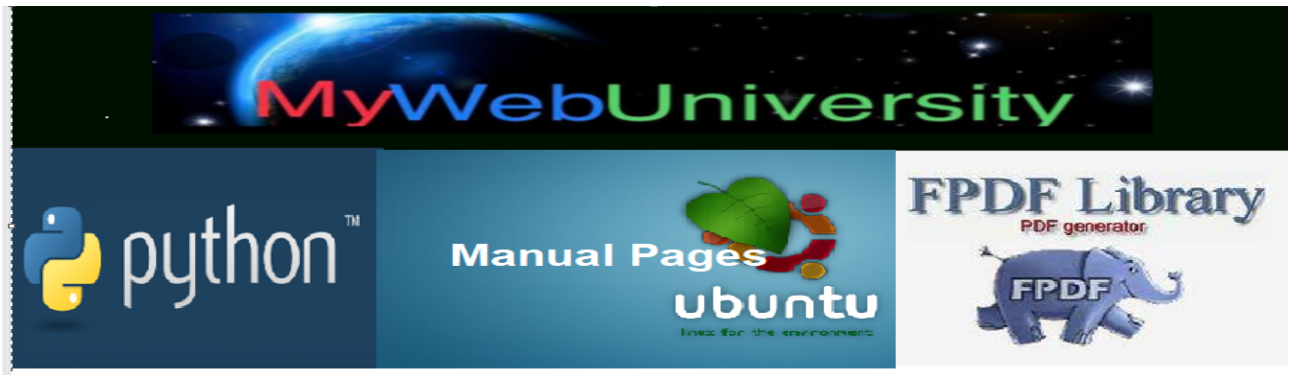

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'bluetoothctl.1'

$ man bluetoothctl.1

bluetoothctl(1) Linux System Administration bluetoothctl(1)

Name

bluetoothctl - Bluetooth Control Command Line Tool

SYNOPSIS

bluetoothctl [-a capability] [-e] [-m] [-t seconds] [-v] [-h]

DESCRIPTION

bluetoothctl(1) interactive bluetooth control tool. The tool works with Bluetooth Classic (BR/EDR) and Bluetooth Low Energy (LE) controllers.

The tool is menu driven but can be automated from the command line. Examples are given in the automation section.

OPTIONS

-a capability, --agent capability

Register agent handler: <capability>

-e, --endpoints

Register Media endpoints

-m, --monitor

Enable monitor output

-t seconds, --timeout seconds

Timeout in seconds for non-interactive mode

-v, --version

Display version

-h, --help

Display help

Commands

list

List available controllers.

Usage > list

show

Controller information.

Usage > show [ctrl]

select

Select default controller.

Usage > select <ctrl>

devices

List available devices, with an optional property as the filter.

Usage > devices [Paired/Bonded/Trusted/Connected]

system-alias

Set controller alias.

Usage > system-alias <name>

reset-alias

Reset controller alias.

Usage > reset-alias

power

Set controller power.

When the controller is powered off, the USB port the controller is attached to is put into a suspend state.

Usage > power <on/off>

advertise

Enable/disable advertising with given type.

If you exit the program advertising will be disabled.

When advertising the controller should advertise with random address but may use its public address if it does not support the feature (address of the device).

A device can advertise if it initiated the connection to another advertising device.

Usage > advertise <on/off/type>

set-alias

Set device alias.

Usage > set-alias <alias>

scan

Scan for devices.

For LE, scanning is an important requirement before connecting or pairing.

The purpose of scanning is to find devices that are advertising with their discoverable flag set (either limited or general). Once you have found the address then you can connect or pair.

Note the following when scanning:

? When scanning the controller will use a random address that is not resolvable so the public address is not leaked. A new random address is created every time scan on is used.

? When turning on scanning the device will start receiving advertising reports of what devices are advertising.

? The filtering of duplicate advertising reports may be enabled depending on the filter? ing settings.

? Device objects found during a scan session will only be persisted if they are con? nected/paired otherwise they are removed after some time.

Usage > scan <on/off/bredr/le>

pair

Pair with device.

This will pair with a device and then trust and connect to it. If the device is already paired this will first remove the pairing.

The command can either be used while the controller is in the connected or not connected state.

If the controller is already connected then the pair command can be used without an argu? ments. If the controller is not connected, the pair command can be given the address of a device with an active scan report and it will initiate the connection before pairing.

Before pairing the agent must be selected to choose the authentication mechanism.

Usage > pair <dev>

pairable

Set controller pairable mode.

This enables/disables pairing. If pairing is disabled then the controller will not accept any pairing requests.

Usage > pairable <on/off>

discoverable

Set discoverable mode.

This enables/disables discoverable mode. If discoverable is disabled then the controller will not respond to any scan requests.

In LE if discoverable if off the controller will just passively scan and not make scan re? quests to advertising devices. If on it will make the advertising requests.

It will use a random address if supported by the controller. The length of time "discover? able on" is valid is determined by discoverable-timeout command.

Usage > discoverable <on/off>

discoverable-timeout

Set discoverable timeout.

The time in seconds that "discoverable on" is valid.

Usage > discoverable-timeout [value]

agent

Enable/disable agent with given capability.

This chooses the local authentication mechanism of the controller. It is needed for pairing and allows you to choose the IO capabilities of the controller.

The valid agent capabilities are: DisplayOnly, DisplayYesNo, KeyboardDisplay, KeyboardOnly, NoInputNoOutput.

Usage > agent <on/off/capability>

default-agent

Set current agent as the default one.

After selecting the agent this will make it the default agent.

Usage > default-agent

trust

Trust device.

Usage > trust <dev>

untrust

Untrust device.

Usage > untrust <dev>

block

Block device.

Usage > block <dev>

unblock

Unblock device

Usage > unblock <dev>

remove

Remove device.

Usage > remove <dev>

connect

Connect device.

This will initiate a connection to a device.

By default this commands tries to connect all the profiles the remote device supports and have been flagged as auto-connectable. In case when the UUID of the remote service is given only that service will be connected. The UUID can be either a short form (16-bit UUID) or a long form (128-bit UUID). There are also some special values for well-known profiles like "a2dp-sink", "a2dp-source", "hfp-hf", "hfp-ag", "ftp" or "spp".

To connect with an LE device the controller must have an active scan report of the device it wants to connect to.

If no advertising report is received before the timeout a le-connection-abort-by-local error will be issued. In that case either try again to connect assuming the device is advertising.

Usage > connect <dev> [uuid]

Example

> connect 1C:48:F9:9D:81:5C

Example

> connect 1C:48:F9:9D:81:5C hsp-hs

Example

> connect 1C:48:F9:9D:81:5C 00001108-0000-1000-8000-00805f9b34fb

Example

> connect 1C:48:F9:9D:81:5C 0x1108

disconnect

Disconnect device.

By default this commands disconnects all profiles and then terminates the connection. In case when the UUID of the remote service is given only that service will be disconnected.

For LE when disconnecting from an active connection the device address is not needed.

Usage > disconnect <dev> [uuid]

info

Device information.

Usage > info <dev>

bearer

Get/Set preferred bearer.

Usage > bearer <dev> [last-seen/bredr/le]

Example get preferred bearer

> bearer <addr>

PreferredBearer: last-seen

Example set preferred bearer to LE

> bearer <addr> le

[CHG] Device <addr> PreferredBearer: le

Changing le succeeded

Example set preferred bearer to BREDR

> bearer <addr> bredr

[CHG] Device <addr> PreferredBearer: bredr

Changing bredr succeeded

Advertise Submenu

See bluetoothctl-advertise(1).

Monitor Submenu

See bluetoothctl-monitor(1)

Scan Submenu

See bluetoothctl-scan(1)

Gatt Submenu

See bluetoothctl-gatt(1)

Admin Submenu

See bluetoothctl-admin(1)

Player Submenu

See bluetoothctl-player(1)

Endpoint Submenu

See bluetoothctl-endpoint(1)

Transport Submenu

See bluetoothctl-transport(1)

Management Submenu

See bluetoothctl-mgmt(1)

Assistant Submenu

See bluetoothctl-assistant(1)

AUTOMATION

Two common ways to automate the tool are to use Here Docs or the program expect. Using Here Docs to show information about the Bluetooth controller.

```
bluetoothctl <<EOF
list
show
EOF
```

RESOURCES

<http://www.bluez.org>

REPORTING BUGS

<linux-bluetooth@vger.kernel.org>

Copyright

Free use of this software is granted under the terms of the GNU Lesser General Public Li? censes (LGPL).