

RewriteMisc

Look somewhere else for that image

If the image isn't in the /images/ directory, look in /images1/, /images2/, and /images3/ for it as well.

```
RewriteEngine On

# RewriteLog /tmp/rewrite.log
# RewriteLogLevel 6

RewriteRule ^/images/(.*) $1
RewriteCond /usr/local/apache/htdocs/images/%{REQUEST_FILENAME} -f
RewriteRule ^ /usr/local/apache/htdocs/images/%{REQUEST_FILENAME} [L]

RewriteCond /usr/local/apache/htdocs/images1/%{REQUEST_FILENAME} -f
RewriteRule ^ /usr/local/apache/htdocs/images1/%{REQUEST_FILENAME} [L]

RewriteCond /usr/local/apache/htdocs/images2/%{REQUEST_FILENAME} -f
RewriteRule ^ /usr/local/apache/htdocs/images2/%{REQUEST_FILENAME} [L]

RewriteCond /usr/local/apache/htdocs/images3/%{REQUEST_FILENAME} -f
RewriteRule ^ /usr/local/apache/htdocs/images3/%{REQUEST_FILENAME} [L]
```

Redirect Everything

I want all requests to my site to get sent to a certain page. Perhaps for a "system is down for maintenance" condition, or perhaps there's really only one page here. So, for example, let's send everything to **page.html**

```
RewriteEngine On
RewriteRule !^/page\.html$ /page.html [PT]
```

Or, if you want all pages mapped to a handler that knows what to do with that page:

```
RewriteEngine On
RewriteCond $1 !=/handler.php
RewriteRule ^(.*) /handler.php?uri=$1 [PT]
```

Connecting to Tomcat

Problem:

We are using the 2.2 branch of Apache HTTP Server as frontend while all requests with a jsp extension should be served by our tomcat backend server on Port 8009.

Recipe:

```
RewriteEngine On
RewriteRule ^/(.+\.jsp)$ ajp://localhost:8009/$1 [P]

ProxyPassReverse / ajp://localhost:8009/
```

Discussion:

Mod_rewrite passes all requests which are ending with .jsp to mod_proxy ([mod_proxy_ajp](#)). We use the Apache JServ Protocol (ajp), this means the modules [http://httpd.apache.org/docs/trunk/mod/mod_proxy_ajp.html] and mod_proxy must be loaded into the server configuration.

Load balancing

Problem:

We have one frontend server which serves static content while the processing of dynamical content is up to two backend servers.

Recipe:

Using `mod_rewrite`, `mod_proxy` and `mod_proxy_balancer`.

```
RewriteEngine On
RewriteRule ^/(.+\.php)$ balancer://myphpcluster/$1 [P]

<Proxy balancer://myphpcluster>
    BalancerMember http://10.0.0.1 smax=10 loadfactor=20
    BalancerMember http://10.0.0.2
    ProxySet maxattempts=2
</Proxy>

ProxyPassReverse / http://10.0.0.1/
ProxyPassReverse / http://10.0.0.2/
```

Discussion:

A brief description may be provided in the future.

For a list of available [ProxyPass](#) parameters see the [mod_proxy documentation](#).

Note: The directive [ProxySet](#) is undocumented. You can use this directive in order to set additional parameters for a balancer or a worker, because the following does not work as expected:

```
# maxattempts=2 and other paramters are not recognized this way
RewriteRule ^/(.+\.php)$ balancer://myphpcluster/$1 [P] maxattempts=2
```

Log executable file access

We wish to create a log file that logs accesses to executable files, wherever they are in the filesystem, and whatever file extension they have.

```
RewriteCond %{LA-U:REQUEST_FILENAME} -x
RewriteRule ^ - [E=exec:1]
CustomLog /var/logs/exec-cgi.log combined env=exec
```

Discussion

The LA-U syntax does a look-ahead to find out what the value of a particular variable will be later on. This requires a sub-request, and that results in something of a performance hit. Consider using the [ScriptLog](#) directive instead. However, the technique used here can be used for a larger class of problems, when a variable isn't available at request time, but will be later on.

Using HTTP Headers for routing

Problem:

Using a custom HTTP header for rewrite rules to allow calls to be dynamically routed

Recipe:

Using `mod_rewrite`, `mod_proxy` and custom headers sent from a client, we have been able to rewrite end points for web service calls routed through a proxy to multiple backends.

```
RewriteEngine on
RewriteLog "logs/rewrite.log"
#RewriteLogLevel 5

# The following rule routes to the first server
RewriteCond %{HTTP:X-WS-Version} Version1
RewriteRule (.*?) http://myserver1.com:7001$1 [P,L]

# The following rule routes to the second server
RewriteCond %{HTTP:X-WS-Version} Version2
RewriteRule (.*?) http://myserver2.com:7001$1 [P,L]
```

Discussion:

We needed this for internal routing of web service calls proxied from a web app server to our backend servers. We wanted a way to have multiple web service engines in behind with the client not knowing anything about where the call was routed to. The service bus could not always handle dynamic routing but the client could set a header value with routable data. As well, this type of routing is far more efficient than having a service bus pull the message apart and route based on that, even on SOAP headers.