indifferent is welcome.

13.1 Changes from version 1.1

- Advice on not using the Virtual Host block directive has been deleted. This appears to work just fine!
- References to label generation options has been updated to reflect range of generators available on the ICRA website
- Whole new section on HTML meta tags and scripting tips added.

13.2 Changes from version 2.0

- Short section on the ICRA label tester added

13.3 Changes from version 2.1

- Removal of RSACi elements in all examples (this coincides with ICRA ending inclusion of RSACi elements in its labels in April 2004).
- Updating of references to ICRA *filter* to ICRA *plus*.
- Examples now use the preferred example.org (cf. foo.com)
- Revised HTML and HTTP together section
- Generalized references to Microsoft servers cf. IIS.
- General editing to match new FAQ layout on ICRA website (to be published April 2004)

14 Index

Age-based classification	13
Banner ads	22
Block directives	10
Directory block	11
Edit Ratings (MS servers)	17
Files block.....	11
Folder	16
gen.....	8
gen false	22
Home directory	16
ht access.....	14
HTTP Header viewer	18
http-equiv	19
l 7	
Label generation.....	28
Label tester.....	26
Labelling non-HTML files	20
Location block	12
mod_header	10
Multiple domains	24
PICS	5
Pics 1.1	7
Precedence	21, 27
r 7	
ratingsv02.html	7
Reducing bandwidth	25
Regular Expressions	13
Terms and conditions	28
Testing labels	26
Virtual directory.....	16
Virtual Host.....	11
W3C.....	5
Who decides what children see and don't see?	5
Who rates the content?	5
Wildcards	13