

# PowerToys Contributor's Guide

Below is our guidance for how to report issues, propose new features, and submit contributions via Pull Requests (PRs). Our philosophy is heavily based around understanding the problem and scenarios first, this is why we follow this pattern before work has started.

1. There is an issue
2. There has been a conversation
3. There is agreement on the problem, the fit for PowerToys, and the solution to the problem (implementation)

## Filing an issue

Please follow this simple rule to help us eliminate any unnecessary wasted effort & frustration, and ensure an efficient and effective use of everyone's time - yours, ours, and other community members':

👉 If you have a question, think you've discovered an issue, would like to propose a new feature, etc., then find/file an issue **BEFORE** starting work to fix/implement it.

When requesting new features / enhancements, understanding problem and scenario around it is extremely important. Having additional evidence, data, tweets, blog posts, research, ... anything is extremely helpful. This information provides context to the scenario that may otherwise be lost.

- Don't know whether you're reporting an issue or requesting a feature? File an issue
- Have a question that you don't see answered in docs, videos, etc.? File an issue
- Want to know if we're planning on building a particular feature? File an issue
- Got a great idea for a new utility or feature? File an issue/request/idea
- Don't understand how to do something? File an issue/Community Guidance Request
- Found an existing issue that describes yours? Great - upvote and add additional commentary / info / repro-steps / etc.

A quick search before filing an issue also could be helpful. It is likely someone else has found the problem you're seeing, and someone may be working on or have already contributed a fix!

## How to tell the PowerToys team this is an interesting thing to focus on

Upvote the original issue by clicking its [+ 😊] button and hitting 👍 (+1) icon or a different one. This way allows us to measure how impactful different issues are compared to others. The issue with comments like "+1", "me too", or similar is they actually make it harder to have a conversation and harder to quickly determine trending important requests.

---

## Contributing fixes / features

Please comment on an issue to let us know you're interested in working on something before you start the work. Not only does this avoid multiple people unexpectedly working on the same thing at the same time but it enables us to make sure everyone is clear on what should be done to implement any new functionality. It's less work for everyone, in the long run, to establish this up front.

## To Spec or not to Spec

A key point is for everyone to understand the approach that will be taken. We want to be sure if anyone does work, we will accept it in. Items that are larger in scope we'll want some type of spec to understand what is being planned and have a discussion. Specs help collaborators discuss different approaches to solve a problem, describe how the feature will behave, how the feature will impact the user, what happens if something goes wrong, etc. Driving

towards agreement in a spec, before any code is written, often results in simpler code, and less wasted effort in the long run.

For such scenarios, once a team member has agreed with your approach, skip ahead to the section headed "Development" section below.

Team members will be happy to help review specs and guide them to completion.

## **Help Wanted**

Once the team has approved an issue/spec approach to solving, development can proceed. If no developers are immediately available, the spec can be parked ready for a developer to get started. Parked specs' issues will be labeled "Help Wanted". To find a list of development opportunities waiting for developer involvement, visit the Issues and filter on [the Help-Wanted label](#).

---

## **Development**

Follow the [development guidelines](#).

### **Naming of features and functionality**

Naming should be descriptive and straight forward. We want names to be clear about functionality and usefulness moving forward.

### **How can I become a collaborator on the PowerToys team**

Be a great community member. Just help out a lot and make useful additions, filing bugs/suggestions, help develop fixes and features, code reviews, and always, docs. Lets continue to make the PowerToys repository a great spot to learn and make a great set of utilities.

When the time comes, Microsoft will reach out and help make you a formal team member. Just make sure they can reach out to you :)

---

## **Thank you**

Thank you in advance for your contribution!