

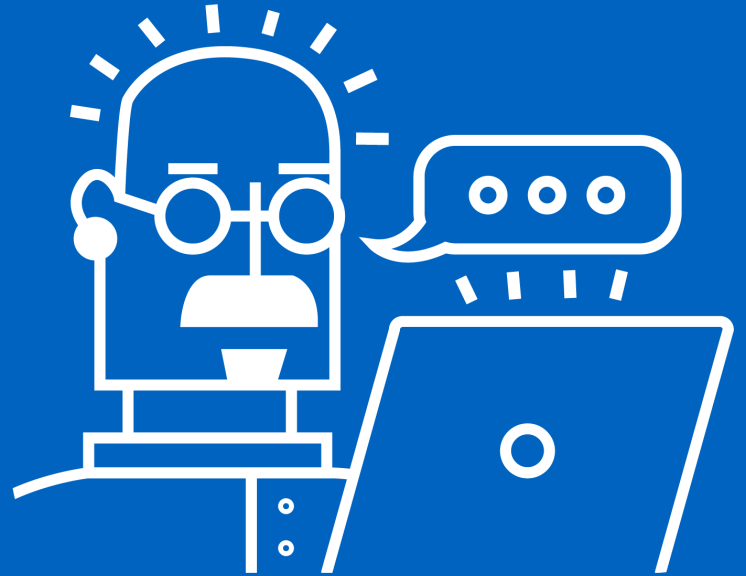
Don't be a `[[Square]]`! Keeping up with wikitext changes.

C. Scott Ananian `[[User:cscott]]` <cananian@wikimedia.org>

MediaWiki Users and Developers Conference 2024

April 17, 2024, Portland, OR

1. Parsoid and wikitext



Changing how articles are written (since 2012)

- *Wikitext* is the markup language used to write articles on MediaWiki wikis.
- The *parser* is the component which turns wikitext into HTML for browsers to display.
- *Parsoid* is the next-generation wikitext parser; used in production since 2012 as the engine behind VisualEditor, our WYSIWYG editor.



Wikitext -[Parsoid]-> HTML

43
44 The '''[MediaWiki Users and Developers Conference](#)''' (MWUD, U&D, [#MWcon](#)), formerly [\[\[EMWCon\]\]](#), will be held from April 17-19, 2024 in [\[w:Portland, Oregon|Portland, Oregon\]](#), United States. The conference is organized by the [\[\[MediaWiki Stakeholders' Group\]\]](#) with the help of its [\[\[MediaWiki Users and Developers Conference 2024#Sponsorship|sponsors and partners\]\]](#).

45
46 The Users and Developers Conference will be a three-day conference featuring discussions of topics related to the usage of [MediaWiki](#) software by and within companies, non-profits, governments, organizations, and communities. If you use [MediaWiki](#) or you're interested in it, we would love to see you attend.

47
48 Sessions will be recorded and streamed. While this will primarily be an in-person conference, we will consider remote presentations on a case-by-case basis.

49 == Event details ==

50 [<mapframe text="Jupiter NEXT" width="200" height="200" zoom="14" longitude="-122.65633866830494" latitude="45.52271669330707" align="left">](#)
51 {
52 {
53 "type": "Feature",
54 "geometry": { "type": "Point", "coordinates": [-122.65633866830494, |
45.52271669330707] },

```
<div class="mw-pt-languages noprint navigation-not-searchable" lang="en" dir="ltr">
  </div>
<div class="noprint" style="float:right;"> </div>
<p>
The
<b>MediaWiki Users and Developers Conference</b>
(MWUD, U&D, #MWcon), formerly
<a href="/wiki/EMWCon" title="EMWCon">EMWCon</a>
, will be held from April 17-19, 2024 in
<a class="extiw" href="https://en.wikipedia.org/wiki/Portland,_Oregon"
title="w:Portland, Oregon">Portland, Oregon</a>
, United States. The conference is organized by the
<a href="/wiki/MediaWiki_Stakeholders%27_Group" title="MediaWiki Stakeholders'
Group">MediaWiki Stakeholders' Group</a>
with the help of its
<a class="mw-selflink-fragment" href="#Sponsorship">sponsors and partners</a>
.
</p>
<p> </p>
<p> </p>
<meta property="mw:PageProp/toc">
<div class="mw-heading mw-heading2 ext-discussiontools-init-section"> </div>
<div class="mw-kartographer-container thumb tleft"> </div>
<ul> </ul>
<div class="mw-heading mw-heading2 ext-discussiontools-init-section"> </div>
<p> </p>
<table> </table>
<p> </p>
<div class="mw-heading mw-heading2 ext-discussiontools-init-section"> </div>
```

Changing how articles are written (since 2012)

- *Wikitext* is the markup language used to write articles on MediaWiki wikis
- The *parser* is the component which turns wikitext into HTML for browsers to display
- *Parsoid* is the next-generation wikitext parser, used in production since 2012, as the engine behind VisualEditor, our WYSIWYG editor
 - Contains a number of new features compared to the previous parser, including support for round-trip conversion from HTML back to wikitext, and preservation of semantic information.
 - In the process of building Parsoid we have been **standardizing, documenting, and cleaning up** the behavior of wikitext.



Changing how articles are read (since 2024)

- Maintaining multiple wikitext parsers is not sustainable!
- The *Parser Unification project* aims to gradually replace the legacy wikitext parser entirely with Parsoid
- In 2024 we began to roll out Parsoid for *read views* on Wikimedia wikis



Edit

Main menu hide

- Main page
- Get MediaWiki
- Get extensions
- Tech blog
- Contribute
- Switch to old look
- Support
- User help
- FAQ
- Technical manual
- Support desk
- Communication
- Development
- Developer portal
- Code statistics

Page Discussion

Read Edit Edit source View history ☆

← → Paragraph ↕ ↻ ☰ + Ω ? ⚠ ☰ ✎ ▼ Publish...

Languages: Bahasa Indonesia ● English ● Sunda ● русский ● 日本語 ●
+ Insert paragraph

The **MediaWiki Users and Developers Conference** (MWUD, U&D, #MWcon), formerly **EMWCon**, will be held from April 17-19, 2024 in **Portland, Oregon**, United States. The conference is organized by the **MediaWiki Stakeholders' Group** with the help of its **sponsors and partners**.

The Users and Developers Conference will be a three-day conference featuring discussions of topics related to the usage of **MediaWiki** software by and within companies, non-profits, governments, organizations, and communities. If you use **MediaWiki** or you're interested in it, we would love to see you attend.

</translate> 🗨 📄 **MediaWiki Users and Developers Conference 2024** </translate>



MediaWiki
USERS & DEVELOPERS
CONFERENCE

<https://commons.wikimedia.org/wiki/Special:UploadWizard>

Read

MediaWiki Users and Developers Conference 2024

- Main page
- Get MediaWiki
- Get extensions
- Tech blog
- Contribute
- Switch to old look
- Support

- User help
- FAQ
- Technical manual
- Support desk
- Communication
- Development
- Developer portal
- Code statistics

[Translate this page](#)

Languages: Bahasa Indonesia ● English ● Sunda ● русский ● 日本語 ●

The **MediaWiki Users and Developers Conference** (MWUD, U&D, #MWcon), formerly **EMWCon**, will be held from April 17-19, 2024 in **Portland, Oregon**, United States. The conference is organized by the **MediaWiki Stakeholders' Group** with the help of its **sponsors and partners**.

The Users and Developers Conference will be a three-day conference featuring discussions of topics related to the usage of **MediaWiki** software by and within companies, non-profits, governments, organizations, and communities. If you use **MediaWiki** or you're interested in it, we would love to see you attend.

Sessions will be recorded and streamed. While this will primarily be an in-person conference, we will consider remote presentations on a case-by-case basis.

Event details [edit | edit source]

MediaWiki Users and Developers Conference 2024



By and for the third-party MediaWiki community

Conference Details

Location: Portland, Oregon, United States
In-person TRN

Changing how articles are read (since 2024)

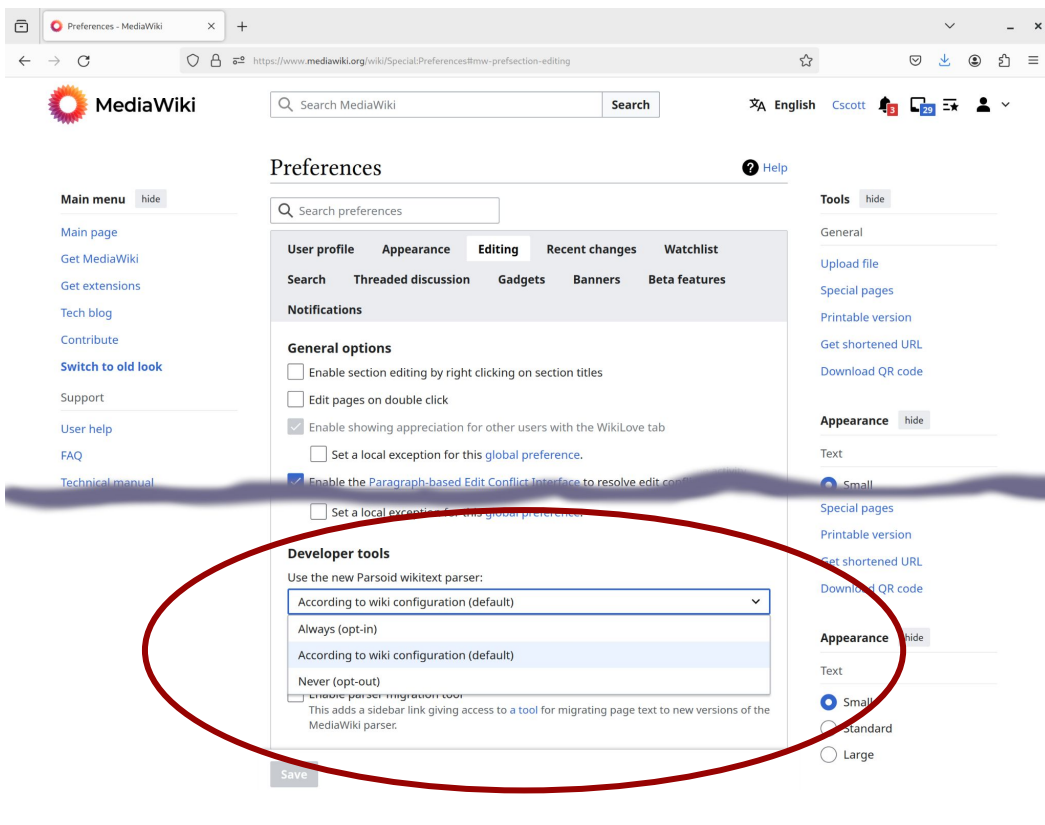
- Maintaining multiple wikitext parsers is not sustainable!
- The *Parser Unification project* aims to gradually replace the legacy wikitext parser entirely with Parsoid
- In 2024 we began to roll out Parsoid for *read views* on Wikimedia wikis

You can join us on this transition!

[[[Extension:ParserMigration](#)]] can be installed on any MW 1.42 wiki and allows site-wide as well as per-user opt-in to use the new Parsoid wikitext parser.



Parsoid opt-in using [\[\[Extension:ParserMigration\]\]](#)



The screenshot shows the MediaWiki Preferences page, specifically the 'Developer tools' section. A red oval highlights the dropdown menu for the 'Use the new Parsoid wikitext parser:' option. The dropdown menu is open, showing four options: 'According to wiki configuration (default)', 'Always (opt-in)', 'According to wiki configuration (default)', and 'Never (opt-out)'. The 'Always (opt-in)' option is selected. Below the dropdown, there is a checkbox for 'Enable parser migration tool' which is checked. The page also shows other preference sections like 'General options' and 'Appearance'.



2.

What do

[[Square brackets]]

have to do with it?



The tale of HTML5 and Tidy

2004: The HTML4-based Tidy library was introduced to MediaWiki and gradually became an integral part of how wikitext was rendered.

2015: It became obvious continued use of Tidy was unsustainable, and work began to shift wikitext rendering to a web-standard HTML5 tree building algorithm (part of RemexHtml).

2018: Transition finally completed, three years later, after fixing ~7% of all pages.

Three keys to success:

1. Tools: ParserMigration and Linter
2. Community engagement
3. Phased deployment

Result: unblocked new innovation (starting with Extension:TemplateStyles)

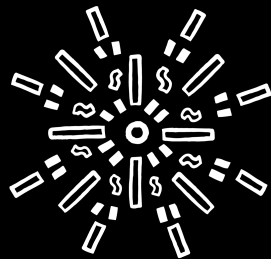
* First step toward Parser Unification *



More: <https://diff.wikimedia.org/2018/07/09/tidy-html5-replacement/>



**The goal of content
migration is to enable
wikitext improvements**



Semantic MediaWiki and `[[syntax]]`

Things which use `[[brace syntax]]` in wikitext:

- `[[Wiki links]]` (2001)
- `[[m:Interwiki links]]` (2002)
- `[[Category: ...]]` (invisible, problematic)
- `[[es:Enlace]]` (invisible, discouraged)
- `[[Property::Value]]` (sometimes visible?)
(Semantic MediaWiki, 2007)

SMW uses `InternalParseBeforeLinks` as a hook to replace `[[...]]` syntax “before” the MW legacy parser sees it; Parsoid does not have parser stages like this and there was never an “official” hook point for `[[...]]` syntax.

Alternatives using standard `{{#...}}` syntax:


- `{{#set:property1=value1|...}}`
 - `Additional template= parameter`
useful to control display
- `{{#declare}}` in templates
- `{{#ask}}` / `{{#show}}` to further control display

Perhaps additional parser functions or helpers can be added; perhaps additional performance work is needed. But let’s go with this for now...

More: <https://phabricator.wikimedia.org/T76278>,
<https://github.com/SemanticMediaWiki/SemanticMediaWiki/issues/4727>,
<https://github.com/SemanticMediaWiki/SemanticMediaWiki/issues/584#issuecomment-60246958>



Tough crowd...

 GRC ATF  Risk Management at GRC-ATF using Mediawiki+
The Purpose Of This Talk

Bottom Line Up-Front Mediawiki has what it takes to do Risk Management for NASA

The purpose of this talk is **NOT**
to explain NASA's Risk Management Process
so that you will fully understand it

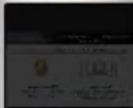
```
[[[#ask: [[Risk Management::MediaWiki]] ]]]
```

The purpose of this talk **IS**
to show you that **MediaWiki Has What It Takes** meet
the software requirements of one of NASA's most
essential processes (Risk Management)

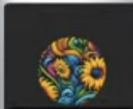
Presenter's screen



Presenter



Presenter'...



#MWCOn ...

Assuming we agree we want to get *there*...

**How do we get there
from here?**



WIKIMEDIA
FOUNDATION

How do we get there from here?

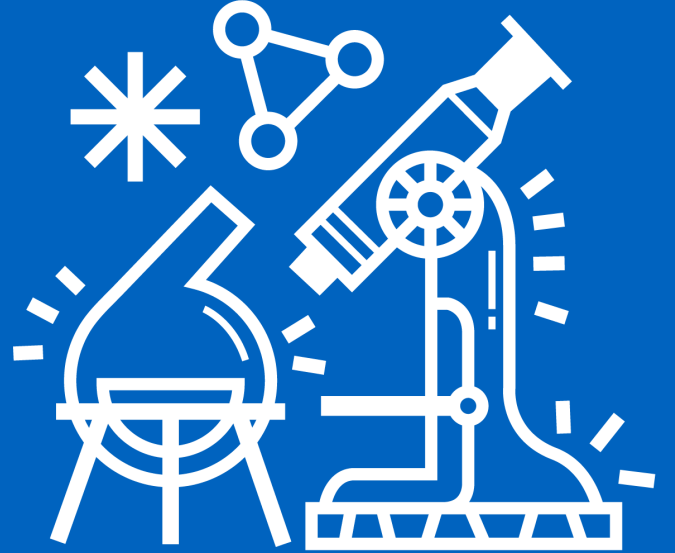
We'll use the conversion of
[[square bracket]]
syntax as a running
example in the
remainder...

...but it's only an example!
Your problems may be
different, but similar tools
can be used.



3.

A survey of tools



Tracking Categories

The original!

Example, in SMW:src/InternalParseBeforeLinks.php:

```
private function canPerformUpdate( $text, $title ) {  
    ...  
    if ( InTextAnnotationParser::hasMarker( $text ) ||  
        InTextAnnotationParser::hasPropertyLink( $text ) ) {  
        $this->parser->addTrackingCategory( "smw-inline-property" );  
        return true;  
    }  
}
```



Tracking Categories, cont.




Editors and linters work from [\[\[Special:TrackingCategories\]\]](#)

Tracking categories

 [Help](#)

[Together](#) Tools 

This page lists tracking categories which are automatically populated by the MediaWiki software. Their names can be changed by altering the relevant system messages in the MediaWiki namespace. See [Category:Tracking categories](#) for tracking categories made by the English Wikipedia, usually added by templates.

Tracking category 	Message name 	Category inclusion criteria 
Articles with missing files (4 P), Pages with missing files (3 C, 90,506 P, 163 F), Templates with missing files (empty)	broken-file-category	The page contains a broken file link (a link to embed a file when the file does not exist).
Articles using duplicate arguments in template calls (14 P), Pages using duplicate arguments in template calls (1 C, 656 P)	duplicate-args-category	The page contains template calls that use duplicates of arguments, such as <code>{{foo bar=1 bar=2}}</code> or <code>{{foo bar 1=baz}}</code> .



Linters Extension

- Created as part of the HTML5/Tidy migration project.
- Creation of a lint can be more complicated, but can track template-generated issues & provide precise location information.

```
private function lintNightModeUnawareBackgroundColor(
    Env $env, Element $node, DataParsoid $dp, ?stdClass $tplInfo
): void {
    // Get inline style attribute value
    $styleAttrValue = DOMCompat::getAttribute( $node, 'style' );

    // Check if background color is set but font color is not
    if (
        ( $styleAttrValue !== null ) &&
        preg_match( '/(^|;)\s*background(-color)?\s*:/i', $styleAttrValue ) &&
        !preg_match( '/(^|;)\s*color\s*:/i', $styleAttrValue )
    ) {
        $tplLintInfo = $this->findEnclosingTemplateName( $env, $tplInfo );
        $lintObj = [
            'dsr' => $this->findLintDSR(
                $tplLintInfo, $tplInfo, DOMDataUtils::getDataParsoid( $node )->dsrc ?? null
            ),
            'templateInfo' => $tplLintInfo,
        ];
        $env->recordLint( 'night-mode-unaware-background-color', $lintObj );
    }
}
```



Linters Extension, cont.

Editors and linters work from [\[\[Special:LintErrors\]\]](#)

Lint errors

 [Help](#)

Tools ▾

This special page displays [lint errors](#). Discuss this special page at [Wikipedia talk:Linter](#). The counts for categories are estimated, not exact. The [firefly linter count](#) shows a condensed, more exact, count of each error in each namespace.

High priority

- [Table tag that should be deleted](#) (1,312 errors)
- [Misnested tag with different rendering in HTML5 and HTML4](#) (17,410 errors)
- [Miscellaneous issues](#) (0 errors)
- [Multiline table in list](#) (1 error)
- [Multiple unclosed formatting tags](#) (0 errors)
- [Paragraph wrapping bug workaround](#) (0 errors)
- [Self-closed tags](#) (0 errors)
- [Old behaviour of link-wrapping font tags](#) (0 errors)



Lint Extension, cont.

Lint errors: Bogus file options

[? Help](#)

Tools ▾

[← Special:LintErrors](#)

These pages have files with bogus options.

Namespace:

Please enter a title prefix or complete title:

Search for a specific page title, or all pages with a common title prefix

Search for a specific page title

Search for all pages that match the title prefix

Filter by tag name

Filter by template state

[|< First page](#) [< Previous page](#) [Next page >](#) [Last page >|](#)

Page title	Bogus file option	Through a template?
User:Roke/links (edit history)	4	Template:Superimpose
User:Anichiniluxury/sandbox (edit history)	Anichini	Template:Infobox company
User:Bartonjr/sandbox (edit history)	frameless	Template:Infobox company
User:Underneath it All/Sandbox (edit history)		



Linters Extension, cont.

```
https://en.wikipedia.org/w/index.php?title=User:Roke/links&action=edit&lintid=3337173

</pre>
</div>

<div style="width: 70%;">
{| align=right
|-
|
|
|{{Superimpose| base = Australia_Locator_Blank.svg
| base_width = 250px
| base_caption = Australia
| float = Locator_Dot.svg
| float_width = 4
| float_caption = Add caption here
| x = 200
| y = 55
|}}
Add caption here
|}
</div>

{{clear}}

: '' See also [[Wikipedia:Shortcuts]] for a huge list of Wikipedia pages, and
[[[:Category:Wikipedia guidelines]] and [[[:Category:Wikipedia style guidelines]]''
{| border="2" colspan="2"
|-
| bgcolor="#c0c0c0" colspan="10" |Community
|-
```



Linters Extension, cont.

Community participation is vital!

Volunteers have created innumerable help pages, guides, FAQs, explanations of how to fix specific issues, etc.

A good entry point is [\[\[en:Wikipedia:Linters\]\]](https://en.wikipedia.org/wiki/Linters)



Here be dragons



We do not generally use automated edits for linting on Wikimedia projects.

As a result, these next tools—which could be very appropriate for 3rd party wikis—have not been developed to the same extent for content migration.

Perhaps tomorrow in the Create Camp we can begin to change that?

dawizard damage



WIKIMEDIA
FOUNDATION

Bots & Frameworks

- Semi-automated edits
 - [AutoWikiBrowser](#): designed to make tedious or repetitive editing tasks quicker and easier
- Automated edits ([\[\[m:Help:Bots\]\]](#))
 - [Pywikibot](#): the original (2003) with a large collection of existing scripts
 - [Mwbot](#): The new upstart: rust-y and (optionally) Parsoid-powered
- Admin tools / extensions
 - [\[\[Extension:MassEditRegex\]\]](#): perform mass edits via a special page
 - [\[\[Extension:Replace Text\]\]](#): similar, plus a command-line script



Bots & Frameworks: AutoWikiBrowser

.NET Desktop application with the following features:

- Regular-expression search-and-replace
- Calling out to custom C# or VB.NET code to process page text
- Semi-offline modes using database dumps



Bots & Frameworks: AutoWikiBrowser

The screenshot displays the AutoWikiBrowser interface, which is used for editing Wikipedia pages. It features a central comparison window with two columns: 'Current revision' on the left and 'Your text' on the right. The text in both columns is identical, showing a paragraph about plain text and rich text. The interface includes a menu bar at the top with options like 'File', 'View', 'List', 'Plugins', 'Options', 'Tools', and 'Help'. Below the menu bar, there are instructions: 'Double click on a line to undo all changes on that line, or single click to focus the edit box to that line.' The main editing area is divided into two panes, each showing a line of text with a corresponding image thumbnail. The bottom of the interface features a 'Make list' panel on the left, a 'Page stats' panel, and a 'Text editor' panel. The 'Text editor' panel includes a list of editors, a 'Save' button, and a 'Find' search box. The bottom status bar shows 'Pages/min: 0 Edits/min: 0 Edits: 0 Skipped: 0 New: 0 en.wikipedia' and the system clock '1:39 PM 1/29/2021'.

AutoWikiBrowser - Default.xml - Text editor

File View List Plugins Options Tools Help

Double click on a line to undo all changes on that line, or single click to focus the edit box to that line.

Current revision

Your text

{{short description|Computer software used to edit plain text documents}}

{{selfref|For the use of text editors with Wikipedia, see [[Wikipedia:Text editor support]].}}

− {{refimprove|date=October 2014}}

[[File:Leafpad-screenshot.png|right|thumb|Editors like [[Leafpad]], shown here, are often included with operating systems as a default [[helper application]] for opening text files.]]

Line 9:

{{main|Plain text|Rich text}}

− There are important differences between plain text (created and edited by text editors) and [[rich text]] (such as that created by [[word processor]]s or [[desktop publishing software]]).

Plain text exclusively consists of character representation. Each character is represented by a fixed-length sequence of one, two, or four bytes, or as a variable-length sequence of one to four bytes, in accordance to specific [[character encoding]] conventions, such as [[ASCII]], [[ISO/IEC 2022:ISO/IEC 2022]], [[UTF-8]], or [[Unicode]]. These conventions define many printable characters, but also [[whitespace character|non-printing characters]] that control the flow of the text, such as [[Space (punctuation)|space]], [[newline|line break]], and [[page break]]. Plain text contains no other information about the text itself, not even the character encoding convention employed. Plain text is stored in [[text file]], although text files do not exclusively store plain text. In the early days of computers, plain text was displayed using a [[monospace font]], such that horizontal alignment and columnar formatting were sometimes done using whitespace characters. For compatibility reasons, this tradition has not changed.

Line 38:

[[File:Colortax.png|thumb|right|Emacs, a text editor popular among programmers, running on [[Microsoft Windows]]]]

[[File:Gedit 3.32 screenshot.png|thumb|[[gedit]] is a text editor shipped with [[GNOME]]]]

Some text editors are small and simple, while others offer broad and complex functions. For example, [[Unix]] and Unix-like

==Plain text vs. rich text==

{{main|Plain text|Rich text}}

There are important differences between plain text (as created and edited by text editors) and [[rich text]] (such as that created by [[word processor]]s or [[desktop publishing software]]).

Plain text exclusively consists of character representation. Each character is represented by a fixed-length sequence of one, two, or four bytes, or as a variable-length sequence of one to four bytes, in accordance to specific [[character encoding]] conventions, such as [[ASCII]], [[ISO/IEC 2022:ISO/IEC 2022]], [[UTF-8]], or [[Unicode]]. These conventions define many printable characters, but also [[whitespace character|non-printing characters]] that control the flow of the text, such as [[Space (punctuation)|space]], [[newline|line break]], and [[page break]]. Plain text contains no other information about the text itself, not even the character encoding convention employed. Plain text is stored in [[text file]], although text files do not exclusively store plain text. In the early days of computers, plain text was displayed using a [[monospace font]], such that horizontal alignment and columnar formatting were sometimes done using whitespace characters. For compatibility reasons, this tradition has not changed.

Rich text, on the other hand, may contain metadata, character formatting data (e.g. [[Font#Characteristics|typeface, size, weight and style]]), paragraph formatting data (e.g. indentation, alignment, letter and word distribution, and space between words), and other information that is not present in plain text. Rich text can be

Pages/min: 0 Edits/min: 0 Edits: 0 Skipped: 0 New: 0 en.wikipedia

1:39 PM 1/29/2021



Bots & Frameworks: pywikibot

The good: pretty easy to write new bots! Large collection of working examples!

[add_text.py](#)

```
156     def treat_page(self) -> None:
157         """Add text to the page."""
158         text = self.current_page.text
159
160         if self.opt.up:
161             text = self.opt.text + '\n' + text
162         elif not self.opt.reorder:
163             text += '\n' + self.opt.text
164         else:
165             text = textlib.add_text(text, self.opt.text,
166                                     site=self.current_page.site)
167
168         self.put_current(text, summary=self.opt.summary, minor=self.opt.minor)
```



Bots & Frameworks: pywikibot

The bad: Sooo many regular expressions...

[textlib.py](#)

```
276 # also finds links to foreign sites with preleading ":"
277 'interwiki': (
278     r'\[[[:?(\%s)\s?:[^\]]*\]\]\s*',
279     lambda site: '|'.join(
280         ignore_case(i) for i in site.validLanguageLinks()
281         + list(site.family.obsolete.keys()))),
282 # Module invocations (currently only Lua)
283 'invoke': (
284     r'\{\{\s*\#(?:%s):[\s\S]*?\}\}',
285     lambda site: '|'.join(
286         ignore_case(mw) for mw in site.getmagicwords('invoke'))),
287 # this matches internal wikilinks, but also interwiki, categories, and
288 # images.
289 'link': re.compile(r'\[[^\]]*\|([^\]]*)?\]\]\s*'),
290 # pagelist tag (used in Proofread extension).
291 'pagelist': re.compile(r'<{\s\S}*?/>'
292     .format(ignore_case('pagelist'))),
293 # Wikibase property inclusions
294 'property': (
295     r'\{\{\s*\#(?:%s):\s*[Pp]\d+.*?\}\}',
296     lambda site: '|'.join(
297         ignore_case(mw) for mw in site.getmagicwords('property'))),
```



Bots & Frameworks: mwbot

`mwbot` aims to provide a batteries-included framework for building bots and tools for MediaWiki wikis. It builds on top of the [mwapi](#) and [parsoid](#) crates, which offer lower-level APIs.

Reading a page

```
let bot = mwbot::Bot::from_default_config().await.unwrap();
let page = bot.page("Rust (programming language)")?;
let html = page.html().await?.into_mutable();
// The lead section is the second p tag in the first section
let lead = html.select("section > p")[1].text_contents();
assert!(lead.starts_with("Rust is a multi-paradigm, general-purpose programming language"));
```



Bots & Frameworks: mwbot

`mwbot` aims to provide a batteries-included framework for building bots and tools for MediaWiki wikis. It builds on top of the [mwapi](#) and [parsoid](#) crates, which offer lower-level APIs.

Editing a page

```
let bot = mwbot::Bot::from_default_config().await.unwrap();
let page = bot.page("Project:Sandbox"?);
let wikitext = "This is a test edit!";
page.save(wikitext, &SaveOptions::summary("test edit!")).await?;
```

`Page.save()` accepts both HTML and wikitext and supports the `{{nobots}}` exclusion mechanism, among other features.



[[[Extension:ReplaceText](#)]]

- Provides a special page, as well as a command-line script, to allow administrators to do a server-side global string find-and-replace.
- Wikitext based
- Relies on SQL query support
 - Does not work if [\\$wgCompressRevisions](#) is true (set in WMF prod)
 - No regexp support if your database doesn't support it (aka not SQLite)
- Included in many wiki farms/hosts and/or packages: [BlueSpice](#), [Canasta](#), [Debian](#), [MyWikis](#), [Open CSP](#), [ProWiki](#), [semantic::core](#), [wiki.gg](#), [WikiForge](#)



[[Extension:ReplaceText]]

Replace text

To replace one text string with another across all regular pages on this wiki, enter the two pieces of text here and then hit "Continue". You will then be shown a list of pages that contain the search text, and you can choose the ones in which you want to replace it. Your name will appear in page histories as the user responsible for any changes.

Original text:	<input type="text" value="\[\[([^\:]+)::([^\:]+)\]\]"/>
Replacement text:	<input type="text" value="{{#set:\$1=\$2}}"/>

Use regular expressions

(Example: values of "a.()c" for "Original text:" and "ac\$1" for "Replacement text:" would replace "abc" with "acb".)*

Tools hide

General

[Upload file](#)



Search in namespaces: Check:

<input checked="" type="checkbox"/> (Main)	<input checked="" type="checkbox"/> Talk	<input checked="" type="checkbox"/> File	<input checked="" type="checkbox"/> File talk
--	--	--	---

[[Extension:ReplaceText]]

User:Cscott/SMWTest: Revision history

 [Help](#)

[User page](#) [Discussion](#)

[Read](#) [Edit source](#) [View history](#) 

[View logs for this page](#)


nd

Filter revisions

Diff selection: Mark the radio buttons of the revisions to compare and hit enter or the button at the bottom.

Legend: **(cur)** = difference with latest revision, **(prev)** = difference with preceding revision, **m** = minor edit.

Compare selected revisions

- [\(cur | prev\)](#) 17:02, 16 April 2024 Cscott (talk | contribs | block) **m** .. (23 bytes) (+4) ..
(Text replacement - "[\[\[^\]]+\]:([\^\]]+)\]\]" to "{{#set:\$1=\$2}}") (undo)
- [\(cur | prev\)](#) 17:00, 16 April 2024 Cscott (talk | contribs | block) .. (19 bytes) (+19) ..
(Add a test of the ReplaceText extension)



[[Extension:ReplaceText]]

Also works on the command-line (although quoting regexps can be tricky):

```
cananian@skiffserv:~/Wikimedia/core$ php maintenance/run.php ./extensions/ReplaceText/maintenance/replaceAll.php --nsall --debug --regex '\[\[([^\:]+)\]\]\]\]\]' '#set:$1=$2'
```

Replacing '\[\[([^\:]+)\]\]\]\]' with '#set:\$1=\$2' as regular expression.

User:Cscott/SMWTest

Help:FAQ/de

Replace instances on these pages? (Y/N) Y

Attribute changes to the user 'CScott'? (Use --user to override) (Y/N) Y

Replacing on User:Cscott/SMWTest... done.

Replacing on Help:FAQ/de... done.

```
cananian@skiffserv:~/Wikimedia/core$
```



[[Extension:MassEditRegex]]

- Provides a special page (no command-line script) to allow administrators to do a server-side global string find-and-replace
- Wikitext based
- Does not require SQL support for regexps; works with compressed revisions



[[Extension:MassEditRegex]]

Mass edit using regular expressions

Tools ▾

Enter one or more regular expressions (one per line) for matching, and one or more expressions to replace each match with. The first match-expression, if successful, will be replaced with the first replace-expression, and so on. See [the PHP function preg_replace\(\)](#) for details.

Pages to edit:

User:

This is a list of:

- Page names (edit these pages)
- Page name prefixes (edit pages having names beginning with this text)
- Category names (edit each page within these categories)
- Backlinks (edit pages that link to these ones)

Search for:

Replace with:

```
/\[\[([^\]]+)\]::([^\]]+)\]\]\]/
```

```
{{#set:$1=$2}}
```

Summary:

Preview of edit summary: *(Updating square-bracket syntax)*

Execute in browser



[[Extension:MassEditRegex]]

Execute

Show preview

Execute in browser

User:CScott - Before

After

(No difference)

User:Cscott - Before

After

(No difference)

User:Cscott/SMWTest - Before

After

Line 1:

– `[[Property::Value]]`

Line 1:

+ `{{#set:Property=Value}}`

User:Cscott/T236867 - Before

After

(No difference)



ReplaceText and MassEditRegex

- How can we communicate better about wikitext changes that might require migration?
- Can we make our substitutions token- or DOM-based to make recognition more robust?
- Can we integrate substitutions as a “database migration” on upgrade?
- Is there a middle ground between “all manual” and “completely automatic” migrations?



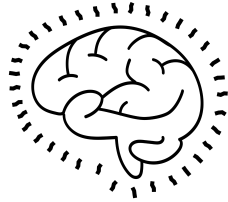
4.

What did I miss?

**What can we
imagine?**



Content Migration is a fact of life; let's try to make it boring and routine.



WIKIMEDIA
FOUNDATION



Contact me: C. Scott Ananian
cananian@wikimedia.org