

# Navigating Trac guide for new people

## core.trac.wordpress.org

As the WordPress Project grows, more developers are inspired to contribute. But how does this happen? The majority of developers use Github and issue pull requests and comments there. But WordPress uses Trac. So, let's get you started so you can create issues and add your voice.

**@floxos90 slides/presentation!**

**<https://slides.leaves-and-love.net/contributing-to-wordpress-core/#/>**

Scope of guide:

Answers the question: How do I use Trac after viewing their first ticket

To Explain to a new contributor how to navigate and use trac for contribution to core.

Assumptions made about the reader:

- 1) Already has a WP.org account
- 2) Already knows what Core team does
- 3) Wants to contribute and has at least found: <https://core.trac.wordpress.org/>
- 4) Already on Slack
- 5) Is a developer

What is Trac?

**Trac** is an [open source](#), Web-based [project management](#) and [bug tracking system](#). It is used by many parts of the WordPress project, including Core, Plugins and Themes.

You can read more about its history as a project here: <https://en.wikipedia.org/wiki/Trac>

Explain this screenshot, what does all this mean?? (random example ticket: <https://core.trac.wordpress.org/ticket/42860>)

The screenshot shows a web browser window displaying a ticket on the WordPress Core Trac website. The URL in the address bar is <https://core.trac.wordpress.org/ticket/42860>. The ticket is titled "#42860 new defect (bug)" and is marked as "Opened 22 hours ago" and "Last modified 10 hours ago". The description of the ticket is "PHP 7.2 warning - Parameter must be an array or an object that implements Countable in /wp-includes/class-wp-query.php on line 3035". The ticket is reported by "lisota" and owned by "lisota". The ticket has a milestone of "4.9.2", a priority of "normal", a severity of "normal", a version of "4.9.1", and a component of "Query". The keywords are "reporter-feedback". The description section contains the text "Upgrading to PHP 7.2 results in the following PHP warning from class-wp-query.php" and "PHP Warning: count(): Parameter must be an array or an object that implements Countable in /wp-includes/class-wp-query.php on line 3035". There is a "Reply" button next to the description. Below the description, there are sections for "Commits (0)", "Attachments (1)", and "Change History (7)". The "Attachments" section shows a patch file "42860-1.patch" (555 bytes) added by "ayeshrajans" 21 hours ago, with a link to the Travis CI build "https://travis-ci.org/Ayesh/wordpress-develop/builds/314423062". There is an "Attach file" button and a "Download all attachments as: .zip" link. The "Change History" section has a dropdown menu with options "Oldest first", "Newest first", "Show only comment text", and "Show only commits/attachments".

---

## Knowledge dump from Felix's Slides

### WordPress Core Trac

- Managing tickets
- Finding tickets
- Creating patches

<> Make WordPress Core

TicketsComponentsHandbook<>🕒📄

## Bug Reports

WordPress uses open source software called Trac to manage bug reports and tasks. If you are looking to report a bug, please [head on over](#).

There are many ways to contribute. We have a detailed [contributor handbook](#) to help get you started. If you want to get started quickly, try one of the [Getting Started](#) reports below — [test a patch](#), or [grab a bug](#) and see if you can reproduce the problem it describes.

If you have encountered a security issue in WordPress, please contact security at [wordpress.org](#). For more, see our [Security FAQ](#).

🕒 Latest Tickets

🔒 Active Tickets

🕒 Trac Timeline

<> Browse Source

👤 Milestones

Search ...

🏠 Recent Updates

💬 Recent Comments

💬 My @Mentions

🔒 No Replies

👤 My Posts

## Regular Chats

Note: All chats happen on [Slack](#).

Weekly Developer Meeting

[Mittwoch, 24. Mai 2017, 22:00 MESZ](#) in [#core](#)

[About the Dev Chat](#)

<https://make.wordpress.org/core/>

<> Make WordPress Core

BlogHandbookTicketsComponents<>🕒📄

New Ticket

Search

## WordPress Trac

Trac is the place to follow along with the development of WordPress. You can view recent code changes in the [Timeline section](#) of this site.

If you are looking to submit a bug report, please [head on over](#).

You will automatically receive notifications for tickets you have reported or participated in. If you would like to receive notifications for any other ticket, click the star next to its summary to "watch" it.

You can [browse the source of WordPress here](#), but you probably also want to [install a development version of WordPress locally](#).

Create a new ticket

## Security Issues

If you have encountered a security issue that isn't addressed in a released version of WordPress, please report it to the [WordPress HackerOne program](#). For more, see our [Security FAQ](#) in the handbook.

## Contributing

There are many ways to contribute to WordPress. We have a detailed [contributor handbook](#) to help get you started.

If you want to get started quickly, [test a patch](#), or [grab a bug](#) and see if you can reproduce the problem it describes. Don't miss the PHP and JS [unit tests](#) — and if you like writing unit tests, there are [some tickets](#) that could use them.

<https://core.trac.wordpress.org>

## Managing tickets

You can search for existing tickets [here](#). For more help, please see the [contributor handbook](#).

Summary:

Description: **B** *I* **A** [🔗](#) [🔗](#) [PHP](#) [—](#) [🔗](#) [🔗](#) [🔗](#) [🔗](#) You may use [WikiFormatting](#) here.

Type: **defect (bug)** [🔗](#)

Version: [🔗](#)  
*If you're filing a bug against trunk, choose 'trunk'. Otherwise, choose the earliest affected version you tested.*

Component: **General** [🔗](#)

Contributor Focuses: [ui](#) [accessibility](#) [javascript](#) [docs](#) [rtl](#)  
[admin](#) [template](#) [multisite](#) [rest-api](#)  
[performance](#)

☐ I have files to attach to this ticket

[Continue to Preview](#)

Summary:

- Brief summary of the ticket description.
- Best practice: If you're filing a bug, describe what's wrong.
  - → Example: "HTML comments in posts aren't handled correctly"
- Best practice: If you're filing an enhancement or feature request, use a call-to-action-type direction.
  - → Example: "Encourage people to change default tagline"

Description: **B** *I* **A**   PHP —    You may use [WikiFormatting](#) here.

- Ticket content: May be as long or as short as you like, but be precise.
- Best practice: Do not use empty phrases, be on point. Tone on Trac is rather formal, so try to avoid things like "I would love to see this!" or similar.
- Pro tip: You may reference another ticket by entering #{ticket\_id}, and it will automatically become a link.
- → Example: "This was previously discussed in #40000."
- Pro tip: You may reference a specific changeset by entering [{changeset\_id}], and it will automatically become a link.
- → Example: "The original bug was fixed in [40000]."

Type:  

- 
- Ticket type: Either defect (bug), enhancement or feature request.
- The vast majority of tickets are for bugs or enhancements.
- It's usually not a good idea to open a feature request ticket out of the blue. New feature ideas should be discussed in advance with others and only be transformed into a ticket once the idea has enough traction to be viable for a possible inclusion in WordPress core.
- → Furthermore, new features need to go through a special feature project workflow. See <https://make.wordpress.org/core/features/> and <https://make.wordpress.org/core/2016/03/31/iterating-on-feature-plugins/> for more information.
- There is a fourth ticket type called task (blessed). It is only available to component maintainers and committers and is intended to be used for task tickets, i.e. tickets that usually span multiple smaller sub-tickets to reach a bigger goal.
-

Version:

*If you're filing a bug against trunk, choose 'trunk'. Otherwise, choose the earliest affected version you tested.*

- This field should only be used when filing a bug.
- It should contain the number of the earliest version where the bug occurred. This is not necessarily the version of WordPress you are using.
- If you're not sure which version to enter, leave the field blank, so someone else can add the information later.
- Pro tip: If you already know where in the code the bug is coming from, you can use `svn blame {file}@r{changeset_id} | less` to work your way backwards and find out the earliest affected version.
- → Example: `svn blame src/wp-includes/post.php@r40000 | less`
- You can then use 400G to go to line 400 and see the changeset number when that line was last changed.

Component:

- The component to which the ticket belongs.
- If it affects multiple components, choose the one it affects the most. You may further specify the related area by using focuses.

Contributor

Focuses:

ui

accessibility

javascript

docs

rtl

admin

template

multisite

rest-api

performance

- A focus is something more general that usually spans across many, if not all WordPress core components.
- While specifying a component is required, focuses are optional.
- If you file a ticket relates to multiple components and one of them is also available as a focus, use a mixture of component and focuses.
- → Example: If you want to file a bug in the REST API taxonomy controller, use the Taxonomy component and the rest-api focus.

## The advanced New Ticket form

You can search for existing tickets [here](#). For more help, please see the [contributor handbook](#).

Summary:

Reporter:

Description: **B** *I* **A**   PHP    You may use [WikiFormatting](#) here.

Milestone:

Type:

Version:

Workflow Keywords:  manual

Priority:

Severity:

Component:

Focuses:

☐ I have files to attach to this ticket

Milestone:

- The milestone to which to assign the tickets. You may specify Awaiting Review, Future Release, the next major release or next minor release.
- When opening a ticket, you should usually keep this on Awaiting Review and ask for some initial feedback - especially if you're not filing a clear bug for which WordPress core is responsible.
- Minor releases should only be used to fix regressions (bugs) from the latest major release or to improve internal functionality with limited to no user facing effect. ~~This has changed a bit since the new release cycle, but it's still something to strive for.~~
- Best practice: Only assign tickets to the upcoming release if sufficient discussions have already happened within the ticket itself or during a Slack meeting and you consider it realistic for the changes to be done in time.
- Best practice: If you close a ticket without a successful resolution, you should remove the milestone from the ticket in case it's ever reopened in the future.

## Workflow Keywords:

- Add -



manual

- Specify one or more keywords to indicate the current status of the ticket and required steps.
- This is supposed to be dynamic throughout the ticket's lifecycle, so you can (and should) change it after opening the ticket as necessary.
- Available keywords are:
  - has-patch: A working patch has been uploaded.
  - needs-patch: The ticket still requires a working patch.
  - needs-refresh: There is a working patch, but the circumstances have changed, so a new patch is required.
  - reporter-feedback: Feedback by the person who opened the ticket requested.
  - dev-feedback: Feedback by developers requested, for example to review a patch.
  - 2nd-opinion: Feedback requested for concepts / ideas, to start a discussion.
  - close: The ticket should probably be closed, but another voice would be welcome.
  - needs-testing: The patch on the ticket needs to be thoroughly tested (usually used for new features).
  - ui-feedback: Feedback requested for the UI changes the ticket deals with.
  - ux-feedback: Feedback requested for the UX changes the ticket deals with.
  - has-unit-tests: Comprehensive unit tests have been uploaded for the ticket.
  - needs-unit-tests: The ticket still requires unit tests.
  - needs-docs: The patch on the ticket still needs proper documentation.
  - needs-codex: The changes from the ticket require changes in the WordPress Codex.
  - has-screenshots: Screenshots for UI issues / changes have been uploaded.
  - needs-screenshots: The ticket still requires screenshots for UI issues / changes.
  - commit: The ticket, patch and everything related has been properly reviewed and is ready for commit.
  - early: The changes proposed in the ticket should be committed early in a release cycle to allow for thorough testing.



- i18n-change: The changes in the ticket include internationalization i18n string changes.
- good-first-bug: The ticket qualifies as a bug that should be relatively easy to fix for new contributors. Experienced contributors should preferably keep off, unless it requires a fix urgently.
- fixed-major: The ticket has already been fixed in trunk, but requires back-porting to an older release.

Priority:


- The priority of the ticket. Can be normal, low, high, lowest or highest omg bbq. No, this is not a joke. 🔥
- Always keep this at normal, unless specific circumstances dictate the value to be different or the ticket changes are nit-picking and not really needed.
- Don't ever use this to mark something as important because you want it to happen!

Severity:


- The severity of the ticket, if it is a bug. Can be normal, major, minor, trivial, critical or blocker.
- Should usually remain normal as well. Critical can be used if the bug needs to be fixed as soon as possible, while blocker indicates that it should be fixed soon so that work on another related ticket can proceed.
- **Again, don't ever use this to mark something as important because you want it to happen! This will be changed appropriately after being reviewed by another person.**

## Ticket Statuses

When you open a new ticket, its initial status will always be set to new.

There are a few more statuses however:

- accepted: The changes proposed by the ticket are accepted for a future merge. Used rarely, as this is mostly implicit by the comments on the ticket.
- assigned: The ticket is assigned to an owner who will take care of the necessary work.
- closed: The ticket has been closed with a specific resolution, either as fixed or a negative resolution.

- reopened: The ticket had already been closed, but was reopened because it was not actually completed properly.
- reviewing: The ticket has an owner who will review the changes.

## Ticket Resolutions

- Resolution set to *fixed*
- Status changed from *assigned* to *closed*

In [40404](#):

Multisite: Introduce an `upgrade_network` capability.

Fixes [#39205](#). See [#37616](#).

**A ticket is automatically closed as fixed, when a commit message contains the phrase "Fixes #{ticket\_id}."**

● resolve as

The resolution will be set. Next status will be 'closed'.

If a ticket should not be fixed for a specific reason, it can be manually be closed with a proper resolution:

- invalid: The ticket was not opened for a valid reason, for example when a bug only occurred because the reporter or a plugin clearly did something wrong.
- wontfix: This may be considered a bug, but is not critical and cannot be fixed for (usually complex) legacy reasons.
- worksforme: The issue described in the ticket cannot be recreated.
- maybelater: This may be considered a bug, but is not critical and involves too heavy changes compared to the benefits. It may be reconsidered later.

### Finding tickets

There are currently thousands open tickets in WordPress Trac.

→ How do you find the ones that you wanna work on?

A good starting point: <https://make.wordpress.org/core/reports/>

Make WordPress Core

Tickets Components Handbook

## Bug Reports

WordPress uses open source software called Trac to manage bug reports and tasks. If you are looking to report a bug, please [head on over](#).

There are many ways to contribute. We have a detailed [contributor handbook](#) to help get you started. If you want to get started quickly, try one of the [Getting Started](#) reports below — [test a patch](#), or [grab a bug](#) and see if you can reproduce the problem it describes.

If you have encountered a security issue in WordPress, please contact security at [wordpress.org](#). For more, see our [Security FAQ](#).

Latest Tickets Active Tickets Trac Timeline Browse Source

Milestones

Search ...

Recent Updates

Recent Comments

My @Mentions

No Replies

My Posts

### Regular Chats

Note: All chats happen on [Slack](#).

Weekly Developer Meeting

[Mittwoch, 24. Mai 2017, 22:00 MESZ](#) in [#core](#)

[About the Dev Chat](#)

## Reports

Reports are lists of tickets to which certain predefined criteria apply. Examples:

- All open tickets that are scheduled for the next major release.
- All open tickets that deal with the Customizer.
- All open tickets that need unit tests.
- All open tickets that are good for new contributors.
- All open tickets that are ready for commit.
- All open bug tickets that have been reported against trunk.

→ Can you name the criteria (from our previous Managing tickets section) that a ticket needs to match in order to fall under each of the above reports respectively?

A good starting point: <https://make.wordpress.org/core/reports/>

Example: All open tickets that are scheduled for the next major release.

- Status: new, accepted, assigned, reopened or reviewing
- Milestone: 4.9

All open tickets that deal with the Customizer.

- Status: new, accepted, assigned, reopened or reviewing
- Component: Customize

All open tickets that need unit tests.

- Status: new, accepted, assigned, reopened or reviewing

- Keywords: must contain needs-unit-tests

All open tickets that are good for new contributors.

- Status: new, accepted, assigned, reopened or reviewing
- Keywords: must contain good-first-bug

All open tickets that are ready for commit.

- Status: new, accepted, assigned, reopened or reviewing
- Keywords: must contain commit

All open bug tickets that have been reported against trunk.

- Status: new, accepted, assigned, reopened or reviewing
- Type: defect (bug)
- Version: trunk

Status: new, accepted, assigned, reopened or reviewing Keywords: must contain good-first-bug

All open bug tickets that have been reported against trunk.

- Status: new, accepted, assigned, reopened or reviewing
- Type: defect (bug)
- Version: trunk

The full list of available reports can be found at <https://core.trac.wordpress.org/report>.

New Ticket

Search

Available Reports

[Return to Last Query](#)

[Custom Query](#)

☐ Show Descriptions

x Clear

SQL reports and saved custom queries

Sort by: [Identifier](#) ▲ [Title](#)

{1} All Tickets

{2} All Tickets, Workflow Oriented

{3} Next Minor Release

{4} Next Minor Release, Workflow Oriented

{5} Next Major Release

{6} Next Major Release, Workflow Oriented

{7} Future Releases

{8} Future Releases, Workflow Oriented

{9} Commit Candidates

{10} Needs Dev / Bug Wrangler Feedback

{11} Needs Reporter Feedback / Steps To Reproduce

{12} Needs Unit Tests

## Dynamic Reports

{30} Tickets with a focus on \$FOCUS

{31} Tickets in the \$COMPONENT component

Some reports require a dynamic parameter. Example usages:

- <https://core.trac.wordpress.org/report/30?FOCUS=rest-api>
- <https://core.trac.wordpress.org/report/31?COMPONENT=Taxonomy>

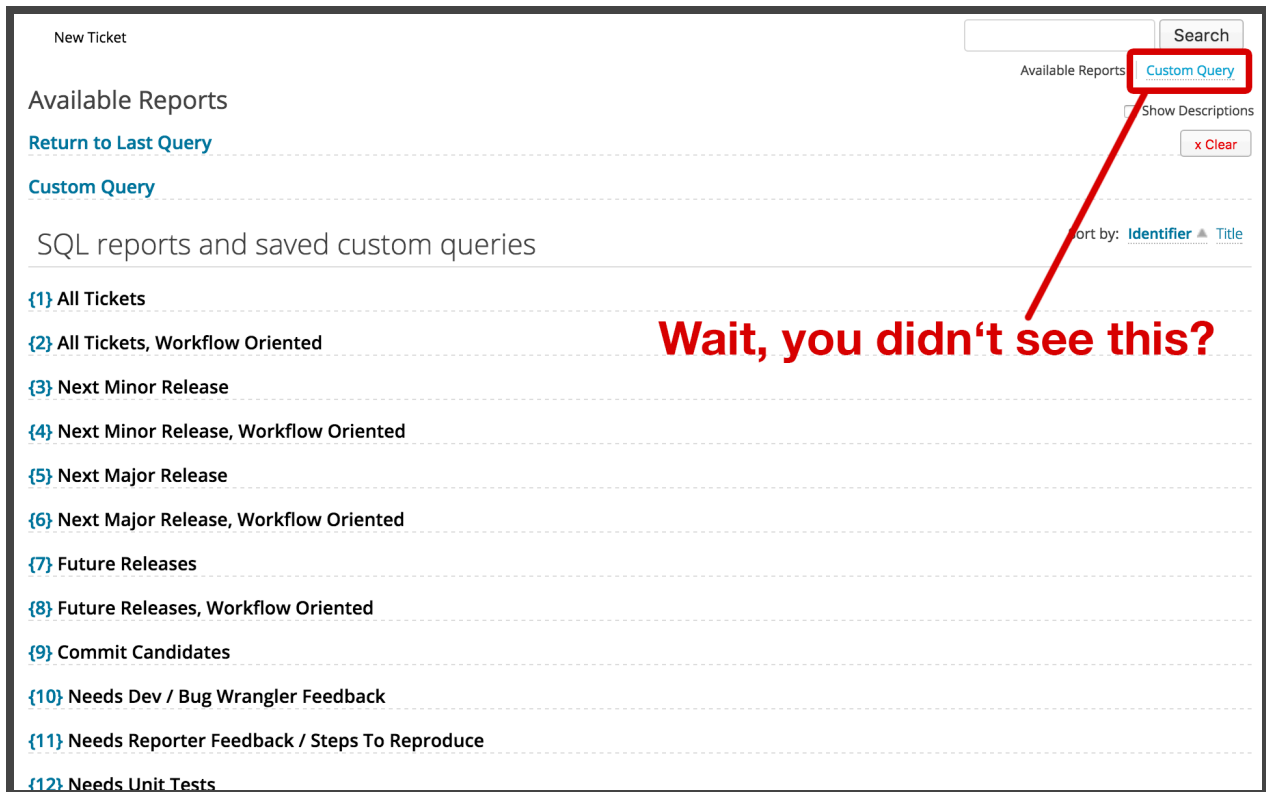
{62} Tickets I Own

Hidden feature: `?USER=$name` pulls up someone else's ticket list.

Some reports support an optional dynamic parameter. Example usages:

- <https://core.trac.wordpress.org/report/62>
- <https://core.trac.wordpress.org/report/62?USER=nacin>

Your most powerful tool: <https://core.trac.wordpress.org/query>



New Ticket

Search

Available Reports

[Return to Last Query](#)

[Custom Query](#)

SQL reports and saved custom queries

Sort by: [Identifier](#) [Title](#)

[{1} All Tickets](#)

[{2} All Tickets, Workflow Oriented](#)

[{3} Next Minor Release](#)

[{4} Next Minor Release, Workflow Oriented](#)

[{5} Next Major Release](#)

[{6} Next Major Release, Workflow Oriented](#)

[{7} Future Releases](#)

[{8} Future Releases, Workflow Oriented](#)

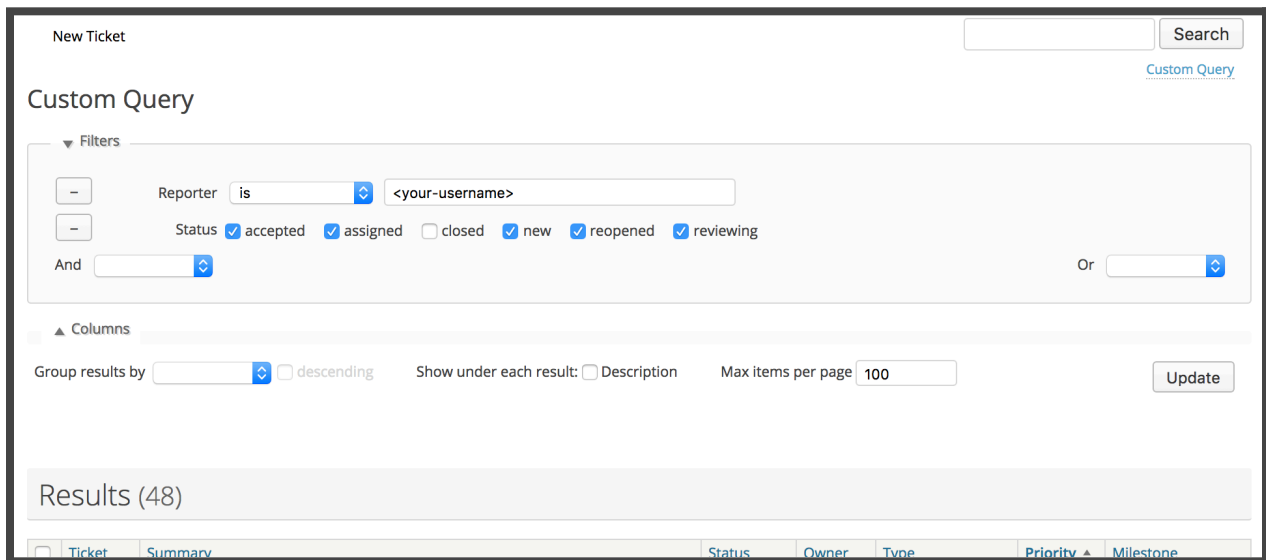
[{9} Commit Candidates](#)

[{10} Needs Dev / Bug Wrangler Feedback](#)

[{11} Needs Reporter Feedback / Steps To Reproduce](#)

[{12} Needs Unit Tests](#)

Wait, you didn't see this?



New Ticket

Search

[Custom Query](#)

Custom Query

Filters

Reporter is

Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And  Or

Columns

Group results by  ☐ descending Show under each result: ☐ Description Max items per page

Update

Results (48)

| <input type="checkbox"/> Ticket | Summary | Status | Owner | Type | Priority | Milestone |
|---------------------------------|---------|--------|-------|------|----------|-----------|
|---------------------------------|---------|--------|-------|------|----------|-----------|

## Query

This page allows you to query by any information available with granular control about relations, columns to show and orderby field.

Example: All open tickets that are scheduled for the next major release.

**Custom Query**

▼ Filters

Milestone

Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And

▲ Columns

Group results by  Show under each result: ☐ Description Max items per page

**Custom Query**

▼ Filters

Milestone

Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And

▲ Columns

Group results by  Show under each result: ☐ Description Max items per page

New Ticket

[Custom Query](#)

**Custom Query**

▼ Filters

Reporter

Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And

▲ Columns

Group results by  Show under each result: ☐ Description Max items per page

Results (48)

| <input type="checkbox"/> Ticket | Summary | Status | Owner | Type | Priority ▲ | Milestone |
|---------------------------------|---------|--------|-------|------|------------|-----------|
|---------------------------------|---------|--------|-------|------|------------|-----------|

Can you generate the following reports by using the custom query feature?

- All open tickets that deal with the Customizer.
- All open tickets that need unit tests.
- All open tickets that are good for new contributors.
- All open tickets that are ready for commit.
- All open bug tickets that have been reported against trunk.

- All open tickets that are in the REST API component or have the rest-api focus.

All open tickets that deal with the Customizer.

The screenshot shows the 'Custom Query' interface. Under the 'Filters' section, the 'Component' dropdown is set to 'is' and 'Customize'. The 'Status' section has checkboxes for 'accepted', 'assigned', 'closed', 'new', 'reopened', and 'reviewing', all of which are checked. There are 'And' and 'Or' dropdown menus for further filtering. Below the filters, the 'Columns' section shows 'Group results by' with a dropdown, a 'descending' checkbox, 'Show under each result: Description' checkbox, and 'Max items per page' set to '100'. An 'Update' button is at the bottom right.

All open tickets that need unit tests.

The screenshot shows the 'Custom Query' interface. Under the 'Filters' section, the 'Keywords' dropdown is set to 'contains' and the text input is 'needs-unit-tests'. The 'Status' section has checkboxes for 'accepted', 'assigned', 'closed', 'new', 'reopened', and 'reviewing', all of which are checked. There are 'And' and 'Or' dropdown menus for further filtering. Below the filters, the 'Columns' section shows 'Group results by' with a dropdown, a 'descending' checkbox, 'Show under each result: Description' checkbox, and 'Max items per page' set to '100'. An 'Update' button is at the bottom right.

All open tickets that are good for new contributors.

The screenshot shows the 'Custom Query' interface. Under the 'Filters' section, the 'Keywords' dropdown is set to 'contains' and the text input is 'good-first-bug'. The 'Status' section has checkboxes for 'accepted', 'assigned', 'closed', 'new', 'reopened', and 'reviewing', all of which are checked. There are 'And' and 'Or' dropdown menus for further filtering. Below the filters, the 'Columns' section shows 'Group results by' with a dropdown, a 'descending' checkbox, 'Show under each result: Description' checkbox, and 'Max items per page' set to '100'. An 'Update' button is at the bottom right.

All open tickets that are ready for commit.



### Custom Query

▼ Filters

☐ Keywords

☐ Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And  Or

▲ Columns

Group results by  ☐ descending Show under each result: ☐ Description Max items per page

All open bug tickets that have been reported against trunk.

### Custom Query

▼ Filters

☐ Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

☐ Type

☐ Version

And  Or

▲ Columns

Group results by  ☐ descending Show under each result: ☐ Description Max items per page

All open tickets that are in the REST API component or have the rest-api focus.

### Custom Query

▼ Filters

☐ Component

☐ Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And  Or

☐ Focuses

☐ Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And  Or

▲ Columns

Group results by  ☐ descending Show under each result: ☐ Description Max items per page

New Ticket Search

[Custom Query](#)

### Custom Query

▼ Filters

Reporter

Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And  Or

▲ Columns

Group results by  ☐ descending Show under each result: ☐ Description Max items per page  Update

Results (48)

| <input type="checkbox"/> Ticket | <a href="#">Summary</a> | Status | Owner | Type | Priority ▲ | Milestone |
|---------------------------------|-------------------------|--------|-------|------|------------|-----------|
|---------------------------------|-------------------------|--------|-------|------|------------|-----------|

Pro tip: Now that you know how to use Trac's powerful query functionality, try to figure out over time which queries are useful to you regularly, and bookmark them for quick access.

### Trac notifications

Once you open a ticket or comment on a ticket, you will automatically receive emails when that ticket is updated. In addition, you may manually subscribe to any ticket you're interested in.

But wait, there's more...

<> Make WordPress Core

Blog Handbook Tickets Components Notifications 

logged in as <your-username>

- On that page you can check components and focuses that you would like to receive all ticket updates for.
- You could also subscribe to the "firehose", so that you receive emails for every update on every ticket.
- → Go ahead if you wanna spend your entire life on WordPress. 🐼

New Ticket

Search

[Custom Query](#)

### Custom Query

▼ Filters

Reporter is

Status ☒ accepted ☒ assigned ☐ closed ☒ new ☒ reopened ☒ reviewing

And  Or

▲ Columns

Group results by  ☐ descending Show under each result: ☐ Description Max items per page

Results (48)

| <input type="checkbox"/> Ticket | Summary | Status | Owner | Type | Priority ▲ | Milestone |
|---------------------------------|---------|--------|-------|------|------------|-----------|
|---------------------------------|---------|--------|-------|------|------------|-----------|

## Creating patches

Patches are the code changes that are uploaded to the related ticket as attachments.

→ How do you do it right?

The workflow for creating a patch

Either download and apply the patch manually and upload your new patch manually, or do it like a pro...

**svn update**

**grunt patch:12345**

**# Do the code changes you need.**

**grunt upload\_patch:12345**

**svn revert -R \***

**12345 should be the ticket number.**

If you wanna upload the patch manually, use `svn diff > 12345.diff` and then upload the file as an attachment to the ticket manually.

## Best practices for writing patches

- Creating a patch should usually be one of the last things you do in a ticket lifecycle. Of course, you can always write a patch to have a better explanation of what you're trying to achieve, but keep in mind that it might not be accepted.
- Be patient, consider and discuss edge-cases. Letting a problem sink in for a while will improve the outcome.
- Name your patch after the ticket number and use the .diff file extension in favor of .patch. If there are multiple patches on the ticket, an index number will automatically be appended once the patch is uploaded.

## Writing a comment for the patch

- After uploading a patch, always leave a comment on the ticket describing what you changed and why you changed it. A patch upload itself does not trigger a notification, and furthermore people wouldn't know the reasoning behind your changes.
- Pro tip: Use [attachment:{filename}] to directly reference a specific patch on the ticket, and it will be automatically transformed into a link.
- Make sure to adjust the keywords as necessary:
  - Add the has-patch keyword if you uploaded the initial patch.
  - Add the has-unit-tests keyword if your patch already contains sufficient unit tests.
  - Use the dev-feedback keyword if the patch is rather complex and could use thorough feedback on the approach.

## Dealing with feedback

- If nobody responds to the ideas from your ticket or reviews your patch, it may be useful to bring that ticket forward in a related chat (more on that in just a bit). Make sure to ask kindly though, and don't annoy people. They're usually busy.
- If the consensus is that your idea is not viable for inclusion in WordPress core, or if your patch does not fulfill the discussed requirements, stay focused. It is easy to get frustrated by such decisions. Try to move the discussion forward in a constructive way.
- If you feel like your idea was rejected without a clear explanation, ask for the reasoning behind it. Contributors' times are often limited, so they unfortunately tend to write brief messages and omit things.
- Always remember to not take anything personally. The other people (probably) don't know you, so why should they have any negative impression?

If you wanna learn more about philosophical and psychological challenges:  
<http://wordpress.tv/2016/09/12/felix-arntz-where-are-the-core-contributors/>