File and directory structure revamp

This is a work in progress. As things are done, please remove from here, and document at Path Structure.

Objectives

- Make everything work natively with cloud platforms such as
  - Google App Engine
  - Windows Azure
- Make it more accessible for new users and theme designers
  - `lib/wiki_plugins/*` has plenty of goodies but is "hidden"
- For a theme to be distributed as .zip with everything in it
- For a smooth transition by still supporting previous paths
- All writable files should be in one of two folders
  - Temp (delete any time)
  - Storage (all things you should backup)
- Make it possible in the future to distribute themes via a web interface (instead of mods)
- Make it multi-site by default (like multiTiki but with cleaner separation of per site content)
- Make backups easier
- Make it easier to be add as an official package of various distros
- Easier versionning. Ex.: Tiki with SVN, while themes would be managed Git
- Security: some PHP files could be stored outside the web accessible root
- Have a standard place (and configurable) for custom PHP, JavaScript, CSS, translations, themes, and theme options (someone could create a theme option of a built-in theme)
- Move db/local.php elsewhere
- Simplify management

Watch out for

- Multi-domain (to consider in structure)
- Various storage places listed at tiki-admin_system.php

Questions

- Is it acceptable to add rewrite rules (.htaccess) as a requirement to use Tiki? (and thus everything goes via route.php)
  - Does any other system (Symfony, Drupal, WordPress, etc.) do this?
  - For Apache httpd, that would require mod_rewrite, which is currently not a documented dependency. It's a very common module, but I think this would deserve consulting the development mailing list.
Brainstorming for the folder structure.

- `_custom/_default` (starts with underscore so we see at the top)
  - `_custom/_default/config/
  - `_custom/_default/config/database-connection.php` (was db/local.php)
  - `_custom/_default/config/??.xml` (was db/config/custom.xml)
  - `_custom/_default/config/tiki.ini` (should be tiki.ini.php so it's not ever web readable)
  - `_custom/_default/lib/
  - `_custom/_default/lib/smarty_tiki/` (custom smarty plugins maybe smarty_custom?)
  - `_custom/_default/lib/wiki-plugins/` (custom wiki plugins)
  - `_custom/_default/lang/
    - `_custom/_default/lang/fr/` (custom translation files)
  - `_custom/_default/themes/
    - `_custom/_default/themes/css/` (global custom files applicable to all themes)
    - `_custom/_default/themes/js/`
    - `_custom/_default/themes/templates/`
    - `_custom/_default/themes/etc...`
    - `_custom/_default/themes/mycustomtheme/` (specific custom theme)
      - `_custom/_default/themes/mycustomtheme/css/`
      - `_custom/_default/themes/mycustomtheme/etc...`
    - `_custom/_default/themes/fivealive-lite/options/mycustomthemeoption` (specific theme option for bundled fivealive-lite theme for example)
    - `_custom/_default/themes/fivealive-lite/options/mycustomthemeoption/css`
      - `_custom/_default/themes/fivealive-lite/options/mycustomthemeoption/etc...`

Same logic for Multi-Tiki: replace `_default` by

- `_custom/example.org/` (for multi-tiki)
  - `_custom/example.org/themes/
    - `_custom/example.org/themes/customthemeforexample.org/
      - `_custom/example.org/themes/customthemeforexample.org/css/` (etc)
  - `_custom/example.org/lang/` (etc)

And "normal" Tiki files:

- `admin/` (move as src/admin/)
- `db/` (rename as config? - I agree with renaming, either conf or "etc". ).
- `doc/`
- `installer/
- `modules/` (should move to lib - also remove modules/cache)
- `src/` (for most of the .php files (no consensus: tiki-read_articles.php, etc renames as read_articles.php)
- `storage/`
  - H5P stuff
To KIL or move

- custom (same as storage?)
- dump
- files We need this for SEFURLs for File Galleries
- maps
- permissioncheck (should be in admin and have writable parts separated into temp or storage)
- tests (should be in admin?)
- tikitests (should be in admin?)
- whelp
- vendor_extra (hopefully KIL-able by Tiki 13);

2 main folder structure

As per Steve's suggestion on the dev list (Jul 21, 2013 10:42 AM) and Marc's reply there could be 2 (or 3) main folders:

**system (or core)**
The "system" or "core" files are the "don't touch unless you really know what your doing" files. These files will be overwritten with an update. The only way file changes survive an update is to check them back into the SVN baseline before a code release.

- admin/
- doc/
- ext/
- files/
- installer/
- modules/
- src/

**local (or custom)**
These files are for example for theme customizations that would take priority over the actual themes stored in the "core" directory structure. This is where entirely custom work would go, things like custom icons. This would also be a good place to store gallery files when not being stored in the database. Regular, and pre-upgrade, MySQL backups could go here as well.

- db/ (rename as config?)
- files/

- example.org/(multitiki)
  - db/ (rename as config?)
  - files/
RightToLeft languages support (RTL):

Add -rtl before the file suffix, for example, if we have theme named "footstrap":

- themes/footstrap/css/footstrap.css
- themes/footstrap/img/footstrap.png

Then for RTL languages there would be:

- themes/footstrap/css/footstrap-rtl.css
- themes/footstrap/img/footstrap-rtl.png

Options

Add options directory and follow the same hierarchy. Ex.:

- themes/footstrap/options/plumb/css/fivealive.css
- themes/footstrap/options/plumb/img/fivealive.png

Ideas

- Custom folder in Tiki root by default (empty maybe with readme and index.php) for user custom files?
- Custom directory for themes?
- Custom directory for modules and plugins (which will become Gadgets)
- For storage/ and temp/, instead of having the base structure, and copied into a subdirectory for multi-tiki, could we keep some symmetry and use a folder called default when multi-tiki is not used/on base install?
- Have a new vendor_asset directory which has a copy of the assets we use (copies with composer post-install script)
- Jean-Marc brought up the question of wiki XML dump feature storing within same space,
  - it increases sizes of backups
  - and has security implications
- Important: we need a strategy for tiki-data storage (ideally outside the root)
Bootswatch

- We should get Bootswatch from https://packagist.org/packages/thomaspark/bootswatch so we'll need a post-install-cmd: https://getcomposer.org/doc/articles/scripts.md#command-events

How to move most PHP files outside the web root?

This would have to be optional. So if we do this, how could it work?

Related links


alias

- File structure revamp