drm-intel Maintainer Guidelines

This document describes the detailed maintainer tasks for drm-intel.

Maintaining drm-intel Branches

The guide assumes a maintainer rotation of one person handling features and fixes for one release. Thus for each branch, you take over from the person maintaining the branch before you, and leave it in a known state for the person after you.

drm-intel-next-queued

Take over when the last drm-intel feature pull has been sent for an upcoming merge window, and drm-intel-next-queued starts targeting the next merge window after that. This happens around -rc5 of the current development kernel.

Tag drm-intel-next-queued periodically, say every 1-2 weeks, depending on the rate of change. Use `dim update-next`. Write a tag summary detailing the changes since the last tag. Send a testing request.

Send drm-intel-next-queued pull requests periodically. Use `dim pull-request-next`. Pull requests are based on the tags generated above. Don't send a pull request immediately after tagging, leave some time for testing. Typically you would send a pull request for each tag, but the tooling allows for accumulation of several tags into one pull request. Dave usually doesn't start pulling features until the pull requests for the previous release have been merged to Linus' tree, so it's normal to accumulate several tags before and during the merge window. Only send your first pull request after -rc1.

Do backmerges as needed, but also don't let drm-intel-next-queued fall too much behind from drm-next; use your discretion. Use `dim backmerge`. Never rebase or force push. Only do backmerges from drm-next. Specifically, don't merge drm-misc

branches or backmerge Linus' tree directly; they all need to go through drm-next. Request drm-misc maintainers to have your dependencies sent to drm-next, and request Dave to have Linus' tree backmerged to drm-next.

Pull in gvt feature pull requests targeting -next. Use `dim apply-pull`.

Usually committers handle most patches, but look for the patches that fall through the cracks otherwise. Ping for reviews, and push reviewed patches.

Track the patches being committed, ensuring the merge criteria and the rest of the process is being properly followed.

Remind people about the last feature pull deadline (-rc5 time frame), especially when there are known features aiming for a release. It's not the maintainer's job to ensure deadlines are met, but to raise awareness. Avoid surprising people.

When sending the last feature pull request, pass the torch to the next in rotation to maintain drm-intel-next-queued. Move on to drm-intel-next-fixes to follow through with the features you handled.

drm-intel-next-fixes

After the last drm-intel feature pull request has been merged to drm-next, somewhere around -rc6 or -rc7 time frame, rebase drm-intel-next-fixes on top of drm-next, and push. Use `dim rebase`.

Cherry pick fixes from drm-intel-next-queued periodically. Use `dim cherry-pick-next-fixes`. The cherry-picks are automated based on Fixes: and Cc: stable tags. Double check that they make sense. Use `dim push-next-fixes` to push. Stop cherry-picking during the merge window, and only pick the fixes that really need to make it to -rc1.

Note: `dim cherry-pick-next-fixes` and `dim cherry-pick-fixes` (see below) use the same logic to pick patches. The former catches everything that the latter catches. Both skip commits that have already been picked. Basically there's a race condition where a commit could be picked for next-fixes when it should really be headed for fixes. Communicate with the current -fixes maintainer.

Send drm-intel-next-fixes pull requests as needed. Check the CI results, and use `dim pull-request-next-fixes`. The fixes will be merged to drm-next, on top of the features heading for the next merge window.

You can send the last pull also during the merge window, even after the main drm pull request has been sent to Linus. Either there will be a follow-up fixes pull, or it will be sent and merged after -rc1 as fixes.

Pull in the (relatively rare) gvt next-fixes pull requests. Use `dim apply-pull`.

Do occasional fast-forward rebases on top of drm-next as needed.

Make sure everything you've picked has been merged upstream. If not, request Dave to merge the missing pull requests to drm-fixes after -rc1.

After -rc1, move on to drm-intel-fixes.

drm-intel-fixes

When you take over, drm-intel-fixes should be at the latest upstream kernel release tag. After -rc1, rebase drm-intel-fixes on top of it. Rebase on the tags only, don't rebase on arbitrary commits in Linus' master. Use `dim rebase drm-intel-fixes vX.Y-rc1`.

Adopt a weekly routine for -rc fixes.

On Monday, if the the last fixes pull has been merged to Linus' tree, rebase drm-intel-fixes on top as a fast-forward rebase. Again, use `dim rebase drm-intel-fixes vX.Y-rcN`. Skip or abort the rebase if it's not a fast-forward rebase, i.e. some of the cherry-picks have not been merged upstream. Try again next week.

After the rebase, give CI time to run the plain upstream tag, to detect issues introduced by Linus' upstream.

Before Wednesday, cherry pick fixes from drm-intel-next-queued. Use `dim cherry-pick-fixes`. The cherry-picks are automated based on Fixes: and Cc: stable tags. Double check that they make sense. Be more and more critical toward

the higher -rc, nearing stable kernel rules for commits, and drop commits that are not needed using `git rebase -i`. Remember, this is the fastest path to Linus' upstream. Use `dim push-fixes` to push.

If cherry-picks fail, look into the failures, and see if the commits are needed. Send mail to intel-gfx and the authors to ask for help and/or backports as needed.

Pull in the gvt fixes pull requests. Use `dim apply-pull`.

Again, let CI crunch through the branch. If everything looks good, send the fixes pull request Thursday at the latest. Due to time zones, a pull request sent on Friday might not make it, missing the Sunday -rc. Use `dim pull-request-fixes`.

After vX.Y release, rebase drm-intel-fixes one last time on that. Use `dim rebase drm-intel-fixes vX.Y`.

It's time to take a break from maintaining branches, until you start over with drm-intel-next-queued.