Android Control Tester

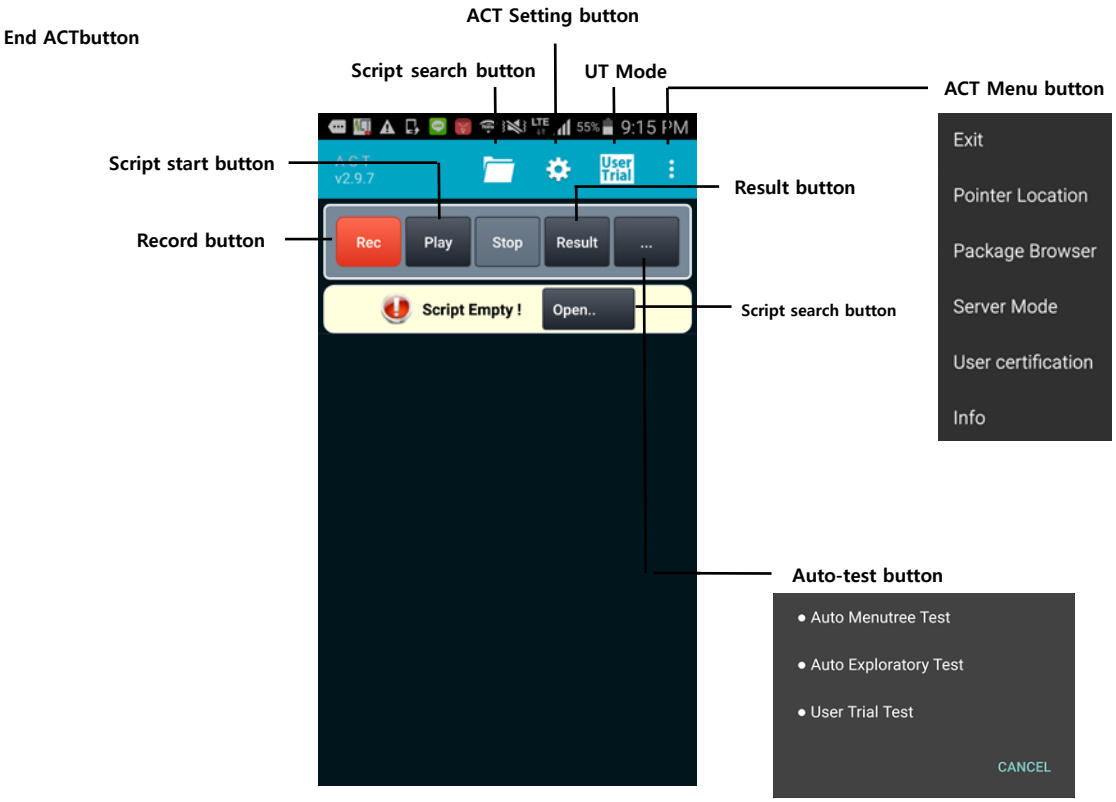
Summary

1. Android Control Tester

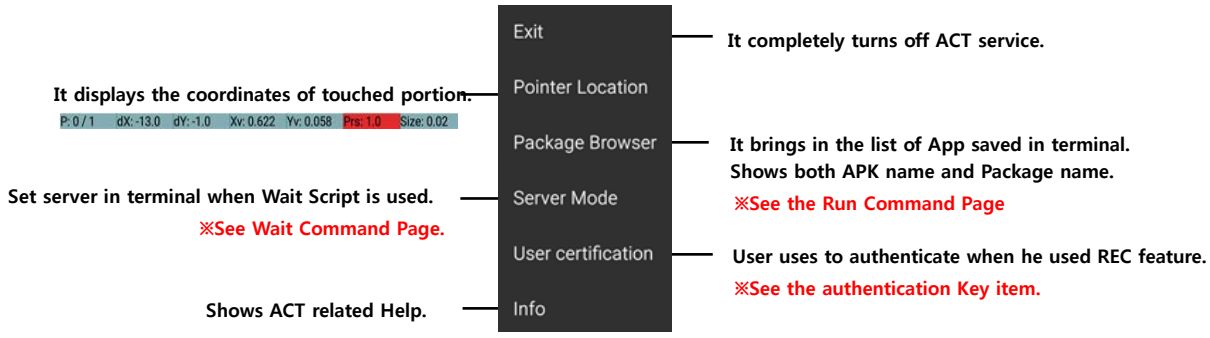
▪ Definition of ACT

This is the tool which enables terminal to execute specific script exclusively. It can be used in several ways by combining various commands and, User's input can be reproduced as it is by recording even without composing a separate script by supporting REC feature.

▪ Composition of ACT2 screen



Composition of ACT2 Menu



2. Sequential (menu tree) exploration test

▪ Definition of sequential exploratory test

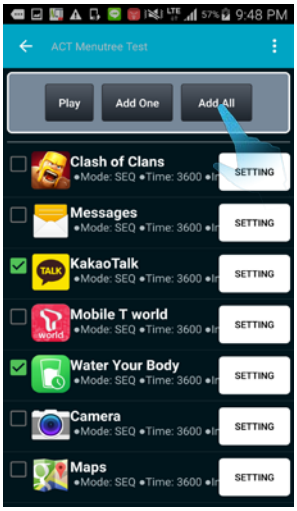
It executes the menu and UI Component of the relevant App sequentially. It can execute the menus which is difficult for a person to test all thus supplementing the weakness of being unable to test by irregular exploration test.

▪ Method of sequential exploration test

Auto exploratory test button > Sequence (Menu tree) exploratory test > Add the App you wish > settings > Play .

Example

Setting – 1 before the test

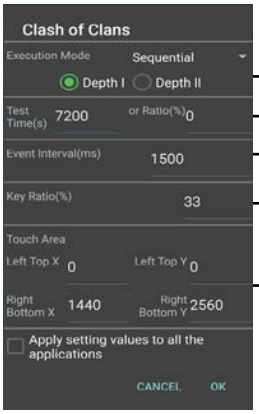


Select the app to be tested. When there are many apps to be tested, Add all > select App > delete

When there are less testing apps

Add one>select App> confirm

Setting – 2 before the test



Set the depth of exploration.

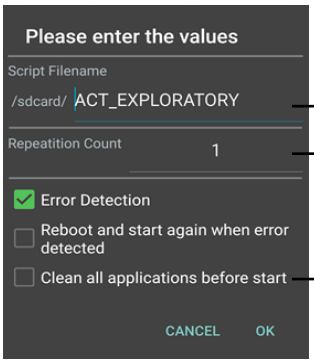
Set the time of exploration test.

Set Sleep time between events.

Set the %age of key to be inputted during random_keypress.

Set the coordinates of screen to be touched at the time of random_tap.

Setting – 3 before the test



Set the storage to save Auto-created script.

Set total repetition.

Set whether to delete data whenever App is executed.

Result of auto script creation

Script gets auto-created as per the above settings.

3. Non sequential exploratory test

▪ Definition of non-sequential exploratory test

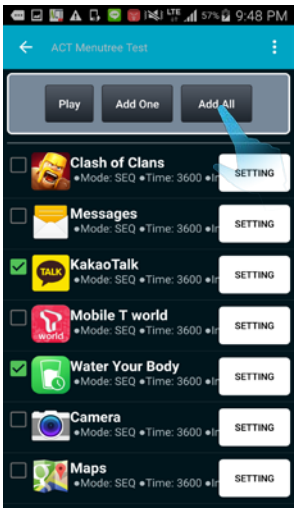
ACT obtains the composition info of the relevant screen automatically and analyses it and, touch by selecting randomly. Executes one of the Key-event randomly from back, menu, volume_up, volume_down after touching and, after that it also scrolls in the screen having list. It can detect issues in various scenarios unlike the sequential test.

▪ Method to execute non-sequential exploratory test

Auto exploration test button > Non-sequential exploration test > Add the desired App > settings > Play .

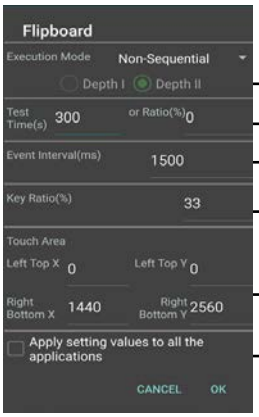
Example

Setting – 1 before the test



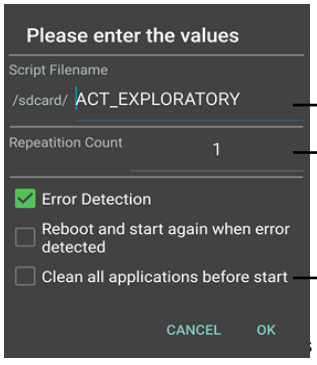
Select the app to be tested.
When there are many testing Apps,
Add all> select App>delete
When there is less testing App,
Add one>select App>confirm

Setting – 2 before the test



Set the depth of exploration.
Set the time of exploration
Set Sleep time between events.
Set the %age of key to be inputted during random_keypress.
Set the coordinates of screen to be touched at the time of random_tap.
Apply this settings to all added application

Setting – 3 before the test



Set the storage to save
Auto-created script.
Set total repetition.
Set whether to delete data whenever App executed.

Result of auto-script creation

4. Authentication Key

▪ Definition of authentication Key

When the user uses 'REC' feature in ACT2, a personal authentication key is needed which is made by combining employee number and name to prevent the leakage of personal info and that of ACT Tool. These are kind of file with pki extension.

▪ Key obtaining method

While preparing a site to issue key,

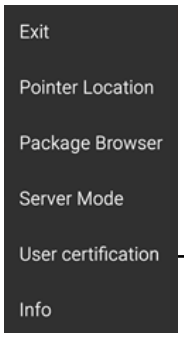
(It can be issues when requested by/as swgo.choi@samsung.com, s-r.choi@samsung.com temporarily.)

▪ Key authentication method

Save the issued authentication Key in terminal before using Rec feature.

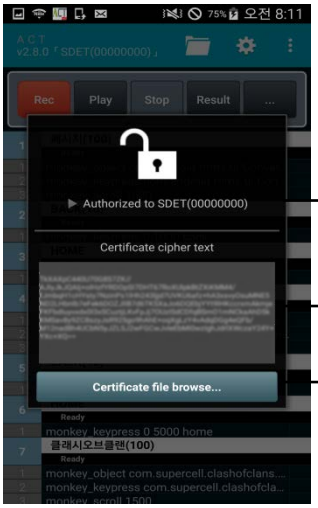
Example

Authentication - **1**



Select User's authentication

Authentication - **2**



User info appears Who has been authorized.

The encrypted personal key is brought

Brings in the personal Key

Authentication – 3

The authenticated personal info is displayed in ACT Main if authentication has been done.

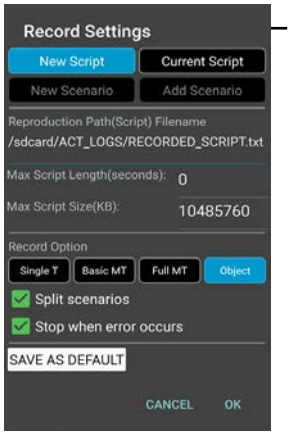
5. Recording

▪ Definition of recording

Recording feature is supported in ACT2. ACT records the motion of user and makes into script.
The made script can be played again. Key authentication has to be done already to use REC feature.
You can terminate by 'Shake Motion' (shaking the terminal 3~4 times strongly) while recording according to 'ACT Setting'.

▪ Recording

Start recording



If you start recording after authentication, the following setting screen appears.
You can added in currently saved file and setting of path etc. can be selected.

Single T	It creates the script of format like the previous ACT1. 'Multi Touch' can't be used.
Basic MT	Supports 'Multi Touch' and, creates the coordinates-based script.
Full MT	Supports Multi Touch and, creates coordinates-based script.
Object	Creates Object-based script and, supports WaitFor, Click Command.

※ Script created by recorded option – start REC > select Message

Single T

```
# Home
touch up 8 1292
touch down 761 2334
12 touch move 761 2334
36 touch move 761 2334
12 touch up 761 2334
```

Basic MT

```
# Home
mtouch 0 1 [ 0 769 2348 s=0.0275 ]
23 mtouch 2 1 [ 0 769 2348 s=0.0314 ]
60 mtouch 2 1 [ 0 769 2349 s=0.0314 ]
25 mtouch 2 1 [ 0 768 2350 s=0.0118 ]
12 mtouch 1 1 [ 0 768 2350 s=0.0118 ]
```

Full MT

```
# Home
mtouch 0 1 [ 0 767 2361 s=0.0314 mj=8 ]
72 mtouch 2 1 [ 0 767 2360 s=0.0314 mj=8 ]
85 mtouch 2 1 [ 0 767 2361 s=0.0314 mj=8 ]
12 mtouch 2 1 [ 0 767 2362 s=0.0118 mj=3 ]
11 mtouch 1 1 [ 0 767 2362 s=0.0118 mj=3 ]
```

Object

```
# Home
click t="₩Q메시지₩E"
c=android.widget.TextView
p=com.sec.android.app.launcher
k=y Pc=android.view.View
Ct="₩Q전화₩E"
Cc=android.widget.TextView
Sc=android.view.View

# 메시지
* run com.android.mms
 66 waitfor p=com.android.mms
 c=com.android.mms.ui.ConversationComposer
```

6. Analysis of test result - 1

▪ Analysis of test result

If ACT test is ended, **ACT_RESULT.xml and log are saved in** /sdcard/ACT_LOGS folder according to settings of ACT Setting. Scenario-wise **CPU usage, Memory usage, Network usage** and progress time of the relevant scenario is displayed in **ACT Main**.

Result > start~**end time total progress time, resource-status of entire scenarios are displayed by graph** in the test result.

Example

ACT Main screen

Scenario wise **Resource measurement result**

Time taken to proceed the relevant script.

CPU usage **RAM increased state (becomes red when increased)** **packet usage**

The no. of successful tests appear.

The no. of failed tests appear.

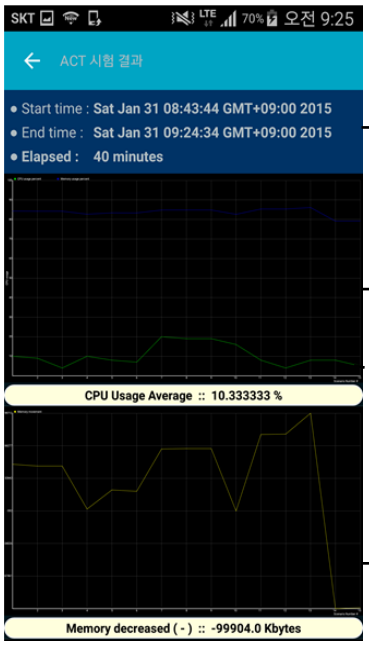
ACT_RESULT.xml file

You can see the detailed data for each testing scenario.
You can know starting time, ending time, no. of repetition and no. of successful, CPU usage, battery usage, memory usage, memory usage rate, packet usage etc.

6. Analysis of test result – 2

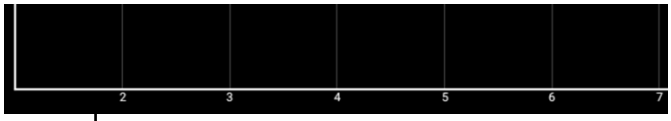
Example

Execution result screen 1



Starting time/ending time/time-taken appear.

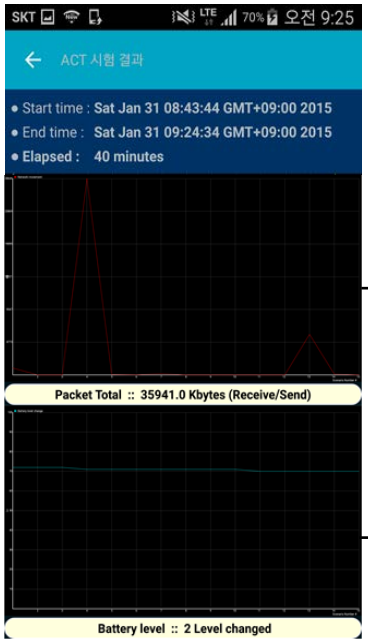
Scenario- wise CPU usage is drawn.



Scenario number is displayed in the bottom.

Scenario-wise relative Memory-variation is drawn.

Execution result screen 2



Scenario wise Network usage is drawn.

Scenario-wise battery-variation is drawn.

7. Sending Intent

▪ Definition of Intent sending

You can perform your desired action by sending intent to ACT from other App or adb command. They're load and execute ACT Script, pause during execution, re-execute etc. Both pause and re-execute can be used while recording or running the script.

▪ Intent type

Type	Intent	Parameter
Start ACT	**com.salab.act.intent.START_ACT**	SCRIPT_FILE or SCRIPT
	ACT is executed if the relevant intent is received and test is started automatically by bringing in the script file according to Parameter or that script is executed by inputting the command directly.	
Pause ACT	**com.salab.act.intent.PAUSE_ACT**	
	ACT gets paused if the relevant intent is received during the execution of ACT (Recording or while running Script)	
Re-execute ACT	**com.salab.act.intent.RESUME_ACT**	
	When ACT has been paused, it is re-executed from the paused part if the relevant Intent is received.	

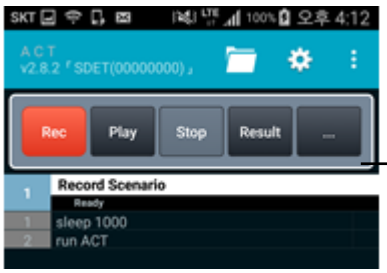
Example

Adb shell command

Operation screen

Adb Command

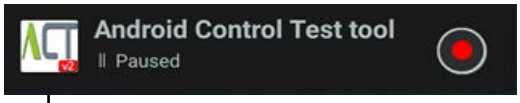
```
am broadcast –a com.salab.act.intent.START_ACT –e
/storage/emulated/0/ACTScript.txt
```



ACT is started and action is in progress by bringing in script file.

Adb Command

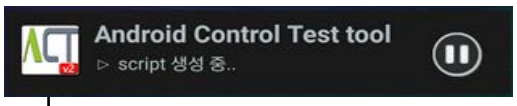
```
am broadcast –a com.salab.act.intent.PAUSE_ACT
```



ACT gets paused.

Adb Command

```
am broadcast –a com.salab.act.intent.RESUME_ACT
```



ACT is restarted.

8. Scenario & Comment and repetition

▪ Definition of Scenario and repetition

is used to separate event-wise scenarios. If **(no. of repetition)** is attached behind scenario, the relevant scenario is repeated as many times as the number inputted. If scenario part is long-pressed in ACT screen, you can perform 'execute only this scenario, execute from this scenario, edit, menu, end'.

If **(no. of repetition)** is attached behind file, the file is repeated as many times as the number inputted. Ex) Test(100).txt

▪ Definition of comment

It is written while processing the comment of script. The script starting with * becomes comment. It is not shown in ACT screen.

Example

Script

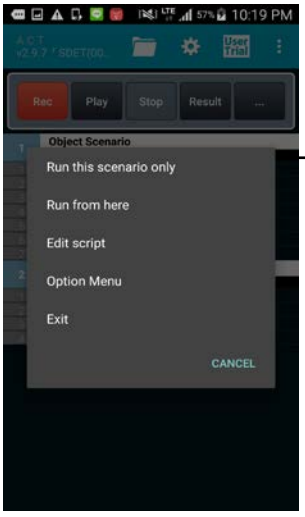
```
# Object Scenario
  sleep 1000
  object device storage
  * Go to my file
  sleep 1000
  press menu
  sleep 2000
  press home
  sleep 2000

# Scenario repeat (100)
  sleep 2000
  press back
  sleep 3000
  press home
```

Operating screen

scenario

comment is processed to enter into my files and it is not seen.

Repeats the relevant scenario 100 times



If you long-press the scenario part, the following pop up window appears.

9. Command

9.1 Key Event

▪ Definition of Key Event

It makes the key value (Parameter) inputted by the user to appear forcefully in the terminal.

In case of doing action like Long press, it sleeps till you press after key down and then you can key up.

▪ Parameter

[**Event type** Keycode or Keyname]

Input the Keycode(Keyname) which makes event to occur. All Key codes can be checked from 'android site'.

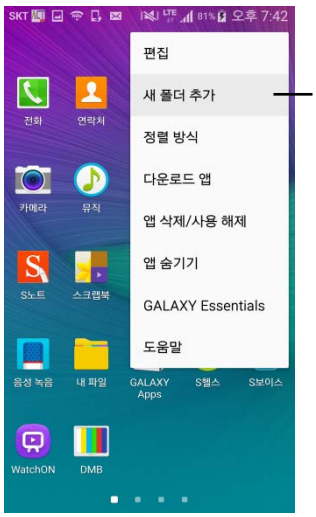
Type of Key event	Parameter	Meaning
key down	**Key Code**	**Press the inputted key..**
key up	**Key Code**	**Leave the inputted key**
press	**Key Code**	**Press and leave the inputted key sequentially.**
type	**Text to be inputted**	**Input the inputted text in terminal.**

Example

Script

```
# Key event Scenario
  sleep 1000
  press menu
  sleep 2000
  press dpad_down
  sleep 2000
  press dpad_down
  sleep 2000
  press dpad_center
```

Operating screen



Menu window

First press the menu key in screen and then focus 2 times to/by pad_down.
click by dpad_center after moving/changing.

9.2 Scroll

▪ Definition of Scroll

It moves the schedule-range of relevant screen up-down or left-right. It scrolls 90% of the entire screen in case of up-down or left-right. There are 2 kinds of Scroll Command – slow scroll and scroll with a bang (fast scrolling) **fling**.

▪ Parameter

[Horizontal : h Vertical : v] [front : f back : b]

In case of horizontal (h) script, f scrolls from right side to left side **and** b scrolls from left side to right side.

In case of vertical (v) script, f scrolls from bottom to top direction and b scrolls from top to bottom direction.

Example

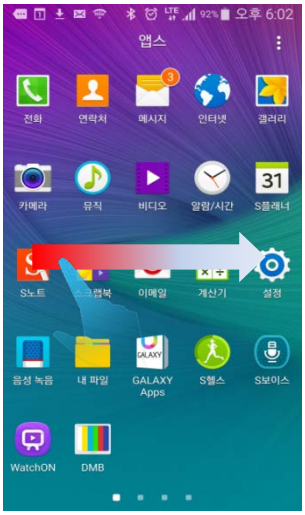
Script

```
# Scroll Scenario
  sleep 1000
  scroll v f
```

Operating screen

Slowly scroll the relevant screen from bottom to top.

```
# Scroll Scenario
  sleep 1000
  fling h b
```



Scroll the relevant screen fastly from left side to right side.

9.3 Touch Event

▪ Definition of Touch Event

It makes the 'Touch' to occur at the position corresponding to coordinates-value(Parameter) inputted by the user. g In case of doing action like Long press, it sleeps till you press after key down and then you can key up.

▪ Parameter

Relevant coordinates **ACT App** > Menu > you can use the coordinates displayed at the top after turning on the pointer display.

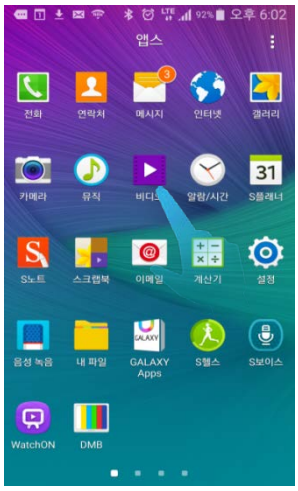
Type of Key event	Parameter	Meaning
touch down	**X Y coordinates**	**Press the inputted coordinates.**
touch up	**X Y coordinates**	**Leave the inputted coordinates.**
Tap	**X Y coordinates**	**Press and leave the inputted coordinates sequentially.**

Example

Script

```
# Touch Scenario
  sleep 1000
  tap 750 1450
```

Operating screen



It Touch down/up the part corresponding the coordinates inputted in the relevant screen.

```
# Touch move Scenario
  sleep 1000
  touch down 1000 1500
  50 touch move 1000 1350
  50 touch move 1000 1200
  50 touch move 1000 1050
  50 touch up 1000 900
```

※the numbers front of the commands means 'sleep'.

Move from touch-down part to touch –up part by dragging the Touch move coordinates similar to scrolling.

1000 900
1000 1050
1000 1200
1000 1350
1000 1500

9.4 Random Event

▪ Definition of Random Event

Random Event is provided for the commands which is needed to run randomly among the commands provided in ACT.
Especially random_tap and random_keypress are mainly used in non-sequential exploration test of ACT.

▪ Parameter

When Parameter is not inputted, the number to be inputted is also created randomly and gets executed.
In case of random_tap, command is executed by setting screen area to the full.
In case of random_keypress, back, menu, volume_up, volume_down are executed randomly.

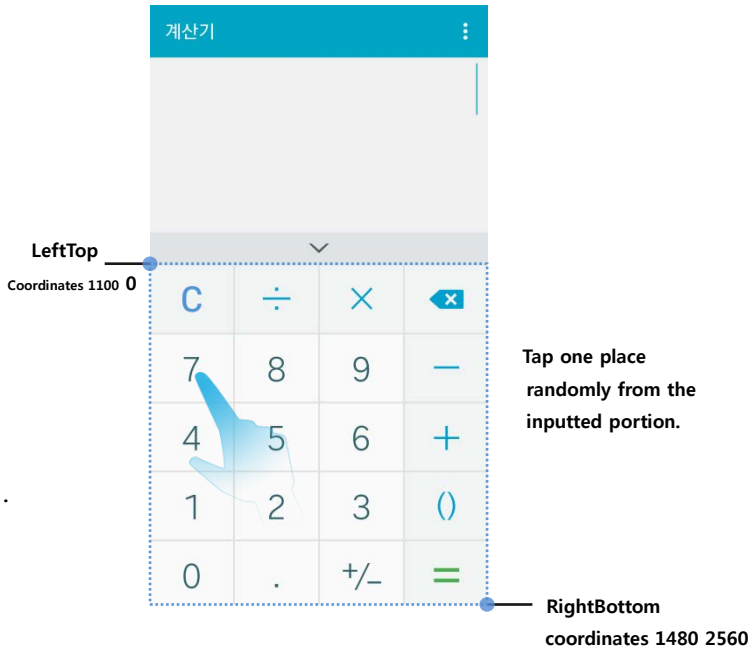
Send type	Parameter	Meaning
random_text	**Number**	**Input the random text as inputted number in terminal.**
random_number	**Number**	**Input the random number as inputted number in terminal.**
random_tap	**Tapping area**	**Random Tap inputted area.**
random_keypress	**The key you want to execute randomly**	**Input the random number inputted Keycodes**

Example

Script

```
#Random Event Scenario
  sleep 1000
  random_tap 1100 0 1480 2560
```

Operating screen



9.5 Object

▪ Definition of Object

Executing the touch action by searching the specific object (Parameter inputted by the user) of the screen without using the coordinates. It can be performed irrespective of User/Eng binary. When it is unable to find the relevant object, it makes to occur 'Object not found' error according to ACT settings and or, moves on to the next script.

▪ Parameter

[**object** text or id name]

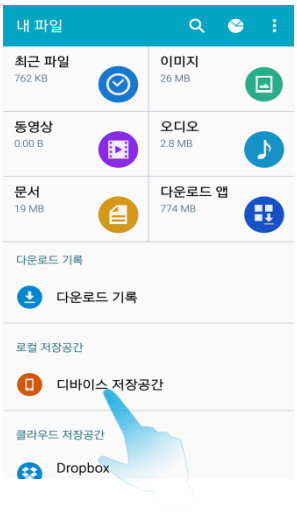
Input text or name of id of object to be touched by searching. It should be inputted exactly as it is shown in the screen (up to the word spacing).

Example

Script

```
# Object Scenario
  sleep 1000
  object device storage
```

Operating screen



Search the text 'device and click that text.

9.6 Click

▪ Definition of Click

Executing the touch action by searching the specific text (parameter inputted by the user) of the screen without using the coordinates. It can be performed irrespective of User/Eng binary. Unlike the Object, click is possible even when the new line character (₩n) is included and, the desired text can be clicked even when there is case like text according to Parameter.

▪ Parameter

[Touching Text ₩E" by searching **click** t="₩Q]

Input the text to be touched by searching.

In case of searching text from **'click** Parameter' it starts with **[t=]**. Also to express the text to be searched, it should be started with "₩Q and to inform the ending of searching text, it should be ended by ₩E".

Many other Parameters, other than text are inputted while recording but when the user uses by composing the script, he can perform the desired action just by using the above Parameter only.

Example

Script

```
# Click Scenario
  sleep 1000
  click t="₩QGALAXY Apps₩E"
```

Operating screen

Search the text 'GALAXY Apps' and click that text.

9.7 WaitFor

▪ Definition of WaitFor

Wait till the activity of specific package appears completely. It is similar to 'Sleep Command' but the timing issue of varying sleep-time depending on environment (Sluggish, network status etc) of the terminal whenever 'Sleep' is executed, can be resolved. 'WaitFor' is not a factitious time rather it proceeds the next script while waiting for complete ending by checking the status of screen.

▪ Parameter

[**waitfor** p= package name c= class name]

Input Package and Activity name to wait till it appears in the screen.

Start with **[t=]** in case of searching text from **click** Parameter. Also to express the searching text, it should be started with "₩Q and to know the ending of searching text, it should be closed by ₩E".

Many other Parameters, other than text are inputted while recording but when the user uses by composing the script, he can perform the desired action just by using the above Parameter only.

Example

Script

```
# WaitFor Scenario
  sleep 1000
  click t="₩QSplanner₩E"
  500 waitfor p=com.android.calendar c=com.android.calendar.AllInOneActivity
```

Operating screen

Click S planner by Click Command.

Next script proceeds continuously while waiting to appear AllInOneActivity of inputted calendar Package.

9.8 App & file execution

▪ Definition of App execution

Executing the app installed in terminal directly. Using '**run'** command.

▪ Parameter

[**run** App name or Package name]

Input the name of App you want to execute or Package name.

Name and Package of relevant app can be searched from **ACT App** > **menu** > **installed App list**.

▪ Definition of file execution

It executes the file inputted by user. '**execute'** command is used. The most suitable App to execute the file is selected from the terminal by using MIME Type provided by Android.

▪ Parameter

[entire path to **execute** file]

Input the entire path of the file which you want to run.

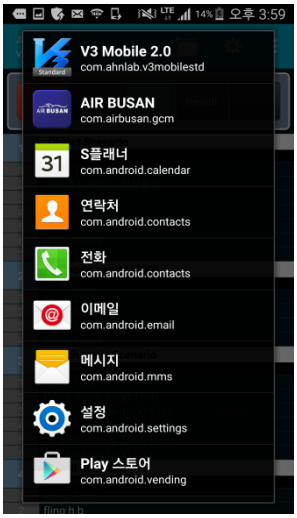
Example

Script

```
#Run Scenario
    sleep 1000
    run ACT
    sleep 1000
    press back
    sleep 1000
    run com.salab.act
```

```
#Execute Scenario
    sleep 1000
    execute /sdcard/Script.txt
```

Operating screen



List of installed App

9.9 Quick Panel settings

▪ Definition of Quick Panel settings

The main values in Quick Panel of the terminal is turned on/off by Script. The panel settings linked with network are excluded and if the settings are changed, it can affect the other settings.

▪ Parameter

[Set setting value or check Setting value]

Turn on/off by using the value of **Setting type** in below table. It can be checked by 'check command' that whether the relevant Panel value is turned on/off normally.

.

Setting type	settings	meaning
wifi	**on / off**	**Turns on/off the WiFi of terminal.**
bluetooth	**on / off**	**Turns on/off the Bluetooth of terminal.**
rotation	**on / off**	**Turns on/off the screen rotation of terminal.**
vibrate	**normal / vibrate / silent**	**Changes by ringtone/vibration/silent for the terminal.**
airplane	**on / off**	**Turns on/off the airplane mode of terminal.**
gps	**on / off**	**Turns on/off the position of terminal.**
multiwindow	**on / off**	**Turns on/off the multiwindow mode of terminal.**
powersaving	**on / off**	**Turns on/off the power saving mode of terminal.**
smartstay	**on / off**	**Turns on/off the smart stay of terminal.**
brightness	**0~255**	**Adjusts the brightness of terminal-screen.** **※Check not supported**

Example

Script

```
# Quick Panel   Scenario
  sleep 1000
  wifi on

# Quick Panel   Scenario
  sleep 1000
  gps on check
```

Operating screen

Wifi is on, GPS is on and checks whether it is turned on actually. Moves on to other script after checking that it is turned on correctly and working.

9.10 Installing and removing App & authority

■ Definition of installing and removing App

Installing internally force-installs the APK saved in internal memory of the terminal.

Removing removes the App installed in terminal forcibly.

■ Parameter

Put the info of App to be installed and deleted. You can search the **Package** of App from **ACT App** > menu > **list of installed App.**

Install	Delete
install	**uninstall.**
Full Path saved by APK	**Package name**
install /sdcard/ACT2_v.2.80_user.apk	**uninstall com.salab.act**

■ Definition of authority

It checks all the Apps saved in terminal and the app using the right to take the personal information and arranging all those rights, save in text file in /ACT_LOGS/ folder.

■ Parameter

[**permission**]

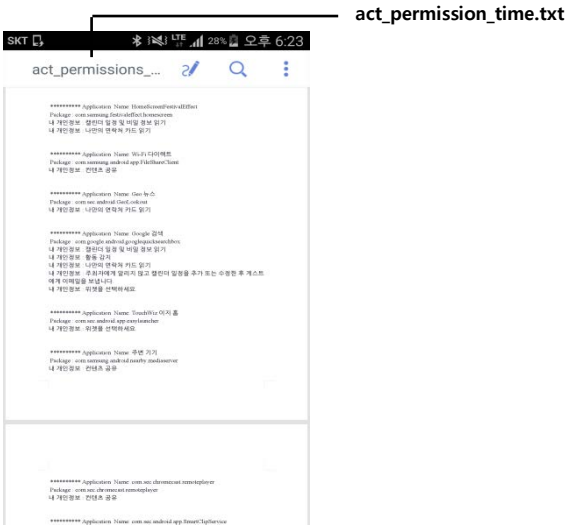
Parameter is not needed in right/authority.

Example

Script

```
#Permission Scenario
    sleep 1000
    permission
```

Operating screen

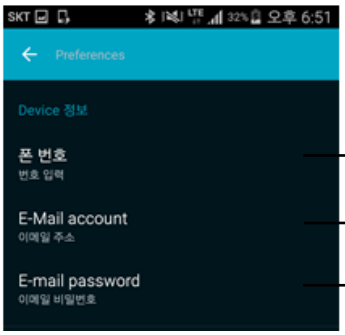


9.11 Send

▪ Definition of Sending

Makes call to the number inputted in terminal and, sends the inputted contents as SMS Email.

Phone no. (auto-input when SIM is detected), Email address (G-Mail) password must be set in device info of ACT Setting before using the relevant command.



Self phone number. (auto-inputted when SIM is detected.)

Input Email account. **※Gmail is essential.**

Input the password of Email account.

▪ Parameter

Blank input is not permitted in details and, 'ACT Test Message' detail is sent automatically when the required detail is not inputted.

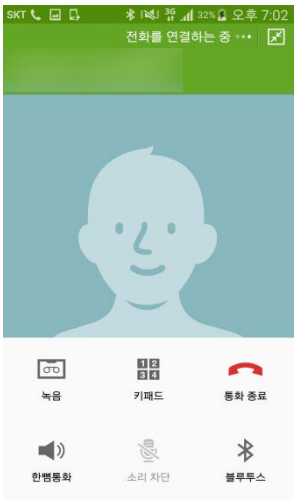
Send type	Parameter	meaning
sendcall	Phone number	Makes call to the inputted number.
retrysendcall	Phone number	Makes call till the inputted number receives.
endcall		Ends the call.
sendsms	Phone number details	Sends the contents to inputted no. as SMS.
sendemail	Mail address details	Sends the contents to inputted mail address as Mail.

Example

Script

```
#Send Scenario
  sleep 1000
  sendcall 01000003082
```

Operating screen



Sends directly without dialing.

9.12 Library execution

▪ Definition of Library execution

Re-usability of script can be raised by calling and executing on script after making the specific script as/by library. Library should exist outside the scenario and declared as/by '**@**'. The declared library can't be seen in ACT screen but, it can be checked and modified by going to edit mode.

▪ Parameter

[@ declared Library name]

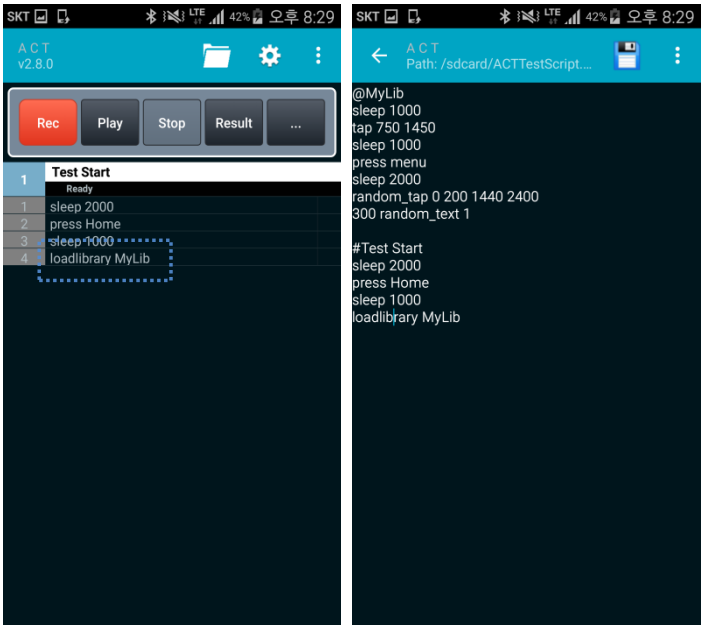
Write the library name written as/by @.

Example

Script

```
@MyLib
  sleep 1000
  tap 750 1450
  sleep 1000
  press menu
  sleep 2000
  random_tap 0 200 1440 2400
  300 random_text 1

#Library Start
  sleep 2000
  press Home
  sleep 1000
  loadlibrary MyLib
```

Operating screen



Library can't be checked from ACT main screen but it can be checked in/from edit mode.

9.13 Wait Event

▪ Definition of Wait Event

When any specific event is required by the terminal, it can be made to occur when the user wants by using 'Wait Event'.

Wait Event is set for server phone and event is sent to the testing terminal from server end. The testing terminal waits continuously till event comes.

Testing terminal side

Input server-end phone number in testing terminal.

Server end

Run **ACT App** > **menu** > **server mode.**

▪ Parameter

Next script is processed when event does not comes till the max. inputted time.

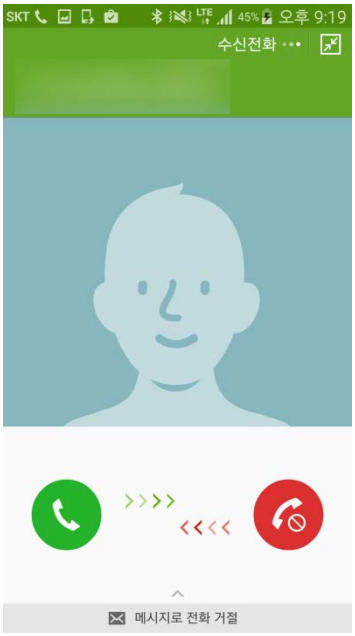
Wait type	Parameter	meaning
wait call	**Max. waiting time (ms)**	**Requests to make call to server phone and waits till it is received.**
wait endcall		**Waits till the server phone ends the call.**
wait sms	**Max. waiting time (ms)**	**Requests to send SMS to server phone and waits till it is received.**
wait email	**Max. waiting time (ms)**	**Requests only to send email to server phone.**
wait alarm	**Max. waiting time (ms)**	**Waits till Alarm rings ※set alarm in test-phone**
wait download	**Max. waiting time (ms)**	**Waits till the download the completed while downloading App from market.**

Example

Script

```
#Wait Scenario
  sleep 1000
  wait call
```

Operating screen



Wait till call is made from the server phone to test-phone.

9.14 Sound Recording

▪ Definition of Sound Recording

The user can record the sound coming out from the terminal till he wants. It is recommended not to exceed max. 1 minute while saving. The saved file is saved in 'ACT_LOGS folder' and it is saved in 'Sound_time.amr' format.

▪ Parameter

[**record_start** or **record_start path**]

Record_start and record_stop should always be done by one set and, recording time can be controlled by sleep.

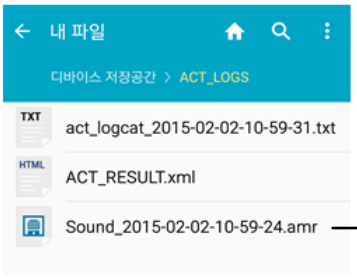
Send type	Parameter	meaning
record_start	**Blank space / path**	**It is saved by the relevant name when inputted by Parameter in ACT_LOGS folder and if not so, then saved as sound_time.amr.**
record_stop		**The inputted number(call) is send till it is received.**

Example

Script

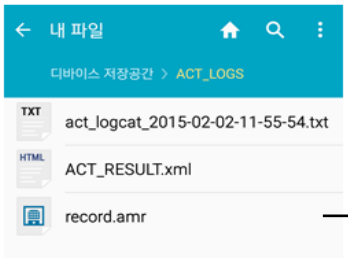
```
#Record Scenario
  sleep 1000
  record_start
  sleep 5000
  record_stop
```

Operating screen



If filename is not inputted, Saved as 'Sound_time.amr'.

```
#Record Scenario
  sleep 1000
  record_start record.amr
  sleep 5000
  record_stop
```



If filename is inputted, Saved in ACT_LOGS folder By the inputted name.

9.15 Screentext

▪ Definition of Screentext

The command inputted by user is executed if the relevant text is present in the screen by searching the specific text (Parameter inputted by the user) when the screen changes. When the relevant Text could not be found, it waits endlessly. It can be run irrespective of User/Eng binary.

▪ Parameter

[**screentext** { text1 : command 1} { text2 : cmd2 } ...]

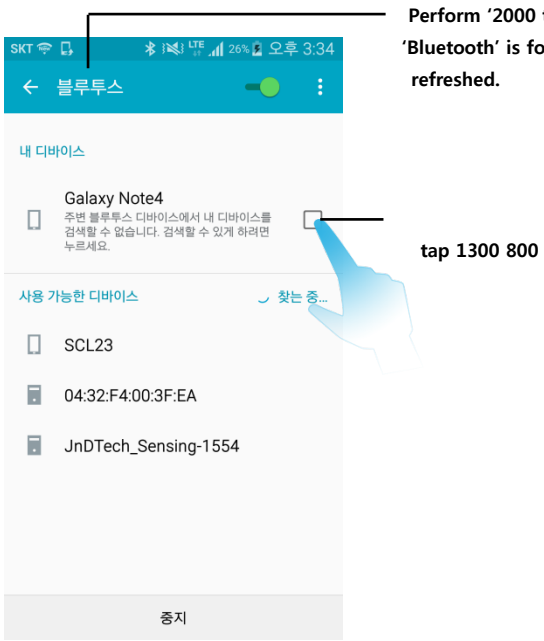
Input the text to be searched. It should be inputted exactly as it is shown (upto the word-spacing) in the screen.

Example

Script

```
# Screen Text Scenario
  sleep 1000
  run setting
  sleep 1000
  tap 770 2000
  screentext { bluetooth : 2000 tap 1300 800 }
  2000 press home
```

Operating screen



9.16 Saving Log & screen capture

■ Definition of saving log

Save the log of AP like Dumpsys, Dumpstate, Logcat etc. Save to/as logcat_time.txt and bugreport_time.txt after creating /sdcard/ACT_LOGS/_execute-date-time folder.

■ Parameter

[**bugreport**]

Parameter is not needed in saving LOG.

■ Definition of screen capturing

The user executes the inputted file.

The most suitable App to run the file gets selected from terminal by using MIME Type which is provided by/in Android.

■ Parameter

[path to save **screenshot** file]

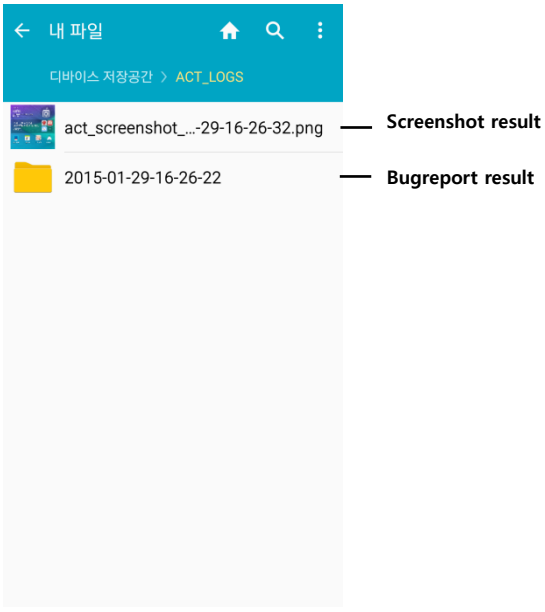
Input the entire path of the position where you want to save. It gets saved in /sdcard/ACT_LOGS folder when path is not inputted.

Example

Script

```
#Save Scenario
  sleep 1000
  bugreport
  sleep 10000
  screenshot
  sleep 5000
```

Operating screen



9.17 Intent & rebooting

■ Definition of sending Intent

The basic/default intent of Android can be sent through ACT.

■ Parameter

[**intent** intent name]

. All the basic/default Intent provided in Android can be used. It can be checked from 'Android Developer Intent'.

■ Definition of rebooting

If it meets the 'Reboot command', rebooting is done and, if rebooting gets done, the subsequent script proceeds continuously. It can be set to proceed continuously after the scheduled time by using delay of Setting.

■ Parameter

[**reboot**]

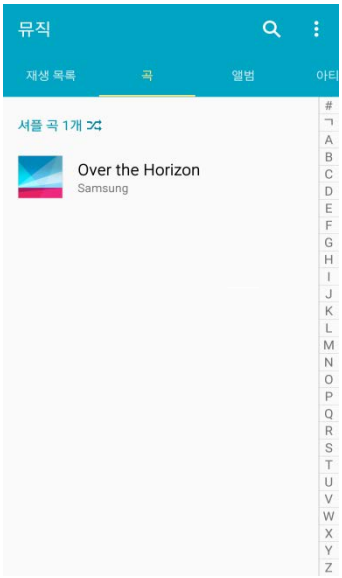
Parameter is not needed in rebooting.

Example

Script

```
#Intent Scenario
  Sleep 1000
  Intent android.intent.action.MUSIC_PLAYER
```

Operating screen



Music Player gets executed.

10. ACT Setting

Definition of ACT setting

ACT Menu > the values set in Setting, can be changed on script. It can be helpful when test has to be done in same condition.

Setting	Script Value	Parameter
Phone number	**$PhoneNumber**	Phone number
Input the no. of terminal. It gets inputted automatically when terminal detects the SIM.		
E-Mail account	**$EMail**	**Email address**
Input while using Email related Command. Only Gmail(@gmail.com) is applied and entire address need to be inputted.		
E-Mail password	**$EMailPW**	**Email password**
Input the password of inputted Email account.		
Phone number at server side	**$ServerNumber**	Phone number
Input the server-side number to be used during Wait Command.		
Crash/ANR detection	**$ErrorDetect**	**true/false**
Save Log and screen-shot as per settings in /ACT_LOGS/errorApp_time folder when ANR or Crash has occurred.		
Whether to ignore script error	**$IgnoreError**	**true/false**
Even if False appears or, there is error in script during the test, it is ignored and next script proceeds.		
Whether to save screenshot	**$SaveScreenshot**	**true/false**
Save screenshot when Crash/ANR has been detected. (Crash/ANR should be turned on.)		
Whether to reboot phone	**$RebootWhenErrorOccurs**	**true/false**
Reboot after completion of Log saving when Crash/ANR has been detected. (Crash/ANR should be turned on.)		
Whether to save Logcat	**$SaveLogcat**	**true/false**
save Logcat in /ACT_LOGS folder after all ACT script is run/proceeded.		
Saving result	**$SaveResultAlways**	**true/false**
Save 'ACT_LOGS/ACT_RESULT.xml' after proceeding of entire ACT script. Scenario-wise time-taken, memory usage, no. of repetition, CPU usage, battery usage etc. can be checked.		
Resource measurement	**$ResourceTracing**	**true/false**
Save CPU, Packet, memory, battery usage whenever each command is executed.		
CPU measurement min. time	**$CPUTraceMinTime**	**time(ms)**
Set the min time to check CPU usage. (Resource Tracing should be turned on.)		
auto start of ACT script after booting	**$AutoStart**	**true/false**
It proceeds from next script after booting is completed when rebooting is done without the ending of test.		
Waiting time after booting	**$DelayTimeAfterReboot**	**time(ms)**
It gives delay time before auto-starting of ACT Script when rebooting is done.		
Big start button	**$BigStartButton**	**true/false**
Start button happens to appear in full screen.		
End point	**$BatteryEndLevel**	**Number**
ACT test is ended automatically when remaining battery level of the terminal becomes the same as the value inputted by user. It is 1%, 5%(default value), 15%, 30%, 50%. 80%, 100%(for charging test), 0% in actual settings.		
Standard temperature	**$TemperatureDetection**	**Number**
Input the temperature which is standard for detection. Detect on the basis of AP temperature and it brings in battery temperature when not supported.		

Setting	Script Value	Parameter
Temperature check interval	**$TemperatureMinTime**	**Time(ms)**
Set the min. tome to check temperature.		
Sleep permission	**$AllowDeepSleep**	**true/false**
If the sleep time of terminal is longer than **screen auto turn off time,** terminal state can be set between Dimming and Deep Sleep.		
Motion detecction	**$ShakeAction**	**true/false**
ACT can be stopped if terminal is shaken 3~4 times strongly while it is running (Rec or script execution).		
Touch coordinates and script display	**$PointerLocation**	**true/false**
The currently running script and touch-coordinates are shown on screen-top.		
Display of progress state	**$NotificationCheck**	**true/false**
Progress status of current script is displayed in notification.		
Mixing the scenario execution order	**$RandomScenario**	**true/false**
Perform by mixing the execution order of scenario randomly.		
ACT WatchCat	**$ACTWatchCat**	**true/false**
Checks the state to get ACT executed normally and, when it is terminated unintentionally, it makes ACT to run again.		
ACT WatchCat interval	**$ACTWatchCatInterval**	**time(ms)**
ACT WatchCat is inputted for the time to check the status of ACT.		