Contributing to the FreeBSD Ports Collection

Sam Lawrance
Mark Linimon
Revision: 45519

FreeBSD is a registered trademark of the FreeBSD Foundation.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and the FreeBSD Project was aware of the trademark claim, the designations have been followed by the "™" or the "®" symbol.

2014-08-26 by jgh.

Abstract

This article describes the ways in which an individual can contribute to the FreeBSD Ports Collection.

Table of Contents

1. Introduction 1
2. What you can do to help 2
3. Creating a new port 2
4. Adopting an unmaintained port 2
5. The challenge for port maintainers 3
6. Finding and fixing a broken port 10
7. When to call it quits 10
8. Resources for ports maintainers and contributors 10
Index 11

1. Introduction

The Ports Collection is a perpetual work in progress. We want to provide our users with an easy to use, up to date, high quality repository of third party software. We need people to donate some of their time and effort to help us achieve this goal.

Anyone can get involved, and there are lots of different ways to do so. Contributing to ports is an excellent way to help “give back” something to the project. Whether you are looking for an ongoing role, or a fun challenge for a rainy day, we would love to have your help!

As a volunteer, what you do is limited only by what you want to do. However, we do ask that you are aware of what other members of the FreeBSD community will expect of you. You may want to take this into account before deciding to volunteer.

2. What you can do to help

There are a number of easy ways you can contribute to keeping the ports tree up to date and in good working order:

- Find some cool or useful software and create a port for it.
- There are a large number of ports that have no maintainer. Become a maintainer and adopt a port.
- If you have created or adopted a port, be aware of what you need to do as a maintainer.
- When you are looking for a quick challenge you could fix a bug or a broken port.

3. Creating a new port

There is a separate document available to help guide you through creating (and upgrading) a port called the Porter's Handbook. The Porter's Handbook is the best reference to working with the ports system. It provides details about how the ports system operates and discusses recommended practices.

4. Adopting an unmaintained port

4.1. Choosing an unmaintained port

Taking over maintainership of ports that are unmaintained is a great way to get involved. Unmaintained ports are only updated and fixed when somebody volunteers to work on them. There are a large number of unmaintained ports. It is a good idea to start with adopting a port that you use regularly.

Unmaintained ports have their `MAINTAINER` set to `ports@FreeBSD.org` . A list of unmaintained ports and their current errors and problem reports can be seen at the FreeBSD Ports Monitoring System.

Some ports affect a large number of others due to dependencies and slave port relationships. Generally, we want people to have some experience before they maintain such ports.

You can find out whether or not a port has dependencies or slave ports by looking at a master index of ports called `INDEX`. (The name of the file varies by release of FreeBSD; for instance, `INDEX-8`.) Some ports have conditional dependencies that are not included in a default `INDEX` build. We expect you to be able to recognize such ports by looking through other ports' `Makefiles`.

4.2. How to adopt the port

First make sure you understand your responsibilities as a maintainer. Also read the Porter's Handbook. *Please do not commit yourself to more than you feel you can comfortably handle.*

You may request maintainership of any unmaintained port as soon as you wish. Simply set `MAINTAINER` to your own email address and send a PR (Problem Report) with the change. If the port has build errors or needs updating, you may wish to include any other changes in the same PR. This will help because many committers are less willing to assign maintainership to someone who does not have a known track record with FreeBSD. Submitting PRs that fix build errors or update ports are the best ways to establish one.

File your PR with category `ports` and class `change-request` . A committer will examine your PR, commit the changes, and finally close the PR. Sometimes this process can take a little while (committers are volunteers, too :).

5. The challenge for port maintainers

This section will give you an idea of why ports need to be maintained and outline the responsibilities of a port maintainer.

5.1. Why ports require maintenance

Creating a port is a once-off task. Ensuring that a port is up to date and continues to build and run requires an ongoing maintenance effort. Maintainers are the people who dedicate some of their time to meeting these goals.

The foremost reason ports need maintenance is to bring the latest and greatest in third party software to the FreeBSD community. An additional challenge is to keep individual ports working within the Ports Collection framework as it evolves.

As a maintainer, you will need to manage the following challenges:

- New software versions and updates. New versions and updates of existing ported software become available all the time, and these need to be incorporated into the Ports Collection in order to provide up-to-date software.
- Changes to dependencies. If significant changes are made to the dependencies of your port, it may need to be updated so that it will continue to work correctly.
- Changes affecting dependent ports. If other ports depend on a port that you maintain, changes to your port may require coordination with other maintainers.
- Interaction with other users, maintainers and developers. Part of being a maintainer is taking on a support role. You are not expected to provide general support (but we welcome it if you choose to do so). What you should provide is a point of coordination for FreeBSD-specific issues regarding your ports.
- Bug hunting. A port may be affected by bugs which are specific to FreeBSD. You will need to investigate, find, and fix these bugs when they are reported. Thoroughly testing a port to identify problems before they make their way into the Ports Collection is even better.
- Changes to ports infrastructure and policy. Occasionally the systems that are used to build ports and packages are updated or a new recommendation affecting the infrastructure is made. You should be aware of these changes in case your ports are affected and require updating.
- Changes to the base system. FreeBSD is under constant development. Changes to software, libraries, the kernel or even policy changes can cause flow-on change requirements to ports.

5.2. Maintainer responsibilities

5.2.1. Keep your ports up to date

This section outlines the process to follow to keep your ports up to date.

This is an overview. More information about upgrading a port is available in the Porter's Handbook.

1. Watch for updates

   Monitor the upstream vendor for new versions, updates and security fixes for the software. Announcement mailing lists or news web pages are useful for doing this. Sometimes users will contact you and ask when your port will be updated. If you are busy with other things or for any reason just cannot update it at the moment, ask if they will help you by submitting an update.

   You may also receive automated email from the `FreeBSD Ports Version Check` informing you that a newer version of your port's distfile is available. More infor-

mation about that system (including how to stop future emails) will be provided in the message.

2. Incorporate changes

   When they become available, incorporate the changes into the port. You need to be able to generate a patch between the original port and your updated port.

3. Review and test

   Thoroughly review and test your changes:

   - Build, install and test your port on as many platforms and architectures as you can. It is common for a port to work on one branch or platform and fail on another.
   - Make sure your port's dependencies are complete. The recommended way of doing this is by installing your own ports tinderbox. See resources for more information.
   - Check that the packing list is up to date. This involves adding in any new files and directories and removing unused entries.
   - Verify your port using portlint(1) as a guide. See resources for important information about using portlint.
   - Consider whether changes to your port might cause any other ports to break. If this is the case, coordinate the changes with the maintainers of those ports. This is especially important if your update changes the shared library version; in this case, at the very least, the dependent ports will need to get a `PORTREVISION` bump so that they will automatically be upgraded by automated tools such as portmaster or portupgrade(1).

4. Submit changes

   Send your update by submitting a PR with an explanation of the changes and a patch containing the differences between the original port and the updated one. Please refer to Writing FreeBSD Problem Reports for information on how to write a really good PR.

Note

Please do not submit a shar(1) archive of the entire port; instead, use diff(1) `-ruN`. In this way, committers can much more

easily see exactly what changes are being made. The Porter's Handbook section on Upgrading has more information.

5. Wait

   At some stage a committer will deal with your PR. It may take minutes, or it may take weeks — so please be patient.

6. Give feedback

   If a committer finds a problem with your changes, they will most likely refer it back to you. A prompt response will help get your PR committed faster, and is better for maintaining a thread of conversation when trying to resolve any problems.

7. And Finally

   Your changes will be committed and your port will have been updated. The PR will then be closed by the committer. That's it!

5.2.2. Ensure your ports continue to build correctly

This section is about discovering and fixing problems that stop your ports from building correctly.

FreeBSD only guarantees that the Ports Collection works on the `-STABLE` branches. You should be running `7-STABLE` or `8-STABLE`, preferably the latter. In theory, you should be able to get by with running the latest release of each stable branch (since the ABIs are not supposed to change) but if you can run the branch, that is even better.

Since the majority of FreeBSD installations run on PC-compatible machines (what is termed the `i386` architecture), we expect you to keep the port working on that architecture. We prefer that ports also work on the `amd64` architecture running native. It is completely fair to ask for help if you do not have one of these machines.

Note

The usual failure modes for non-`i386` machines are that the original programmers assumed that, for instance, pointers are `int`s, or that a relatively lax older gcc compiler was being used. More and more, application authors are reworking their code to remove these assumptions — but if the author is not actively maintaining their code, you may need to do this yourself.

These are the tasks you need to perform to ensure your port is able to be built:

1. Watch for build failures

   Regularly check the automated ports building cluster, pointyhat, and the distfiles scanner to see if any of the ports you maintain are failing to build or fetch (see resources for more information about these systems). Reports of failures may also come to you from other users or automated systems via email.

2. Collect information

   Once you are aware of a problem, collect information to help you fix it. Build errors reported by `pointyhat` are accompanied by logs which will show you where the build failed. If the failure was reported to you by a user, ask them to send you information which may help in diagnosing the problem, such as:

   - Build logs
   - The commands and options used to build the port (including options set in `/etc/make.conf`)
   - A list of packages installed on their system as shown by pkg_info(1)
   - The version of FreeBSD they are running as shown by uname(1) `-a`
   - When their ports collection was last updated
   - When their `INDEX` file was last updated

3. Investigate and find a solution

   Unfortunately there is no straightforward process to follow to do this. Remember, though: if you are stuck, ask for help! The FreeBSD ports mailing list is a good place to start, and the upstream developers are often very helpful.

4. Submit changes

   Just as with updating a port, you should now incorporate changes, review and test, submit your changes in a PR, and provide feedback if required.

5. Send patches to upstream authors

   In some cases, you will have to make patches to the port to make it run on FreeBSD. Some (but not all) upstream authors will accept such patches back into their code for the next release. If so, this may even help their users on other BSD-based systems as well and perhaps save duplicated effort. Please consider sending any applicable patches to the authors as a courtesy.

5.2.3. Investigate bug reports and PRs related to your port

This section is about discovering and fixing bugs.

FreeBSD-specific bugs are generally caused by assumptions about the build and runtime environments that do not apply to FreeBSD. You are less likely to encounter a problem of this type, but it can be more subtle and difficult to diagnose.

These are the tasks you need to perform to ensure your port continues to work as intended:

1. Respond to bug reports

   Bugs may be reported to you through email via the Problem Report database. Bugs may also be reported directly to you by users.

   You should respond to PRs and other reports within 14 days, but please try not to take that long. Try to respond as soon as possible, even if it is just to say you need some more time before you can work on the PR.

   If you have not responded after 14 days, any committer may commit from a PR that you have not responded to via a `maintainer-timeout`.

2. Collect information

   If the person reporting the bug has not also provided a fix, you need to collect the information that will allow you to generate one.

   If the bug is reproducible, you can collect most of the required information yourself. If not, ask the person who reported the bug to collect the information for you, such as:

   - A detailed description of their actions, expected program behavior and actual behavior
   - Copies of input data used to trigger the bug
   - Information about their build and execution environment — for example, a list of installed packages and the output of env(1)
   - Core dumps
   - Stack traces

3. Eliminate incorrect reports

   Some bug reports may be incorrect. For example, the user may have simply misused the program; or their installed packages may be out of date and require updating.

Sometimes a reported bug is not specific to FreeBSD. In this case report the bug to the upstream developers. If the bug is within your capabilities to fix, you can also patch the port so that the fix is applied before the next upstream release.

4. Find a solution

   As with build errors, you will need to sort out a fix to the problem. Again, remember to ask if you are stuck!

5. Submit or approve changes

   Just as with updating a port, you should now incorporate changes, review and test, and submit your changes in a PR (or send a follow-up if a PR already exists for the problem). If another user has submitted changes in the PR, you can also send a follow-up saying whether or not you approve the changes.

5.2.4. Providing support

Part of being a maintainer is providing support — not for the software in general — but for the port and any FreeBSD-specific quirks and problems. Users may contact you with questions, suggestions, problems and patches. Most of the time their correspondence will be specific to FreeBSD.

Occasionally you may have to invoke your skills in diplomacy, and kindly point users seeking general support to the appropriate resources. Less frequently you will encounter a person asking why the `RPM`s are not up to date or how can they get the software to run under Foo Linux. Take the opportunity to tell them that your port is up to date (if it is, of course!), and suggest that they try FreeBSD.

Sometimes users and developers will decide that you are a busy person whose time is valuable and do some of the work for you. For example, they might:

- submit a PR or send you patches to update your port,
- investigate and perhaps provide a fix to a PR, or
- otherwise submit changes to your port.

In these cases your main obligation is to respond in a timely manner. Again, the timeout for non-responsive maintainers is 14 days. After this period changes may be committed unapproved. They have taken the trouble to do this for you; so please try to at least respond promptly. Then review, approve, modify or discuss their changes with them as soon as possible.

If you can make them feel that their contribution is appreciated (and it should be) you will have a better chance persuading them to do more things for you in the future :-).

6. Finding and fixing a broken port

There are two really good places to find a port that needs some attention.

You can use the web interface to the Problem Report database to search through and view unresolved PRs. The majority of ports PRs are updates, but with a little searching and skimming over synopses you should be able to find something interesting to work on (the `sw-bug` class is a good place to start).

The other place is the FreeBSD Ports Monitoring System. In particular look for unmaintained ports with build errors and ports that are marked `BROKEN`. It is OK to send changes for a maintained port as well, but remember to ask the maintainer in case they are already working on the problem.

Once you have found a bug or problem, collect information, investigate and fix! If there is an existing PR, follow up to that. Otherwise create a new PR. Your changes will be reviewed and, if everything checks out, committed.

7. When to call it quits

As your interests and commitments change, you may find that you no longer have time to continue some (or all) of your ports contributions. That is fine! Please let us know if you are no longer using a port or have otherwise lost time or interest in being a maintainer. In this way we can go ahead and allow other people to try to work on existing problems with the port without waiting for your response. Remember, FreeBSD is a volunteer project, so if maintaining a port is no fun anymore, it is probably time to let someone else do it!

In any case, the Ports Management Team (`portmgr`) reserves the right to reset your maintainership if you have not actively maintained your port in some time. (Currently, this is set to 3 months.) By this, we mean that there are unresolved problems or pending updates that have not been worked on during that time.

8. Resources for ports maintainers and contributors

The Porter's Handbook is your hitchhiker's guide to the ports system. Keep it handy!

Writing FreeBSD Problem Reports describes how to best formulate and submit a PR. In 2005 more than eleven thousand ports PRs were submitted! Following this article will greatly assist us in reducing the time needed to handle your PRs.

The Problem Report database.

Pointyhat is the ports build cluster. You can use Pointyhat to check port build logs across all architectures and major releases.

The FreeBSD Ports Monitoring System can show you cross-referenced information about ports such as build errors and problem reports. If you are a maintainer you can use it to check on the build status of your ports. As a contributor you can use it to find broken and unmaintained ports that need to be fixed.

The FreeBSD Ports distfile scanner can show you ports for which the distfiles are not fetchable. You can check on your own ports or use it to find ports that need their `MASTER_SITES` updated.

The ports tinderbox is the most thorough way to test a port through the entire cycle of installation, packaging, and deinstallation. It features a command-line interface but also can be controlled via a web interface. Please see `ports/ports-mgmt/tinderbox` . More documentation is located at the marcuscom tinderbox home page.

portlint(1) is an application which can be used to verify that your port conforms to many important stylistic and functional guidelines. portlint is a simple heuristic application, so you should use it *only as a guide*. If portlint suggests changes which seem unreasonable, consult the Porter's Handbook or ask for advice.

The FreeBSD ports mailing list is for general ports-related discussion. It is a good place to ask for help. You can subscribe, or read and search the list archives. Reading the archives of the FreeBSD ports bugs mailing list and the FreeBSD CVS ports commit list may also be of interest.

Index

C

contributing to ports, 1