

RewriteFlags CO

```
cookie|CO=Name:Value:Domain[:Lifetime[:Path]]  
(set cookie)
```

This sets a cookie on the client's browser. The cookie's name is specified by NAME and the value is VAL. The domain field is the domain of the cookie, such as '.apache.org', the optional lifetime is the lifetime of the cookie in minutes, and the optional path is the path of the cookie.

For example, if you wanted to set a cookie to show that a user came in through the "front door", you could do so with:

```
RewriteEngine On  
RewriteRule ^/index\.html$ - [CO=frontdoor=yes:.drbacchus.com]
```

And if you wanted to use this to force someone to come through the front door, you could:

```
RewriteEngine On  
RewriteCond %{HTTP_COOKIE} !frontdoor=yes  
RewriteRule . /index.html [R,L]  
RewriteRule /index\.html - [CO=frontdoor=yes:.drbacchus.com]
```

This ruleset rewrites any request to /index.html if the frontdoor cookie isn't set to 'yes'. Note that the pattern in the [RewriteRule](#) is just . which will match any request.

Log the referer, even after it's not the referer

The goal of this recipe (WARNING - STILL UNTESTED) is to log the referer, even after it's not the referer any more. That is, someone comes to your site, and you log their referer. However, as soon as they start browsing around your site, the referer is your site. We want to know where they came from, for the entire time they are on our site.

```
RewriteEngine On  
  
# Not a local referer  
RewriteCond %{HTTP_REFERER} !my\.host\.com [NC]  
RewriteRule .? - [CO=refer:%{HTTP_REFERER}:my.host.com]  
  
# Log the referer  
RewriteCond %{HTTP_COOKIE} refer  
RewriteRule .? - [E=log:refer]  
LogFormat %{Cookie}i myrefererlog  
CustomLog /var/log/myreferers.log myrefererlog
```

This is untested, and so is just concept code, and may not work. In particular, I'm concerned about situations where there are multiple cookies - I don't think I'll get the right thing. But it's worth experimenting with.

There is an open PR (31781) - you cannot use a colon with the CO-Flag (contained in %{HTTP_REFERER}), escaping the colon [CO=refer:a\:bc:localhost] quoting the values etc. doesn't work.

Got it working that way:

```
# Not a local referer
RewriteCond %{HTTP_REFERER} !127\.0\.0\.1 [NC]
# strip the colon
RewriteCond %{HTTP_REFERER} ^(https?):([^\:]+)
# skip subrequests
RewriteRule ^ - [CO=refer:%1_%2:127.0.0.1,NS]

# extract the value of that cookie (knocks out the multiple cookie problem)
# either we are at the end of the ENV Cookie or an other name/value pair follows
RewriteCond %{HTTP_COOKIE} refer=(.+)($|;\ )
# save the value in the ENV log (I think PR 31781 applies the E-Flag as
# well, - making an env with name a:aa or value b:bb impossible, but untested by myself)
RewriteRule ^ - [E=log:%1,NS]
LogFormat %{log}e myrefererlog
CustomLog /var/log/myreferers.log myrefererlog
```