# Contributing to Glances

Looking to contribute something to Glances ? \*\*Here's how you can help.\*\*

Please take a moment to review this document in order to make the contribution

process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of

the developers managing and developing this open source project. In return,

they should reciprocate that respect in addressing your issue or assessing

patches and features.

## Using the issue tracker

The [issue tracker](https://github.com/nicolargo/glances/issues) is

the preferred channel for [bug reports](#bug-reports), [features requests](#feature-requests)

and [submitting pull requests](#pull-requests), but please respect the following

restrictions:

\* Please \*\*do not\*\* use the issue tracker for personal support requests. A official Q&A exist. [Use it](https://groups.google.com/forum/?hl=en#!forum/glances-users)!

\* Please \*\*do not\*\* derail or troll issues. Keep the discussion on topic and

respect the opinions of others.

## Bug reports

A bug is a \_demonstrable problem\_ that is caused by the code in the repository.

Good bug reports are extremely helpful, so thanks!

Guidelines for bug reports:

0. \*\*Use the GitHub issue search\*\* &mdash; check if the issue has already been

reported.

1. \*\*Check if the issue has been fixed\*\* &mdash; try to reproduce it using the

latest `master` or `develop` branch in the repository.

2. \*\*Isolate the problem\*\* &mdash; ideally create a simple test bed.

3. \*\*Give us your test environment\*\* &mdash; Operating system name and version

Glances version...

Example:

> Short and descriptive example bug report title

>

> Glances and psutil version used (glances -V)

>

> Operating system description (name and version).

>

> A summary of the issue and the OS environment in which it occurs. If

> suitable, include the steps required to reproduce the bug.

>

> 1. This is the first step

> 2. This is the second step

> 3. Further steps, etc.

>

> Screenshot (if usefull)

>

> Any other information you want to share that is relevant to the issue being

> reported. This might include the lines of code that you have identified as

> causing the bug, and potential solutions (and your opinions on their

> merits).

## Feature requests

Feature requests are welcome. But take a moment to find out whether your idea

fits with the scope and aims of the project. It's up to \*you\* to make a strong

case to convince the project's developers of the merits of this feature. Please

provide as much detail and context as possible.

## Pull requests

Good pull requests—patches, improvements, new features—are a fantastic

help. They should remain focused in scope and avoid containing unrelated

commits.

\*\*Please ask first\*\* before embarking on any significant pull request (e.g.

implementing features, refactoring code, porting to a different language),

otherwise you risk spending a lot of time working on something that the

project's developers might not want to merge into the project.

First of all, all pull request should be done on the `develop` branch.

Glances uses PEP8 compatible code, so use a PEP validator before submitting

your pull request. Also uses the unitaries tests scripts (unitest-all.py).

Similarly, when contributing to Glances's documentation, you should edit the

documentation source files in

[the `/doc/` and `/man/` directories of the `develop` branch](https://github.com/nicolargo/glances/tree/develop/docs) and generate

the documentation outputs files by reading the [README](https://github.com/nicolargo/glances/tree/develop/docs/README.txt) file.

Adhering to the following process is the best way to get your work

included in the project:

1. [Fork](https://help.github.com/fork-a-repo/) the project, clone your fork,

and configure the remotes:

```bash

# Clone your fork of the repo into the current directory

git clone https://github.com/<your-username>/glances.git

# Navigate to the newly cloned directory

cd glances

# Assign the original repo to a remote called "upstream"

git remote add upstream https://github.com/nicolargo/glances.git

```

2. Get the latest changes from upstream:

```bash

git checkout develop

git pull upstream develop

```

3. Create a new topic branch (off the main project development branch) to

contain your feature, change, or fix (best way is to call it issue#xxx):

```bash

git checkout -b <topic-branch-name>

```

4. It's coding time !

Please respect the following coding convention: [Elements of Python Style](https://github.com/amontalenti/elements-of-python-style)

Commit your changes in logical chunks. Please adhere to these [git commit

message guidelines](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html)

or your code is unlikely be merged into the main project. Use Git's

[interactive rebase](https://help.github.com/articles/interactive-rebase)

feature to tidy up your commits before making them public.

5. Locally merge (or rebase) the upstream development branch into your topic branch:

```bash

git pull [--rebase] upstream develop

```

6. Push your topic branch up to your fork:

```bash

git push origin <topic-branch-name>

```

7. [Open a Pull Request](https://help.github.com/articles/using-pull-requests/)

with a clear title and description against the `develop` branch.

\*\*IMPORTANT\*\*: By submitting a patch, you agree to allow the project owners to

license your work under the terms of the [LGPLv3](COPYING) (if it

includes code changes).