

**NAME**

`boltctl` – control the thunderbolt device manager

**SYNOPSIS**

**`boltctl`** *authorize* *DEVICE*  
**`boltctl`** *config*  
**`boltctl`** *domains*  
**`boltctl`** *enroll* *DEVICE*  
**`boltctl`** *forget* *DEVICE*  
**`boltctl`** *info* *DEVICE*  
**`boltctl`** *list*  
**`boltctl`** *monitor*  
**`boltctl`** *power*

**DESCRIPTION**

*boltctl* is the command line interface to interact with *bold*, the system daemon that manages Thunderbolt 3(TM) devices. It can be used to query the state of devices as well as manage them.

Devices can be globally identified via their unique identifier (uuid). All commands that take a *DEVICE* identifier expect this unique id.

If no command is given, it is equivalent to *boltctl list*.

**OPTIONS****`--version`**

Print version information and exit.

**`-U` | **`--uuid`** {*full* | *short* | *alias* | *N*}**

Control how UUIDs are printed. Since they are somewhat sensitive data it is not advisable to share them publicly in full length. Instead *short* or *alias* can and should be used when sharing the output of *boltctl*.

*full*

Print all UUIDs in full length.

*short*

Truncate all UUIDs so only the first 13 characters are printed.

*alias*

All UUIDs are replaced by a random string that is derived from the UUID, therefore the devices can be uniquely identified without revealing the original UUID.

*N*

If a integer *N* is specified, all UUIDs are truncated to only show up to *N*.

**COMMANDS****`authorize` [**`-F`** | **`--first-time`**] *DEVICE***

Authorize a currently unauthorized device identified via its unique id (uuid) *DEVICE*. If a key is stored in the database it will be used, given the security level of the domain supports secure device connection. Use *boltctl list* to find out the uuid of a device.

**`-F` | **`--first-time`****

Normally, when attempting to authorize an already authorized device **`boltctl`** will do nothing and return a successful status code. When using this option, the attempt will fail and result in a negative exit code if the device is already authorized.

**`config` **`--describe`** [*global* | *domain* | *device*]**

List global, domain, or all (if nothing is specified) properties. The format is 3 columns: permission, name, description. Permission indicates if the property is only readable or can also be written.

**config** *KEY* [*VALUE*]

Get or set, if *VALUE* is specified, a global property.

**config** <*domain|device*>.KEY *TARGET* [*VALUE*]

Get or set, if *VALUE* is specified, a domain or device property, where *TARGET* is the unique id of the domain or the device.

**domains** [-v | --verbose]

List all currently active Thunderbolt domains. A Thunderbolt domain represents the Thunderbolt controller hardware. There will be one domain (and host device) for each Thunderbolt controller present in the system. The *security* property shows the security level of the controller. If *iommu* support is active (see the **boltd** man page) it will be indicated by a +*iommu* suffix for "secure" or "user" mode, or just plain *iommu* in case the security level is "none" (sl0). *bootacl* shows the used and total slots of the boot access control list (BootACL) and the content of all non-empty entries. NB: if BootACL is unsupported it will show 0 for both (0/0). The *online* property shows if the thunderbolt controller is currently powered by the firmware. **NB:** if the controller is currently offline the BootACL list will reflect what *boltd* estimates the list will look like once the controller is back online and local changes have been synchronized to the controller. This might not be accurate if the list was modified in the meantime, e.g. from a different installation or OS.

**enroll** [--policy *policy*] *DEVICE*

Authorize and record the device with the unique id *DEVICE* in the database. If the domain supports secure connection a new key will be generated and stored in the database alongside the device name and vendor name. The key, if created, will be used in the future to securely authorize the device.

**--policy** {*default* | *auto* | *manual*}

Specify the policy to be used for the newly enrolled device.

*default*

Use the global default policy of the daemon; this can be changed, but is normally also *auto*.

*auto*

Automatically authorize this device whenever it is connected.

*manual*

Do **not** automatically authorize the device; instead require manual authorization via **boltctl authorize**.

**forget** *DEVICE*

Remove the information about the device with the unique id *DEVICE* from the database. This includes the key, if one was previously generated. If you pass **--all** instead of the *DEVICE* all devices are removed instead of just one.

**info** *DEVICE*

Display information about the device with the unique id *DEVICE*.

**list** [-a | --all]

List and print information about all connected and stored devices.

**-a** | **--all**

Normally, the only the device type that will be shown is peripherals. Therefore the device that represents the host itself will be omitted. Using this option will instead include all device types in the list.

**monitor**

Listen for and show changes in connected devices.

**power** [-t | --timeout *seconds*] [-q | --query]

Power up the Thunderbolt controller. If the Thunderbolt controller is not in "native enumeration mode" it can be completely powered down by the host firmware/BIOS. On supported systems there is an interface to "force" power the thunderbolt controller. If supported this command will request the daemon to do so. The daemon will keep track of all client requests and will release the force power override when the last request is released.

**-t** | **--timeout** *seconds*

Release the force power request after the specified amount of *seconds* and exit.

**-q** | **--query**

Query the current force power status of the daemon.

#### **AUTHOR**

Written by Christian Kellner <ckellner@redhat.com>.