TikiInstallFeatureDev

For the moment this module is not yet planned, but here are few musings on the whole idea

Reasons to have this module/feature

(I still do not know the difference in tiki-speak when you are used in your dayjob to call it a module, but ... whatever)

Prerequisites:

- TikiCoreWishlist

Consequences:

- TikiPackager
- TikiPackageRemover
- TikiDesintegration

AdminMayhem

Tiki is growing with every release. More and more features and modules are being added and so the codebase keeps on growing as well.

The profiles that are currently under development are a great way to aleviate the admin's job from setting all the options for the features he/she wants to have on his/her site(s).

However, this is not enough for all ...

TikiFeatureBloat

Not everybody needs 500+ avatar images (since they do not use the forums), "N" languages or "M" different styles.

And then I am not speaking of the amount of code that remains unused in your htdocs or public_html or similar directories and just sits there, taking up (precious hosting space).

BudgetHosting

There are quite a few users out there that have a tiki site on a free hosting site and/or a small budget hosting. I'm speaking of the <= 100 MB hosting sites. Apart from the fact that they have a MB quota, they often have a file quota as well. For them, current incarnations of tiki are becoming unviable, since a release from the 1.7.3 tarball is taking up a lot of space (3000+ files and 24MB). Add to that session files and cached templates and you have little room left for anything else on their site.

Fixed directories and security

Apart from the fact that not everyone is as happy with opening up directories in their public web space for file

uploads, some things just do not belong in the public section of the webspace (most notably: db, lib, templates_c, etc ...).

It is also sometimes needed to place these directories in a mutual place for a multisite tiki running from the same code.

Current way to fix

So, most ppl I know that use tiki move those sensitive directories around, remove the files they do not need, place some .htaccess files to restrict other sensitive dirs (for instance the templates_c and lib, which cannot be moved around) and maintain their site(s) like this.

As you clearly can see, with the current rate of releases this is a bitch to maintain.

Proposed solution

Tiki Core and all other modules/features

A little refactoring is required (or much), and it needs to made clear what files are used by all modules and which features/modules are inherent in the core functionality of tiki. All the other features/modules need to be divided to their respective files and it needs to become documented what files they use, and on which features (other than 'tikicore') they depend upon.

Install module

Feature preferences pages

An install module takes over from the Admin feature-preferences pages. The page to enable/disable specific tiki-features that were already installed is pulled from the database (and placed appropriately under the section they belong to in no specific order, or alphabetically).

During the install process maybe a new page is installed to manage specific settings as well.

naming conventions could prove golden here.

Functionality

So, how do you go about it then?

Installing a feature would involve the next steps

- upload a .tar.gz or .zip (library) with a specific naming scheme and content.
- select the module/feature you wish to install/update/upgrade (or the one just uploaded)
- examines the library for structure, useability with current version of tiki etc.
- shows a report of the contents of the library and asks for confirmation if nothing is wrong with it or shows a dialog with things why it cannot comply (wrong tiki version, dependent modules are not installed, ...) and offers to make a backup of the database.
- executes the 'install-script', placing the files from the library in the associated directories, running a DB script

This could be refined by allowing only certain files to be overwritten (templates anyone?).

Making Patches

Not only would it be possible to make these modules/features be patched in between versions of tiki, it would also allow to make an update patch between tiki (core) versions possible.

Removing installations

This one is even more trickier then an install or update. It should be possible to deinstall certain features, but with the exception that anything in the database is to be maintained in it. (after all, there could be other features/modules still installed that are in need of that table/data), or when they are still needed by other modules/features, leave the files as well (or abort the whole uninstall or display the dependencies).

Making distributions

In fact, it could also be noted that you could also do the opposite as well. Select the files that belong to a certain feature/module, name it, package it along with a script and distribute this as well. (for backups for instance when you have customised a feature).

Naming conventions

e.g.:
tiki-core-1.7.3-1.0-install.tar.gz or **tiki-core-1.7.3-1.0-update.tar.gz** or **tiki-core-1.7.3-1.0-backup.tar.gz**

in short:
A-B-C-D-E.F

Where:

- A = 'tiki' to distinguish
- B = module/feature name (e.g.: galaxia, forum, wiki, webmail, cms, ...) uniquely and possibly maintained and issued by tikiwiki project managers.
- C = tikiwiki version this is supposed to run under.
- D = every version of tikiwiki this is reset to 1. After that, intermediate quick fixes are
- E = *install* if it is an entire feature/module, *update* if it is a patch since the last release and *backup* is used internally to store the feature's pages and/or table data in during an install.
- F = the extension (either .tar.gz or .zip?)

Script

How the script is organised is to be seen. For now, I propose some sort of blocked text file.

It is divided up into sections (example): tiki-myFeature-1.7.3-1.0-install.tar.gz

```
//FEATURE DESCRIPTION
```

myFeature is a feature that does something. This text is displayed in the install screen so that we know what a feature contains and can abort installing it anyhow.

//REQUIRED BEFORE INSTALL
tiki-core-1.7.3
tiki-myFeature-1.7.3-1.0

//DB SCRIPT
myFeature1.7.3-1.0.sql

//INSTALLATION FILES
$templates/myFeature.tpl
$lib/$feature/myFeature.php

the directories $templates and $lib are part of the configuration of the install feature, pointing to the directories where respectively tikiwiki/lib and tikiwiki/templates are and their names on the server.

$feature is a macro that in this case is substituted by myFeature

predefined directories

- $lib : the lib directory
- $templates : the templates directory
- $tiki : main tiki directory (same as not specifying a directory)
- $modules : tiki modules directory
- $images : tiki images directory
- $img : tiki img directory

modus operandi

The current incarnation of tiki contains a class to use tar files.