

# TikiIntegrator

Some content has been migrated to doc:[Integrator](#) and child pages

## Overview

**Tiki Integrator** is a feature to allow easy integration of third party pages into the Tiki (with look and theme of Tiki page). Integrator may have several repositories. 'Repository' is just a path on the local filesystem where a set of files is stored. The files can be stored in the Tiki file gallery. These files will be processed later by integrator's rules. Such rules usually do two actions:

1. remove all before HTML `<~BODY~>` and after `</BODY~>` tags
2. fix hyperlinks and reference to images so they will point to the Tiki Integrator wrapper to be correctly displayed from inside Tiki Integrator does not do it by itself -- all rules and repositories must be configured by admin. To process the file (note that it works with local HTML files by opening and processing them) just apply configuration rules for a given repository and insert the result inside tiki-center div to integrate a page into tiki.

origins of this idea

---

Originally this feature was developed to access and view doxygened pages from inside Tiki. After some time, the following idea came into my head...  As many other programmers I have a lot of repositories (directories) with third party documentation (usually static \*.html files) which I need to have fast access to from one point -- my intranet Tiki site which is used as a team groupware tool. So Tiki Integrator should help me to do this

## Internals Design

What I plan to implement

---

- Integrator may have a set of repositories and an admin interface that should allow to manipulate the repositories
  - usual operations are: add/remove/edit repository and list repositories
- Integrator may setup several rules for one repository
- Rules can be two types
  - simple -- use underlying *str\_replace*
  - advanced -- use regular expressions
- To be integrated, pages get processed (filtered through) by the wrapper which is applying rules defined for the given repository
- It is possible to load an add on CSS file (which can be redefined by current theme)
  - How do I use the existing CSS found in Tiki?
    1. at first the wrapper will search for files redefined for current theme (let it be prefixed with current theme name)
    2. then a general file (configured for this specific repository) will be searched

DB Schema

---

- Two tables are required to implement this feature
  - *tiki\_integrator\_repositories* -- Integrator repositories
  - *tiki\_integrator\_rules* -- integration rules for repositories
- The following fields are needed in *tiki\_int\_repositories*

- *repID* -- repository identifier
- *name* -- human readable repository name
- *path* -- path relative to root of local webserver
- *start\_page* -- page to be displayed if file argument for wrapper script is missing
- *styles\_file* -- CSS file to be loaded
- *visibility* -- is repository visible in users list (this allows to create "rules only" repositories -- i.e. repositories with the only goal: be rule sources for others... like a "skeleton" file)
- *cacheable* -- is pages in repository should be cached
- *description* -- short human readable text about this repository
- The following fields are needed in *tiki\_int\_rules*
  - *ruleID* -- rule identifier
  - *repID* -- repository identifier
  - *ord* -- rules will be applied according to this order
  - *srch* -- text to search
  - *repl* -- text to replace
  - *type* -- simple ('n' value) or advanced ('y' value)
  - *casesense* -- case sensitivity (y/n) if simple rule
  - *rxmod* -- regex modifiers (see PHP docs) if advanced rule
  - *description* -- rule comment (what is this )

## API

---

- There is a class called `TikiIntegrator` with all needed methods to manage repositories and rules. See <lib/integrator/integrator.php>
- *...is this really interesting for somebody except me?*  -- if yes dive into the source, then ask me for details...

## What has been implemented?

---

- Easy rules copying among repositories
- Every rule may be enabled/disabled
- Predefined set of rules for doxygened documentation (just copy it into your repository )
- Rule preview -- allows to play with rules and see results
  - HTML code view panel
  - Results view panel
- Can intergate remote pages (http(s)://...). This will allow also to integrate locally located scripts (i.e. they should get interpreted by web server before processed by wrapper)
- Special *meta-variables*, I don't know how to name this , may be used in *replace* field
  - `{repID}` ~~~~ ID of current repository (very helpful for links fixing )
  - `{path}` ~~~~ path of configured repository (very helpful for images/scripts fixing )
- Intergated pages can be cached. This helps to avoid processing repository pages each time they accessed, after 1st touch page will be cached and used all next times. Sometimes it helps to reduce CPU load at processing pages... [mose](#) hopes that it will help him 
  - Allow to refresh individual pages
  - Admin can clear whole repository cache
    - It also clear automaticaly on rules change or repository removing
  - Per-repository cache timeouts -- useful when repository periodicaly regenerated. Typical usage: web stats of any kind or doxygened API documentation of highly active project (like Tiki )
    - ... so the best value of cache expiration will be regeneration interval in seconds
  - Tiki Integrator utilize cache related functions of `TikiLib` MEGA-class but, **do not call**

`refresh_cache()` (this method also used in cache control admin panel) for integrated pages! URLs associated with cached data actually can't be accessed for that data by that methods! ~~~~ Use *refresh* button below of integrated page or *clear cache* link on repository edit form.

- You may see integrator in action here: [tiki-list\\_integrator\\_repositories.php](#)

## Requirements

1. In `php.ini` `track_errors` should be `On` to be able to show integration error messages in a convenient way (-+`$php_errormsg`+ PHP variable used).



```
;Store the last error/warning message in $php_errormsg (boolean).
track_errors = On
```

2. `tiki_p_view_integrator` (see general permissions) needed to view configured repositories
3. `tiki_p_admin_integrator` (see general permissions) needed to configure repositories and rules

## Micro HOWTO

As a little tutorial I'll teach you how to integrate [doxygened](#) documentation.

1. at first you need to produce it and to put it into your document root... let it be `/var/www/html/mydoxydoc`. Also consider to fix background of generated pictures -- to make it transparent (not all themes in Tiki have white background). You may do this by one simple command:



```
cd /var/www/html/mydoxydoc;
find . -name '*.png' -exec convert -transparent '#FFFFFF' '{}' '{} ' ';
```

As you may noticed, [ImageMagick](#) needs to be installed on your system.

2. Next you need to define the *repository* using [Admin tool](#). Specify the following [\[+\]](#)
3. OK your first repository has been created. Now it's time to define some **integration rules**. Click on your repository in the table below ...
4. ... and on the new page (*Edit Rules for Repository*) press *copy rules* link... [\[+\]](#)
5. Choose **Doxygened (1.3.4) Documentation** and press the *Copy* button
  - here is a small cheat: by default (just after install) Tiki Integrator already has rules for doxygened repositories (as most usual type of programmer documentation)... in other cases you will need to make such rules for your repository by yourself... but fortunately it is easier than you may think
6. Press/click *view repository* and enjoy
  - it's recommended to use one of the Notheme themes
  - ... also you may ask the author of your preferred theme to provide `doxygen.css` file too

## Todo and Future Plans

- **Need smbd to describe this in documentation** ~~~~ this is **my** technical paper... *official formatted* pages FeatureAdmin/FeatureDev/??? I'm still waiting for authors

- Wiki plugin to integrate other page content
- Easy import/export of integration rules. Need to choose in what way:
  1. XML file
  2. SQL file
  3. PHP serialize/unserialize
- Need a way to see files in repository (smth like ``ls`` or `dir`)
- Repository can be a set of remote files ~~~~ imagine what repository have **only** remote files and that files have no any links. In current concept such files can be integrated only as separate repositories (what is lack). Need a way to add remote files to repository and later ability to list all files in repository...
- Handle HTTP POST/GET methods
  - in what way?
  - what to do with posted data?
    - Posting data to the tracker maybe? The tracker has the most flexibility in terms of field labeling and allows for smaller instances of a specific data set.
  - where to get required data?
    - There's plenty of data
  - how to describe it?
  - what integrated page should look like after *filtering*?
  - ... so many questions w/o (currently) answers... ~~~~ any suggestions?
- Rules sharing (a la symlinks) -- rules can be shared among repositories i.e. it is enough to fix one rule to see effect in all repositories which is use this rule... (I'm tired to fix all my doxygened repositories when smth improved in my rules :) -- *hey ppl! is this interesting for smbd else?*

## FAQ

- Will there be a way to use the file system in tiki to allow users to upload their html files to the tiki file gallery and then have the integrator work its magic?  Yes. Files from gallery should be configured as *remote*.
- Can the integrated pages be edited like a wiki page?  No! tiki-integrator.php just wraps for other (real but located in other place) files.