Termite2 Client - Console Commands

This document presents and explains the available Termite2 commands that can be performed on the Termite2 console interface. The available commands are presented in 3 groups. **Network commands** are those used to create and model the emulated network (these commands also exists on the previous system, Termite). **Emulator commands** are used to interact and managed the emulators used on the emulated network. **Other commands** can be see as miscellaneous commands with varied uses.

For each group, the commands are identified by their names, followed by their syntax and a brief description/explanation. See example bellow:

- **cmd**
  - Syntax (cmd <arg1> <arg2> <arg3> …).
  - Description and explanation of command cmd.

Network commands:

- **newdevice**
  - newdevice <node>
  - Creates a new virtual device/node on the emulated network with <node> as its name/id.
- **deletedevice**
  - deletedevice <node>
  - Deletes the <node> from the emulated network.
- **binddevice**
  - binddevice <node> <emu_id>
  - Binds the <node> to the emulator <emu_id>[1].
- **unbinddevice**
  - unbinddevice <node> <emu_id>
  - Unbinds the <node> to the emulator <emu_id>.
- **move**
  - move <node1> (<node2>, <node3>, …)
  - Emulates the movement of <node1> to the vicinity of the nodes <node2> and <node3>. This makes the nodes <node2> and <node3> neighbours off <node1>. Note that the command does not make <node2> and <node3> neighbours off each other.
    To emulate a node moving out of the vicinity of all other do “move <node1> ( )”.
- **creategroup**
  - creategroup <go_node> (<node2>, <node3>, …)
  - Emulates the creation on a peer-to-peer group where the <go_node> (group owner node) creates a group with the nodes <node2> and <node3> as members (himself included). For the

[1] <emu_id> is the id given by the Termite2 Client to the emulators. To see the emulators, their id’s and other data use the command “list emus”. This command is explained on Other commands group.

group creation to work the member nodes nodes (<node2> and <node3>) must be in the vicinity of the <go_node>. Note that is possible to create empty groups by not providing members, example: “creategroup <go_node> ( )”.

- **joingroup**
  - joingroup <node> (<go_node1>, <go_node2>, …)
  - Emulates <node> joining the group(s) <go_node1> and <go_node2> (…). Note that <node> must be in the vicinity of the go nodes <go_node1> and <go_node2>.
- **leavegroup**
  - leavegroup <node> (<go_node1>, <go_node2>, …)
  - Emulates <node> leaving the group(s) <go_node1> and <go_node2> (…).
- **deletegroup**
  - deletegroup <go_node>
  - Deletes the group corresponding to the <go_node>.
- **ping**
  - ping
  - Pings all emulators in order to check if WiFi-Direct is on (through the Termite2 API on the applications).
- **commit**
  - commit
  - Commits the emulated network state/topology on the Termite2 Client to all emulators that are bound to nodes.

Emulator(s) commands:

- **refresh**
  - refresh
  - Refreshes the available emulator detected by the Termite2 Client.
- **installed**
  - installed
  - Returns all the AVD’s (Android virtual devices) installed on the connected Termite2 Server.
- **createavds**
  - createavds <t2server-ip> <nº> <avd_name> <api>
  - This command creates <nº> AVD’s with the name <avd_name> and api number <api> on the Termite2 Server on the address <t2server-ip>[2]. An example of this command can be “createavds 192.168.1.2 3 test 21”, this creates 3 AVD’s, test1, test2 and test3 all with the Android api 21 on the Termite2 Server at 192.168.1.2 .

[2] <t2server_ip> is the address that identifies the termite2 Server targeted by the command. To see the address of all the Termite2 Severs use the command “list servers”. This command is explained on Other commands group.

- **destroyavds**
  - destroyavds <t2server-ip> <avd_name1> <avd_name2 > … / all
  - This command deletes the AVD's <avd_name1> and <avd_name2 > (…) on the Termite2 Server on the address <t2server-ip>. If you want to delete all AVD's on a Termite2 Server machine you can use the argument "all" instead of passing the AVD names (for example "destroyavds 192.168.1.2 test1 test2" or "destroyavds 192.168.1.2 all").
- **startemus**
  - startemus <t2server1-ip> <all/nº> <app-pack> | <t2server2-ip> <all/nº> <app-pack> | ...
  - This command starts <nº> emulators (or all available if we use "all", available means installed AVDs) on the Termite2 Server on the address <t2server1-ip> and optionally, it automatically starts the application with the package name <app-pack> on the emulators after they start. We can chain multiple startemus commands for multiple Termite2 Servers by using the operator "|", for example: " startemus 192.168.1.2 2 test.package | 192.168.1.3 all " starts 2 emulators running the application with package test.package on the Termite2 Server - 192.168.1.2 and at the same time start all available emulators (with no app) on the Termite2 Server - 192.168.1.3.
    Note if we try to start a number (<nº>) of emulators that exceeds the number of installed AVDs on the Termite2 Server, the command resolves to starting the maximum number of emulators possible. For example if we try to start 5 emulators but there is only 3 AVDs installed, the command starts all 3.
- **stopemu**
  - stopemu <emu_id>
  - Stops chosen emulator <emu_id> .
- **stopall**
  - stopall // stopall <t2server1-ip> <t2server2-ip> …
  - This command stops all emulators on the Termite2 Server at <t2server1-ip> and <t2server2-ip> (…). If the command is performed without arguments we stop all emulators on all connected Termite2 Servers.
- **installapp**
  - installapp <t2server-ip> <apk-name> <avd_name1> <avd_name2> / all | …
  - This command installs the apk <apk-name>[3] on the online emulators[4] <avd_name1> <avd_name2> (…) running on the Termite2 Server at <t2server-ip>. If we want to install the apk on all the online emulators on the Termite2 Server at <t2server-ip> we can simply use the "all" argument instead of the AVD names.
    Similar to the startemus command, we can use the operator "|" to chain multiple installapp commands on multiple Termite2 Servers, for example:
    "installapp 192.168.1.2 test.apk test1 test2 | 192.168.1.3 test.apk all" install the apk "test.apk" on the emulators test1 and test2 at the Termite2 Server - 192.168.1.2 and at the same time it installs the same apk on all online emulators at the Termite2 Server - 192.168.1.3.

[3] This file of type .apk must be stored on the folder ~/Termite2Server/apks on the respective Termite2 Server(s).

[4] Online emulators are emulators that are already running on the Termite2 Server machine(s)

- **startapp**
    - startapp <t2server-ip> <app-pack> <avd_name1> <avd_name2> / all | …
    - This command starts the app with the package name <app-pack> on the online emulators <avd_name1> <avd_name2> (…) running on the Termite2 Server at <t2server-ip>. If we want to start the apk on all the online emulators on the Termite2 Server at <t2server-ip> we can simply use the "all" argument instead of the AVD names.
      Similar to the startemus and installapp commands, we can use the operator "|" to chain multiple startapp commands on multiple Termite2 Servers, for example:
      "startapp 192.168.1.2 test.package test1 test2 | 192.168.1.3 test.package all" starts the app corresponding to the app app package "test.package" on the emulators test1 and test2 at the Termite2 Server with the address 192.168.1.2 and at the same time starts the same app on all online emulators at the Termite2 Server with the address 192.168.1.3.

Other commands:

- **list**
    - list emulators / servers / devices / groups / neighbours / binds / tnetwork / scripts / history
    - This command lists various types of information based on the argument; "emulators" shows the emulators currently detected by the Termite2 Client; "servers" shows the IP addresses of the connected Termite2 Server(s); "devices" presents the virtual nodes on the emulated network; "groups" shows the peer-to-peer groups that exist on the emulated network; "neighbours" list the neighbours of each virtual node on the network; "binds" shows the binds that currently exist between nodes and emulators; "tnetwork" shows the state of each virtual node on the network (their neighbours and groups); "scripts" presents the name of all the scripts files available to be loaded with the load command; "history" shows the command history.
- **load**
    - load <scrip_name.txt>
    - This command loads an txt file[5] that contains Termite2 commands (one per line) and executes them in order. This is useful too easily execute a large number of commands or save/load various operations.
      For example, we can create a scrip that starts multiple emulators on various Termite2 Servers. Then an application is installed and started on all emulators. In this example, this allows us too easily set up emulators ready to be used on the emulated network.
      While the file is being loaded (commands are being processed) you can type "stop" on the Termite2 Client command line to stop the load command.
- **wait**
    - wait <time> ms/s/m
    - This command stalls the program for a period of <time>. This time can be in milliseconds "ms"(default, if no argument is passed), seconds "s" or minutes "m".
      This command is more useful if used on a script that runs multiple commands on the Termite2 Client (using the command load) and applies waiting times between some commands. For example imagine an txt file with the following commands, on per line: newdevice a, newdevice b, move a (b), wait 5 s, move a ( ).

[5] The file as to be of type .txt and must be stored on the folder ~/Termite2-Cli/scripts

Loading this file (load command) will cause the creation of the nodes a and b, then emulates node a coming to the vicinity of node b, waiting 5 seconds and the node a moves away from the vicinity of node b.

- **localscript**
  - localscript <file_name> <arg1> <arg2> …
  - This command allows us to execute the script file <file_name>[6] with the arguments (for the script) <arg1> and <arg2> (…) from the Termite2 Client console. This can be used to execute scp scripts that transfers files to the Termite2 Server(s) are running (for example apk files to them install on emulators managed by the Termite2 Server(s)).
- **runscript**
  - runscript <t2server-ip> <file_name> <arg1> <arg2> …
  - This command is similar to the localscript command, but the script is run on the Termite2 Server at <t2server-ip>. The script file must be of type .sh or .bat and must be stored on the folder ~/Termite2Server/scripts on the respective Termite2 Server.
    This command is useful if you want to run scripts on Termite2 Server machines remotely form a single control point (the Termite2 Client).
- **time**
  - time set / log / show / reset
  - This command works like a stop watch. Use “set” to initiate the time and then use “log” to log and see how much time as passed in milliseconds. Use “show” to display all logged values since the last “set”. Use “reset” to reset the time watch and the “show” logged values.
- **cls**
  - cls
  - Clear the console screen.
- **help**
  - help commands/cmds
  - Presents a list with all available commands. If you pass the argument “commands” or “cmds” the commands are listed with a brief explanation.
- **quit**
  - quit
  - Terminates the Termite2 Client.

[6] The file as to be of type .sh or .bat and must be stored on the folder ~\Termite2-Cli\scripts