

UPDATE 23rd May, 2005

I've released the first proper version of the Tiki URLt package. This is essentially everything you need to be able to use URL transformations, thus acheiving Search Engine Friendly URLs.

The solution is actually really neat, with some helpful features, including:

- Super-fast HTML parsing and URL translation
- Advanced cached database reads for greater efficiency (configurable!)
- Handles errors gracefully (eg. Page Not Found, Translation Error)
- Can handle non-PHP content (configurable MIME types)
- Works on any web server (Apache, IIS, etc) or can be used with Apache mod_rewrite
- Written to the latest Tiki guidelines (ie. ADODB, PHPdoc, etc)
- Tested and working on Tiki 1.8.5 and 1.9.0 (but should also work with other versions)

Current Known Issues

(1) The HTML parser in the output filter does not handle JavaScript links correctly. This is not ideal, but it is unlikely the parser will ever really be able to translate links embedded in JavaScript.

(2) Smarty seems to submit the same template information multiple times. This means that URLs are double-translated, so the "stop on match" facility is somewhat diminished in value. A Tiki bug has been raised for the problem, although it may be a broader Smarty issue.

(3) Tiki programs that return HTTP redirects (ie. "302 Location:" headers) are not available to the output filter. As a result, the redirects do not bear translated URLs. It is unclear how to solve this issue (are outgoing headers available to PHP4? If so, how!?)

(4) Numerous Tiki programs do not output consistent URLs. For example, a blog page post is identified as "tiki-view_blog_post.php?blogId=1&PostId=1". However, the link "back to blog" is "tiki-view_blog_post.php?find=&blogId=1&PostId=1...". Whilst only a slight difference in ordering, and no doubt easily done, this causes immense problems for URL translations. It is unclear how this can be overcome in a sensible Tiki-wide way.

(5) Blogs are a real pain in the neck for SEFURL work. I'm not sure what the solution is, short of some major blog hackery.

(6) Actually making regex transformations work is a really difficult process. Users require intimate knowledge of regexes, and how web pages are delivered and how requests are made. As a result, this is a very advanced feature, which only the very cunning should consider!