Admin Wizard

See also: Help

The Admin Wizard should be *useable for novice admins, installing Tiki for the first time*. It should make it easier to detect and to activate Tiki features.

This is a description of how the wizard works internally, and how to extend it. The initial proposal/specification can be found at the end.

The wizard pages can be changed without changing the core wizard functionality. This includes adding, editing, removing and ordering.

Please also see lib/wizard/readme.txt

Wishes

Open & Pending

<table>
<thead>
<tr>
<th>LastModif</th>
<th>Resolution Status</th>
<th>Submitted By</th>
<th>Volunteered</th>
<th>Title</th>
<th>Category</th>
<th>Version</th>
<th>Feature</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Apr 16</td>
<td>Please retest</td>
<td>Gary Cunningham-Lee</td>
<td></td>
<td>Layouts aren't listed on Configuration</td>
<td>Regression</td>
<td>15.x</td>
<td>Admin Interface (UI) Wizards - Admin Wizard</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wizard - Set up Look and Feel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Login</th>
<th>Full Name</th>
<th>LastModif</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No records found

Enhancement proposals
1.1. Add a screen for General settings

Some general settings are missing (from the "Admin home > General" admin panel, or other spurious admin panels):

1. default email footer
2. menu icons
3. default number of items in listings
4. SEFURL & SEFURL postfilter
5. proxy info (checkbox, + name, port, username 6 pass). Needed for profiles admin panel to work listing profiles when the web site is behind a proxy
6. temp folder.

1.2. Add to the login screen

1. cookie name (to prevent cookie collision on sister Tikis in subdirectories with the same domain in the same hosting)
2. registerKey
3. cookie_consent_feature

1.3. Add a screen for Comments preferences

Comments are important enough in most sites to receive feedback from registered users or even from anonymous. There are a bunch of preferences to show to new admins to choose their options.

Pref for Anti-bot captcha could be also shown there, to allow the use case of allowing feedback from anonymous users.

1.4. Add a link to set permissions for each feature in all screens

Like this for comments:

Permissions

1.5. Add preferences related to Categories

Add preferences related to Categories in the Screen about Categories, instead of just a few links to define categories there.

And when doing so, search for "categ*" in the search box of the admin panel, so that you can see the other preferences scattered around the admin panels related to categories (like "Categorize pages within the structure", or categorize forums, etc).

Consider adding a duplicate of those prefs to the Categories admin panel also, like in all the other cases in which they are duplicated for easier access when they affect categories and wiki, etc., and they are expected to be found in either one admin panel.
How it works

The Admin Wizard can be triggered at login time, or it can be started manually from the Admin panel.

The wizard triggers are hooked up the login process. When a new login is detected, wizardlib's onLogin method is called. Thus all login events are sourced through onLogin.

onLogin will the distribute the event to the correct wizard. Currently there is only the Admin Wizard. It is implemented in tiki-wizard_admin.php.

Manual triggering starts here...

tiki-wizard_admin.php does permission checking and prepares the environment. This includes creating and ordering the included wizard pages. Then wizardlib's showPages method is called with the created pages.

wizardlib showPages does the wizard page handling, i.e. displaying the "current" page and enabling back/next navigation, in addition to random page access.

Each wizard page must inherit from the Wizard abstract class. It defines the 2 required processing methods

- onSetupPage
- onContinue

Wizard pages should/must first call the parent method, before other processing is done.

There is one more function

- isEditable

If this returns false, the wizard will

1. Display the text "Next" instead on "Save and Continue" for the continue button
2. not save specified {preferences=blabla}

wizardlib showPages

showPages will as the pages are stepped through, do the following call sequence:

Page1.onstartup (assume always visible)
  .. User clicks Save and Continue on Page1
Page1.onContinue
Page2.onstartup
  if the page is not visible, i.e. onstartup returns false, the next page it processed.
  .. User clicks Save and Continue on Page2
Page2.onContinue
Page3.onstartup
  ... and so on
When the last page has been processed, the wizard returns to the URL specified in the initial call.

The first page of the wizard must always be visible, and it must not conditionally hide itself.

wizardlib page order

Current page order (sub-pages are indented)
A subpage adds more detail to the main selection

- Language
- LookAndFeel
- DateTime
- EditorType
  - Wysiwyg
  - TextArea
- Wiki
  - AutoTOC
  - Category
  - Structures
  - JCapture
- Files
  - FileStorage
- Features
- Community
- Search
- Advanced
  - Namespace
- Profiles

Revised: Look and Feel moved to front. Language should precede Look and Feel

jb's preferred order as of r48287

- look_and_feel
- language
- date_time
- editor_type
- wysiwyg
- text_area
- wiki
- auto_toc
- structures
- jcapture
- files
- files_storage
- features
- community
- search
The category setting is quite early in the current wizard (sub page to the Wiki page). It is placed this early because other options have category dependent options, e.g. feature_wiki_categorize_structure. Thus it is best to know if categories are enabled or not. If categories are not enabled, the category specific options are hidden.

If categories are moved to the end, this will not be possible.
The main category option is currently on the Wiki wizard page. The sub-category page is only there to make the user aware of where to configure the categories. Ideally the main wizard page should be followed by the corresponding details pages.

The language page should go first, in case the user doesn't speak English.

**onSetupPage**

Load the current settings, and fill the template fields.  
Do **not** set preferences or other values in onSetupPage. This is done in OnContinue.

The page handler must decide if the page should be displayed or not. It can also prepare the looks of the page, or what should be included. 
If the method returns true, the page is displayed.

**onContinue**

The page handler processes the new settings, which are passed in the $_REQUEST variable.

Normally regular Tiki preference settings are used. They will be automatically saved, and requires no processing in onContinue. However, custom form elements can be also included, and may need to be processed.

**How to disable the login wizard**

It is at the time of writing, not sure if the admin wizard will be triggered at login or not. However, the current implementation activates it at login time.

If, for some reason, the Admin Wizard is not wanted at login time, it can be disabled.

1) Set the preference: wizard_admin_hide_on_login = 'n'
This will hide the wizard at login time. However, the Admin Wizard can still be re-activated by an Admin. 
The preference can be set in Admin Wizard itself (Show on login)

2) Disable the login routing
In lib/wizard/wizardlib.php onSetupPage, add as the first line:
return false;
This will stop the login triggers from triggering the Admin Wizard, no matter what setting the admin decides to use.
How to create new wizard pages

The wizard pages are placed in the lib/wizard/pages sub directory. For Admin Wizard pages, start the file name with admin_.

The templates to the wizard pages, are places in the templates/wizard folder.

To add a new page

1. Create a new pages file, e.g. pages/admin_mypage.php
2. Create the template page, e.g. templates/wizard/admin_mypage.tpl
3. instantiate and link the template page in tiki-wizard_admin.php

Wizard page design guidelines

Proposed guidelines

1) Keep each wizard page simple.
   Do not overload with information or options. If it's too much, consider using several pages to collect the option settings.

2) Keep the number of wizard pages low.
   Stepping through 100 setup pages is just too much...

3) The main page should not set detailed preferences in the background. This can cause problems if the wizard is used during upgrades. The wizard may over-write customized settings.
   Instead, it may be better to use a separate wizard page to add the details.

4) Avoid presenting options that "do not apply in this case".
   It may be better to break the page up into 2 separate "cases".

5) Allow the user to complete the wizard
   This means that calls which will redirect or "die" must be avoided. The wizard should handle these cases differently and allow the user to complete the wizard. A solution is to check for the condition in the onStartPage function, then either hide "invalid" template elements, or return false which will hide the page.

6) Open windows to collect additional information in a new tab/window.
   This case applies when an existing, non-wizard panel is used to collect more info. If opened in the same window, the user is required to use the browser "Back" button, which doesn't work well in the wizard. When a new window/tab is used, this window can simply be closed afterwards, and the user can continue stepping through the wizard pages.

7) The first wizard page should always be displayed and provide a "don't show" one-click button
   Do not display the first page conditionally. Especially if the wizard is "forced" upon the user, such as the admin wizard which hooks into the login process. In such cases it is best to provide a one-click "hide and don't show any more", so experience user can easily skip the wizard and keep it hidden.

Wizard page structure

The Tiki Wizard allows pages to be skipped, thus it is possible to hide pages based on previous steps in the
wizard.
Thus it is possible to pop details wizard pages, only for the activated features.

To use this structure

1. Have main feature on the main page
2. Define a wizard page for each main feature to set the details,

and an onSetupPage method, which only activates the page, when the main feature is selected.

The Tiki wizard will automatically skip all non-viewed pages. So, only the activated features prompt for more details.

**Tiki Homepage changes**

The Tiki homepage previously consisted of a group plugin, which showed one content to admin and another content to anonymous users. This approach has the drawback that the page is difficult to edit especially in Wysiwyg mode.

The homepage has now been split.
Only the public part of the homepage now remains. The admin part, showing the different profiles, has been added as the last page in the Admin Wizard.

Additionally an HTML version of the new homepage has been created. This version is installed, if the user chooses wysiwyg HTML mode during the wizard setup. If the user has enabled inline editing, the homepage will be fully editable, as soon as the installation has completed.

**Proposal / Initial Specification**

A first-run admin wizard should be created to ease the Tiki setup for major settings.

In trunk 12, the installation process was extended with an extra step to specify the main Wysiwyg settings. This works fine for manual installations, but not for 1-click installers, where the admins do not use the Tiki installer. A first-run wizard is a better solution, which can handle both manual and 1-click installations.

The Admin Wizard should be **useable for novice admins, installing Tiki for the first time**.

The pages should not be overloaded with options, but enable novice admins to enable/change major system settings. Detail settings are done in the regular admin panel. Links can be provided in the wizard pages.

**Admin Wizard features**

- Simplify default home page (which is not currently WYSIWYGfriendly) [New Tiki Homepage Draft]. It is possible to make move the admin part of the current homepage to the admin wizard, and only use the public part of the homepage. Thus no group plugin, which should make is WYSIWYG friendly.
- Possible to rerun it at any time
- Possible to save or cancel out from any wizard page
- Editor/Wysiwyg setup, like in the trunk installation procedure. See [Tiki12](#) for a screenshot.
  - Namespaces, with namespace separator. Maybe include option to show/hide the namespace
from the name when displayed. (Structure paths, Modules: LastVisitedPages, LastChanges)...or simply turn the display off?

- Date/Time settings
- External File storage locations
  - File Gallery
  - Wiki Attachments

Notes

The wizard may function a little like the installer.

Whether the Admin Wizard should be run at login or not can be a preference: "Hide admin wizard at login". If not set, the admin wizard is started when an admin logs in.

New installations do not set this "hide flag", thus the wizard will be run when the admin logs in. The admin can toggle the value of the "Hide admin wizard at login" preference in the Admin Panel.

A direct link to the Admin Wizard should be somewhere in the Admin Panel.

Need a "paging" mechanism, like the installer. A "step" in the wizard, will probably not depend on previous steps. So, the admin user should have the possibility to go directly to any step of the wizard. Having a tabbed interface may work well.

Current Admin homepage

[+] Related:

- Upgrade Wizard
- Profiles Wizard