

Question	Answer (Human)	Answer (GPT)	Answer (GPT/RAG) [1]	Characteristics Answer (Human)	Characteristics Answer (GPT)	Characteristics Answer (GPT/RAG)	Which answer provides the most helpful information?	Which answer provides the most comprehensive information?	Which answer is the most concise without being incomplete?	Which answer do you prefer to see in practice?	Regarding the answer you prefer to see in practice, we encourage you to emphasize all particular aspects that are (ir)relevant in the answer(s) and aspects you think I missed.
<p><suppressed_user> Hi team . Hope everybody is doing well. We have a couple of [crash reports] (https://bugzilla.mozilla.org/show_bug.cgi?id=1873619) on Android, coming from 'builds so', with the below stack. Is somebody familiar with this?</p> <p>...</p> <p>Reason: SIGSEGV / SEGV_MAPERR 0 'Builds so' libutils.so@0x0000000000000000 1 boot.oat boot.oat@0x0000000000000000 2 ? @0x0000000000000000</p> <p>Thanks in advance</p>	<p>The stacks look rather bad. Maybe <suppressed_user> would have some ideas of what could be going wrong. The devices the crashes are on all have the same manufacturer, so maybe that means something.</p>	<p>Hi <suppressed_user>.</p> <p>The crash reports you're seeing are likely due to a segmentation fault in 'libutils.so'. This is a known issue and has been reported in Bugzilla under (Bug 1873619)https://bugzilla.mozilla.org/show_bug.cgi?id=1873619). You may want to follow the updates on that bug report for further information and potential fixes.</p>	<p>Hi , the crash reports indicate a segmentation fault (SIGSEGV) in libutils.so, which is a common library in Android's native code. The specific address (builds.so@0x0000000000000000) suggests an issue within that library, but without more context, it's challenging to pinpoint the exact cause. This could be due to memory corruption, invalid memory access, or a bug in the library itself. It would be helpful to gather more information, such as the Android version, device model, and steps to reproduce the crash. Additionally, checking for updates or patches for libutils so might provide a solution. [2]</p>	Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise	Helpful, Comprehensive	Answer (GPT)	Answer (GPT/RAG)	Answer (GPT)	Answer (GPT)	Missing: the advice from Answer (GPT/RAG) to gather as much environment information as possible, and steps to reproduce
<p>Does someone here know what one can do about this? After upgrading to Firefox 121, it appears macOS is unwilling to allow Firefox to be started from the command-line, thus making e.g. unit tests via Karma and such impossible.</p> <p>https://github.com/karma-runner/karma-firefox-launcher/issues/328#issuecomment-188513046</p> <p>macOS is not allowing the program to run? Not even for "--version", or with no arguments.</p> <p>\$ Applications/Firefox.app/Contents/MacOS/Firefox-bin Killed: 9</p>	<p>Based on vaguely related reports at https://apple.stackexchange.com/questions/258623/how-to-fix-killed-9-error-in-macos and https://bugzilla.mozilla.org/show_bug.cgi?id=1704126, I'm guessing this has something to do with the way Mozilla signed the Firefox binary, and/or the way Firefox replaced its own binary (e.g. mv instead of cp or something like that)</p>	<p>To address the issue of Firefox 121 being unable to start from the command line on macOS, try running the following command to remove the quarantine attribute: 'xattr -d com.apple.quarantine Applications/Firefox.app'. If this does not resolve the issue, consider downgrading Firefox or checking for updates on the related GitHub issue for any new fixes or workarounds.</p>	<p>The issue you're encountering is likely due to macOS's Notarization and Hardened Runtime requirements, which prevent Firefox from being started from the command line without proper entitlements. To work around this, you can either disable System Integrity Protection (SIP), which is not recommended due to security risks, or use a local build or a try build of Firefox that doesn't enforce these restrictions. For local builds, ensure you follow the instructions for creating a debuggable build and signing local macOS builds in the Firefox developer documentation. [3]</p>	Helpful	Helpful, Comprehensive, Concise	Helpful, Comprehensive	Answer (GPT)	Answer (GPT)	Answer (GPT)	Answer (GPT)	
<p><suppressed_user> Can I enable a logger f'd normally use via 'MOZ_LOG' for a mochitest suite on 'try' so? I just pass in the env somehow?</p>	<p>Yes, with "--env" in the 'mach try' invocation, IRC f'd hope we have doc for this</p>	<p>Yes, you can enable a logger via 'MOZ_LOG' for a mochitest suite on try by setting the environment variable in the task configuration. You can do this by adding "mochazest" options in your 'try' syntax, such as 'try -b do-p-all -u mochitest --env MOZ_LOG=<module>-<level>'.</p>	<p>Yes, you can enable a logger for a mochitest suite on try by setting the MOZ_LOG environment variable. To do this, modify the buildBrowserEnv method in the Python file for the test suite you are interested in and add the necessary environment variable settings. For example, you can add browserEnv["MOZ_LOG"] = "example_logger:3" to set the logging level for example_logger. Additionally, you can set MOZ_LOG_FILE to specify the log file path. This will ensure that the logs are captured and available for download from TreeHerder after the try run.</p>	Helpful, Concise	Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT)	Answer (GPT)	
<p><suppressed_user> My local mozilla-central builds are failing today with an 'error[E0635]: unknown feature 'stdsimd'', on two separate machines, but I don't know what's changed recently to cause this... anyone able to point me in the right direction?</p>	<p>You're using nightly rust, is why</p>	<p>The 'error[E0635]: unknown feature 'stdsimd'' indicates that the 'stdsimd' feature is not recognized. This could be due to recent changes in the Rust compiler or the Mozilla codebase. To resolve this, ensure you are using the latest version of Rust by running 'rustup update', and also update your local mozilla-central repository by running 'mach bootstrap' followed by 'mach build'. If the issue persists, check the Mozilla build documentation or the recent commit history for any related changes.</p>	<p>[4]</p> <p>The error error[E0635]: unknown feature stdsimd typically indicates that the Rust compiler version you're using does not support the stdsimd feature. To resolve this, ensure that your Rust toolchain is up-to-date by running rustup update. Additionally, verify that your rust-toolchain file in the mozilla-central directory specifies the correct Rust version. If the issue persists, you might want to check recent changes in the mozilla central repository or consult the build documentation for any updates regarding Rust dependencies.</p>		Helpful, Comprehensive, Concise		Answer (GPT)	Answer (GPT)	Answer (GPT)	Answer (GPT)	
<p><suppressed_user> Hello guys, I have send a patch (my first here) for review more than two weeks ago but still didn't get any reviews. Is there anything that I did wrong? Should I add more reviewers? Ping people?</p>	<p>It looks like you did the right thing on the patch. You could try pinging the reviewer here or needinfo them in bugzilla. We do also have a gtc-reviewer reviewer group but that's probably not quite the right thing for imageb things.</p>	<p>Hi <suppressed_user>, it's common for reviews to take some time. You can gently ping the reviewers on the Bugzilla bug page or add one or two more relevant reviewers to get attention. Be sure to provide context and express your willingness to make any needed changes.</p>	<p>If your patch has been waiting for a review for more than two weeks, you should follow the "Getting attention" section guidelines. Add a comment to the bug saying "review ping!", check the "Need more information from" box, and add the reviewer's name. If there is still no response within a day or two, you can ask for help on Matrix in the #introduction.mozilla.org or #developers.mozilla.org channels. Additionally, you can tag more reviewers or use review groups like #geckoview-reviewers in Phabricator to notify more people.</p>	Helpful, Comprehensive, Concise		Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	
<p><suppressed_user> I have a problem that's hard to debug -- ./mach test toolkit/components/extensions/test/xpcshell_ext_webRequest_response Body.js -- I get a timeout on the initial test in parallel. On the retry (Retrying tests that failed when run in parallel) to run the xpcshell tests sequentially, I get an assertion. What I really want it to get an rr trace of the assertion, but if I use --debuggerrr, it never times out the initial test, so we never get to the one that fails.</p> <p>Any suggestions?</p> <pre>12345670:01.02 SUITE_START: xpcshell - running 2 tests 0:01.02 TEST_START: xpcshell-e10s.toni: toolkit/components/extensions/test/xpcshell_ext_webRequest_response Body.js 0:01.02 TEST_START: xpcshell-remote.toni: toolkit/components/extensions/test/xpcshell_ext_webRequest_response Body.js 5:01.02 TEST_END: TIMEOUT 5:01.02 INFO xpcshell return code: None 5:01.02 TEST_END: TIMEOUT 5:01.02 INFO xpcshell return code: None</pre> <p>(and then on the retry after a timeout I get the IOThread assertion)</p>	<p>Maybe rr chaos mode helps? Something like --debugger-args="record-chaos"</p>	<p>You can try running the test with "--log-tpid" to get more detailed logs, which might help identify the root cause of the timeout and the assertion failure. Additionally, consider using "rr record" to capture the trace of the initial run, even if it doesn't fail, and then use "rr replay" to debug the subsequent assertion. This way, you might capture more context leading up to the failure.</p>	<p>To debug the issue, try running the test with the --sequential flag to avoid parallel execution, which might help isolate the problem. Additionally, you can use the MOZ_DEBUG_CHILD_PROCESS=1 environment variable to pause the child process and attach a debugger to it. If the test still times out, consider adding more logging to identify where it hangs. If you need an rr trace, ensure the test runs long enough to capture the failure by adjusting timeouts or adding delays. Finally, verify if adding Firefox-asdfrr = browser to the xpcshell.toni manifest resolves any resource access issues.</p>	Helpful, Concise	Helpful, Comprehensive		Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	

Question	Answer (Human)	Answer (GPT)	Answer (GPT/RAG) [1]	Characteristics Answer (Human)	Characteristics Answer (GPT)	Characteristics Answer (GPT/RAG)	Which answer provides the most helpful information?	Which answer provides the most comprehensive information?	Which answer is the most concise without being incomplete?	Which answer do you prefer to see in practice?	Regarding the answer you prefer to see in practice, we encourage you to emphasize all particular aspects that are (ir)relevant in the answer(s) and aspects you think I missed.
<suppressed_user> Does Firefox have that suspended sleeping tab feature and is it app or formerly platform scope code?	Not entirely sure of the latter, but there is browser.tabs.unloadOnLowMemory	Yes, Firefox has a suspended sleeping tab feature, which is primarily implemented at the application level (app scope). This means the feature is handled within the Firefox browser's own codebase rather than the underlying platform.	Yes, Firefox has a suspended sleeping tab feature, which is implemented at the platform level within the browser's codebase. This feature allows the browser to be discarded or suspended to save resources, and it is managed by the browser's session store and tab management components. [8]	Concise	Concise	Helpful, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	I found Answer (GPT) to be confusing, especially with the context of Answer (GPT/RAG). They almost seem contradictory. Though I think Answer (GPT) is thinking of "all of Firefox" as the "application", and perhaps the operating system as the "platform". I like the way that Answer (GPT/RAG) phrased it much more.
<suppressed_user> I would like to look at some WPT test results. How do I find a recent m-c landed version - or similar - in Treeherder?	https://wpt.fyi/results/?labels=experimental&label=master&allred=1 is likely easiest. Our results to the dashboard already come from treeherder. But if you want logs, you could to m-c in treeherder and just click on a recent job like https://treeherder.mozilla.org/jobs?repo=mozilla-central&selectedTaskRun=10K772J2UG6z7CphIRZdGA0	To find recent WPT test results for a mozilla-central (m-c) landed version in Treeherder, go to the Treeherder website, navigate to the Treeherder web interface, select the "central" repository (m-c), and look for recent pushes. You can filter results by using the search bar to look for specific changesets or keywords.	To find recent WPT test results for a mozilla-central (m-c) landed version in Treeherder, go to the Treeherder website, use the search bar to filter for the specific push or revision you're interested in, and look for the "wpt" or "WPT" jobs in the results. You can click on these jobs to view detailed logs and results for the Web Platform Tests. [9]	Helpful, Comprehensive, Concise	Concise	Helpful, Concise	Answer (Human)	Answer (Human)	Answer (Human)	Answer (Human)	I think that having direct links (as long as they are correct links) is always going to be more helpful than plain-word descriptions.
<suppressed_user> How can I reproduce non-unified build failures locally?	If you know the specific directory, the hacky way to avoid a giant rebuild is to edit the moz.build file for the directory where the errors are from UNIFIED_SOURCES to SOURCES, or something like that.	To reproduce non-unified build failures locally, you can add the 'ac_add_options --disable-unified-build' line to your 'mozconfig' file. This will force a non-unified build, allowing you to identify and address the specific failures that occur in this mode.	To reproduce non-unified build failures locally, set the following flag in your mozconfig file: ac_add_options --disable-unified-build. This will configure the build system to compile files individually rather than in unified chunks, helping you identify and debug issues that arise in a non-unified build environment. [10]	Concise	Helpful, Concise	Helpful, Concise	Answer (GPT)	Answer (GPT)	Answer (GPT)	Answer (GPT)	Answer's (GPT/RAG) and (GPT) were similar, but I liked Answer (GPT) most. Hard to choose.
<suppressed_user> Hi everyone, I'm setting up Firefox locally but is it normal that bootstrap.py is taking almost an hour?	Depend to network condition if first fetch the code. Huge code base	Yes, it's normal for 'bootstrap.py' to take a significant amount of time, especially on the first run, as it installs all necessary dependencies and tools for building Firefox. The duration can vary based on your system's performance and internet speed.	Yes, it is normal for bootstrap.py to take a significant amount of time, ranging from 40 minutes to two hours, depending on your internet connection and system performance, as it downloads a lot of data and dependencies required for setting up Firefox. [11]	Concise	Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answers (GPT/RAG) and (GPT) were similar. I suppose I like Answer (GPT/RAG) this time.
<suppressed_user> If I call downWindow.restore() on a minimized window, should I expect a 'activate' event (always, on all desktop platforms)?	Not necessarily. The right event is probably sizemodchange. But on Linux the compositor might decide to in fact not restore the window	Yes, calling 'downWindow.restore()' on a minimized window should trigger an 'activate' event on all desktop platforms where this behavior is consistently supported by the implementation.	Yes, calling downWindow.restore() on a minimized window should trigger an 'activate' event on all desktop platforms, as restoring a window typically brings it back to the foreground and activates it. [12]	Concise	Helpful, Concise	Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG) seems most helpful. Answer (Human) seems to contradict the other two. I'm assuming Answers (GPT/RAG) and (GPT) are correct.
<suppressed_user> Hi everyone. So recently I have run into this problem when I try to pull from the Mozilla build because I am trying to start on a good first bug. My issue is that I try to use 'hg pull' but I get an error saying 'bash: hg.exe: command not found', but when I try to install mercurial with 'pip install Mercurial' it outputs that 'Requirement already satisfied: Mercurial in c:\mozilla-build\python3\lib\site-packages (6.7.1)'. So I have it installed but it is not recognizing it. Am I missing something? Also as I am typing this I remembered that, before, when my PC did recognize that I had Mercurial I tried updating Mercurial but it got interrupted in the middle of the process by my anti-virus. Could that maybe be the reason I am having this issue? Thank you in advance and let me know if this question is better to be asked in another channel.	Use start-shell.bat from mozilla build tool to enter the command to pull. Also, need to navigate to the folder where the code store in mozilla build tool shell first before execute hg pull	It looks like the issue might be with your PATH environment variable not including the directory where 'hg.exe' is located. To fix this, add the directory containing 'hg.exe' to your PATH. You can do this by navigating to System Properties > Environment Variables and then editing the PATH variable to include the path to 'hg.exe'. Additionally, check that your Mercurial installation wasn't corrupted by the interrupted update; you might want to reinstall Mercurial to ensure it's set up correctly.	It sounds like your Mercurial installation is not in your system's PATH. You can resolve this by adding the directory containing hg.exe to your PATH environment variable. Since you mentioned that Mercurial is installed in c:\mozilla-build\python3\lib\site-packages, you should add c:\mozilla-build\python3\Scripts to your PATH. You can do this by running the following command in your terminal: setx PATH "%PATH%;c:\mozilla-build\python3\Scripts". After doing this, restart your terminal and try running hg pull again. If the issue persists, you might want to reinstall Mercurial to ensure the installation is complete and not corrupted by the antivirus interruption. [13]	Concise	Helpful, Concise	Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	
<suppressed_user> I modified the dav1d source and changed how it calculates the amount of logical processors. Is there a way to check that I didn't break core count detection?	That isn't particularly problematic because Firefox selects the number of threads for 'libdav1d' itself, and doesn't use the default but happy to reach a path I need be	You can verify that your modification didn't break core count detection by running tests that specifically check for core count functionality, such as unit tests or integration tests related to processor detection in dav1d. Additionally, you can manually check the core count on different systems and compare the results against the expected values to ensure accuracy.	To verify that you haven't broken the core count detection in dav1d after modifying the source code, you can run the dav1d executable with the --cpu option, which will print out the detected CPU features and core count. Additionally, you can compare the output with the expected core count on your system using tools like lscpu on Linux or wmic cpu get NumberOfLogicalProcessors on Windows. [14]	Helpful, Comprehensive, Concise	Helpful, Concise	Helpful, Concise	Answer (Human)	Answer (Human)	Answer (Human)	Answer (Human)	The key point is that Firefox picks the number of threads, not dav1d. Answer (GPT) is just very generic advice about running tests. Answer (GPT/RAG) is more helpful, with specifics about how to test, but doesn't include the key point that Answer (Human) does.
<suppressed_user> I have been using the mozilla-build shell and I have been pulling in the mozilla-unified folder. Is that the correct folder? I thought that was the cloned repo. Also the error or 'command not found' messages show up when I try to execute 'hg --version' command too so I think their might be an issue in my installation of Mercurial. Also, I should mention that I cannot find the hg.exe file when I type ls in my File Explorer search bar, all that comes up in tmp0jbn4k_o.hg.exe.manifest	Probably you need to reinstall everything by running bootstrap.py again	It seems like your Mercurial installation might be incomplete or corrupted. Ensure you have installed Mercurial properly by following the installation guide: https://www.mercurial-scm.org/wiki/Download . Verify that the 'hg.exe' file exists in your system PATH, and try reinstalling if necessary. The 'mozilla-unified' folder is indeed the correct repository.	It sounds like there might be an issue with your Mercurial installation. The mozilla-unified folder is indeed the correct folder for the cloned repository, but if you're getting 'command not found' errors when trying to execute hg --version, it indicates that Mercurial is not properly installed or not in your system's PATH. First, ensure that Mercurial is installed correctly by reinstalling it. If it is installed, make sure the path to hg.exe is included in your system's PATH environment variable. You can verify this by opening a new command prompt and typing where hg to see if it locates the hg.exe file. If not, add the directory containing hg.exe to your PATH and try again. [15]	Helpful, Concise	Helpful	Helpful, Comprehensive	Answer (Human)	Answer (GPT/RAG)	Answer (Human)	Answer (Human)	Answer (GPT/RAG) has a lot of good information about installing Mercurial, but in practice Mercurial is installed via bootstrap.py so it's easiest to just run that again, hence my preference for Answer (Human). Answer (GPT/RAG) does also have the helpful information that mozilla-unified is the right directory, it would have been nice for Answer (Human) to include that!
<suppressed_user> As someone who's new to mercurial (coming from Git), how would I get a version of the Firefox repository, to build FF from source code, as it was (in the main/central branch) on a certain date?	You can actually use 'git'.	To get a version of the Firefox repository from a specific date using Mercurial, you can use the following command: 'hg update -d "YYYY-MM-DD"'. This will update your local repository to reflect the state of the central branch as it was on the specified date.	To get a version of the Firefox repository as it was on a certain date using Mercurial, you can use the hg update command with the -d option to specify the date. First, clone the repository using hg clone https://hg.mozilla.org/mozilla-central/ firefox-source and navigate into the directory with cd firefox-source. Then, update to the desired date with hg update -d "YYYY-MM-DD", replacing YYYY-MM-DD with the specific date you are interested in. This will update your working directory to the state of the repository as it was on that date. [16]	Helpful	Helpful, Comprehensive, Concise	Helpful, Comprehensive	Answer (GPT)	Answer (GPT/RAG)	Answer (GPT)	Answer (GPT)	Answer (GPT) answers the question without adding the extra (probably unnecessary) information about how to clone the repo like Answer (GPT/RAG) does. Answer (Human) is maybe more helpful from a general sense (if the reader asked would rather use git than mercurial) but doesn't have any details.
<suppressed_user> can I convert nsString to nsCString?	Yes, CopyUTF16toUTF8, or NS_ConvertUTF16toUTF8 is generally the conversion you want	Yes, you can convert an 'nsString' to 'nsCString' in Mozilla Firefox code using the 'Assign' method, like this: nsCString cString; Assign(NS_ConvertUTF16toUTF8(nsString));. This approach ensures proper conversion from UTF-16 to UTF-8 encoding.	Yes, you can convert an nsString (UTF-16) to an nsCString (UTF-8) using the NS_ConvertUTF16toUTF8 helper class. For example, you can do nsCString cString; cString.Assign(NS_ConvertUTF16toUTF8(utf16String)); where utf16String is your nsString instance. [17]	Helpful, Concise	Helpful	Helpful, Comprehensive	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT)	Answer (GPT/RAG)	Answer (GPT/RAG) has the key function name and no other irrelevant information.

Question	Answer (Human)	Answer (GPT)	Answer (GPT/RAG) [1]	Characteristics Answer (Human)	Characteristics Answer (GPT)	Characteristics Answer (GPT/RAG)	Which answer provides the most helpful information?	Which answer provides the most comprehensive information?	Which answer is the most concise without being incomplete?	Which answer do you prefer to see in practice?	Regarding the answer you prefer to see in practice, we encourage you to emphasize all particular aspects that are (ir)relevant in the answer(s) and aspects you think it misses.
<suppressed_user> Do we have instructions somewhere how to get patches/versions from phabtry using git?	Yeah, add a try remote as described in https://github.com/glandium/cinnabar/wiki/Mozilla-A-git-workflow-for-Gecko-development and do git cinnabar fetch try <suppressed_user> && git checkout FETCH_HEAD or equivalent	Yes, you can find instructions for getting patches or versions from Phabricator and Try using Git in the Mozilla Developer Guide(https://mozilla-version-control-tools.readthedocs.io/en/latest/mozilla-view/install-git.html).	Yes, you can find instructions on how to get patches and versions from Phabricator and Try using Git in the Mozilla documentation. Specifically, you can use the moz-phab tool to manage patches. To apply a patch from Phabricator, use the command moz-phab patch D-revision id- or moz-phab patch <URL>. For submitting patches to Try, you can use ./mach try fuzzy to select jobs and push to Try. Detailed steps are available in the Phabricator user guide (https://moz-conduit.readthedocs.io/en/latest/phabricator-user.html).	Helpful, Concise		Helpful	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (Human)	Answer (GPT/RAG)	Answer (GPT/RAG) is helpful except it doesn't cover how to get the try build like Answer (Human) does.
<suppressed_user> How could I use https://firefox-source-docs.mozilla.org/dom/use-counters.html#deprecated-dom-operations-for-a-dictionary-member ?	Do you want to check for presence or for a particular value? <suppressed_user>: not sure if we support [UseCounter] in dict members, but you can always use a manual use counter. See Document::WarnOnceAbout etc	You can use the deprecated DOM operations use counters for a dictionary member by modifying the relevant IDL (Interface Definition Language) file to include the [UseCounter] extended attribute on the dictionary member. This will track its usage in the same way as other deprecated operations, and you can monitor the data collected through the Firefox telemetry system. For more detailed instructions, refer to the [use counters documentation](https://firefox-source-docs.mozilla.org/dom/use-counters.html#deprecated-dom-operations).	To use a deprecated DOM operation as a dictionary member, you need to decide the use counter in nsDeprecatedOperation::list using the DEPRECATED_OPERATION (DeprecationReference) macro, ensuring the parameter matches the deprecation note in the IDL file. After adding the use counter, run ./mach gen-use-counter-metrics and build as normal. This will generate the necessary enum in enum class UseCounter, which you can then use to track the usage of the deprecated operation. For detailed steps, refer to the use-counters documentation. [18]	Helpful, Concise	Helpful	Helpful, Comprehensive	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG) has useful information, although I don't entirely understand if it answers the question; Answer (Human) might be more helpful for this.
<suppressed_user> hi all, I'm working on making some changes to the new tab hover preview feature and I have come up against a strange issue. I'm trying to set up a MutationObserver to listen to changes on other elements in the frontend to decide if the tab preview should be disabled or not, but when I try to set it up (even with a no-op callback function), I get an 'Unexpected NS_ERROR_FAILURE' in the console. There is a javascript traceback pointing to my call, but there doesn't seem to be any way to get a more descriptive error message. Has anyone here run into something like this before?	NS_ERROR_FAILURE is always the least helpful thing. Do you have a patch you can share (e.g. via work-in-progress phabricator upload)? And are you testing on an artifact build or a compiled build? In the latter case you might be able to get somewhere with a C++ debugger like MS Visual Studio or XCode or ghidra (depending on what platform you're on) generally MutationObserver is a web-exposed API so I'd expect any exceptions to come in the form of 'normal' DOM exceptions so what you're seeing is likely a bug but it's difficult to say exactly what's happening without being able to reproduce / more details :-)	Yes, this issue can occur if the target element of the 'MutationObserver' is not properly connected to the DOM or if there are permissions issues. Ensure that the element exists and is fully loaded before setting up the observer. Additionally, verify that no security policies (like Content Security Policy) are blocking the operation. You might also want to check if the element is within a shadow DOM, as this can cause issues with observers if not handled correctly.	It sounds like the "Unexpected NS_ERROR_FAILURE" you're encountering might be due to the MutationObserver being set up on a node that isn't fully available or accessible at the time of the observer's initialization. Ensure that the target node exists and is fully loaded before setting up the observer. You can use document.adoptedStyleSheets (DOMContentLoader...) or window.onload to ensure the DOM is fully loaded before initializing the MutationObserver. Additionally, double-check that the node you're observing is not within a shadow DOM or another context that might restrict access. [20]	Comprehensive, Concise	Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT)	Answer (GPT/RAG)	This [Answer (GPT/RAG)] gives a small code example of what might work to fix the issue and also points out other possible issues.
<suppressed_user> Do we have anything like clock tick (https://searchfox.org/mozilla-central/source/browser/components/websites/webkit0/Discovers/StreamFreeDesktop.js#2850) for xpcshell tests? I see something we can change the time with to test timing / expiration situations	irc siron has it, but it doesn't work for most cases as it can't mock the setTimeout in the js.mjs scope as that's a different setTimeout	Yes, for xpcshell tests, you can use 'Services.prefs.setPref("network.cookie.lifetimePolicy", 2);' and 'Services.prefs.setPref("network.cookie.lifetime.days", <number_of_days>)' to simulate time-based changes, or use 'TakeTimers' provided by 'siron' to manipulate time. Additionally, 'Services.prefs.setPref' is commonly used to adjust time-dependent settings for testing expiration scenarios.	Yes, for xpcshell tests, you can use fakeTimers from the sinon library to manipulate time and test timing/expiration scenarios. You can install sinon via npm and then use sinon.useFakeTimers() to control the clock in your tests. This allows you to advance time programmatically and test how your code behaves with different time intervals. [21]	Helpful, Concise		Helpful, Comprehensive, Concise	Answer (Human)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (Human)	This [Answer (Human)] gives the necessary package but also points out that it probably won't work (though more context in the question is needed).
<suppressed_user> Does anyone know how I can find out who is currently working on Native Tab Grouping? https://connect.mozilla.org/discuss/native-tab-grouping-more-customizable-tab-bar/id/p303 This status was updated to 'In Progress'. I've also been working on this and would love to sync up.	Ask in Firefox Desktop Development, probably they know	You can find out who is currently working on Native Tab Grouping by checking the relevant Bugzilla tickets for the feature and reaching out to the assignees or contributors listed there. Additionally, you can ask in the appropriate Mozilla developer channels or forums, such as the Firefox dev mailing list or the #firefox-desktop Slack channel, for the most up-to-date information on the current developers involved.	To find out who is currently working on Native Tab Grouping, you can check the status and assignees of the relevant bug or feature request in Mozilla's Bugzilla or GitHub repositories. Additionally, you can join the Mozilla community channels, such as the #introduction room on Matrix or the relevant project-specific channels, and ask directly there. This way, you can connect with the developers involved and coordinate your efforts. [22]	Helpful, Concise	Helpful, Comprehensive	Comprehensive	Answer (GPT)	Answer (GPT)	Answer (Human)	Answer (GPT)	Multiple specific sources of the information are given, though it would be more helpful if a specific bug could be provided, of course.
<suppressed_user> If I'm using 'mach run', how can I change log level? Want to get debug logs	I think you can [set the 'MOZ_LOG' environment variable](https://firefox-source-docs.mozilla.org/xpcom/logging.html#enabling-logging-using-environment-variables)	To change the log level when using 'mach run', and get debug logs, set the 'RUST_LOG' environment variable to 'debug' before running the command. For example, you can use 'RUST_LOG=debug mach run'.	To change the log level to debug when using ./mach run, you can set the remote log level preference to Debug by running the following command: % mach run --setpref "remote.log.level=Debug" --remote-debugging-port [23]	Helpful, Comprehensive, Concise	Concise	Concise	Answer (Human)	Answer (Human)	Answer (Human)	Answer (Human)	This [Answer (Human)] is the only correct answer (though admittedly logging is a multi-layered thing).
<suppressed_user> So it occurs to me I perhaps should have come here in the first place but for the past few days I have been attempting to build a new mozilla application largely based on experience with doing the same on a full xul codebase. It has been going surprisingly well save for a few build system bits that are looking for specific applications and don't have a fallback. Specifically, I want to ask if anyone else has come attempting the same in recent years and do you guys think it would be desirable to get ahold of a build system peer or simply file a bug to sort out these couple of things so that the rich technology crammed in the codebase can explode out again into many nifty web and mozilla-enabled projects?	I think fees for those would be worth putting at least on a bug, specially if they have been already written :) Feel free to send the patches the way of the #build reviewer group and cc me or so	If you're facing build system issues with your new Mozilla application, it's advisable to file a bug report and submit patches to the #build reviewer group for review. This approach ensures that any necessary fixes can be tracked and addressed effectively. You can also CC relevant peers to expedite the process.	Yes, it would be beneficial to file a bug report for the build system issues you're encountering, especially if you already have fixes or patches. This will help ensure that the problems are tracked and addressed systematically. Additionally, you can reach out to a build system peer or the #build reviewer group for more immediate assistance and to get your patches reviewed. Including specific details and citing relevant people will facilitate quicker resolution and integration of your contributions. [24]	Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise	Helpful, Comprehensive	Answer (Human)	Answer (GPT)	Answer (Human)	Answer (Human)	
<suppressed_user> anyone know if it's possible to make mozregression use beta builds or mozilla-beta artifacts? We have a bug that only reproduces on beta for some reason, mysteriously, it reproduces on 119 beta, 120, 121, 122, but not on nightly or release builds of any of those versions	would passing in the --repo mozilla-beta work? Maybe check with mozregression.mozilla.org if no one here knows	Yes, it's possible to make mozregression use beta builds by specifying the --repo mozilla-beta flag when you run mozregression. This flag will ensure that mozregression fetches and tests beta builds instead of nightly or release builds.	Yes, you can use MozRegression with beta builds by specifying the --repo mozilla-beta option. This will allow you to track down the regression using flag when you investigate. For example, you can run mozregression --repo mozilla-beta --good 119 --bad 122 to find the regression range within the beta builds. [25]	Helpful, Concise	Helpful, Comprehensive, Concise	Comprehensive	Answer (Human)	Answer (GPT)	Answer (GPT)	Answer (Human)	Aside from giving the solution, this Answer (Human) also gives an avenue for further investigation if that doesn't work (or if there are more related questions wrt mozregression).
<suppressed_user> I have one more question about random generation. https://searchfox.org/mozilla-central/source/libm/RandNum.cpp#89-129 Why does the windows part double-not the RtlGenRandom() function on line 94?	That's just a handy way to convert from a 'a' = false, anything else = 'true' variable to a proper 'bool'	The double negation (!!) applied to the RtlGenRandom() function on line 94 in 'RandNum.cpp' ensures that the result is explicitly converted to a boolean value ('true' or 'false'). This is a common C++ idiom to guarantee that the result of a function returning a non-zero value is interpreted as 'true' and zero as 'false', providing clear and predictable boolean logic.	The double negation (!!) in 'RtlGenRandom()' is used to convert the return value of RtlGenRandom from a BOOLEAN type to a standard bool type in C++. This ensures that the result is either true or false, which is more idiomatic and safer to use in C++ boolean expressions. [26]	Helpful, Concise		Helpful, Comprehensive	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (Human)	Answer (Human)	The question is missing some context e.g. about why the person is asking the question. Guessing at the original cause of the question, the user was perhaps confused about the signatures of 'RtlGenRandom' and the calling function, etc. Given this is a simple question the concise answer and leaving room for follow up questions if the answer missed the mark is probably better. I didn't mark the first question as "Helpful" as I think it's perhaps a bit misleading. In this particular scenario, the '!' is I believe technically unnecessary (as there would be an implicit coercion from BOOLEAN to bool for the return value anyway), but it doesn't hurt anything.

Question	Answer (Human)	Answer (GPT)	Answer (GPT/RAG) [1]	Characteristics Answer (Human)	Characteristics Answer (GPT)	Characteristics Answer (GPT/RAG)	Which answer provides the most helpful information?	Which answer provides the most comprehensive information?	Which answer is the most concise without being incomplete?	Which answer do you prefer to see in practice?	Regarding the answer you prefer to see in practice, we encourage you to emphasize all particular aspects that are (ir)relevant in the answer(s) and aspects you think it missed.
<suppressed_user> ac_add_options – with– branding=“browser/branding/official” is not working when added to top of a mozconfig added to root firefox source directory and then building... any ideas?	Unintuitively, it's –enable-official-branding	Ensure that the 'ac_add_options – with– branding=“browser/branding/official” line is not commented out and placed correctly in the 'mozconfig' file. Additionally, check that all necessary branding files are present and paths are correctly specified. You may also need to run 'mach build' after configuring 'mozconfig' to apply changes properly.	Ensure that the ac_add_options –with-branding=“browser/branding/official” line is placed correctly in your mozconfig file and that the MOZCONFIG environment variable is set to the absolute path of this mozconfig file before running the build commands. Additionally, verify that there are no conflicting branding options elsewhere in the mozconfig file. If the issue persists, try running ‘mach clobber’ to clean the build directory and then rebuild. [27]	Helpful, Comprehensive, Concise			Answer (Human)	Answer (Human)	Answer (Human)	Answer (Human)	Answers other than Answer (Human) missed the reason why this didn't work, which appears to be that the question asker was incorrect about the flag which needs to be used. The other answers were advice for how to debug if you incorrectly configured your mozconfig, which would just waste someone's time in a situation like this where they're using the wrong flag.
<suppressed_user> what are background profiles?	Gonna be honest idk what they do. Something runs in the background and I delete the profile for it. I periodically get errors that it can't find the profile.	Background profiles in Mozilla Firefox are configurations used to collect performance data in the background, without interrupting the user's browsing experience. They help developers identify performance issues by capturing data on how Firefox behaves during regular use.	Background profiles in Mozilla Firefox are specialized user profiles used by background tasks to perform operations without user interaction. These profiles can be either ephemeral, meaning they are temporary and do not persist after the task completes, or non-ephemeral, meaning they are persistent and their data is saved for future use. The configuration and behavior of these profiles are managed through the profiles.ini file, which records the profile locations and ensures that non-ephemeral profiles are reused across background tasks. [28]								Fundamentally my responses here depend on what exactly the user is asking about. A "background profile" probably has multiple meanings in different situations. My answers are assuming that the user meant "background task profile", but perhaps the user meant something else, and a clarifying question would be more useful. I think all 3 answers are bad and/or misleading, though (GPT/RAG) is the “best”. None of them provide the feedback the user probably wants which I'm guessing is roughly: > Firefox occasionally needs to perform tasks, such as profile maintenance or updates, in the background. In some cases, this is done with a headless Firefox instance running with a "background task profile".
<suppressed_user> does anyone have good docs or interesting content about using penmosco to debug memory leaks?	Somebody delivered a recorded tutorial on using Penmosco recently. I don't think it was specific to memory leaks, but maybe that will help. Or do you already know what was covered? That feels like a bug. -suppressed_user- https://mozilla.hotelier.com/Panopto/Pages/Viewer.aspx?d=ae8f5c08-4cd3-4723-85cd-b12d0106d725, just in case.	You can find good documentation and content about using Penmosco to debug memory leaks on the official Penmosco website, particularly in their guides and tutorials section. Additionally, the Mozilla Developer Network (MDN) and Mozilla's own documentation may have useful information and examples related to memory leak debugging with Penmosco.	To debug memory leaks in Firefox using Penmosco, you can start by recording an r trace with AVX-512 disabled (using a script like penmosco-record). Once you have the trace, upload it to Penmosco for processing. Mozilla employees can refer to the Penmosco install and configuration doc for credentials, while contributors can use a free Penmosco account for up to 5 uploads. Detailed instructions for setting up and using Penmosco can be found on the Penmosco website and in the Mozilla documentation. [29]	Helpful, Concise			Answer (Human)	Answer (Human)	Answer (Human)	Answer (Human)	The (GPT/RAG) and (GPT) answers completely missed the question being asked, which wasn't how to use penmosco, but specifically how to use it for debugging memory leaks (which is not generally how penmosco is used as it is a time-travelling debugger). While I don't think answer (Human) is super effective either in terms of giving the contributor advice, referring to other mozilla-specific documentation may be more useful here.
<suppressed_user> Should I use the passive voice in technical documents? or rather... should the passive voice be used in technical documents?	came across this last week https://evanish.com/blog/how-i-edited-infinite-passive-voice-where the authors makes some point against passive voice	In technical documents, it's generally better to use the active voice because it is clearer and more direct, making it easier for readers to understand who is performing the action. However, the passive voice can be appropriate when the focus is on the action or result rather than the doer, or when the doer is unknown or irrelevant.	In technical documents, it's generally better to use the active voice because it makes sentences clearer and more direct. Active voice helps identify who is performing an action, which can reduce ambiguity and improve readability. However, passive voice can be useful in certain contexts, such as when the actor is unknown or irrelevant, or when you want to emphasize the action itself rather than the actor. [30]	Helpful, Concise	Helpful, Comprehensive, Concise	Helpful, Comprehensive	Answer (GPT)	Answer (GPT/RAG)	Answer (GPT)	Answer (Human)	This is a bit of a non-technical question. All answers seem fine. I appreciate the (Human) providing citations backing up the claims. In all of these responses so far, including code references to back up claims would be valuable if this was coming from an AI so that it's possible to check the responses.
<suppressed_user> does anyone know what component is responsible for creating the media key popups on windows? I have a cool idea for a small enhancement, but I'm not sure where to file it.	I think it's windows itself via https://searchfox.org/mozilla-central/source/widget/windows/WidgetSMTCPProvider.cpp	The component responsible for creating the media key popups on Windows in Mozilla Firefox is 'widget/windows/MediaControlPopup.cpp'. You can file your enhancement idea under the "Toolkit" product, specifically in the "Widget: Win32" component on Bugzilla.	The component responsible for creating the media key popups on Windows is the WindowsSMTCPProvider class, which is part of the MediaKeyEventSourceFactory in the Mozilla codebase. You can file your enhancement idea under the "Core:: AudioVideo" component in Bugzilla, specifying that it pertains to the WindowsSMTCPProvider.				Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	For answer (GPT), there is no file named 'MediaControlPopup.cpp'. It appears the AI hallucinated it unless it was removed recently. Answer (Human) doesn't answer "where do I file it", and instead just answers that the popup is displayed by Windows through WindowsSMTCPProvider. Answer (GPT/RAG) gives a component, but it's not clear that it's the correct component (it would probably eventually be triaged to the correct place). Given the asker doesn't appear to be interested in the actual code, and just where to file a bug answer (GPT/RAG) is probably the most useful. For then, though the extra details like "part of the MediaKeyEventSourceFactory" are probably more misleading and will lead to a more confusing bug report. Given this I'm marking none as helpful.
<suppressed_user> is there a way to draw a string on a canvas in vertical mode?	In Firefox, yes: the writing-mode and text-orientation of the canvas element will be used when drawing the string: data:text/html;<suppressed_user>-note that this doesn't work the same in Blink/WebKit browsers, though, so you might not want to depend on it.	Yes, you can draw a string on a canvas in vertical mode by using the 'rotate' method of the canvas context. First, set the rotation to 90 degrees or -90 degrees using 'ctx.rotate(Math.PI / 2)' or 'ctx.rotate(-Math.PI / 2)'. Then use the 'fillText' method to draw the string. Ensure you adjust the position accordingly since the rotation will affect the canvas coordinates.	var canvas = document.getElementById('canvas'); var ctx = canvas.getContext('2d'); ctx.font = "20px Arial"; ctx.save(); ctx.translate(50, 100); // Move the context to the desired position ctx.rotate(Math.PI / 2); // Rotate the context 90 degrees counter-clockwise ctx.fillText("Vertical Text", 0, 0); ctx.restore(); This code will draw the text "Vertical Text" vertically on the canvas. [32]	Helpful, Comprehensive, Concise			Answer (Human)	Answer (Human)	Answer (Human)	Answer (Human)	Only Answer (Human) is correct at all. The others explain how to write rotated text, which is not the same at all.
<suppressed_user> Hello, I have a stack of two commits, one is landing right now so the phab issue is closed. I want to update the second one with "moz-phab" but it complains that "You can not request review of this revision because it has already been closed. You can only request review of open revisions: -- is there a trick to push just the second one until things gets in sync? I have not found the right combo.	Have you tried -single	To update the second commit with 'moz-phab', you should rebase the second commit on top of the main branch after the first commit has landed. This can be done by running 'hg rebase -d default' (or the appropriate branch name) from the second commit, and then use 'moz-phab submit' to push the updated commit for review.	To update the second commit with moz-phab, you should create a new revision for the second commit since the original one is already closed. You can do this by using the moz-phab submit command again, which will create a new revision for the second commit. Ensure that your working directory is clean and only contains the changes for the second commit before running the command. [33]	Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise		Answer (Human)	Answer (Human)	Answer (GPT)	Answer (Human)	-single is the answer. Waiting to rebase would also work, but would have to wait until the patch lands on central and require a rebase.
<suppressed_user> Could someone point me to canonical docs for the review request syntax that phabricator uses? I.e. "r?adant" vs "r?adant" etc. Not looking for answers about these specific examples, but a more general set of docs that I can bookmark and/or share with others.	There's a section in the doc : https://mozilla-version-control-tools.readthedocs.io/en/latest/development/contributing.html#commit-creation-guidelines	You can find the canonical documentation for Phabricator's review request syntax on Mozilla's Phabricator User Guide page, specifically under the section "Requesting Code Reviews" which explains the different notations like "r? user", "muser", and "mrelay". Here's the link: [Phabricator User Guide: Requesting Code Reviews]https://moz-conduit.readthedocs.io/en/latest/phabricator-user.html#requesting-code-reviews.	To update the canonical documentation for the review request syntax used in Phabricator in the Mozilla Conduit documentation, specifically under the section for commit message syntax (https://moz-conduit.readthedocs.io/en/latest/phabricator-user.html#commit-message-syntax). This section explains the different syntaxes for requesting reviews, including single reviewers (r=reviewer), multiple reviewers (r=reviewer1,reviewer2), blocking reviews (r=reviewer), and review groups (r!=review-group). [34]	Helpful, Comprehensive		Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG) gives correct information but a dead link that points to a section that doesn't exist. Answer (Human) points to a link that works, but doesn't explain the difference. Answer (GPT) hallucinates the link and doesn't provide the information

Question	Answer (Human)	Answer (GPT)	Answer (GPT/RAG) [1]	Characteristics Answer (Human)	Characteristics Answer (GPT)	Characteristics Answer (GPT/RAG)	Which answer provides the most helpful information?	Which answer provides the most comprehensive information?	Which answer is the most concise without being incomplete?	Which answer do you prefer to see in practice?	Regarding the answer you prefer to see in practice, we encourage you to emphasize all particular aspects that are (re)relevant in the answer(s) and aspects you think it missed.
<suppressed_user> was able to compile but the long time build firefox is slower than the firefox produced with artifact build 7 how can i enable optimization for performance on speedometer 2.1 ?	For speedometer you probably want an -enable-release build with PQO See the 'shippable' mozconfigs in the tree	To enable optimization for performance on Speedometer 2.1 in your Firefox build, you should ensure that you're compiling with optimization flags. Use -mach build -enable-release to build with optimizations for speed. Additionally, make sure that you are not using debug builds, as they are not optimized for performance.	To optimize performance for Speedometer 2.1, ensure you are using an optimized build by adding ac_add_options -enable-optimize to your mozconfig file. Additionally, enable Profile Guided Optimization (PGO) by adding ac_add_options MOZ_PGO=1, and set the rust optimization level to 2 with ac_add_options RUSTC_OPT_LEVEL=2. These settings will help produce a build that is more performant and closer to release builds. [35]	Helpful, Comprehensive, Concise		Comprehensive	Answer (Human)	Answer (Human)	Answer (Human)	Answer (Human)	The right Answer is (Human). Answer (GPT) just does something that doesn't exist. Answer (GPT/RAG) almost gets it right, but doesn't - enable-release or anything like that. So you really want to look at the shippable mozconfigs in the tree :)
<suppressed_user> could you expand on what 'stand-in' means in color handling in LookAndFeel files? from [this comment in Phab](https://phabricator.services.mozilla.com/D19737#diff-1098513) - I'm still learning the platform side of this code and would appreciate a hint	It means roughly 'hardcoded, non-platform colors' they were originally for But now we use them also both for the non-native-theme, and as a fallback when the native theme doesn't have relevant colors (windows in dark mode for example)	In the context of color handling in LookAndFeel files, 'stand-in' refers to a temporary or placeholder value used until the actual color value is available or can be determined. This is typically employed during the initialization phase or when a color value is being fetched asynchronously and ensures that the application can continue to function without interruptions while waiting for the definitive color value.		Helpful, Comprehensive, Concise			Answer (Human)	Answer (Human)	Answer (Human)	Answer (Human)	The AI answers are rather garbage on this one, just wrong info in both cases.
<suppressed_user> Hey - off hand, do you remember if we have a way for privileged JS in a content process to access the WindowGlobalParent of a subframe so that it can send messages down to an actor in that subframe? Or do all messages need to be marshalled through the parent process?	All 'WindowGlobalParent' exist in the parent process So there is no 'WindowGlobalParent' in a content process You need to get the parent process involved	Privileged JS in a content process cannot directly access the 'WindowGlobalParent' of a subframe; all messages need to be marshalled through the parent process to communicate with an actor in the subframe.	In the Firefox architecture, privileged JS in a content process cannot directly access the WindowGlobalParent of a subframe to send messages. Instead, all messages need to be marshalled through the parent process to the JSWindowActor framework, which facilitates communication between the parent and child processes via IPC actors. [37]	Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	It would've been good to point out like Answer (Human) did, that all WindowGlobalParent objects are parent-process-specific. However Answer (GPT/RAG) is probably a better answer over-all.
<suppressed_user> When does an assertion like 'NS_WARN_IF(...)' throw rather than just warn?	Never	NS_WARN_IF(...) in Mozilla Firefox code does not throw. It only logs a warning message if the condition inside the parentheses is true.	The NS_WARN_IF macro does not throw an exception; it only issues a warning in debug builds if the condition is true. If you need to handle the error condition, you should follow it with appropriate error handling code, such as returning an error code. [38]	Concise	Comprehensive, Concise	Helpful, Comprehensive	Answer (GPT/RAG)	Answer (GPT)	Answer (GPT)	Answer (GPT)	
<suppressed_user> Out of curiosity, is it possible to target the specific commit for the latest release build, eg- 124.0.1, yet not have it build as a release configuration? I'm just trying to think of ways to get around this build issue at the moment as a temporary measure. I do need some C++ work though, so unfortunately artifact builds are out of the question.	Yes, you can build however you want, even a debug build	Yes, you can target a specific commit for the latest release build (e.g., 124.0.1) and build it without the release configuration by using the following steps: checkout the desired commit from the repository, then configure the build options to disable the release configuration. This can be done by setting appropriate build flags or modifying the build configuration files to match your requirements, ensuring that you avoid the release-specific optimizations and settings.	Yes, you can target a specific commit for the latest release build, such as 124.0.1, without building it as a release configuration by checking out the specific commit using git checkout -b 'commit-hash' and then modifying the build configuration to a non-release mode, such as debug. This way, you can work on the C++ code without the constraints of a release build. [39]	Concise	Comprehensive	Comprehensive	Answer (GPT)	Answer (GPT)	Answer (GPT/RAG)	Answer (GPT)	
<suppressed_user> Does anyone know if we encode the hg revision into local builds at all, in a way that can be discovered? (I know it's on about: buildconfig for official builds, but we don't show the revision there in local builds)	You could try looking for the implementation of one of these maybe? https://searchfox.org/mozilla-central/search?q=buildid&path=diffCase=false&ex=	Yes, in local builds, the Mercurial (hg) revision is encoded into the build and can be discovered by running -mach build-backend - backend=RecursiveKleiser which generates a 'buildid.h' file containing the revision information.	Yes, the hg revision is encoded into local builds and can be discovered. During the build process, a script writes build information, including the commit hash and commit date, to a file named build-info.h located in the output directory (\$OUT_DIR). This file is included in the program during compilation, allowing you to access the commit hash and date in the compiled binary. [40]		Helpful, Comprehensive, Concise	Helpful, Comprehensive	Answer (GPT)	Answer (GPT)	Answer (GPT/RAG)	Answer (GPT)	
<suppressed_user> Hi folks, I'm setting up a new macbook, and had no luck getting hg to pull mozilla-unified. bootstrap.py timed out, so I ran 'hg pull https://hg.mozilla.org/mozilla-unified -debug' twice, and each time it 'got stuck on' adding xulrunner/toolbar/redirect.cpp revisions' and sat there until it timed out. (using git seems to have worked fine). Just wondering if that's a sign of a problem with the hg repo?	Yes, I've encountered exactly the same issue on a Linux machine. I also found the same issue when trying to use a mercurial bundle (rather than cloning the repo). Curiously, it only seemed to happen with a 'compressed' bundle - the 'uncompressed' one worked fine.	It sounds like you might be experiencing network issues or a problem specific to that file in the Mercurial repository. You could try pulling the repository from a different network or using a different internet connection. Additionally, running 'hg pull' with the -config flag to increase the timeout limit or using a mirror of the repository might help resolve the issue.	It sounds like your network connection might be unstable, causing the hg pull command to time out. To work around this, you can download a Mercurial bundle of the mozilla-unified repository from https://hg.cdn.mozilla.net/ (select the 'zstd (max)' column for mozilla-unified). After downloading, initialize a new repository with hg init mozilla-unified, move the bundle file next to the mozilla-unified directory, rename it to bundle.hg, and run hg unbundle - bundle.hg inside the mozilla-unified directory. This method allows you to resume the download if interrupted and should help you avoid the timeout issue. [41]	Helpful		Helpful, Comprehensive	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT)	Answer (GPT/RAG)	
<suppressed_user> Why is the installer after .mach package in installsea? There's nothing else in install, why not just place it in install?	SeaMonkey and Netscape legacy it is an interesting little artifact of a time when when I was your age give or take I'm getting old... so according to research I can only barely remember conducting I seem to recall that is distrib/installsea was the location for the xpinstall based installer the real one not the .jam, and well tooling is a pain to update so when toolkit happened and switch to raise they just had it spit the installer out into the same place. that way the upload scripts work, but like I said this is a conclusion based on research I did years ago to decide if that actually had any technical requirements outside mozilla so I could achange it I have an unhealthy fascination with the full suite era xpinstall technology I think xpinstall was all but relegated to the add-ons manager duties but still existed in cpo code in the 1.9.1 era which is the version I first compiled and tinkered with, so I only ever got the truest tastes of it in a developer context to this day still want more, id rust xpinstall, might be one of the few things to learn rust for with any priority over more advanced cpo. hmmmmmmmmmmmm maybe one day	The installer is placed in the 'installsea' directory after a .mach package to maintain a clear and organized directory structure, separating different types of installation artifacts. This separation helps in managing the build process more effectively by categorizing related files and scripts in specific subdirectories, even if the 'install' directory itself is currently sparsely populated.	The installer is placed in the installsea directory after running .mach package to maintain a clear and organized structure, especially since the sea directory is specifically designated for self-extracting archives. This separation helps in managing different types of build artifacts and ensures that the directory structure remains consistent and easy to navigate, particularly when dealing with multiple types of installers and other build outputs. [42]		Helpful, Comprehensive, Concise	Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT)	Answer (GPT/RAG)	

Question	Answer (Human)	Answer (GPT)	Answer (GPT/RAG) [1]	Characteristics Answer (Human)	Characteristics Answer (GPT)	Characteristics Answer (GPT/RAG)	Which answer provides the most helpful information?	Which answer provides the most comprehensive information?	Which answer is the most concise without being incomplete?	Which answer do you prefer to see in practice?	Regarding the answer you prefer to see in practice, we encourage you to emphasize all particular aspects that are (ir)relevant in the answer(s) and aspects you think it missed.
<suppressed_user> In JavaScript, how can I know in which process I'm running it?	Services.appinfo.processID? Or processType Depending on what you want	<p>You can determine which process you are running in JavaScript by checking the type of the global 'window' object. In the main process (chrome), the 'window' object is not defined, whereas in a content process (tab), the 'window' object is defined. Use the following code to check:</p> <pre>--javascript if (typeof window === "undefined") { console.log("Running in the main process"); } else { console.log("Running in a content process"); } }</pre>	<p>In JavaScript, particularly within the context of Mozilla Firefox development, you can determine the process type by checking the Services.appinfo.processType property. If it equals CiNativeXULRuntime, you are in the parent process. Otherwise, you are in a child process. Here's a simple function to check:</p> <pre>function isParentProcess() { return Services.appinfo.processType === CiNativeXULRuntime.PROCESS_TYPE_DEFAULT; }</pre> <p>This function returns true if you are in the parent process and false if you are in a child process. [43]</p>	Concise		Helpful, Comprehensive	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (Human)	Answer (GPT/RAG)	
<suppressed_user> Wenn showing function executions in permosco, how can I print the value of arguments while following pointers?	Clic on the value and it will show the data flow to this value, backwards, in the entire execution.	<p>In Permosco, you can print the value of arguments while following pointers by using the "inspect" command. Navigate to the function call of interest, click on the function name to bring up the execution view, and then use the "inspect" command on the arguments, which allows you to dereference and follow pointers to see their values.</p>	<p>To print the value of arguments while following pointers in Permosco, you can use the print command in the debugger console. For example, if you have a pointer arg and you want to see the value it points to, you can use print *arg. If arg is a pointer to a pointer, you can use print **arg to follow the pointer chain. Additionally, you can use the pttype command to print the type of the variable, which can help you understand the structure of the data you are inspecting. [44]</p>								
<suppressed_user> Does anyone have opinions on the value of the Poison/Interposer on Mac? It is disabled on Apple Silicon and disabling it on x64 Mac should allow us to turn on some additional security hardening. It is something we don't enable on Release. I'm trying to understand the value better. Also, why it's called 'Poison'?	I think the basic idea is to have a line in the sand during XPCOM shutdown where we don't do any I/O after that, so we can do an exit without worrying about something getting saved to disk. I feel like the project has been stalled out for many years. I don't remember if we actually do the quick exit or not. I don't know if it is doing anything useful on MacOS or not. As a practical matter, if it is already not enabled on Apple Silicon surely that is a sign we don't care too much about it..	<p>The Poison/Interposer on Mac is used to detect and prevent illegal I/O operations in the Firefox process. It's valuable for catching bugs during development but is disabled on Apple Silicon and Release builds for performance reasons. Disabling it on x64 Mac could allow for additional security hardening. The name "Poison" reflects its function of identifying and stopping improper operations, much like a poison would neutralize a threat.</p>	<p>The Poison/Interposer on Mac is designed to detect and report unintended I/O operations that could lead to performance issues or security vulnerabilities. It is called "Poison" because it intercepts (or intercepts) I/O operations to identify and "poison" (mark or block) unsafe or unintended access patterns, leading to a controlled crash or alert. Disabling it on x64 Mac could allow for additional security hardening, but it is not enabled on Release builds due to potential performance impacts. The value lies in its ability to catch and diagnose problematic I/O operations during development and testing phases. [45]</p>		Helpful, Comprehensive, Concise	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	Answer (GPT/RAG)	<p>This is some arcane code that I've looked at maybe 5 years ago, but I think Answer (GPT/RAG) sounds the most correct. Answer (Human) is too informal, but does sound like a Mozillaian chatting about something. Answer (GPT) sounds confident but I think it's wrong, but it's been awhile since I've read this code, and it's not documented in the header file.</p>	
<suppressed_user> Hi! Does anybody know how is extensionProcessScriptLoader.js injected into the Content Process?	Via this interface: https://searchfox.org/mozilla-central/source/toolkit/components/extensions/mozextExtensionProcessScriptId	<p>'extensionProcessScriptLoader.js' is injected into the Content Process using the XPCOM component and the 'nsIContentProcess' interface. This is achieved by registering the script in the appropriate manifest file and ensuring that the script is loaded during the initialization phase of the content process, typically through the use of the 'Services.moz.loadFrameScript' function.</p>	<p>The extensionProcessScriptLoader.js is injected into the content process via the ExtensionProcessScript module, which is imported and utilized within the content process scripts. This module is responsible for managing the loading and execution of extension content scripts, ensuring they are preloaded and executed at the appropriate stages of document loading. The injection mechanism typically involves the use of ChromeUtils.importESModule to load the necessary scripts and then hook into the content script execution process to monitor and log script injections, as demonstrated in the provided test setup. [46]</p>	Concise						<p>These responses feel like LLM bullshit to me. It looks like the ParentProcessMessageManager is doing this work via Services.pwm.loadProcessScript. This is not code I've read before, but this would be my intuition of the correct answer. I'm not sure if one of these is a real answer from a human, but it reads like plausible LLM content with incorrect content.</p> <p>https://searchfox.org/mozilla-central/source/toolkit/mozapps/extensions/AddOnManager.sys.mjs#106-111</p>	
<suppressed_user> Is there any really simple place to store an int that isn't prefs? I want to store a timestamp preferably with more than seconds precision (so I don't want to just /1000 it)	No, there isn't. content prefs get blown away for permanent private browsing / 'don't remember history', right?	<p>You can store the timestamp in the 'nsIXULRuntime' service using the 'getServiceDeveloperMode' method, which provides a place to store and retrieve custom data without using prefs. This method allows you to store a double, which can handle high precision timestamps.</p>	<p>SharedPreferences prefs = context.getSharedPreferences("MyPrefs", Context.MODE_PRIVATE); SharedPreferences.Editor editor = prefs.edit(); editor.putLong("timestamp", System.currentTimeMillis()); editor.apply();</p> <p>To retrieve the timestamp:</p> <pre>long timestamp = prefs.getLong("timestamp", 0);</pre> <p>This method avoids dividing by 1000 and retains millisecond precision. [47]</p>							They are all terrible responses.	
<suppressed_user> Someone told me about a try flag for only generating builds and not running any tests. But how I can't find it. Anyone?	There's probably a preset of sorts but I just usually do much try fuzzy and select the builds I want	<p>You can use the '-b do' flag with 'mach try' to generate builds without running any tests. For example, you can run 'mach try -b do' to achieve this.</p>	<p>You can use the SharedPreferences in Android to store a long value representing the timestamp in milliseconds, which provides more precision than seconds. Here's a simple example:</p>								They are all wrong. The first is using outdated syntax, the second is wishy washy, and maybe correct enough to solve the problem. The 3rd has a real flag, but it isn't the problem the user is actually having.
<suppressed_user> I don't think git-cinnabar supports partial checkouts does it?	Unsure!	<p>No, git-cinnabar does not support partial checkouts. It is designed to facilitate interaction with Mercurial repositories using Git commands, but partial checkouts, which allow you to clone only a subset of files or directories, are not a feature it provides.</p>	<p>No, git-cinnabar does not support partial checkouts. It is designed to work with full checkouts of Mercurial repositories using Git, and does not have the capability to handle sparse or partial checkouts. [48]</p>		Helpful, Comprehensive, Concise	Helpful, Concise	Answer (GPT)	Answer (GPT)	Answer (GPT)	Answer (GPT)	I don't know if this answer is correct or not, as I've never had to deal with it.