**rewrite.ini** parser

Rewriter is one of the strongest url rewriter bound to Smarty framework by Sanjaal Corps. This rewriter is an optional installation to polymorph or to change the URL details when it matches certain conditions. This helps to create several dynamic pages with single controller.

Our goal was to provide individual rewrite.ini for for every subdomain. **rewrite.ini** should be always located at /library/services/**domain-name**/rewrite.ini in lowercase filename. And, absence of a rewrite.ini file means no usage of url rewriting.

It uses a simple .ini file, human readable and modifiable, with sections. Each section is composed of four declarations. You should make this file by hand-typing it. Here is an example of a real **rewrite.ini** file:

[contact\_email]

pattern = "/^(contact|email)-(.\*)$/is";

vars = "contact\_quick";

force = "go=now";

page = "contact-form.php";

mime = "text/html";

And, below are the details of these sections and subsections:

**[contact\_email]**

This is a beginning of a section named contact\_email. Anything below it is a subsection under this section, until the parser meets the beginning of the other section. We consider contact\_email as a possible handler.

**pattern = "/^(contact|email)-(.\*)$/is";**

This is the first declaration on how to rewrite the url match. It can vary widely according to your needs on how you should capture your URLs. (.\*?) is a set of capture, and every capture should be matched with **vars** as below. You should be much smarter in regular expressions to write these patterns.

**vars = "contact\_quick";**

This is a comma separated listing of name of variables matching every captures using **patterns**.

**force = "go=now";**

This is a future provision to force adding certain variables when there is a match. Here, if **/contact-*email*** matches, it also tries to add **$go = “now”;** variables. Currently, it is meaningless. But your rewrite.ini file will not validate without a **force** declaration.

**page = "contact-form.php";**

This tells the framework to use a new $page call other than the default one (as seen in the url). Whatever the $page was earlier, it will now perform as the supplied page for every match. URL being matched are never entered into the database – hence there is no record. But when it is rewritten to a different $page value, it will automatically change the controller and page name. So, there is a data.

**mime = "text/html";**

It is a proposed configuration for enforcing MIME headers.

Finally, in action, the link: <url>/contact-sanjaal is rendered as: <url>/contact-form.php?**contact\_quick**=**sanjaal**&**go**=**now**.

**Please note that**, the ordering of sections is an important issue, when similar looking urls are being matched and rewritten. This matching is done in first-come-first-service pattern.