Project Manager, Team Firefox
Mozilla Foundation
650 Castro Street, Suite 300
Mountain View, CA 94041-2021

Dear Project Manager:

I would like to thank you for your substantial contributions throughout the past ten years towards pushing the boundaries of web and browser technologies forward. Your efforts in FIrefox have provided a powerful, open source alternative for those who were dissatisfied with the web browser that their computers came with, and opened up many opportunities for users to truly have the browser they want thanks to its vast extensibility.

Enclosed is a proposal involving a feature which I respectfully request that you implement. It asks that Firefox's download manager be separated from the web browser itself as a standalone process, so that if at any time the web browser were to crash, the user's downloads would be able to continue on uninterrupted. While Firefox is very stable in its current state, I understand that no software is completely immune to crashing, and believe that this functionality would greatly mitigate that issue.

I thank you once again for your consideration and time. Please feel free to contact me if you have any questions or need any additional information.

Sincerely,

Sandbox Firefox’s Download Manager

I. Overview of Issues

The Issue

According to The Guardian, users download 17 gigabytes (GB) of data every month on their home broadband alone. These downloads can range from smaller downloads such as documents to much larger downloads such as entire movies or music albums. Often, these larger downloads can run for as long as several hours at a time, and must run uninterrupted from start to completion. If a download fails before completion, it must be restarted, wasting the time and bandwidth of the failed session. These interruptions are generally difficult to mitigate, since they mainly occur for one of the following reasons: being disconnected from the Internet, having the operating system crash, or having the browser crash—all of which are impossible to completely prevent.

The Users Affected

While issues involving being disconnected from the Internet or having the entire operating system crash are beyond the reach of Team Firefox, having the browser crash is very much relevant to Firefox itself. According to Mozilla Crash Reports, in a group of 100 active daily users, somewhere between two to seven crashes occur every day. Firefox itself has over 450 million users globally, all of which are at risk of having Firefox crash. If the statistic of two to seven crashes occurs every day within a group of 100 users, then at 450 million users, this means that approximately 90,000 to 315,000 crashes occur every day, raising the potential for interrupted downloads by a substantially larger margin. Should any of those crashes occur while a user downloads something, that download will be lost. However, crashes do not happen uniformly across all users; several kinds of users run a higher risk of having Firefox crash on them: those with many third-party extensions installed and those not using the latest stable build of Firefox.

One of the causes of Firefox crashes is from the proliferation of third-party extensions whose stability Firefox cannot guarantee. These users with many extensions installed do not solely rely on code put out by Mozilla. It is also possible that extensions that run stably on their own do not integrate well with other extensions; one extension may modify a resource that another extension might need. Integration issues like these are more prevalent for users who have many extensions installed at a time.

Not every user is using the most recent stable build of Firefox either. Those users who do not miss out on the latest bug fixes or patches, which may help prevent crashes. In some cases, these users are developers or testers who choose to use developer versions of Firefox, such as the beta (Aurora) or nightly (Nightly) builds. These users understand that they exchange stability for access to Firefox's more cutting edge features, though a crash can impact them and potentially interrupt a download in the same ways that it would for someone using Firefox's most recent stable build.

There also exist many users who are using an outdated version of Firefox who do not, or sometimes even cannot, upgrade to a current stable release, especially given Firefox's rapid release cycle. Some users may be reluctant to upgrade because they might feel that it causes too much work for them. Others may avoid upgrading because of the potential risk of breaking their currently installed extensions, which would drastically detract from their experiences with newer versions of Firefox. Users who use Firefox in a work environment may also be limited by their IT department, who may mandate that workstations must only run a company-approved version of Firefox. All of these scenarios can potentially prevent a user from obtaining the most recent bug fixes and patches for Firefox.

II. Solutions

Proposed Solution: Sandbox the Download Manager

One of Firefox's main features is its download manager, which lists all of a user's previous downloads in a separate window, allowing the user to quickly look back on old downloads or to manage a currently running download. This download manager runs under the same application process as Firefox does; if the Firefox process terminates for any reason, the download manager will as well, interrupting all current downloads.

While users download data, they cannot directly interact with that data—a stark contrast to when they browse the web and constantly interact with the data presented to them. This shows the two primary activities involved in Internet use: downloading data and browsing webpages. Firefox currently separates these concerns by having the download manager in a separate window from the browser itself, but be further separated. These two components should be separated into independent system processes running on a computer. This means that the web browser and the download manager would run as separate, independent entities that would interact with each other via inter-process communication (IPC).

This redesign will impact much of how Firefox's backend works with downloading software, safeguarding the download manager from crashing should the browser process crash. If Firefox were to crash in this separated environment, then the browser process would terminate on the user and close all of their webpages. Meanwhile, the download manager process would continue unaffected because of its existence as a completely separate process. This would completely remove one of the potential ways that a download might be interrupted: Firefox crashing.

Existing Solution: Pausing Downloads

Firefox currently has a mechanism for ‘pausing’ a download, temporarily stopping it and protecting the download from any interruptions during that time provided the Firefox process does not terminate. Since pausing involves an explicit action from the user, pausing works best when the user can anticipate an upcoming interruption and can be there to pause the download. For unpredictable interruptions, pausing a download is ineffective. In most cases, interruptions occur without warning, which makes pausing usable only a small fraction of the time, and only if those interruptions do not crash the operating system or Firefox itself. Pausing downloads therefore fails to address the issue of Firefox crashing.

III. Budget

Developers

Firefox’s browser and download manager are backend components which affect FIrefox’s internal software architecture. Firefox’s backend software engineers would be the best fit for implementing these changes. Additionally, Firefox’s open source nature means that other developers have the opportunity to contribute to these changes, particularly those well-versed in C++ as Firefox’s backend is written in. These developers should be comfortable with systems programming, especially with inter-process communication. A crucial part of implementing this separation involves coordinating the browser and the download manager to be able to continue to interact with each other similarly to the way they do now. This involves a large rewrite of the download manager, which would have to accommodate its change to a separate process. In particular, tightly-coupled code between the browser and the download manager will need to be replaced with a more modular design that employs inter-process communication.

Quality Assurance (QA)

This redesign will require careful testing to ensure its release as a stable, polished feature because of the work involved with redesigning the download manager as a separate process. Despite the large testing burden, much of the testing infrastructure already exists, and the redesigned download manager only needs to leverage it. Firefox’s automated crash reporters and testing tools bundled with its pre-release builds will be instrumental in showing where crashes or failures occur the most. Users running the Aurora or Nightly builds can continue to send feedback to Firefox the way they do now, especially since they will be the first to receive the new updates. Mozilla can also take advantage of QMO, the Mozilla QA team, especially its Desktop Firefox, Browser Technologies, and Automation teams. The Desktop Firefox team and Browser Technologies could collaborate to determine a set of suitable tests to run, while the Automation team could automate them to have them constantly running with each trunk build of Firefox containing the new download manager. Testing these new changes should be relatively easy with the current testing infrastructure in place.

IV. Conclusion

Internet use primarily consists of two activities: downloading data and browsing webpages. Downloading data can be a sensitive operation, since any interruption will force it to restart, wasting bandwidth and time from the failed attempt. These interruptions can come from either being disconnected to the Internet or from having the operating system or browser crash. No software is completely impervious to crashing, especially large projects such as Firefox, though it is possible to take steps to mitigate these crashes.

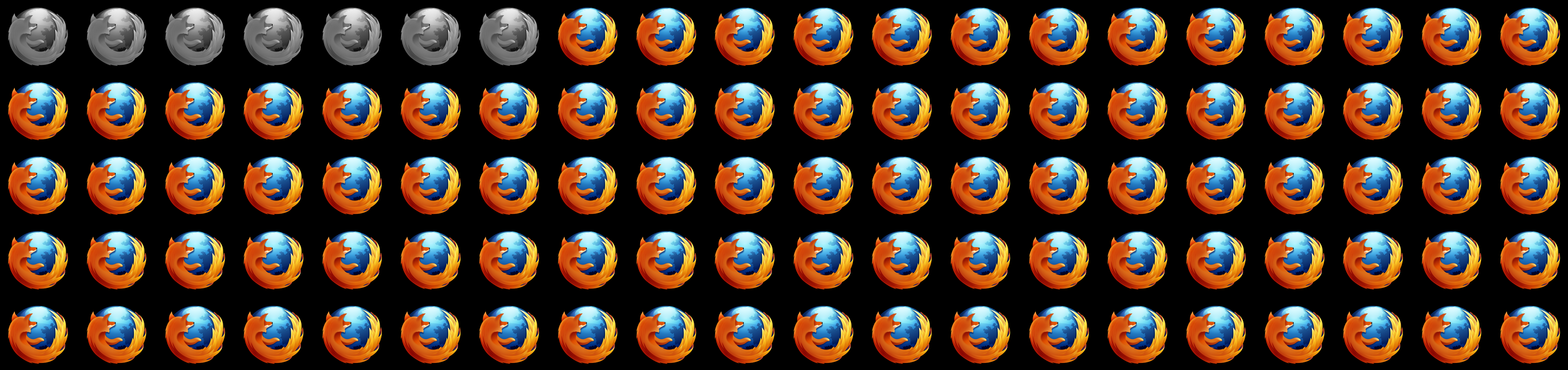
Sandboxing Firefox’s download manager by separating it from Firefox’s browser as an independent system process would allow downloads to continue running, even if the browser itself crashes. This solution would save wasted time and bandwidth, completely removing one of the causes of interrupted downloads: Firefox crashing.

Sandboxing Firefox’s Download Manager

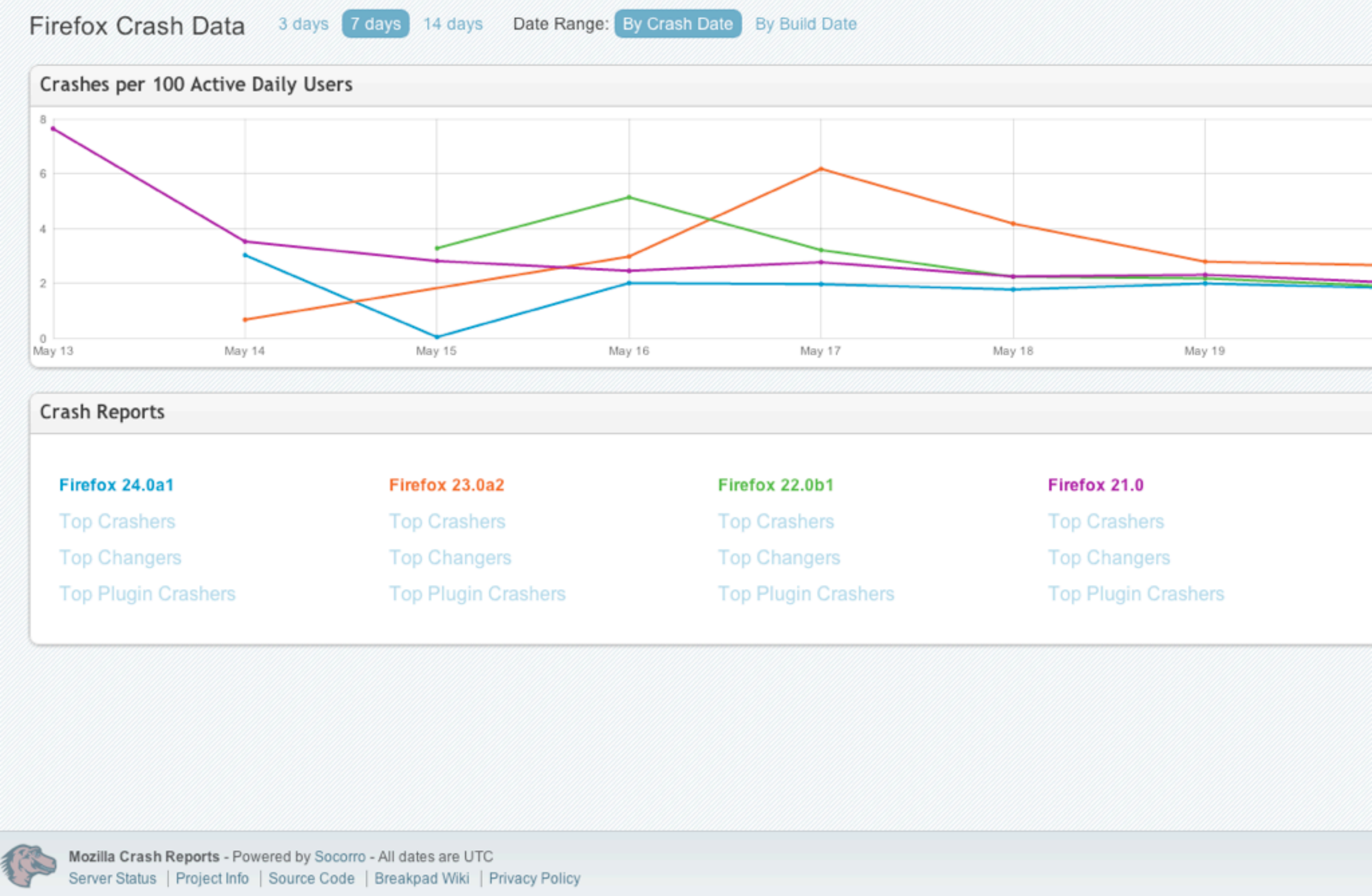
Background of Issues

For every **100 active daily users**…

For every **100 active daily users**...



Between **2 to 7 crashes occur every day**.



Firefox Crash Data

3 days | 7 days | 14 days

Date Range: By Crash Date | By Build Date

Crashes per 100 Active Daily Users

Crash Reports

Firefox 24.0a1

Top Crashers

Top Changers

Top Plugin Crashers

Firefox 23.0a2

Top Crashers

Top Changers

Top Plugin Crashers

Firefox 22.0b1

Top Crashers

Top Changers

Top Plugin Crashers

Firefox 21.0

Top Crashers

Top Changers

Top Plugin Crashers

Mozilla Crash Reports - Powered by Socorro - All dates are UTC

Server Status | Project Info | Source Code | Breakpad Wiki | Privacy Policy

Firefox has **450 million users**.

If 7 crashes happen for every 100 users, then **315,000 crashes** will happen for every 450 million users.

That’s a lot of crashes.

People are frequently downloading content:

People are frequently downloading content:

- music
- books
- movies
- software
- photos

The average user downloads
17 GB of data every day
on their home broadband alone.

The Users Affected

Some users are more prone to crashing:

- users who are using many Firefox extensions
- users who are not using the most recent stable build of Firefox

Users who are using many Firefox extensions.

- Extensions are developed by a third party
- third parties are neither screened nor monitored by Firefox for quality assurance

Some users do not (or cannot) use the most recent stable build of Firefox:

- developers on Aurora or Nightly builds
- users who feel it is too much work to upgrade
- users for whom upgrading will break some of their Extensions for
- employees who are forced to use a given version of Firefox on their workstations

An Existing Solution

Firefox support pausing downloads, but it is inadequate for:

- unpredictable connection issues
- operating system crashes
- Firefox crashes

The Solution

Firefox currently involves
browsing the web and **downloading data**.

Separate the process for **browsing the web** and the process for **downloading data** into independent processes.

The Benefits

Because now the operating system also acknowledges browsing the web and downloading data as separate processes, they are **no longer dependent on each other**.

If the browser crashes,
the download manager can still run.

The Budget

Backend software developers needed:

- C++
- systems programming
- inter-process communication (IPC)

Testing infrastructure needed:

- Aurora Channel Feedback
- Nightly Channel Feedback
- Mozilla Crash Reporter
- QMO (Mozilla QA)

Thank you.