

```
#####
##
##      _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
##      / _ \ / _ _ _ _ _ _ _ _ _ _ / / / / / / / / / / / / / / / / / /
##      / / \ \ / _ _ // _ // _ \ _ \ \ / \ / \ / \ / \ / \ / \ / \ /
##      / / \ \ \ _ \ \ _ \ \ _ \ / / / / / / / / / / / / / / / /
##      \ _ \ / / _ > _ > \ _ > \ \ _ \ / / / / / / / / / / / /
##      \ \ / \ \ / \ \ / \ \ / \ \ / \ \ / \ \ / \ \ / \ \ / \ \ / \ \
## -----
##      Designed and Developed by Brad Jones <brad @="bjc.id.au" />
## -----
#####
```

```
<IfModule mod_rewrite.c>
  # Turn rewriting on
  RewriteEngine on

  # Set the base location
  # Assuming this file is writeable by the webserver, possibly dangerous I
  # guess but if your happy to accept the risk especially while in development
  # this will be set to the correct path by the AssetMini View Helper class.
  RewriteBase /assets/

  # If the files don't already exist, lets use PHP to create them.
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule ^(.*)$ min.php?$1 [QSA,L]

  # If the browser accepts gzip and the requested file exists with
  # a .gz appended, then rewrite the request to the .gz file
  RewriteCond %{HTTP:Accept-Encoding} gzip
  RewriteCond %{REQUEST_FILENAME}.gz -f
  RewriteRule (. *\. (css|js))$ $1\.gz [L]

  # But then we need to reset the Content-Type and Content-Encoding headers.
  <FilesMatch ". *\. js\. gz$" >
    ForceType application/x-javascript
    Header set Content-Encoding gzip
  </FilesMatch>
  <FilesMatch ". *\. css\. gz$" >
    ForceType text/css
    Header set Content-Encoding gzip
  </FilesMatch>
</IfModule>
```

```
    header set Content-Encoding gzip
</FilesMatch>
</IfModule>
```