Bugs & Wish list

Add 'Deprecated' preference filter (split "experimental") | Tiki Wiki CMS Groupware :: Development
Add 'Deprecated' preference filter (split "experimental")

Status
● Open

Subject
Add 'Deprecated' preference filter (split "experimental")

Version
14.x
19.x

Category
- Feature request

Feature
Administration

Resolution status
Confirmed

Submitted by
aexoxea

Volunteered to solve
Loïc Yabili

Lastmod by
aexoxea

Rating
★ ★ ★ ★ ★ (2) ❓

Related-to
- ◑ 15.x+: Expose back preference filters on Admin UI for new tiki admins

Description
Currently, there are four preference filters available in the Control Panel interfaces: "Basic", "Advanced", "Experimental" and "Unavailable".

As a user new to Tiki, I have on several occasions noted features that looked interesting, but which I have not enabled as I have subsequently found it is planned to deprecate them. In some cases this was obvious (e.g. a 'warning' icon with the deprecation message on hover next to the feature checkbox), in others it wasn't (i.e. I only found out about it from doc.t.o or related places).

On the assumption that it will help with planned deprecations if new installations don't enable and start using deprecated features, I would propose that a "Deprecated" preference filter be added, that relevant controls be assigned to it, and that it be disabled by default in new installations.

I understand the criteria for such might be controversial (indeed, I hope I'm not reigniting a heated discussion by posting this!) and so would understand if this is WONTFIXED. I also recognise that it is less appropriate for a new user to suggest criteria here than for those that have had Tiki installations for some time and/or are directly involved in development, which is why I haven't listed any in this initial report.

Importance

8

Easy to solve?
8

Priority
64

Demonstrate Bug (Tiki 19+)

Please demonstrate your bug on show2.tiki.org

Version: trunk ▼

Demonstrate Bug (older Tiki versions)

Ticket ID
5769

Created
Sunday 22 November, 2015 13:04:08 GMT-0000
by aexoxea

LastModif
Saturday 21 September, 2019 07:09:09 GMT-0000

Comments

gezza 23 Nov 15 10:20 GMT-0000

sounds reasonable

anyone knows what the already existing "Unavailable" preference tag is supposed to be used for? maybe we can just rename that, otherwise introduce a new tag

---

found info about "unavailable" here: Create a new preference
so probably it should be a new tag

Jonny Bradley 23 Nov 15 11:08 GMT-0000

As you found by the look of it, unavailable means your server doesn't support that feature, such as full-text search on InnoDB tables in old version of mysql.

Marc Laporte 04 May 18 06:42 GMT-0000

Excellent idea!

Let's do for Tiki19 and and update the developer documentation

  ○ experimental should be for things maybe on the way in

- deprecated on the way out

This information will help us optimize some workflows:

- i18n.tiki.org shouldn't promote to translate deprecated features
- documentation team knows to not invest time on these pages
- There should be a way to warn end users in which version it will be removed. In some cases, exact version of removal is undecided.

Once we have this functionality, we should deprecate a bunch of things in Tiki19 and announce "planned removal for v22"

Torsten Fabricius 04 May 18 12:09 GMT-0000

Excellent.

+1 from me

If possible and convenient (ask the major devs) I would vote for a backport to 18.x for a release of one of the next 18.x minor versions, as this is a functionality that should be in the LTS version aswell (18LTS - 19 - 20 - 21 LTS).

Jean-Marc Libs 22 May 18 12:53 GMT-0000

-1 for backporting.
We don't want to make Tiki LTS unusable by hiding features which will be replaced but the replacement does not work yet, or is impossible to figure out (hello, tubular feature, I'm looking your way).

This is great, but new. We might not get it right the very first time.

Marc Laporte 21 Aug 18 08:31 GMT-0000

I agree about limiting backports of new features and focusing on keeping trunk more stable but I don't understand the connection here. Tabular Tracker: was there a backport that caused an issue? Did it lead to the deprecation (and sort of hiding) of another feature?

I see two aspects

1- Adding a better way to inform users of future plans so they pick feature to use (this task)

2- How we use this new deprecated tag and thus, have guidelines on how the feature replacement should be (quality level) before we remove. This will be on a case by base basis (a different discussion for each deprecation)

Once #1 is added / tested / refined, I think it's a suitable candidate for backporting because it's very

little code / risk. I am not advocating that we do so though.

Let's first finish all in trunk, revisit once it's stable/complete enough. Tiki19 will be out soon!

Marc Laporte 22 May 18 10:52 GMT-0000

@ Serge Kishiko: Thank you for https://sourceforge.net/p/tikiwiki/code/66469/
Next steps:

1. please grep the prefs and add this new tag to existing prefs that have deprecated as a warning
2. add a way for us to indicate when a feature will be removed. This is to help end users plan. They will know what they loose when they upgrade. Possible values:
   - empty (and thus undermined)
   - 19, 20, 21, 22, etc.

Serge Kishiko 23 May 18 18:27 GMT-0000

The first step is done: https://sourceforge.net/p/tikiwiki/code/66482/.
Looking for the way to achieve the last step...

Serge Kishiko 28 May 18 17:54 GMT-0000

I've just finished the last step. I pushed changes on my GitLab fork for test:
https://gitlab.com/Kishiko/tiki/commit/56d7a6fecc987ec4c53c1c111a835ccd59030337

Marc Laporte 21 Aug 18 08:43 GMT-0000

looks good to me. Please add to trunk

Serge Kishiko 30 May 18 19:14 GMT-0000

(Removed). See next comment.

Serge Kishiko 30 May 18 20:02 GMT-0000

List of prefs having "experimental" tag:

1. article_use_new_list_articles
2. bigbluebutton_dynamic_configuration
3. category_defaults
4. comments_inline_annotator
5. connect_feature
6. connect_frequency
7. connect_server
8. connect_last_post
9. connect_server_mode
10. connect_guid
11. cookie_consent_feature
12. cookie_consent_name
13. cookie_consent_expires
14. cookie_consent_description
15. cookie_consent_question
16. cookie_consent_button
17. cookie_consent_alert
18. cookie_consent_mode
19. cookie_consent_dom_id
20. cookie_consent_disable
21. feature_areas
22. feature_galleries
23. feature_machine_translation
24. feature_reports
25. feature_file_galleries_templates
26. feature_faqs
27. feature_quizzes
28. feature_live_support
29. feature_tell_a_friend
30. feature_html_pages
31. feature_htmlpurifier_output
32. feature_quick_object_perms
33. feature_user_encryption
34. feature_user_encryption
35. feature_password_domains
36. feature_webmail
37. feature_slideshow_pdfexport
38. feature_invoice
39. feature_wysiwyg
40. feature_usermenu
41. feature_integrator
42. feature_draw_hide_buttons
43. feature_draw_separate_base_image
44. feature_draw_in_userfiles
45. feature_docs
46. feature_use_fgal_for_user_files
47. feature_use_fgal_for_wiki_attachments
48. feature_editcss
49. feature_view_tpl

50. feature_edit_templates
51. feature_wiki_import_html
52. feature_webdav
53. feature_loadbalancer
54. feature_port_rewriting
55. feature_accounting
56. feature_syntax_highlighter
57. feature_syntax_highlighter_theme
58. feature_dummy
59. feature_time_sheet
60. feature_htmlfeed
61. feature_futurelinkprotocol
62. feature_inline_comments
63. feature_absolute_to_relative_links
64. fgal_fix_mime_type
65. fgal_viewerjs_feature
66. fgal_viewerjs_uri
67. forum_inbound_mail_parse_html
68. forum_inbound_mail_parse_html
69. geo_google_streetview
70. geo_google_streetview_overlay
71. ldap_create_user_ldap
72. metatag_revisitafter
73. mobile_feature
74. mobile_perspectives
75. namespace_enabled
76. session_silent
77. site_layout_per_object
78. theme_styleguide
79. unified_search_textarea_admin
80. user_assigned_modules
81. wiki_list_rating
82. wiki_simple_ratings
83. wikiplugin_list_gui
84. wikiplugin_list_gui
85. wikiplugin_list_convert_trackerlist
86. wikiplugin_maximum_passes
87. wysiwyg_inline_editing

Marc Laporte 30 May 18 20:07 GMT-0000

Will be removed after Tiki18 and before Tiki19:
fgal_viewerjs_feature
fgal_viewerjs_uri
feature_htmlfeed
feature_futurelinkprotocol
feature_time_sheet

There are probably others, but at least, you have something to start with

Serge Kishiko 30 May 18 23:23 GMT-0000

Done here: https://sourceforge.net/p/tikiwiki/code/66560/

Marc Laporte 31 May 18 13:45 GMT-0000

This is accurate:

    'warning' => tra('This feature will be removed after Tiki18 and before Tiki19'),

But if we later want a report of all prefs that will removed per version, it will be a problem.

Let's be more programmatic. I propose to add two optional parameters:

- "Last Supported Version": This preference (and all the related code) will be removed after TikiXX.
- "Is superceded by": Superceded by preference XYZ (with a link)

An example:
https://sourceforge.net/p/tikiwiki/code/HEAD/tree/trunk/lib/wiki-plugins/wikiplugin_swiper.php
will replace
https://sourceforge.net/p/tikiwiki/code/HEAD/tree/trunk/lib/wiki-plugins/wikiplugin_slider.php
(last supported version: Tiki 21)

Serge Kishiko 01 Oct 18 15:07 GMT-0000

Infos added about the replacement of wikiplugin_slider by wikiplugin_swiper: r67765

Marc Laporte>Serge Kishiko 01 Oct 18 15:20 GMT-0000

Replied to Infos added...
hmmmm

Seems odd to put the code there.

@Jonny or **@Victor Tolbert**?

Victor Emanouilov>Marc Laporte 03 Oct 18 06:30 GMT-0000

Replied to
hmmmm... Yes, why don't you put it directly in lib/wiki-plugins/wikiplugin_slider.php::wikiplugin_slider_info - it returns the whole list of prefs for that plugin? Is there any reason to put it in the dynamic code that builds all the plugin prefs?

Serge Kishiko>Victor Emanouilov 03 Oct 18 16:27 GMT-0000

Replied to Yes, why... I thought about it but as I saw a lot of wikiplugin_* prefs were built dynamically despite their own file, reason why I did like that... So, should I change it?

Marc Laporte>Serge Kishiko 07 Oct 18 04:00 GMT-0000

Replied to I thought...
Yes. Will be cleaner.

When someone reads wikiplugin_slider.php, they have the full picture

Marc Laporte 21 Aug 18 08:46 GMT-0000

Also please make sure it works in Preferences report so we have a clear indication of plans.

drsassafras 03 Nov 18 07:34 GMT-0000

prefDoc now supports deprecated.

drsassafras 03 Nov 18 07:50 GMT-0000

A thought about implementation. Right now the warning is used to describe the "will be removed by" feature. This has a few disadvantages.

1. Its a language string that needs to be translated for every new version of tiki.
2. What happens when a pref that already has a warning becomes deprecated? Lines start to get blurred there. I can almost guarantee that it won't show correctly in prefdoc. If the same sorts of icons are ever implemented in tiki preferences that prefdoc uses, it won't show correctly their either, because

there is no way to tell the difference between an EOL notice and a warning.
3. Language strings may be broken when combining a warning and a EOL. I know if one is careful, it's posable to do it without breaking them, but life isn't always like that & we don't have a script that can detect improper string merging.
4. We don't have the planned version of removal in metadata, so I can't make a list of what will be removed in 18, 19 etc.

Replacing that with something like 'remove-by' => '19.1', would overcome those limitations.

my 2 cents.

Marc Laporte 11 Nov 18 13:56 GMT-0000

Agreed. It should be a variable. I am wondering what would be best label.

When I see:
'remove-by' => '19.1',
I think: So it's removed before 19.1? So it appears in 19.0 or not?

Should it be by version or by branch? Ex.:
'last-supported-branch' => '18x',

Attachments

filename	created	hits	comment	version	filetype

No attachments for this item

The original document is available at
https://dev.tiki.org/item5769-Add-Deprecated-preference-filter-split-experimental