```
What:           /sys/bus/pci/drivers/.../bind
Date:           December 2003
Contact:        linux-pci@vger.kernel.org
Description:
                Writing a device location to this file will cause
                the driver to attempt to bind to the device found at
                this location.  This is useful for overriding default
                bindings.  The format for the location is: DDDD:BB:DD.F.
                That is Domain:Bus:Device.Function and is the same as
                found in /sys/bus/pci/devices/.  For example:
                # echo 0000:00:19.0 > /sys/bus/pci/drivers/foo/bind
                (Note: kernels before 2.6.28 may require echo -n).

What:           /sys/bus/pci/drivers/.../unbind
Date:           December 2003
Contact:        linux-pci@vger.kernel.org
Description:
                Writing a device location to this file will cause the
                driver to attempt to unbind from the device found at
                this location.  This may be useful when overriding default
                bindings.  The format for the location is: DDDD:BB:DD.F.
                That is Domain:Bus:Device.Function and is the same as
                found in /sys/bus/pci/devices/. For example:
                # echo 0000:00:19.0 > /sys/bus/pci/drivers/foo/unbind
                (Note: kernels before 2.6.28 may require echo -n).

What:           /sys/bus/pci/drivers/.../new_id
Date:           December 2003
Contact:        linux-pci@vger.kernel.org
Description:
                Writing a device ID to this file will attempt to
                dynamically add a new device ID to a PCI device driver.
                This may allow the driver to support more hardware than
                was included in the driver's static device ID support
                table at compile time.  The format for the device ID is:
                VVVV DDDD SVVV SDDD CCCC MMMM PPPP.  That is Vendor ID,
                Device ID, Subsystem Vendor ID, Subsystem Device ID,
                Class, Class Mask, and Private Driver Data.  The Vendor ID
                and Device ID fields are required, the rest are optional.
                Upon successfully adding an ID, the driver will probe
                for the device and attempt to bind to it.  For example:
                # echo "8086 10f5" > /sys/bus/pci/drivers/foo/new_id

What:           /sys/bus/pci/drivers/.../remove_id
Date:           February 2009
Contact:        Chris Wright <chrisw@sous-sol.org>
Description:
                Writing a device ID to this file will remove an ID
                that was dynamically added via the new_id sysfs entry.
                The format for the device ID is:
                VVVV DDDD SVVV SDDD CCCC MMMM.  That is Vendor ID, Device
                ID, Subsystem Vendor ID, Subsystem Device ID, Class,
                and Class Mask.  The Vendor ID and Device ID fields are
                required, the rest are optional.  After successfully
                removing an ID, the driver will no longer support the
                device.  This is useful to ensure auto probing won't
                match the driver to the device.  For example:
                # echo "8086 10f5" > /sys/bus/pci/drivers/foo/remove_id

What:           /sys/bus/pci/rescan
Date:           January 2009
Contact:        Linux PCI developers <linux-pci@vger.kernel.org>
Description:
                Writing a non-zero value to this attribute will
                force a rescan of all PCI buses in the system, and
                re-discover previously removed devices.
                Depends on CONFIG_HOTPLUG.

What:           /sys/bus/pci/devices/.../remove
Date:           January 2009
Contact:        Linux PCI developers <linux-pci@vger.kernel.org>
Description:
                Writing a non-zero value to this attribute will
                hot-remove the PCI device and any of its children.
                Depends on CONFIG_HOTPLUG.

What:           /sys/bus/pci/devices/.../rescan
Date:           January 2009
Contact:        Linux PCI developers <linux-pci@vger.kernel.org>
Description:
                Writing a non-zero value to this attribute will
                force a rescan of the device's parent bus and all
                child buses, and re-discover devices removed earlier
                from this part of the device tree.
                Depends on CONFIG_HOTPLUG.

What:           /sys/bus/pci/devices/.../reset
Date:           July 2009
Contact:        Michael S. Tsirkin <mst@redhat.com>
Description:
                Some devices allow an individual function to be reset
                without affecting other functions in the same device.
                For devices that have this support, a file named reset
                will be present in sysfs.  Writing 1 to this file
                will perform reset.

What:           /sys/bus/pci/devices/.../vpd
Date:           February 2008
Contact:        Ben Hutchings <bhutchings@solarflare.com>
Description:
                A file named vpd in a device directory will be a
                binary file containing the Vital Product Data for the
                device.  It should follow the VPD format defined in
                PCI Specification 2.1 or 2.2, but users should consider
                that some devices may have malformatted data.  If the
                underlying VPD has a writable section then the
                corresponding section of this file will be writable.

What:           /sys/bus/pci/devices/.../virtfnN
Date:           March 2009
Contact:        Yu Zhao <yu.zhao@intel.com>
Description:
```

```
                This symbolic link appears when hardware supports the SR-IOV
                capability and the Physical Function driver has enabled it.
                The symbolic link points to the PCI device sysfs entry of the
                Virtual Function whose index is N (0...MaxVFs-1).

What:           /sys/bus/pci/devices/.../dep_link
Date:           March 2009
Contact:        Yu Zhao <yu.zhao@intel.com>
Description:
                This symbolic link appears when hardware supports the SR-IOV
                capability and the Physical Function driver has enabled it,
                and this device has vendor specific dependencies with others.
                The symbolic link points to the PCI device sysfs entry of
                Physical Function this device depends on.

What:           /sys/bus/pci/devices/.../physfn
Date:           March 2009
Contact:        Yu Zhao <yu.zhao@intel.com>
Description:
                This symbolic link appears when a device is a Virtual Function.
                The symbolic link points to the PCI device sysfs entry of the
                Physical Function this device associates with.

What:           /sys/bus/pci/slots/.../module
Date:           June 2009
Contact:        linux-pci@vger.kernel.org
Description:
                This symbolic link points to the PCI hotplug controller driver
                module that manages the hotplug slot.
```