

"Recommended revisions", present and future

Yaron Koren

MediaWiki Users & Developers Workshop

April 14, 2025 · Sandusky, Ohio

Common questions for any MediaWiki admin

- Which extensions can I use?
- Which extensions *should* I use?
- For those extensions, what version should I use?

This also applies for:

- wiki farm administrators
- MediaWiki package maintainers

To a lesser extent, it applies to skins as well.

Complicating factors

- Some extensions are poorly-maintained
- Every version of every extension has bugs
- Some extensions require other extensions
- Conversely, some extensions don't work well with one another

Resources for finding good extensions and skins

- mediawiki.org
- Guides, e.g. *Working with MediaWiki*
- [WikiApiary](https://wikiapiary.org/)

Canasta

- <https://canasta.wiki>
- MediaWiki package that I'm one of the developers of - more information about it in Jeffrey's talk
- New version with every LTS MediaWiki version (1.39, 1.43, etc.)
- Like every other package and wiki farm, requires decisions on which extensions and skins to include - and which revisions of those to use

The Canasta solution: contents.yaml

```
extensions:
- AbuseFilter:
  bundled: true
  additional steps:
  - composer update
  - database update
- AdminLinks:
  branch: master
  commit: 4a66a151e3e6bff64b26efbf4491833549b025e5 # v. 0.6.3
- AdvancedSearch:
  commit: d879c5105bbdc574d6c7e95fa4cd20d7b6de6dcd
- AJAXPoll:
  commit: fbfd07ef43063e806026d5bd3f95493e2b189378
- AntiSpooof:
  commit: a010c2d53618aa99b8b8a32d9665de9000e53188
  additional steps:
  - composer update
  - database update
```

Canasta's contents.yaml spawned the "Recommended revisions" project

- https://www.mediawiki.org/wiki/Recommended_revisions - overview
- https://www.mediawiki.org/wiki/Recommended_revisions/1.43 - YAML for MW 1.43
(Looks a lot like Canasta's contents.yaml)

Hopefully, other packages and wiki farms will start using "Recommended revisions"

...and hopefully Canasta itself will use this page directly, starting with the next version (MW 1.47)

Note about using "Recommended revisions"

- Always specify a revision ID!
- **Bad:**
`https://www.mediawiki.org/wiki/Recommended_revisions/1.43?action=raw`
- **Good:**
`https://www.mediawiki.org/wiki/Recommended_revisions/1.43?oldid=7621705&action=raw`
- We need a revision ID to try to guarantee both security and consistency.

Can these recommendations really be "one-size-fits-all"?

- No - there is a good chance that different packages/farms/installations will want different extensions (and to a lesser extent, skins).
- However, the YAML syntax we have developed now supports inheritance, so you can use *some of* another set.

Inheritance

- A YAML file can specify that it is inheriting from another YAML file - and only need to specify the differences.
- e.g. "Package X inherits from the file used by wiki farm Y, which in turn inherits from Recommended revisions/1.43".
- This prevents data duplication, and helps keep software versions standardized.

But there's more...

- It turns out that this YAML listing was the "missing ingredient" for a web-based installer!

Web-based skin and extension installation

- e.g., like WordPress has had for decades:

The screenshot displays the WordPress 'Add Plugins' page. At the top, there is a search bar with a 'Keyword' dropdown and a 'Search plugins...' input field. Below the search bar, there are tabs for 'Featured', 'Popular', 'Recommended', and 'Favorites'. A paragraph explains that plugins extend WordPress functionality and can be installed from the WordPress Plugin Directory or uploaded as .zip files. The main content area features four plugin cards, each with an icon, title, description, 'Install Now' button, 'More Details' link, star rating, and installation statistics.

| Plugin Name | Icon | Description | By | Rating | Active Installations | Last Updated | Compatibility |
|--------------------------------|------|---|-----------------------------|-------------|----------------------|--------------|---|
| BuddyPress | | BuddyPress helps site builders & developers add community features to their websites, with user profiles, activity streams, and more! | The BuddyPress Community | ★★★★☆ (318) | 200,000+ | 2 months ago | Compatible with your version of WordPress |
| Health Check & Troubleshooting | | Health Check identifies common problems, and helps you troubleshoot plugin and theme conflicts. | The WordPress.org community | ★★★★☆ (92) | 80,000+ | 1 week ago | Compatible with your version of WordPress |
| Gutenberg | | A new editing experience for WordPress is in the works, with the goal of making... | Gutenberg Team | | | | |
| bbPress | | bbPress is forum software, made the WordPress way. | The bbPress Community | | | | |

A long-time dream for MediaWiki users

- First attempt: Google Summer of Code project by Jeroen De Dauw in 2010
(https://www.mediawiki.org/wiki/Requests_for_comment/Distribution_and_deployment)
- Project ultimately failed. :(

Big issue: security risks

- Any attempt to "find" extensions in real time is a bad idea.
- How do you prevent a harmful extension (malicious or otherwise) from showing up on everyone's dashboard, where downloading it is just a click away?

The solution: use a trusted resource

- A resource like the "Recommended revisions" YAML file ensures some level of confidence in both the set of extensions/skins available, and the revision used for each one.

Latest project: the Springboard extension

Developed two weeks ago (!) at the Wikimedia Hackathon in Istanbul by Naresh Kumar Babu, Jayanth Vikash Saminathan and Sanjay Thiyagarajan, with design input from me.



Springboard

- <https://www.mediawiki.org/wiki/Extension:Springboard>
- Still in beta, but it works!
- Makes use of a YAML file for the set of installable extensions/skins
- By default, uses "Recommended revisions/1.43" for MW 1.43 - there is still no default for other MW versions.

Springboard

Demo!

What does this mean for Recommended revisions?

Hopefully, there will be "synergy", and a positive feedback loop:

More usage of Springboard, more packages + wiki farms using these files →

more "eyeballs" on files →

better, and more, YAML files (ideally, one for each MW version)

This has already happened with Springboard - development on Springboard led to more parameters added to the YAML file, like "required extensions" and "Wikidata ID".

What about governance?

- Anyone can edit the “Recommended revisions” page.
- It shouldn’t affect any installations – that’s what the “oldid=” is for!
- If this thing becomes more popular, and there are disputes, some sort of mechanisms for decision-making will have to emerge.

Comments/questions