

Maintainer Entry Profile

The Maintainer Entry Profile supplements the top-level process documents (submitting-patches, submitting drivers...) with subsystem/device-driver-local customs as well as details about the patch submission life-cycle. A contributor uses this document to level set their expectations and avoid common mistakes; maintainers may use these profiles to look across subsystems for opportunities to converge on common practices.

Overview

Provide an introduction to how the subsystem operates. While MAINTAINERS tells the contributor where to send patches for which files, it does not convey other subsystem-local infrastructure and mechanisms that aid development.

Example questions to consider:

- Are there notifications when patches are applied to the local tree, or merged upstream?
- Does the subsystem have a patchwork instance? Are patchwork state changes notified?
- Any bots or CI infrastructure that watches the list, or automated testing feedback that the subsystem uses to gate acceptance?
- Git branches that are pulled into -next?
- What branch should contributors submit against?
- Links to any other Maintainer Entry Profiles? For example a device-driver may point to an entry for its parent subsystem. This makes the contributor aware of obligations a maintainer may have for other maintainers in the submission chain.

Submit Checklist Addendum

List mandatory and advisory criteria, beyond the common "submit-checklist", for a patch to be considered healthy enough for maintainer attention. For example: "pass checkpatch.pl with no errors, or warning. Pass the unit test detailed at \$URI".

The Submit Checklist Addendum can also include details about the status of related hardware specifications. For example, does the subsystem require published specifications at a certain revision before patches will be considered.

Key Cycle Dates

One of the common misunderstandings of submitters is that patches can be sent at any time before the merge window closes and can still be considered for the next -rc1. The reality is that most patches need to be settled in soaking in linux-next in advance of the merge window opening. Clarify for the submitter the key dates (in terms of -rc release week) that patches might be considered for merging and when patches need to wait for the next -rc. At a minimum:

- Last -rc for new feature submissions: New feature submissions targeting the next merge window should have their first posting for consideration before this point. Patches that are submitted after this point should be clear that they are targeting the NEXT+1 merge window, or should come with sufficient justification why they should be considered on an expedited schedule. A general guideline is to set expectation with contributors that new feature submissions should appear before -rc5.
- Last -rc to merge features: Deadline for merge decisions Indicate to contributors the point at which an as yet un-applied patch set will need to wait for the NEXT+1 merge window. Of course there is no obligation to ever accept any given patchset, but if the review has not concluded by this point the expectation is the contributor should wait and resubmit for the following merge window.

Optional:

- First -rc at which the development baseline branch, listed in the overview section, should be considered ready for new submissions.

Review Cadence

One of the largest sources of contributor angst is how soon to ping after a patchset has been posted without receiving any feedback. In addition to specifying how long to wait before a resubmission this section can also indicate a preferred style of update like, resend the full series, or privately send a reminder email. This section might also list how review works for this code area and methods to get feedback that are not directly from the maintainer.

Existing profiles

For now, existing maintainer profiles are listed here; we will likely want to do something different in the near future.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\maintainer\linux-master) (Documentation) (maintainer) maintainer-entry-profile.rst, line 99)
```

Unknown directive type "toctree".

```
.. toctree::  
    :maxdepth: 1  
  
    ../doc-guide/maintainer-profile  
    ../nvdimm/maintainer-entry-profile  
    ../riscv/patch-acceptance  
    ../driver-api/media/maintainer-entry-profile  
    ../driver-api/vfio-pci-device-specific-driver-acceptance
```