Fing CLI FAQs

Fing CLI vs. Fing Kit

What is the difference between Fing CLI and FingKit?

FingKit is a product designed to provide device scanning and recognition technology. It's composed of two pieces of software:

1. The local client as device scanner
2. The device recognition API providing the recognition technology

For Desktop and Embedded systems, the local client (1) is distributed as part of the Fing CLI. The FingKit documentation is available here. Paragraph 6 and 7 in the guide cover the part related to Desktop/Embedded version.

What are the FingKit requirements?

For Desktop and Embedded systems, the FingKit requirements coincide with those of the Fing CLI. The FingKit requires a license to gain the access to our Device Recognition API. Click here or write to our sales team on sales@fing.io to get pricing information.

How can I launch the FingKit using the Fing CLI?

Read the full PDF user guide. Since now afterwards we refer to the Fing CLI assuming that FingKit is just an option of the tool.

Installation

Getting Started

Fing command-line is available for Windows, Mac OS X, Linux-based operating systems, OpenWRT.

It can be installed on a Raspberry PI (mounting a standard Raspbian) and on any system with Docker containerization system.

The installable files may be downloaded here:

https://www.fing.io/download-free-ip-scanner-desktop-linux-windows-osx/

What are the system requirements?

Fing runs on the following Operating Systems:

- Windows Vista, Windows 7, Windows 8.x, Windows 10
- OS X Maverick (10.9), OS X Yosemite (10.10), OS X El Capitan (10.11), OS X Sierra (10.12), OS X High Sierra (10.13)
- Debian >=5.0, Ubuntu >=8.x, CentOS >=5.0, Fedora >=10, Slackware and Gentoo. Packages are available for 32 and 64 bit architectures.
- Raspbian
- OpenWRT Chaos Calmer on standard target/subtarget architectures
- Anything where the docker toolkit has been installed

What are the user requirements?

You don't need to be a Network Administrator nor a Domain Administrator to run Fing.

An ***administrative user account*** is required to install and run the software; this is a security restriction enforced on every operating system to make sure the network is appropriately protected.

If you are unsure about the type of user account you are using on a machine, please refer to the list below.

Windows

On Windows, you must be logged as a User having Local Administrator privileges. If you are unsure of your user type, click on:

```
 Start | Settings | Control Panel | User accounts.
```

A dialog will pop-up and you shall see the account's type (Standard or Administrator).

You shall be listed as 'Administrator'.

To run the software, a Command-line tool cmd.exe shall be executed as a User Administrator.

OS X

On OS X, you must be logged as a User having Administrator privileges.

If you are unsure of your user type, click on your User Name in the menu-bar and select "Account Preferences...". A dialog will prompt a list of all users and types (Standard, Admin, System).

You shall be listed as either 'Admin' or 'System'.

Linux

OpenWRT

Raspberry PI

You must be logged as the ***root*** user. Other equally-powerful accounts and the installation through `sudo` command are correct, too.
N.B. Find more about privileged access here.

Docker

You already have root permission inside the docker. You must have the permission to start docker instances on the host system.

What are the steps of the installation procedures?

The installation is straightforward on all platform: download the package and install it. Below some specification on how to install on the different operating system.

Windows

Double click on the `.exe` file to trigger the installer and follow the mandatory steps.

If you do not have WinPcap packet capture library in your system, you will be also prompted to install it. When asked to enable the WinPcap 'NPF' Windows service, you can decide whether:

- to install and start the NPF service, thus allowing fing (and other packet capture tools) to be used by any non-admin user in the system
- not to install the NFP service, so that only admin user accounts will be allowed to use fing

As soon as installation is complete, you can open a command prompt and launch the software:

```
fing
```

OS X

After having downloaded the `dmg` file for your OS X, open the `pkg` file, and follow the instructions provided by the installer.

When installed properly, open a Terminal and type:

```
sudo fing
```

Linux

The software has been packed using Debian (DPKG) and Read Hat (RPM) package manager. It's also available as tarball (TGZ) for Linux distros exploiting different package management systems.

Architecture supported:

- `i686` and `amd64` (intel 32/64 bit)
- `arm` and `arm64` (arm soft float 32/64 bit)
- `armhf` (arm hard float 32 bit)
- `mips` and `mips64` (mips big endian 32/64 bit)
- `mipsel` and `mipsel64` (mips little endian 32/64 bit)

Download the package most suitable for your distribution and open a Terminal window.

You may refer to this [List of Linux Distributions]() to find out the package format your platform requires. Supposing you want to install fing on a x86_64-based Linux distribution, type:

```
sudo dpkg -i fing-amd64.deb    # DPKG
sudo rpm  -i fing-amd64.rpm    # RPM
sudo installpkg fing-amd64.tgz # TGZ
```

In general, the package of the fing CLI are distributed using the following standard:

```
fing-<arch>.<format>
```

After the installation has successfully completed, open a Terminal and type:

```
sudo fing
```

The command 'fing' without any additional parameter executes a discovery on your Local Area Network with a plain text output sent to console. This is only the simplest of reports Fing may generate.

OpenWRT

On OpenWRT platform you can use its own package manager (OpenwrtPacKaGe Manager) and download the proper package:

```
fing-<target>-<subtarget>.ipk
```

And install it with this instruction:

```
sudo opkg install fing-<target>-<subtarget>.ipk
```

When installed properly, you can execute the tool by typing on shell:

```
sudo fing
```

Raspberry Pi

On Raspberry Pi you can install Fing CLI using:

- `fing-armhf.deb` / `fing-armhf.tgz` for Raspberry Pi 1 and 2
- `fing-arm64.deb` / `fing-arm64.tgz` for Raspberry Pi 3

Download the ***Rasbpian*** distribution at Raspian official site.
We test and recommend the latest lite version ***RASPBIAN JESSIE LITE***, although other distributions may work as well.

To install Fing CLI correctly, you shall execute, type:

```
sudo apt-get install chkconfig

# Option 1: dpkg
wget [TODO: PUT THE LINK TO THE PROPER PACKAGE]
sudo dpkg -i fing-armhf.deb

# Option 2: tgz
wget [TODO: PUT THE LINK TO THE PROPER PACKAGE]
sudo installpkg fing-armhf.tgz
```

After the installation has successfully completed, open a Terminal and type:

```
sudo fing
```

The command 'fing' without any additional parameter executes a discovery on your Local Area Network with a plain text output sent to console. This is only the simplest of reports Fing may generate.

Docker

Before all, you have to create locally the fing image using the following command:

```
docker import [TODO: PUT THE LINK TO THE DOCKER IMAGE TARBALL] docker-fing
```

Then you can run the container:

```
docker run --name fing --net=host docker-fing /usr/bin/fing
```

Which is the proper command to invoke the Fing CLI?

Windows

```
fing
```

OS X

Linux

OpenWRT

Raspberry PI

```
sudo fing
```

Docker

```
docker run --name fing --net=host docker-fing /usr/bin/fing
```

Uninstallation

What are the steps of the uninstallation procedures?

Windows

To uninstall Fing on Windows platform, use the Start menu to locate the Fing folder. An Uninstall menu item is available; selecting it will run the Uninstaller application that will remove the program from the system.

Linux

If you have installed fing using dpkg, type:

```
sudo dpkg -r fing
```

If you have installed fing using rpm, type:

```
sudo rpm -e fing
```

Otherwise, type:

```
sudo /usr/local/bin/fing-uninstall.sh
```

OS X

Open a Terminal window and execute:

```
sudo /usr/local/bin/fing-uninstall.sh
```

Before you uninstall a kit running as a service, make sure you have shut down any other service that loads the kit at the machine's start-up. See Fing Kit as a Service for further details.

OpenWRT

Open a shell and type:

```
sudo opkg remove fing
```

Raspberry Pi

Login to your raspberry and execute:

```
sudo /usr/local/bin/fing-uninstall.sh
```

Docker

You have to be sure that there are not running instances of fing image:

```
docker rm -f $(docker ps -q -f “ancestor=docker-fing”)
```

then type:

```
docker rmi docker-fing
```

Please refer to Docker official documentation for any deepening.

Configuration Files

Which is the format of the configuration files?

Fing configuration files are all in the same format: properties format (i.e. a list of declarations like `settings=value`)

Where are configuration files stored?

Windows

The application data folder is:

```
%APPDATA%\Fing\conf
```

You also have a shortcut to that folder using :

```
Start -> Fing -> Fing Configuration
```

OS X

Linux

OpenWRT

Raspberry PI

Docker

Configuration files are placed in:

```
/etc/fing
```

N.B. Fing create the configuration files from standard template at the first launch! They do not exist after the installation.

Which customization con I do with the configuration files?

There are a lot of possible customizations that may be performed using the Fing properties file. As an example, to change the separator used in the Comma-Separated-List export, simply open your `fing.properties` configuration file and add/edit the setting below:

```
overlook.fing.formatter.csv.separator = ,
```

Commands

Name	Description	Short Option	Long Option	Argument(s)
Info	Show network information	-i	--info	-
Discovery	Run a network discovery	-n	--netdiscover	Network/Host List
Service Scan	Scan services on a host/network	-s	--servicescan	Network/Host List
Ping	Ping a list of host	-p	--ping	Hosts List
Traceroute	Trace the route to the target	-T	--traceroute	Hosts List
Wake On Lan	Wake a local device	-w	--wol	Hosts List

How to get the network information on the host?

The Fing Kit can provide detailed information about the status of your network card and other useful information on your local network:

- Local Network Interfaces:
  - Type (Ethernet, WiFi, etc.)
  - Hardware Address
  - IPv4
  - IPv6
- Default gateway
- DNS Server

```
<cmd> --info
```

How to run a network discovery?

An ***administrative user account*** is required to run the software on ***all platforms***. The network discovery provides a complete view of any network: fing engine automatically detects the underlying network type and uses the best technique to take the picture of the target network.

The best results are achieved on Local Area Networks, both on wired Ethernet connection and wireless Wi-Fi connections, where Fing can make use of the dedicated data-link layer discovery engine which is faster one more accurate. Fing's engine can detect all the hosts present in the network, even behind a Firewall!

Discoveries performed on non-local networks (or non-ethernet networks) are handled by a network layer discovery engine, which relies on TCP/IP network layer, i.e. ICMP (ping) and TCP queries. When you start a discovery fing tells you the actual engine which is being used; in case of specific needs, it is possible to configure and tune each engine for optimal results, creating dedicated discovery profiles in the related configuration properties file:

```
discovery.properties
```

When you start Fing without arguments, it takes the nearest of your available networks and starts performing a discovery on it, reporting stuff directly on console. But you can perform discovery on any network, by providing your target network to fing in the command line:

```
<cmd> -n 192.168.1.0/24
<cmd> -n www.domotz.com/24
```

If you do not provide any output parameter, fing uses default ones (as specified in its `fing.properties` configuration file). But for a running discovery you can setup as many output formats as you need, by providing a command line argument like:

```
<cmd> -o setupFormat1 setupFormat2 ... setupFormatN
```

The output format setup syntax is pretty simple: there are 2 main categories of output flows, table and log. The table flow produces a network table dump every time a discovery round is completed, while the log flow logs each network event as soon as it's detected.

If you want to change the discovery round frequency, edit the related `round.interval` setting in `discovery.properties` configuration file. Note that for each profile you declare you must provide both configurations for data-link and network layer discovery classes.

The log flow allows you to log network events in real-time, on the console itself or in a specific file. Currently there are two formats supported for log flows: text and CSV.

E.g. to start fing producing textual log in console and a CSV log in a file:

```
<cmd> -o log,text log,csv,my-network-log.csv
```

The table flow instead produces a network table view refreshed each time a discovery round finishes. The most popular formats are text and HTML but here it is the complete list: stext (short text for 80-columns console), text (plain text), html, csv, and xml.

E.g. on Windows to make fing report network discovery to an HTML file on your:

```
fing -o table,stext table,html,"%USERPROFILE%\Desktop\network.html"
```

The network table report contains details for each host found in the network, and it's refreshed in real-time at each round: IP address, MAC address, hostname and host friendly name. The latter is a friendly name you can associate to the hosts by means of the `hosts.properties` configuration file, where you are able to define your custom names for hosts (by IP address or MAC address) and for networks.

By default, when you close fing the discovery session is lost, unless you want to save session data into a session file; in this case fing can be closed and restarted when you need, without losing any discovery session data. To exploit discovery session feature you have to simply provide fing the session file name to use.

E.g. on Windows to make fing generate an HTML report and save session data in a folder named report:

```
fing -n 192.168.1.0/24 -o table,html,c:\report\lan.html --session
               c:\report\lansessiondata
```

How to run a service scan?

The service discovery feature, also known as service scan, quickly detects active TCP services on a target host or network. Service discovery also gives its best with ethernet-based networks, where TCP SYN scan technique can be applied to audit active services on any host in a few seconds.

You can scan a local or remote host but also entire networks. You may specify a maximum number of ports for the scan to make it faster:

```
<cmd> -s 192.168.1.1         # (single host)
<cmd> -s 192.168.1.1 -m 1000 # (single host with max ports)
<cmd> -s www.fing.io         # (domain)
<cmd> -s www.google.com/24   # (entire network)
```

By default discovered services are reported on console as a plain text output, but you can choose between different output formats: like text, CSV, XML and HTML. So it is actually possible not only to use it as a command line administrative tool, but also integrated with your 3rd party applications.

```
<cmd> -s host -o html,report.html
<cmd> -s host -o xml,scan.xml
```

How to measure network round-trip time?

The ping feature is capable of performing multiple quick pings to a list of target hosts. You can scan a set of local or remote hosts.

```
<cmd> -p 192.168.1.1 www.domotz.com www.fing.io
```

By default ping summary is reported on console as a plain text output, but you can choose between different output formats: like text, CSV, XML and HTML. So it is actually possible not only to use it as a command line administrative tool, but also integrated with your 3rd party applications.

The output format setup syntax is pretty simple: there are 2 main categories of output flows, table and log. The table flow produces a network table dump every time a discovery round is completed, while the log flow logs each network event as soon as it's detected.

```
<cmd> -p host -o html,ping.html or fing -p host -o xml,ping.xml
```

How to trace the route of packets to destination host?

The traceroute feature is capable of for displaying the route (path) and measuring transit delays of packets sent to a target host. You can scan a set of local or remote hosts.

```
<cmd> -T 8.8.8.8
```

The summary is reported only on console as a plain text output.

```
-------------------------------------------------------
|          === Fing 5.1.0 - www.fing.io ===           |
|-----------------------------------------------------|
| #hop  | host        |   avg |  loss |   min |   max |
|-----------------------------------------------------|
|     1 | 192.168.7.1 | 1.831 |     - | 1.371 | 2.474 |
|     2 | 10.0.1.1    | 2.223 |     - | 2.007 | 2.514 |

                            ***************

|    11 | 8.8.8.8     | 30.49 |     - | 30.22 | 30.92 |
-------------------------------------------------------
```

How to wake device on LAN or WAN?

Wake-on-LAN (WOL) is an Ethernet computer networking standard that allows a computer to be turned on - or woken up - by a network message.

WOL is implemented at hardware level by the motherboard of a computer (BIOS) and the network interface (firmware). Consequently, it is independent of the Operating System (and NIC drivers) running on the hardware, so it works on Windows, OS X and Linux machines. Some operating systems can control Wake-on-LAN behaviour via hardware drivers. The WOL message must be sent on the local network. To wake up machines from an external network (Wake-on-WAN), WOL Gateways must be configured to receive WOL packets from outside networks, and forward them to the local network.

Fing is able to send WOL magic packets to wake local or remote computers that have the WOL feature enabled. The feature works with devices that are in your connected local network, but you may also send remote WOL signals outiside your local network to hosts/routers configured to be WOL gateway services by specifing the host address and port.

The syntax is very simple:

```
<cmd> --wol
```

Each target must have the following syntax:

```
MAC[@network[:port]]
```

or

```
MAC[@host[:port]]
```

Examples:

- On docker, to send WOL to MAC 010203040506 in current LAN

```
fing --wol 010203040506
```

- On OS X, to send WOL to MAC 010203040506 and 112233445566 in current LAN, silently:

```
sudo fing --wol 010203040506 112233445566 –silent
```

- On Raspberry Pi, to send WOL to MAC 01:02:03:04:05:06 in network 192.168.0.1/24

```
sudo fing --wol 01:02:03:04:05:06@192.168.0.1/24
```

- On Linux, to send a remote WOL for MAC 01:02:03:04:05:06 to the host myremoterouter.com configured on UDP port number 9

```
sudo fing --wol 010203040506@myremoterouter.com:9
```

Fing CLI / Kit as a Service

What does it mean running Fing CLI as a Service?

A **service** is a computer program that runs as a background process, rather than being under the direct control of an interactive user. Fing CLI can run continuously in background and write the output – results and logs – in the respective folder.

What are the system requirements?

Every Operating System has its own service management framework as part of the core features. You do not need to install anything to run Fing CLI as a service

Is it possible to run the kit as a service?

Yes, this section has been designed exactly for that use case indeed.

How can I configure my system to run the Fing Kit as a Service?

Fing Kit requires a configuration file at this location:

- Windows: "`%APPDATA%\Fing\conf\kit.properties`"
- Linux/Unix/OSx: "`/etc/fing/kit.properties`"

The file is created after the first execution of `fing`:

```
enrichment.enabled=true
license=TO_BE_EDITED_WITH_THE_PROPER_LICENSE_KEY
output.folder=/var/data/fing/kit/
refresh.interval=3600000
rounds=0
```

Edit `kit.properties` and insert your license ID.

After that you can setup your system.

Windows

The command-line interface of Fing is a simple but powerful tool to integrate Fing in your daily work. It's easy to perform network discoveries in the background, or generate XML reports at regular intervals. If executed as a Windows Service, Fing automatically starts when you power on your computer, and automatically stops when you shut it down. As all Windows services, Fing Services can be managed from the `Windows Service Control Panel`.

You must use absolute paths for every file you provide in the command line (both output files and session file). Fing Service will run as **LocalSystem** user and will use the Home Folders of the user that registers the service to read and store configuration files, log files and any other accessory file.

To manually install it:

```
  fing --installservice FingKit --kit "%APPDATA%\Fing\conf\kit.properties"
```

The corresponding Fing network discovery with device recongition is installed as service and immediately started. Later, if you want to later uninstall the service, just type:

```
  fing --uninstallservice FingKit
```

N.B. Note that, at the moment, Fing service removal is not supported when completely uninstalling Fing. Hence, if you are in the process of upgrading or removing Fing software from your computer, you should stop and uninstall running Fing services as an initial step.

OS X

Copy or link the launchd script from

```
/usr/local/lib/fing/launchd/com.fing.fingkit.FingKit.plist
```

to the folder

```
/Library/LaunchDaemons
```

The full instruction is:

```
sudo cp /usr/local/lib/fing/launchd/com.fing.fingkit.FingKit.plist \
                  /Library/LaunchDaemons
```

To manually load and start the service in background, type:

```
sudo launchctl load /Library/LaunchDaemons/com.fing.fingkit.FingKit.plist
```

To unload:

```
sudo launchctl unload /Library/LaunchDaemons/com.fing.fingkit.FingKit.plist
```

Linux

OpenWRT

The two main services manager for Linux operating systems and also the Raspbian for your PI are `System V` (Red-hat, Centos, and other RPM based distributions support it) and `Upstart` (Ubuntu and Debian).

On OpenWRT the standard should be System V.

The steps to do are really smooth:

- System V

.
Copy or link the init.d script from `/usr/local/lib/fing/init.d/fing-kit` to your `/etc/init.d` folder.

Then add to your services with:

```
chkconfig --add fing-kit
```

To manually control the service:

- start:

```
service fing-kit start
```

- stop:

```
service fing-kit stop
```

- restart:

```
service fing-kit restart
```

- Upstart

Copy the upstart script from `/usr/local/lib/fing/upstart/fing-kit.conf` to your `/etc/init` folder.

To manage the service with upstart:

- start:

```
sudo start fing-kit
```

- stop:

```
sudo stop fing-kit
```

Raspberry PI

The Raspberry also has System V. We have a specific init file for it that you have to copy or link in `/etc/init.d`:

```
sudo cp /usr/local/lib/fing/init.d/fing-kit.raspberry /etc/init.d/fing-kit
```

Make sure you have installed chkconfig:

```
sudo apt-get install chkconfig
```

Then add to your services with:

```
sudo chkconfig --add fing-kit
```

To manually control the service:

- start:

```
sudo systemctl start fing-kit
```

- stop:

```
sudo systemctl stop fing-kit
```

- restart:

```
sudo systemctl restart fing-kit
```

Docker

The docker container can be run in background using a simple option:

```
docker run --name fing --net=host --detach \
      --mount type=bind,source=~/conf,target=/etc/fing \
      --mount type=bind,source=~/data,target=/var/data/fing \
      docker-fing /usr/bin/fing --kit /etc/fing/kit.properties
```

Before running the command you have to create the file `kit.properties` in `~/conf/kit.properties` as explained in the kit documentation.

N.B. The path refers to the Docker file system. The mount options is required to have the configuration and the output folder on your system.

Can I run other Fing CLI command as service?

Yes. Below some explanation

Windows

Change the option '`--kit "%APPDATA%\Fing\conf\kit.properties"`' with those you need as explained in this section.

OS X

Follow these steps:

1. Copy the plist file

```
sudo cp /usr/local/lib/fing/launchd/com.fing.fingkit.FingKit.plist \
      /Library/LaunchDaemons
```

2. The plist file has been design to run the kit. You have to edit the option in the file.

```
 <key>ProgramArguments</key>
 <array>
          <string>/usr/local/bin/fing</string>
          <string>--kit</string>
          <string>/etc/fing/kit.properties</string>
  </array>
```

3. Load the daemon as explained here.

Linux

OpenWRT

Raspberry Pi

If your system has system V as service manager. You need to follow the procedure here and then update the file `/etc/init.d/fing-kit`. You have just to update the variable options with those required by the command you want to launch.

If your system exploit upstart, you have to do the same in the `fing-kit.conf` file inside `/etc/init`

Docker

The easier the better:

```
docker run --name fing --net=host --detach docker-fing /usr/bin/fing …
```

Interactive mode

The interactive mode is will guide you through Fing tools with a step-by-step procedure. Fing is a command-line tool, and you would need to provide a (possibly long) list of arguments to specify your settings. In the interactive mode, the tool itself will guide through the available features and configurations, from basic operations to complex discoveries, in just few seconds. The feature configures Fing and starts your discovery and additionally displays the whole command-line arguments, according to your chosen options.

How can I start fing in interactive mode?

Windows

On Windows, you can start interactive mode from fing shortcut in

```
Start menu | Applications
```

Alternatively, you may execute a Command-Line shell using

```
Start menu | Execute
```

and type

```
cmd
```

a Shell window will open, allowing you to enter the command:

```
fing --interactive
```

OS X

Linux

OpenWRT

Raspberry Pi

You shall open a Terminal window and execute:

```
sudo fing --interactive
```

Docker

You shall open a shell on the host system and execute:

```
docker run --name=fing –net=host docker-fing /usr/bin/fing --interactive
```

What tools are available?

Once started, the interactive mode will provide the following options:

- ***(D)***iscover

- ***(S)***can
- ***(P)***ing
- ***(T)***raceroute
- display ***(I)***nfos

Each letter will start the corresponding Fing tool, guiding you through a series of question regarding the networks to scan, the output formats, the output destinations and such. At the end of each tool, the interactive mode will print the full set of parameters - should you need to run the same command directly in the future - and will ask you if you wish to execute it now.

```
=== You have completed the procedure ===

      The equivalent command is:
      fing -p localhost -o csv,console
```

Other

How to extract data from CSV files?

Fing for Console may output the content of the discovery in a classical Comma-Separated-Output, which actually uses a semicolon to separate the content. You may generate the CSV output using the appropriate output format, as in the example below:

```
<cmd> -r1 -n10.255.0.0/16 -o table,csv,fingscan.csv
```

The file contains the discovery data is in the following order:

1. IP address
2. Custom Name of the Node (if you have assigned custom names in the `.properties` file)
3. The state (UP/DOWN)
4. The timestamp of last change
5. The host name (if you have enabled DNS)
6. The Hardware address (if you are discoverying a local network)
7. The Hardware vendor

You may eliminate one of the column with simple scripts. On Unix and Mac Os X, you may use an "awk oneliners", like:

```
awk -F ";" '{print $1, $2, $3, $5, $6, $7}' yourfile.csv
```

On Windows, you may use the PowerShell, like:

```
Get-Content yourfile.csv | %{ $_.Split(';')[1]; }
```

which extracts the first column. There is also the import-csv cmdlet which provides many features.