NAME

devlink-dev − devlink device configuration

SYNOPSIS

devlink [*OPTIONS*] **dev** { *COMMAND* | **help** }

OPTIONS := { **−V**[*ersion*] | **−n**[*no-nice-names*] }

devlink dev show [*DEV*]

devlink dev help

devlink dev eswitch set *DEV* [**mode** { **legacy** | **switchdev** }] [**inline-mode** { **none** | **link** | **network** | **transport** }] [**encap-mode** { **none** | **basic** }]

devlink dev eswitch show *DEV*

devlink dev param set *DEV* **name** *PARAMETER* **value** *VALUE* **cmode** { **runtime** | **driverinit** | **permanent** }

devlink dev param show [*DEV* **name** *PARAMETER*]

devlink dev reload *DEV* [**netns** { *PID* | *NAME* | *ID* }] [**action** { **driver_reinit** | **fw_activate** }] [**limit no_reset**]

devlink dev info [*DEV*]

devlink dev flash *DEV* **file** *PATH* [**target** *ID*]

DESCRIPTION

devlink dev show - display devlink device attributes

DEV - specifies the devlink device to show. If this argument is omitted all devices are listed.

Format is:
BUS_NAME/BUS_ADDRESS

devlink dev eswitch show - display devlink device eswitch attributes
devlink dev eswitch set - sets devlink device eswitch attributes

mode { **legacy** | **switchdev** }

Set eswitch mode

legacy - Legacy SRIOV

switchdev - SRIOV switchdev offloads

inline-mode { **none** | **link** | **network** | **transport** }

Some HWs need the VF driver to put part of the packet headers on the TX descriptor so the e-switch can do proper matching and steering.

none - None

link - L2 mode

network - L3 mode

transport - L4 mode

encap-mode { **none** | **basic** }

Set eswitch encapsulation support

none - Disable encapsulation support

basic - Enable encapsulation support

devlink dev param set - set new value to devlink device configuration parameter

name *PARAMETER*

Specify parameter name to set.

value *VALUE*

New value to set.

cmode { **runtime** | **driverinit** | **permanent** }

Configuration mode in which the new value is set.

runtime - Set new value while driver is running. This configuration mode doesn't require any reset to apply the new value.

driverinit - Set new value which will be applied during driver initialization. This configuration mode requires restart driver by devlink reload command to apply the new value.

permanent - New value is written to device's non-volatile memory. This configuration mode requires hard reset to apply the new value.

devlink dev param show - display devlink device supported configuration parameters attributes

name *PARAMETER* Specify parameter name to show. If this argument is omitted all parameters supported by devlink devices are listed.

devlink dev reload - perform hot reload of the driver.

DEV - Specifies the devlink device to reload.

netns { *PID* | *NAME* | *ID* } - Specifies the network namespace to reload into, either by pid, name or id.

action { **driver_reinit** | **fw_activate** } - Specifies the reload action required. If this argument is omitted *driver_reinit* action will be used. Note that even though user asks for a specific action, the driver implementation might require to perform another action alongside with it. For example, some driver do not support driver reinitialization being performed without fw activation. Therefore, the devlink reload command returns the list of actions which were actrually performed.

driver_reinit - Driver entities re-initialization, applying devlink-param and devlink-resource values.

fw_activate - Activates new firmware if such image is stored and pending activation. If no limitation specified this action may involve firmware reset. If no new image pending this action will reload current firmware image.

limit no_reset - Specifies limitation on reload action. If this argument is omitted limit is unspecificed and the reload action is not limited. In such case driver implementation may include reset or downtime as needed to perform the actions.

no_reset - No reset allowed, no down time allowed, no link flap and no configuration is lost.

devlink dev info - display device information.

Display device information provided by the driver. This command can be used to query versions of the hardware components or device components which can't be updated (*fixed*) as well as de vice firmware which can be updated. For firmware components *running* displays the versions of firmware currently loaded into the device, while *stored* reports the versions in device's flash. *Running* and *stored* versions may differ after flash has been updated, but before reboot.

DEV - specifies the devlink device to show. If this argument is omitted all devices are listed.

devlink dev flash - write device's non-volatile memory.

DEV - specifies the devlink device to write to.

file *PATH* - Path to the file which will be written into device's flash. The path needs to be relative to one of the directories searched by the kernel firmware loaded, such as /lib/firmware.

component *NAME* - If device stores multiple firmware images in non-volatile memory, this parameter may be used to indicate which firmware image should be written. The value of *NAME* should match the component names from **devlink dev info** and may be driver-dependent.

EXAMPLES

devlink dev show

Shows the state of all devlink devices on the system.

devlink dev show pci/0000:01:00.0

Shows the state of specified devlink device.

devlink dev eswitch show pci/0000:01:00.0

Shows the eswitch mode of specified devlink device.

devlink dev eswitch set pci/0000:01:00.0 mode switchdev

Sets the eswitch mode of specified devlink device to switchdev.

devlink dev param show pci/0000:01:00.0 name max_macs

Shows the parameter max_macs attributes.

devlink dev param set pci/0000:01:00.0 name internal_error_reset value true cmode runtime

Sets the parameter internal_error_reset of specified devlink device to true.

devlink dev reload pci/0000:01:00.0

Performs hot reload of specified devlink device.

devlink dev flash pci/0000:01:00.0 file firmware.bin

Flashes the specified devlink device with provided firmware file name. If the driver supports it, user gets updates about the flash status. For example:
Preparing to flash
Flashing 100%
Flashing done

SEE ALSO

devlink(8), **devlink-port**(8), **devlink-sb**(8), **devlink-monitor**(8),

AUTHOR

Jiri Pirko <jiri@mellanox.com>