

This feature is an enhancement to the wiki rename feature.

- automatically set up an alias to old wiki page
- option is off by default 😊
- outside links to wiki page don't get a 404

Suggested fix

- create a new database table (we'll call it `pageAliasesTable` in this document) to hold the old and new page names
- if `index.php` finds that page doesn't exist, search the the "old" column of the table for a match to the supplied page name
- generate an "HTTP/1.1 301 Moved Permanently" response, giving the new location (obtained from the "new" field of that database entry) to the web client. (see [w3.org STD1](http://w3.org/STD1) for error 301 implementation details)

How I propose the rename feature would work:

rename algorithm (pseudocode)

```
if (the new_name already exists as a valid page) { promptForAlternativeName return } // if there used
to be a page with this name if (the new_name exists in the "old" column of the table) { // if we are
moving the page back to an older location if (the "new" field for that row matches the existing_name) {
database.pageAliasesTable.deleteTheRow // this catches files that have been renamed more than once
searchAndReplaceAllInstancesOf( existing_name with \\ new_name in the "new" column)
renameThePage database.pageAliasesTable.addRow(old="old_name", \\ new="new_name") return }
else { promptForAlternativeName return } // this catches files that have been renamed more than once
searchAndReplaceAllInstancesOf( existing_name with \\ new_name in the "new column" )
renameThePage database.pageAlaisesTable.addRow(old="old_name", \\ new="new_name") // end
```

The create/editpage and deletepage features would also need changing to check if the page used to exist and either prompt for an alternative name or update the (proposed) `pageAliasesTable` table.

A nice additional touch would be to add a row to the database with "new" set to null whenever a page is deleted permanently (perhaps add a "permanent" checkbox to the delete page). Then `index.php` could generate an [HTTP/1.1 410 Gone](http://tools.ietf.org/html/rfc2616#section-10.4.5) error rather than 404. This causes smart clients to delete bookmarks to the page and smart external wikis to mark the link as invalid.

You could even implement a feature very much like blog pings/trackbacks so that a local Tiki could validate and keep track of links to an external Tiki page and the external Tiki could provide a backlink to our (trackable) local Tiki page (when our page is renamed or deleted, the external Tiki can take the appropriate action). How cool would that be! It could become a strong incentive for admins to deploy further Tiki instances in preference to alternative Wikis.

See also: [WikiDev#Renaming Wiki Pages](#)