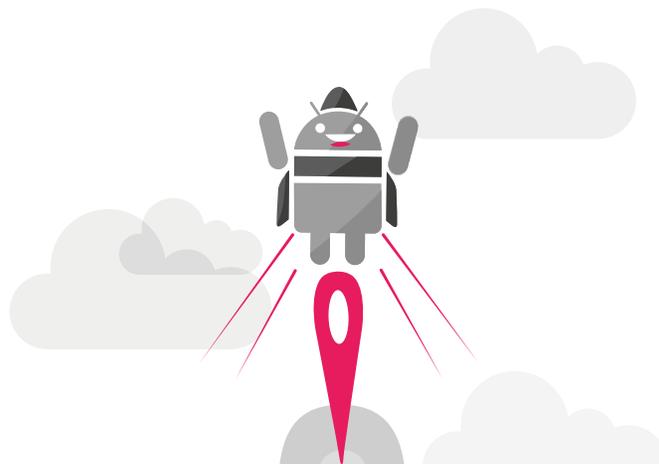


GENYMOTION^{oo}

Shell Guide

Version 2.12



No part of this document may be reproduced or transmitted in any form or by any means, without prior written permission of Genymobile.

Android is a trademark of Google Inc.

Table of contents

Overview	4
Starting Genymotion Shell	5
Interacting with Genymotion Shell	6
Interacting with virtual devices	7
Devices	7
Battery	7
GPS	8
Rotation	9
Android	9
Disk I/O	10
Network	10
Phone	11
Interacting with Genymotion	14
Using Genymotion Shell from a command prompt	15

Overview

Genymotion provides a shell to script the modification of sensor statuses.

This guide explains how to use Genymotion Shell and lists the available commands.

In this guide, the following instructional icons are used:



Notes, tips or additional information.



Situations that could cause performance issues or data losses.

Starting Genymotion Shell

To start Genymotion Shell, perform the action corresponding to your operating system:

- **Windows:** run `C:\Program Files\Genymobile\Genymotion\genyshell.exe`.
- **macOS:** click  from the **Applications** directory or run `/Applications/Genymotion Shell.app/Contents/MacOS/genyshell`.
- **Linux:** run `<Genymotion installer path>/genymotion-shell`.

Interacting with Genymotion Shell

With the commands below, you can perform various basic actions related to the use of Genymotion Shell:

- `help`: displays the list of available commands.
- `pause`: pauses the execution of Genymotion Shell. The value must be set in seconds.
- `version`: returns Genymotion Shell version.
- `exit` or `quit`: closes Genymotion Shell.

Interacting with virtual devices

To modify sensor statuses and values of a virtual device, Genymotion Shell provides commands grouped by category corresponding to the component of the virtual device they interact with. This chapter lists available commands of each category.

Devices

"Devices" commands interact or return results about virtual devices themselves. Available commands are:

- `devices list`
Lists available virtual devices and provides details such as current status, IP address and name.
- `devices refresh`
Refreshes the virtual device list.
- `devices select`
Selects a virtual device. Add the ID of the virtual device you wish to select (as displayed in the device list).
- `devices ping`
Sends a ping message to check that the virtual device is responding.
- `devices factoryreset`
Resets the virtual device to factory state. Add the ID of the device you wish to reset (as displayed in the device list). Adding the keyword `force` at the end skips the warning message about data loss.

Battery

"Battery" commands interact with the Battery widget. Available commands are:

- `battery getmode`
Returns the current battery mode of the selected virtual device.
- `battery setmode`
Sets the battery mode of the selected virtual device. The mode can be:
 - `host`: the virtual battery values reflect the host battery values (if any);
 - `manual`: the virtual battery charge level and state of charge can be manually set.

- `battery getlevel`
Returns the current battery charge level of the selected virtual device. If the battery mode is "host", the returned value is the host value.
- `battery setlevel`
Sets the battery charge level of the selected virtual device. The value must range from 0% to 100%. This command forces the activation of the manual mode.
- `battery getstatus`
Returns the current battery state of charge of the selected virtual device.
- `battery setstatus`
Sets the battery state of charge of the selected virtual device. Values can be:
 - `discharging`: simulates that the power supply is unplugged and the battery is discharging.
The value ranges from 0% to 100%.
 - `charging`: simulates that the power supply is plugged in and the battery is charging.
The value ranges from 0% to 100%.
 - `notcharging`: simulates that the power supply has just been unplugged and the battery is not discharging yet.
The value ranges from 0% to 100%.
 - `full`: simulates that the battery is fully charged.

GPS

"GPS" commands interact with the GPS widget. Available commands are:

- `gps getstatus`
Returns the activation status of the GPS signal reception.
- `gps setstatus`
Sets the activation status of the GPS signal reception. Values can be:
 - `enabled`: the GPS signal reception is enabled;
 - `disabled`: the GPS signal reception is disabled.
- `gps getlatitude`
Returns the current latitude value (if the GPS is activated and a latitude value is set) or 0.
- `gps setlatitude`
Sets the latitude value (and forces the activation of the GPS if deactivated). The value must range from -90° to 90°.
- `gps getlongitude`
Returns the current longitude value (if the GPS is activated and a longitude value is set) or 0.

- `gps setlongitude`
Sets the longitude value (and forces the activation of the GPS if deactivated). The value must range from -180° to 180° .
- `gps getaltitude`
Returns the current altitude value (if the GPS is activated and an altitude value is set) or 0.
- `gps setaltitude`
Sets the altitude value (and forces the activation of the GPS if deactivated). The value must range from -20m to 10000m.
- `gps getaccuracy`
Returns the current accuracy value (if the GPS is activated and an accuracy value is set) or 0.
- `gps setaccuracy`
Set the accuracy value (and forces the activation of the GPS if deactivated). The value must range from 0m to 200m.
- `gps getbearing`
Returns the current bearing value (if the GPS is activated and a bearing value is set) or 0.
- `gps setbearing`
Sets the bearing value (and forces the activation of the GPS if deactivated). The value must range from 0° to 359.99° .

Rotation

The "Rotation" command interacts with the rotation feature. The command is:

- `rotation setangle`
Sets the rotation angle value of the virtual device. The value must range from 0° to 359° .

Android

"Android" commands interact with the Identifiers widget. Available commands are:

- `android version`
Returns the Android version of the selected virtual device.
- `android getandroidid`
Returns the Android ID number of the selected virtual device.



This is only possible with an Indie or a Business license.

- `android setandroidid`
Sets the Android ID number of the selected virtual device.
Values can be:

- `random`: a random Android ID number is generated;
- `custom`: you must enter a value containing 16 hexadecimal digits.



This is only possible with an Indie or a Business license.



If several users exist on the Android system, the Android ID can only be set for the first user.

- `android getdeviceid`

Returns the device ID (IMEI/MEID) number of the selected virtual device.



This is only possible with an Indie or a Business license.

- `android setdeviceid`

Sets the device ID (IMEI/MEID) number of the selected virtual device.

Values can be:

- `none`: no device ID number is generated.
- `random`: a random device ID number is generated.
- `custom`: you must enter a value containing alphanumeric characters, dots, dashes and/or underscores.



This is only possible with an Indie or a Business license.

Disk I/O

"Disk I/O" commands interact with the Disk I/O widget. Available commands are:

- `clearcache`

Clears the disk cache.

- `getreadratelimit`

Returns the read rate limit in KB/sec. If no read rate limit is set, the read rate is unlimited.

- `setreadratelimit`

Sets the disk read rate limit in KB/sec. The value must range from 1 to 2097151. Use 0 to disable read rate limit.

Network

"Network" commands interact with the Network widget. Available commands are:

- `network getprofile`

Returns the current network profile of the selected virtual device.

- `network setprofile`
Sets the network profile of the selected virtual device. Values can be:
 - `no-data`: no network connection is simulated.
 - `gprs`: a GPRS network connection is simulated.
 - `edge`: an Edge network connection is simulated.
 - `3g`: a 3G network connection is simulated.
 - `4g`: a 4G network connection is simulated.
 - `4g-high-losses`: a 4G network connection with packet losses set to 10% is simulated.
 - `4g-bad-dns`: a 4G network connection with a DNS delay set to 3000ms is simulated.
 - `wifi`: a Wi-Fi network connection is simulated.

Phone

"Phone" commands interact with the Phone widget. Available commands are:

- `phone call <phone_number>`
Simulates an incoming phone call from a given phone number.
- `phone sms <phone_number> <message>`
Simulates an incoming text message from a given phone number.
- `phone baseband <command>`
Sends commands directly to the baseband. This has been implemented in order to ease script creation. For more information on the phone baseband command and its capabilities, please refer to section [Baseband commands](#) below.

Baseband commands

The `phone baseband` commands send raw data directly to the baseband. In this section, you will find all commands available in Genymotion Shell to directly interact with the baseband.



Commands below return results in your logcat.

- `gsm list`
Lists all incoming and outgoing phone calls and their state.
- `gsm call <phone_number>`
Simulates a new incoming call from a given phone number.
- `gsm busy <remote_number>`
Hangs up an outgoing call (performed from the integrated Dialer application) and reports the remote phone as busy.

- `gsm hold <remote_number>`
Simulates that an incoming or outgoing phone call (performed from the integrated Dialer application) is put on hold.
- `gsm accept <remote_number>`
Simulates that the outgoing phone call (performed from the integrated Dialer application) is answered.
- `gsm cancel <phone_number>`
Simulates that an incoming or outgoing phone call is hung up.
- `gsm data`
Modifies data connection state. Values can be:
 - `unregistered` or `off`: no network available;
 - `home` or `on`: on local network, non-roaming;
 - `roaming`: on roaming network;
 - `searching`: searching networks;
 - `denied`: emergency calls only.
- `gsm voice`
Modifies voice connection state. Values can be:
 - `unregistered` or `off`: no network available;
 - `home` or `on`: on local network, non-roaming;
 - `roaming`: on roaming network;
 - `searching`: searching networks;
 - `denied`: emergency calls only.
- `gsm status`
Displays the current status of the GSM emulation.
- `gsm signal <rssi> <ber> <rs_snr>`
Changes the reported strength and error rate on next (15s) update.
 - `rssi`: the value must range from 0 to 31. Returns 99 if the value is unknown.
 - `ber`: the value must range from 0% to 7%. Returns 99 if the value is unknown.
 - `rs_snr`: the value must range from -200 to 300.
- `cdma ssource <ssource>`
Sets the current CDMA subscription source. Values can be:
 - `nv`: the subscription is read from non-volatile RAM;
 - `ruim`: the subscription is read from RUIM.
- `cdma prl_version <version>`
Dumps the current PRL version.
- `sms send <phone_number> <message>`
Simulates an incoming text message from a given phone number.

- `sms pdu <hexstring>`

Simulates a new incoming text message defined in the PDU format.

Interacting with Genymotion

The commands below allow you to perform actions related to the Genymotion application via Genymotion Shell:

- `genymotion capabilities`
Returns Genymotion features available on the selected virtual device.
- `genymotion clearcache`
Clears temporary files and logs.
- `genymotion version`
Returns the Genymotion version of the selected virtual device.
- `genymotion license`
Returns information about the Genymotion license (type, validity).

Using Genymotion Shell from a command prompt

You can connect to Genymotion Shell and interact with a virtual device from a command prompt using the commands defined in section [Interacting with virtual devices](#).

Options below are specific to the command prompt:

- `-h`: displays the list of available options.
- `-r ip_address`: connects to a given virtual device.
- `-c "command"`: runs the command on the selected virtual device and returns corresponding values.
- `-f file`: runs the content of the file on the selected virtual device and returns corresponding values line by line.