

# LPFC Driver Release Notes

## Important

Starting in the 8.0.17 release, the driver began to be targeted strictly toward the upstream kernel. As such, we removed `#ifdefs` for older kernels (pre 2.6.10). The 8.0.16 release should be used if the driver is to be run on one of the older kernels.

The proposed modifications to the transport layer for FC remote ports and extended attribute support is now part of the upstream kernel as of 2.6.12. We no longer need to provide patches for this support, nor a *full* version which has old and new kernel support.

The driver now requires a 2.6.12 (if pre-release, 2.6.12-rc1) or later kernel.

Please heed these dependencies....

The following information is provided for additional background on the history of the driver as we push for upstream acceptance.

Cable pull and temporary device Loss:

In older revisions of the lpfc driver, the driver internally queued i/o received from the midlayer. In the cases where a cable was pulled, link jitter, or a device temporarily loses connectivity (due to its cable being removed, a switch rebooting, or a device reboot), the driver could hide the disappearance of the device from the midlayer. I/O's issued to the LLDD would simply be queued for a short duration, allowing the device to reappear or link come back alive, with no inadvertent side effects to the system. If the driver did not hide these conditions, i/o would be errored by the driver, the mid-layer would exhaust its retries, and the device would be taken offline. Manual intervention would be required to re-enable the device.

The community supporting kernel.org has driven an effort to remove internal queuing from all LLDDs. The philosophy is that internal queuing is unnecessary as the block layer already performs the queuing. Removing the queues from the LLDD makes a more predictable and more simple LLDD.

As a potential new addition to kernel.org, the 8.x driver was asked to have all internal queuing removed. Emulex complied with this request. In explaining the impacts of this change, Emulex has worked with the community in modifying the behavior of the SCSI midlayer so that SCSI devices can be temporarily suspended while transport events (such as those described) can occur.

The proposed patch was posted to the linux-scsi mailing list. The patch is contained in the 2.6.10-rc2 (and later) patch kits. As such, this patch is part of the standard 2.6.10 kernel.

By default, the driver expects the patches for block/unblock interfaces to be present in the kernel. No `#define` needs to be set to enable support.

## Kernel Support

This source package is targeted for the upstream kernel only. (See notes at the top of this file). It relies on interfaces that are slowly migrating into the kernel.org kernel.

At this time, the driver requires the 2.6.12 (if pre-release, 2.6.12-rc1) kernel.

If a driver is needed for older kernels please utilize the 8.0.16 driver sources.

## Patches

Thankfully, at this time, patches are not needed.