

Firmware cache

When Linux resumes from suspend some device drivers require firmware lookups to re-initialize devices. During resume there may be a period of time during which firmware lookups are not possible, during this short period of time firmware requests will fail. Time is of essence though, and delaying drivers to wait for the root filesystem for firmware delays user experience with device functionality. In order to support these requirements the firmware infrastructure implements a firmware cache for device drivers for most API calls, automatically behind the scenes.

The firmware cache makes using certain firmware API calls safe during a device driver's suspend and resume callback. Users of these API calls needn't cache the firmware by themselves for dealing with firmware loss during system resume.

The firmware cache works by requesting for firmware prior to suspend and caching it in memory. Upon resume device drivers using the firmware API will have access to the firmware immediately, without having to wait for the root filesystem to mount or dealing with possible race issues with lookups as the root filesystem mounts.

Some implementation details about the firmware cache setup:

- The firmware cache is setup by adding a devres entry for each device that uses all synchronous call except `:c:func:request_firmware_into_buf`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\firmware\[linux-master][Documentation][driver-api][firmware]firmware_cache.rst, line 26); [backlink](#)

Unknown interpreted text role "c:func".

- If an asynchronous call is used the firmware cache is only set up for a device if the second argument (uevent) to `request_firmware_nowait()` is true. When uevent is true it requests that a kobject uevent be sent to userspace for the firmware request through the sysfs fallback mechanism if the firmware file is not found.
- If the firmware cache is determined to be needed as per the above two criteria the firmware cache is setup by adding a devres entry for the device making the firmware request.
- The firmware devres entry is maintained throughout the lifetime of the device. This means that even if you `release_firmware()` the firmware cache will still be used on resume from suspend.
- The timeout for the fallback mechanism is temporarily reduced to 10 seconds as the firmware cache is set up during suspend, the timeout is set back to the old value you had configured after the cache is set up.
- Upon suspend any pending non-uevent firmware requests are killed to avoid stalling the kernel, this is done with `kill_requests_without_uevent()`. Kernel calls requiring the non-uevent therefore need to implement their own firmware cache mechanism but must not use the firmware API on suspend.