



How to review the MediaWiki Math extension

Prerequisites

In the Math extension a mix of PHP and JavaScript is used. In addition to that texvc (the component that sanitizes the user input TeX strings and converts Mediawiki specific TeX commands to standard TeX/LaTeX) is written in OCAML.

However, this feature was rewritten in PHP and will be replaced once we manage to conquer the current review backlog.

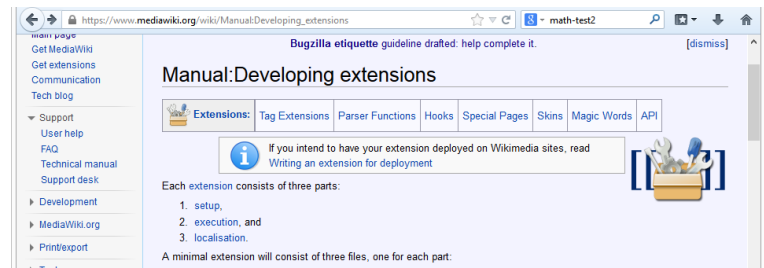


Figure 1 Figure 1 Manual about developing extensions

If you want to contribute you should have good knowledge about either PHP or JavaScript or both. In addition the more background knowledge you have about MediaWiki extension interface the better. At least you should read the one page manual about Extension Development.

Development environment

There are various ways to set up an environment for testing and development. The simplest solution is to use the virtual machine based vagrant environment which is described in detail at the Mediawiki-Vagrant manual page. Simply follow the Quick-Start guide

1. Get Git
2. Get VirtualBox
3. Get Vagrant (version 1.2.2 or higher is required!)
4. Get the code and create your machine:

```
git clone https://gerrit.wikimedia.org/r/mediawiki/vagrant
cd vagrant
vagrant enable-role math
vagrant up
```

This process usually takes up to 20 minutes with modern hardware and a broadband connection.

Now you can log into the system via the `vagrant ssh` command. If you terminate your session e.g. with `exit` or `STRG+D` you can stop your via `vagrant halt` command.

If vagrant is running you can navigate with your browser to

<http://localhost:8080>

in order to log into the testing environment with the user *vagrant* and the password *vagrant*.

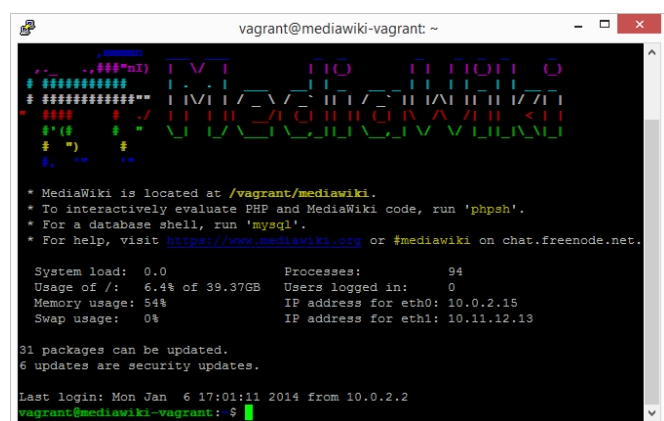


Figure 2 Vagrant development environment



Starting the review process

At first you have to create a development account [at the wikitech page](#).

In a next step you have to import your public key in [your account settings](#). Log into vagrant via vagrant ssh and follow the steps “Add SSH Public key” as described in the account settings page.

1. From the Terminal or Git Bash, run `ssh-keygen`
2. Confirm the default path `.ssh/id_rsa`
3. Enter a passphrase or leave it blank. Remember this passphrase, as you will need it to unlock the key whenever you use it.
4. Open `~/.ssh/id_rsa.pub` and copy & paste the contents into the box below, then click on "Add". Note that `id_rsa.pub` is your public key and can be shared, while `id_rsa` is your private key and should be kept secret.

Now you can link this account to your code. In the vagrant vm navigate to `/vagrant/mediawiki/extensions/Math` and run `git review -s` and enter your new development account user name when prompted.

Now go navigate to the [gerrit web interface](#) with your host browser.

Select the change you want to review and copy the Change-ID from the last row of the commit message e.g.

I8e582ee6deadc2750d30b

49f0c6de31ed9c16a50. To

checkout a revision that

contains the proposed

change go to your vagrant

shell and navigate again to

`/vagrant/mediawiki/exten`

`sions/Math` in that folder

run `git checkout -d $CHANGEID` e.g. `git checkout -d I8e582ee6deadc2750d30b49f0c6de31ed9c16a50`.

Subject	Status	Owner	Project	Branch	Updated	CR	V
Open Changes							
Add VisualEditor support		Catrope	mediawiki/extensions/Math	master (visualeditor)	9:51 AM	✓	
Validate TeX input for all renderers, not just texvc		Physikerwelt	mediawiki/extensions/Math	master (bug/49169)	Jan 4	✓	
Remove call to deprecated ParserOptions::getMath		Physikerwelt	mediawiki/extensions/Math	master (realMaster)	Jan 4	✓	
Merged Changes							
new option to skip xml validation: setXMLValidation	Merged	Physikerwelt	mediawiki/extensions/Math	dev (setXMLValidation)	Jan 6	✓	✓
Fix merge problems	Merged	Physikerwelt	mediawiki/extensions/Math	dev (dev)	Jan 5	✓	✓
allow to maintain a development branch without to review all the translation ...	Merged	Physikerwelt	mediawiki/extensions/Math	refs/meta/config	Jan 5	✓	✓

Figure 3 The Gerrit web interface

Now you can test the new functionality in your web-browser at <http://localhost:8080>. A good starting point for a set of test formulas can be found at the Wikipedia help page for [displaying a formula](#). You can copy and paste some of the equations there to your local testing environment.

If you found an error during this process you have two options:

- a) You can either post a negative (-1) comment describing how to reproduce the problem and what went wrong in the Gerrit web interface.
- b) Or you fix the change by modifying the source if it's an obvious problem. After the correction you have to submit a new change set (see below).



How to review the MediaWiki Math extension

Submitting a new change set

If you decided to correct the error you can submit a new change set by the following procedure.

1. Navigate to `/vagrant/mediawiki/extensions/Math`
2. Run `git status` and check if all the changes you made are intended.
3. Run the automated tests via `php /vagrant/mediawiki/tests/phpunit/phpunit.php ./tests/`
4. Run `git commit -a -amend`
5. If you change something fundamental update the commit message (but never modify the change-Id)
6. You will be asked to specify your name and email address which will be associated with the commit and stored in the git repository. Please be sure to protect your privacy and be aware that you cannot modify this information after you run the review command.
7. Run `git review -R` (-R means don't rebase)
8. You will be notified by eMail that a new change set was uploaded.

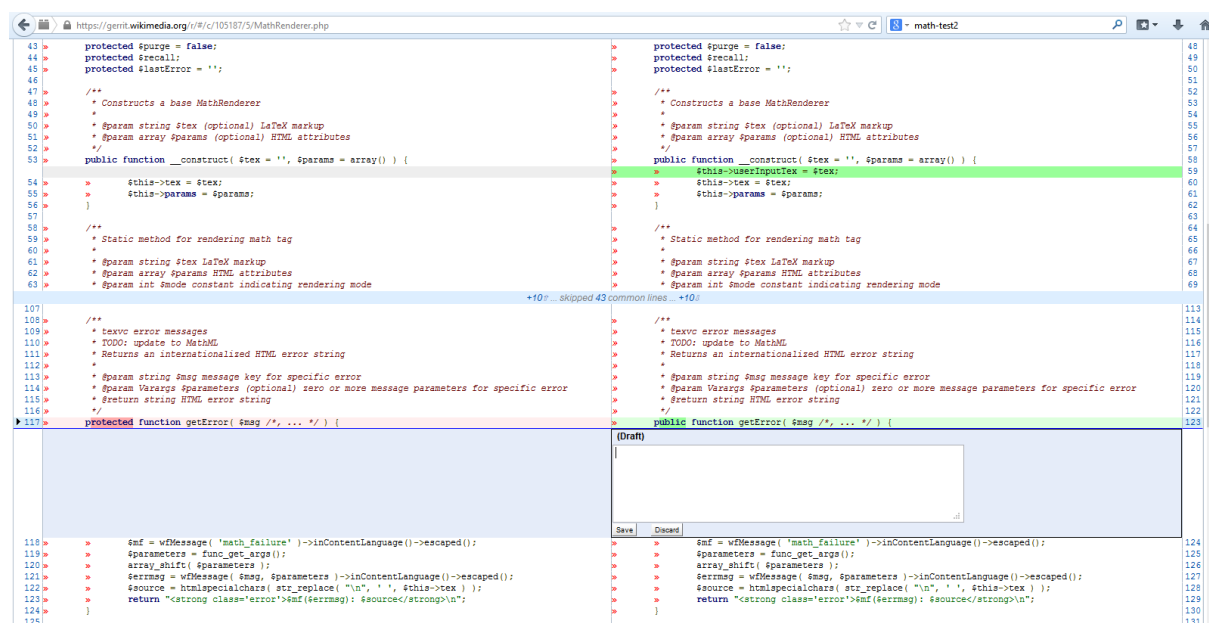


Figure 4 Reviewing in detail. You can post comment to each line individually or leave overall comments in the cover message



Code review in a narrower sense

After you have verified the function correctness it's time for the code review in a narrower sense. Here you review the sources using the Gerrit web UI¹. For that there is a [very good guide](#). Essentially a (+2) review means²:

- *I've inspected this code, and*
- *the code makes sense, and*
- *the code works and does something that we want to do, and*
- *the code doesn't do anything that we don't want it to do, and*
- *the code follows our development guidelines, and*
- *the code will still make sense in five years.*

There are a lot of rules and special conventions which looks very complicated at a first glance. However, using your common sense helps to follow most of the rules automatically. Furthermore, in some cases it makes sense to deviate from the style guide, i.e. if it helps to clarify the function of the code.

If you created a new account you don't have +2 rights, but you can add +1 what basically means "Looks good to me, but someone else must approve". On the other side you don't need to verify all the requirements for +2. So indicate in your commit message what you have done.

+2 rights for the math extension are granted to these accounts:

- thedj
- physikerwelt
- everybody with global +2 rights

In addition to that the following users contributed recently to the Math extension:

- Jiabao
- Frédéric Wang

Questions and answers

Often it's helpful to communicate interactively with other developers. The main communication channel is the irc channel #mediawiki. In addition to that there are various mailing lists and a bugzilla instance. An overview about all the opportunities can be found at the mediawiki page about [communication](#). For math specific questions there is also a channel #mathjax on freenode and a #mathml channel on mozilla.

For questions concerning this document just write me an [email](#).

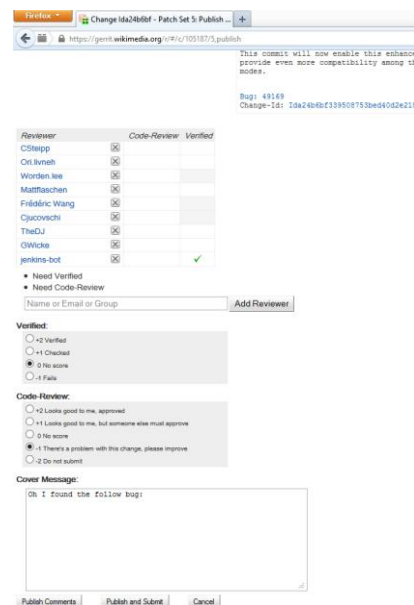


Figure 5 Reviewing a change

¹ It should also be possible with eclipse but I never managed to set that up.

² https://www.mediawiki.org/wiki/Gerrit/Code_review#Review_guidelines



About the author

Moritz Schubotz alias Physikerwelt is research associate in the Database Systems and Information Management Group in the Department of Software Engineering and Theoretical Computer Science at Technische Universität Berlin. Currently he manages the Database Lab course, supports the E-Learning initiative at TU Berlin and makes innovative big data analytics techniques available for small and medium enterprises in Europe in the scope of the DOPA project. He aims to improve the situation for scientific employees in the self-government at TU Berlin.

In his spare time he works on a PhD thesis about making mathematical content searchable. His vision is to find instantiations of mathematical concepts independent of the concrete representation in human readable documents. Beside that he is maintainer of the Math extension at Wikipedia. In that context the goal is to use HTML5 (i.e. MathML) to integrate mathematical notation to wikipages without to include ugly PNG images.

(Version history)

Version 0.1: 6.1.2014