# LIBNVDIMM Maintainer Entry Profile

## Overview

The libnvdimm subsystem manages persistent memory across multiple architectures. The mailing list is tracked by patchwork here: https://patchwork.kernel.org/project/linux-nvdimm/list/ ...and that instance is configured to give feedback to submitters on patch acceptance and upstream merge. Patches are merged to either the 'libnvdimm-fixes' or 'libnvdimm-for-next' branch. Those branches are available here: https://git.kernel.org/pub/scm/linux/kernel/git/nvdimm/nvdimm.git/

In general patches can be submitted against the latest -rc; however, if the incoming code change is dependent on other pending changes then the patch should be based on the libnvdimm-for-next branch. However, since persistent memory sits at the intersection of storage and memory there are cases where patches are more suitable to be merged through a Filesystem or the Memory Management tree. When in doubt copy the nvdimm list and the maintainers will help route.

Submissions will be exposed to the kbuild robot for compile regression testing. It helps to get a success notification from that infrastructure before submitting, but it is not required.

## Submit Checklist Addendum

There are unit tests for the subsystem via the ndctl utility: https://github.com/pmem/ndctl Those tests need to be passed before the patches go upstream, but not necessarily before initial posting. Contact the list if you need help getting the test environment set up.

### ACPI Device Specific Methods (_DSM)

Before patches enabling a new _DSM family will be considered, it must be assigned a format-interface-code from the NVDIMM Sub-team of the ACPI Specification Working Group. In general, the stance of the subsystem is to push back on the proliferation of NVDIMM command sets, so do strongly consider implementing support for an existing command set. See drivers/acpi/nfit/nfit.h for the set of supported command sets.

## Key Cycle Dates

New submissions can be sent at any time, but if they intend to hit the next merge window they should be sent before -rc4, and ideally stabilized in the libnvdimm-for-next branch by -rc6. Of course if a patch set requires more than 2 weeks of review, -rc4 is already too late and some patches may require multiple development cycles to review.

## Review Cadence

In general, please wait up to one week before pinging for feedback. A private mail reminder is preferred. Alternatively ask for other developers that have Reviewed-by tags for libnvdimm changes to take a look and offer their opinion.