

# Create a new preference

Preferences are essentially configuration options with a unique key that can be accessed from all around Tiki to affect the behavior (and can be overridden by [perspective](#).) The preferences can contain arbitrary values. However, by convention, it's better for them to contain strings or numbers. Boolean options are stored as **y** or **n**.

To create a new preference you must:

1. Select a key and set the default value for it
2. Provide a descriptor for it
3. Add it to the administration panels, then clear caches so can you see it
4. Check for it in the code

New Tiki sites will have the new preferences. Existing Tiki sites need to rebuild their preference index before the preference can be found with the *preference search* field. In Tiki 13 and after, this is done Through Admin → Admin Home → the Settings circle on the left tab → Rebuild Admin Index. In Tiki 12 and before, you need to add *?prefrebuild* after your domain name like <http://example.org?prefrebuild>

The descriptors are defined in functions inside of **lib/prefs/\*.php**. The file in which they are contained is defined by the naming of the preference. For preferences containing underscores, the first token maps to the file name. Otherwise, the preference is stored in **global.php**. No need to say your preferences should contain underscores.

**Only major new features should start with `feature_` and go into `lib/prefs/feature.php`**

In fact it's better to start everything with the feature name so all the related preferences are in one file. See [prefs/bigbluebutton.php](#), [prefs/mobile.php](#) or [prefs/payment.php](#) for better examples.

## An example. It's a list definition:

```
'wiki_page_regex' => array(
    'name' => tra('Wiki link format'),
    'description' => tra('Character set used when detecting wiki links within pages.'),
    'type' => 'list',
    'options' => array(
        'complete' => tra('Complete'),
        'full' => tra('Latin'),
        'strict' => tra('English'),
    ),
),
```

For this preference, as the name begin with `wiki_`, the definition will be put in `lib/prefs/wiki.php`  
Preference name without `_` must go in `lib/prefs/global.php`

The descriptors will be used to index the preferences, because Tiki needs a search for configuration options, and to generate the user interface for the configuration option. It can be used inside the administration panels, in

the perspective editor and throughout the application to provide controls to enable the features when they are checked.

In most cases, you can refer to other descriptors to obtain the result you want. There are more details available in [Preferences](#) about the exact options and behaviors. When selecting the name, remember that it can be used in various contexts. The description should not be a direct repetition of the name. Provide more details and attempt to use different words to ease the searching.

To add the preference to an administration panel, please see [Code Howto: Add a preference to an admin panel](#)

Checking for the preferences or using them is just a matter of accessing the right key in the **\$prefs** global array. When checking for a boolean value, it is recommended to always check for the positive value.

#### Checking for positive values

```
// Good
if ($prefs['example_mypref'] == 'y') {
}

// Good
if ($prefs['example_mypref'] != 'y') {
}

// Bad
if ($prefs['example_mypref'] == 'n') {
}
```

When a feature is required to access a complete page, you should use the standard check that will interrupt execution:

#### Standard check

```
require_once 'tiki-setup.php';
$access->check_feature('example_mypref');
```

Checks can also be made within the plugin and module descriptors, preventing them from executing altogether.

For configuration options containing strings or numbers rather than simple on/off flags, they can simply be used as normal values.

## Descriptors

To see/modify the look & feel of the preferences in the admin panels:

<https://sourceforge.net/p/tikiwiki/code/HEAD/tree/trunk/templates/prefs/>

### name

The name of the preference

# description

A 1-line description. This should be meaningful out of context

**name** and **description** are indexed for searching. The quality of the text will impact the usefulness of the search.

# default

Default value (useful for reset)

The default values used to be set by an array in **lib/setup/prefs.php** but in recent versions, they are set with the rest inside of **lib/prefs/\*.php**

# type

Dynamic preferences are dealing with different field types:

- **flag**: shows a checkbox
- **list**: prints a dropdown with a single selection. This field type requires the options array
- **multicheckbox**:
- **multilist**: prints a dropdown with multiple choice possible. This field requires, too, the options array.
- **password**:
- **radio**: radio
- **text**: shows a textfield. This field type accepts 2 properties:
  - **size**: the size of the input field (in characters). Default size is "80"
  - **filter**: the type of data (digits or .....). Default is "accept everything"
- **textarea**: shows a textarea. This field type accepts 2 properties:
  - **size**: the amount of lines displayed in the textarea.
  - **filter**: the type of data (digits or .....). Default is "accept everything"

# options (array)

if relevant for the type

# size

Size attribute of the input field HTML tag (if relevant for the type)

# filter

digits, lang, url, text, ...

# help

wiki page name on doc.tiki.org, which is accessible via question mark pop-up help

# dependencies (array)

If another preference needs to be activated for this one to work. Dependencies are only displayed, they are not actually enforced.

# extensions (array)

PHP extensions required

## detail

Some more information about the preference.

## warning

Will add a hard to miss notice

## hint

**hint** can be used on any preference. The text will be displayed as a note below close to the field.

## shorthint

**shorthint** can be used on text. The text will be displayed on the right of the text box

## keywords

Put synonyms and misspellings that will be caught by the preference search engine in tiki-admin.php and for [Perspectives](#)

## perspective

Preferences can be overridden per [perspective](#). But when this is not wanted/suitable/logical/safe:

**To make it impossible to change a preference per perspective**

```
'perspective' => false,
```

## public

Indicates the preference is a safe value that can potentially be presented to users. Suitable for site titles and other simple values.

Used by the [profile installer](#) and [PluginShowPref](#).

**To allow presenting the preference to a user**

```
'public' => true,
```

## parameters

**A parameter like Autocomplete**

```
'parameters' => array(  
    'autocomplete' => 'off'
```

```
),
```

See also [the commit that introduced this feature](#).

## admin

URL of the admin panel for this feature (if applicable)

This is for a link from the pref to the admin panel. To know where the pref is located, run the `prefreport` script.

## module

filter for module related to this feature (if applicable)

## permission (array)

filter for permissions related to this feature (if applicable)

## plugin

filter for plugins related to this feature (if applicable)

It is better to add the plugin as a preference in the admin panel, near the relevant preferences. Just add a line like the following to `templates/admin/*.tpl`

```
{preference name=wikipugin_kaltura}
```

## view

URL of the list of objects for this feature (if applicable)

## separator

When there are several values, how are they separated? usually with a comma (,)

## translatable

Mark the preference as available for translation

## tags (array)

basic, advanced, experimental, deprecated etc. for use in the admin panel filter and search.

Unavailable will appear to site admin if the system requirements are not met or preference is blocked via [System Configuration](#)

- On a fresh install, basic and new are shown. The user can temporarily (until he changes page) or permanently (until he changes the setting to something else) pick any combination of tags.

- Preferences that are different than the default override the filter and are always shown
- Search engine results respect the pref filter.
- If a pref is not tagged, it is considered as "advanced"
- A pref can have more than one tag.
- Via [System Configuration](#), a web host can limit which prefs are available.
- Jonnyb is working a way for Tiki instances to report [Feedback about prefs](#)

## Tags: what they mean

- **basic**: works well and is an option you typically want to set, for beginners. Stuff you'll want in your first 2 hours using Tiki. 20% of prefs used 80% of the time. This is visible on fresh installs of Tiki8+. Ideally, a user should never find a bug in stable version of Tiki using only basic features. If a bug is found and can't be fixed quickly, the "basic" tag should be removed.
- **advanced**: Less commonly used (long tail) preferences, for power users, programmers, sysadmins and web designers. The other part of the 80-20. If a pref is not tagged, it is advanced. Admins need to choose advanced in tiki-admin.php
- **experimental**: known to be tricky to configure, undocumented, unfinished, new or buggy
- **unavailable**: normally in this version of Tiki but restricted by server admin. See: [System Configuration](#) or can also be caused by missing server requirements (ex.: PHP extensions, storage engine, etc.)
- **deprecated**: Should still work, but this will be removed in a future version of Tiki. Usually means there is a better way of doing it.

## Changing pref tags

- Changing a pref (ex: from basic to advanced) is very easy. If you think a pref is not properly tagged, just change it 😊 Here is an example: <http://sourceforge.net/p/tikiwiki/code/46878>