

App Inventor 2: Mobile Forensic Toolkit

20166405 박성미
20146425 정연석



01.

해쉬란?

02.

Mobile Forensic
Toolkit

03.

앱인벤터
확장기능 만들기

무결성의 원칙

- 수집 증거가 위·변조 되지 않았음을 증명
 - 수집 당시의 데이터 hash 값과 법정 제출 시점 데이터의 hash 값이 같다면 hash 함수의 특성에 따라 무결성을 입증

디지털 포렌식 개관, 박종혁 (2013)



<http://my.bryanqiang.com/createext.html>

Tools you will need:

1. Git Bash (a software for download git code sources)
2. JDK (java 7)
3. a java code editor (Jedit, EditPlus, etc)
4. basic java programming experience
5. Apache ant (a software for compile extensions)

<https://docs.google.com/document/d/1xk9dMfczvjbwD-wMsr-ffqkTIE3ga0ocCE1KOb2www/pub#h.3kn0fmfydnq1>

1b. The Class Hierarchy

All components fit within a class/interface hierarchy, a subset of which is shown in Figure 2. All components implement the `Component` interface, which consists mostly of useful constants. Every component extends either the `VisibleComponent` or `AndroidNonVisibleComponent` abstract classes, and some also implement the `ComponentContainer` interface. `Form` (displayed to the user as "Screen"), which holds zero or more instances of `AndroidViewComponent`, the super components. Exceptions include `Sprite`, which can only be contained in a `Canvas`.

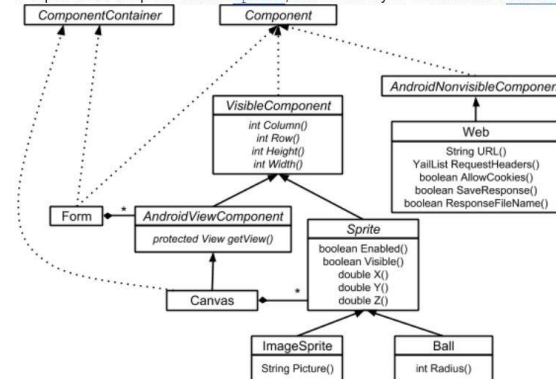



























Figure 2: An incomplete subset of the component class/interface hierarchy. Interfaces and abstract methods are shown in *italics*.



ngmi > appinventor-sources > appinventor > components > build > extensions

ngmi > appinventor-sources > appinventor > components > build > extensions			ngmi > appinventor-sources > appinventor > lib		
Name	Date modified	Type	Name	Date modified	Type
 com.sungmi.binaryConverter.aix	11/25/2018 6:22 PM	AIX Fil	 gwt_incorper	11/15/2018 8:45 PM	File folder
 com.sungmi.runtimeLogs.aix	11/25/2018 6:22 PM	AIX Fil	 gwt_incubator	11/15/2018 8:45 PM	File folder
 com.sungmi.sungmiHasher.aix	11/25/2018 6:22 PM	AIX Fil	 gwt_query	11/15/2018 8:45 PM	File folder
			 gwt_svg	11/15/2018 8:45 PM	File folder
			 jdk5	11/15/2018 8:45 PM	File folder
			 jedis	11/15/2018 8:45 PM	File folder
			 json	11/15/2018 8:45 PM	File folder
			 jts	11/15/2018 8:45 PM	File folder
			 junit	11/15/2018 8:45 PM	File folder
			 junit4	11/15/2018 8:45 PM	File folder
			 junit-addons	11/15/2018 8:45 PM	File folder
			 kawa	11/15/2018 8:45 PM	File folder
			 keyczar	11/15/2018 8:45 PM	File folder
			 leaflet	11/15/2018 8:45 PM	File folder
			 log4j	11/15/2018 8:45 PM	File folder
			 oauth	11/15/2018 8:45 PM	File folder
			 objectify-3.1	11/15/2018 8:45 PM	File folder
			 org-apache-commons-codec	11/22/2018 4:53 AM	File folder
			 org-apache-commons-io	11/20/2018 10:48 PM	File folder

ngmi > appinventor-sources > appinventor > components >

Name
 binaryConverter
 runtimeLogs
 sungmiHasher

SungmiFileHasher.java

```
1 package com.sungmi.sungmiHasher;
2
3 //import libraries
4
5 import android.content.Context;
6 import android.util.Log;
7 //import android.app.Activity;
8 //import android.os.Environment;
9
10 import com.google.appinventor.components.annotations.*;
11 import com.google.appinventor.components.common.ComponentCategory;
12 import com.google.appinventor.components.runtime.*;
13 import com.google.appinventor.components.runtime.collect.Sets;
14 import com.google.appinventor.components.runtime.util.BoundingBox;
15 import com.google.appinventor.components.runtime.util.ErrorMessages;
16 import com.google.appinventor.components.runtime.util.FileUtil;
17 import com.google.appinventor.components.runtime.util.MediaUtil;
18 import com.google.appinventor.components.runtime.util.PaintUtil;
19 import com.google.appinventor.components.common.ComponentConstants;
20 import com.google.appinventor.components.common.PropertyTypeConstants;
21 import com.google.appinventor.components.common.YaVersion;
22 import org.apache.commons.codec.DecoderException;
23 import org.apache.commons.codec.binary.Hex;
24 import org.apache.commons.codec.digest.DigestUtils;
25
26 import java.io.FileNotFoundException;
27 import java.io.FileInputStream;
28 import java.io.InputStream;
29 import java.io.IOException;
30
31
32 ////////////////annotations for basic initialization////////////////////
33
34 @DesignerComponent(version = YaVersion.SUNGMI_TEST_VERSION,
35     description = "my testings", //this is your extension's description
36     category = ComponentCategory.EXTENSION, //the category is extension
37     nonVisible = true, //only can be nonvisible
38     iconName = "images/web.png") //picture path
39
40 @SimpleObject(external = true)
41 @UsesLibraries(libraries = "org.apache.commons.codec.jar")
42 @UsesPermissions(permissionNames = "android.permission.WRITE_EXTERNAL_STORAGE, android.permission.READ_EXTERNAL_STORAGE")
43
```



```
public class SungmiFileHasher extends AndroidNonvisibleComponent //set your class name in XXX for replacement.
implements Component {
```

```
    public static final int VERSION = 0; //version number
    private ComponentContainer container; //define the component
    private Context context; //define the context
    private static final String LOG_TAG = "sungmi EXTENSION STARTS!!!!";
    private DigestUtils DigestUtils;
    private Hex Hex;
    private DecoderException DecoderException;
    //private final Activity activity;
    //private boolean isRepl = false;
```

```
    public SungmiFileHasher (ComponentContainer container) {
        super(container.$form());
        this.container = container;
        context = (Context) container.$context(); //define the context
        Log.d(LOG_TAG, "sungmiHasher created" ); //record
    }
```

```
////////////////////////////////////add your code here////////////////////////////////////
```

```
@SimpleFunction(description = "reads file as binary and hashes with md5")
```

```
public String SungmiMD5(String fileName) {
    String md5String = null;

    //String md5String = null;
    try {
        //md5String = DigestUtils.md5Hex(new FileInputStream(fileName));
        md5String = new String(Hex.encodeHex(DigestUtils.md5(new FileInputStream(fileName))));




































        //String s = new String(Hex.encodeHex(DigestUtils.md5(data)));
    } catch (IOException e){
        Log.d(LOG_TAG, "IO exception");
    }
    Log.d(LOG_TAG, "md5String");
    return md5String;
}
```




```
Sungmi@DESKTOP-FQTSSL MINGW64 ~/appinventor-sources/appinventor (master)
$ ant extensions
Buildfile: C:\Users\Sungmi\appinventor-sources\appinventor\build.xml

extensions:
```

```
BUILD SUCCESSFUL
Total time: 3 minutes 3 seconds
Picked up _JAVA_OPTIONS: -Xmx1024m
```

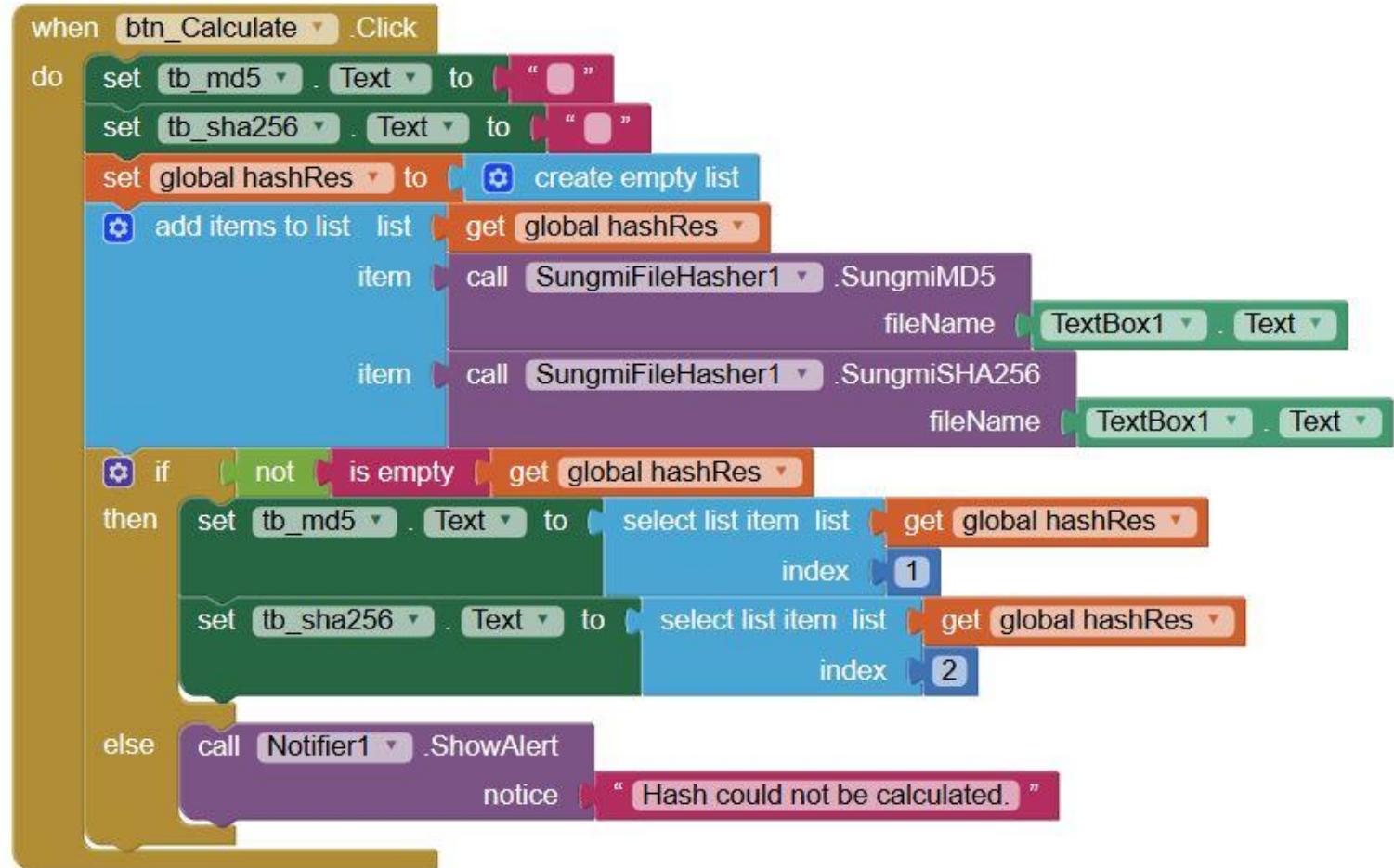
Extension		
<u>Import extension</u>		
	MemoryInfo	 
	KIO4_GetAllIp	 
	ClientUDP	 
	NetworkTools	 
	SpecialTools	 
	PhoneInfo	 
	SungmiFileHasher	 
	TaifunFile	 
	TaifunTM	 
	TaifunWiFi	 
	RootChecker	 
	UrsAI2UDP	 

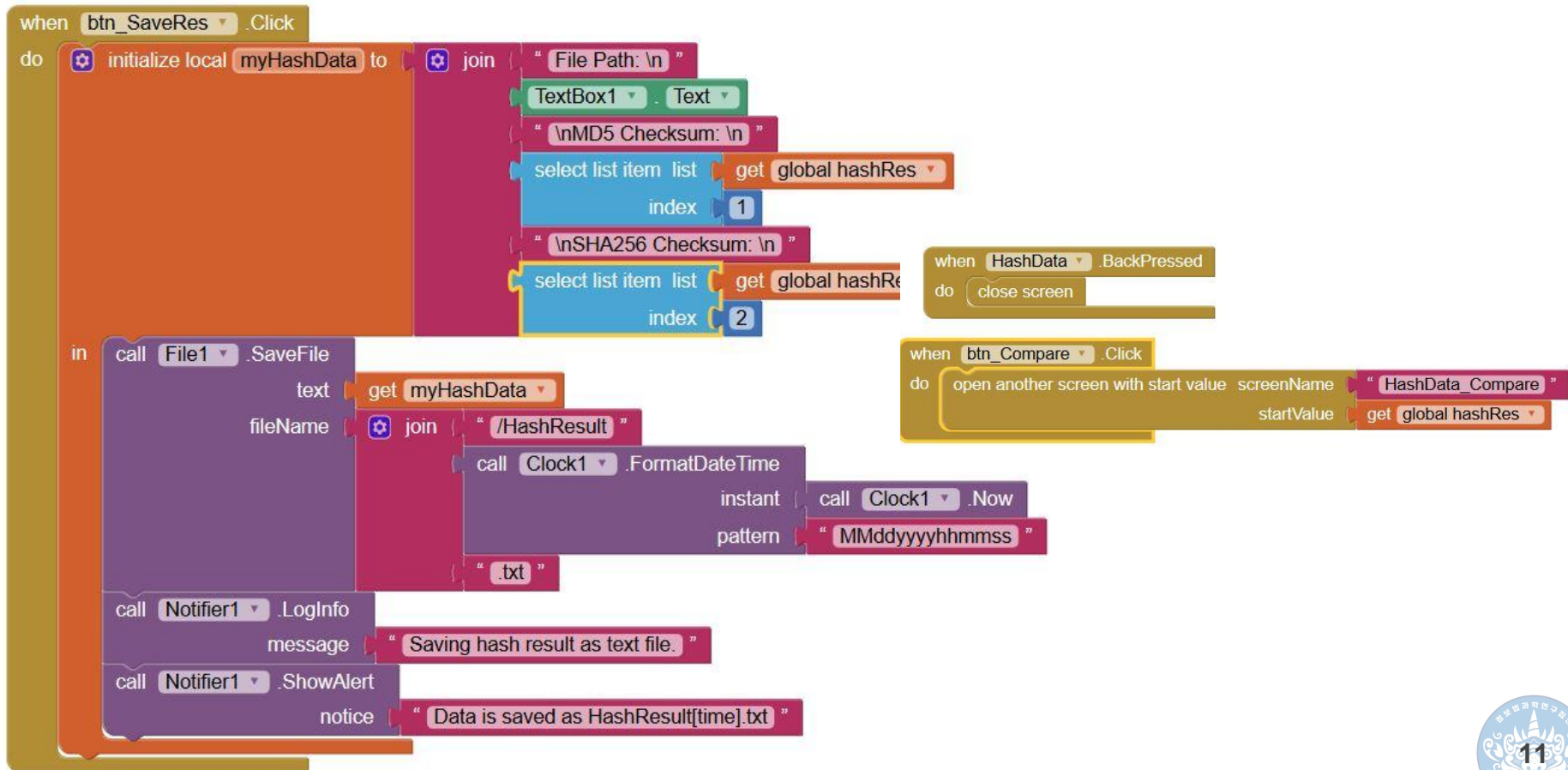



```
when btn_SelectFile .Click
do
  set ActivityStarter1 . Action to " android.intent.action.GET_CONTENT "
  set ActivityStarter1 . DataType to " */* "
  if is empty call ActivityStarter1 .ResolveActivity
  then call Notifier1 .ShowAlert
    notice " There is no file explorer application. "
  else call ActivityStarter1 .StartActivity
```

```
when ActivityStarter1 .AfterActivity
  result
do
  set ActivityStarter1 . DataUri to ActivityStarter1 . ResultUri
  set TextBox1 . Text to call TaifunFile1 .GetFileName
    contentUri ActivityStarter1 . DataUri
```

initialize global hashRes to create empty list



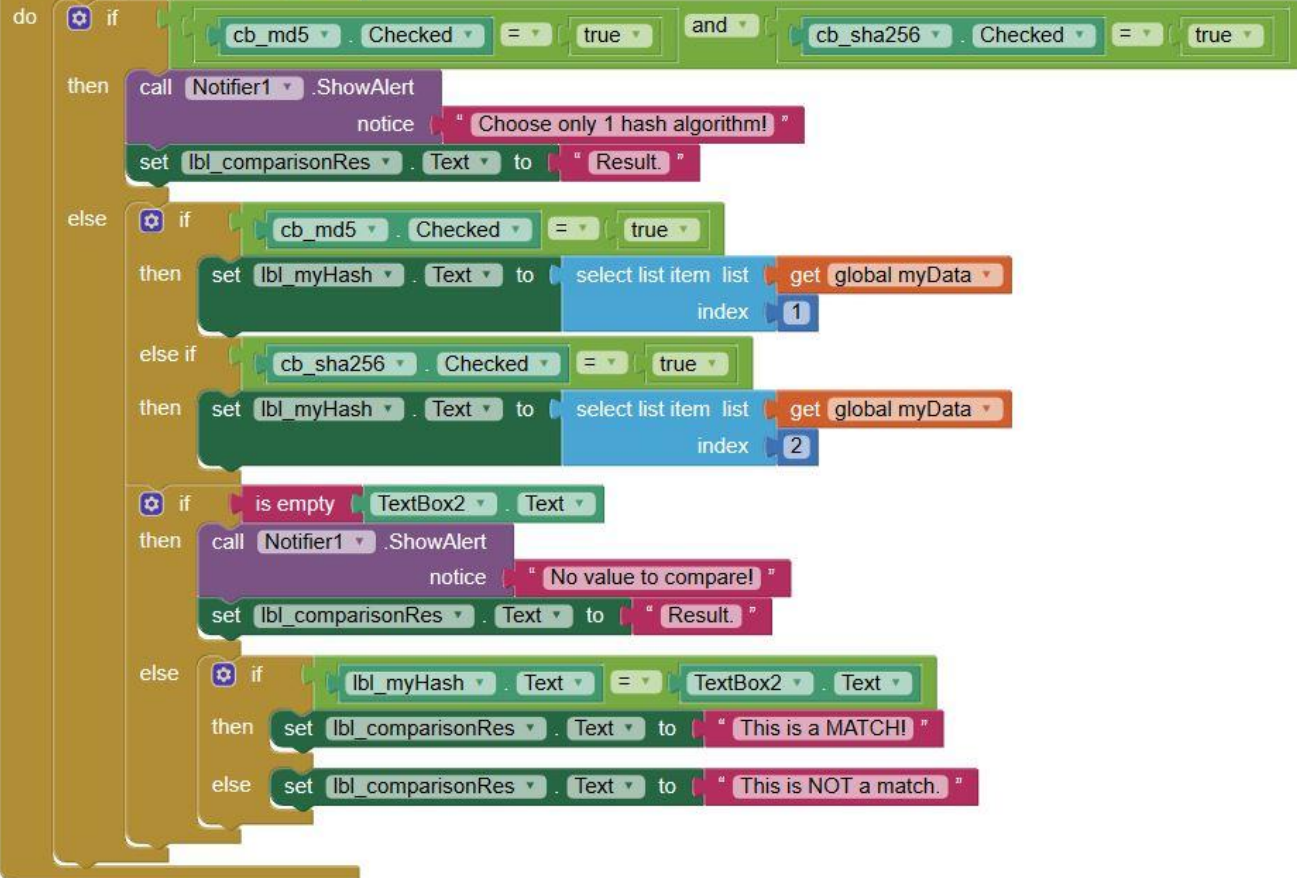


```
when HashData_Compare ▾ .BackPressed  
do close screen
```

```
initialize global myData to get start value
```

```
when HashData_Compare ▾ .Initialize  
do  
  set cb_md5 ▾ . Checked ▾ to false ▾  
  set cb_sha256 ▾ . Checked ▾ to false ▾  
  set lbl_myHash ▾ . Text ▾ to "..."  
  set TextBox2 ▾ . Text ▾ to ""  
  set lbl_comparisonRes ▾ . Text ▾ to "Result."
```

when btn_CompareHex .Click



네트워크 포렌식

- 기기의 연결 상태 및 상세 정보를 제공

자동으로 연결되는 AP와 AP의 맥주소를 찾고,

추후 AP에 대한 로그 감식을 할 수 있다.




































호스트 스캐닝을 통해 지정한 아이피 대역에 있는 호스트를 검색한다.

시스템 정보 조회

- 기기의 하드웨어 리비전 및 소프트웨어 버전 제공

하드웨어 버전, 안드로이드API, 부트로더, 루트유무, SIM 및 셀룰러 정보,
소프트웨어 버전, RAM / ROM 사용량 등등

Import extension

	NetworkTools	 
	SpecialTools	 
	PhoneInfo	 
	MemoryInfo	 
	ClientUDP	 
	KIO4_GetAllIp	 
	RootChecker	 
	TaifunWiFi	 
	TaifunTM	 
	TaifunFile	 
	SungmiFileHasher	 
	UrsA12UDP	 

Extension

- 다양한 서드파티 확장 기능 존재

자바 기반으로 작성된 *.aix 파일들.

개인 개발자 및 단체 개발자들이 다수 제작.

유료화모델도 존재.

ex) UDP Server 및 System Log의 경우 추가 비용을 지불해야 함.

(실제 기능에서 배제)

when btn_OpenSysInfo .Click

do if call RootChecker1 .IsRooted = false

then set lblroot1 .Text to " Not Rooted "

set lblroot1 .TextColor to blue

else set lblroot1 .Text to " Rooted "

set lblroot1 .TextColor to red

set lblversion .Text to PhoneInfo1 .AndroidAPI

set lblram .Text to join call MemoryInfo1 .MemoryFree
" MB / "
call MemoryInfo1 .MemoryTotal
" MB "

set lblname .Text to PhoneInfo1 .DeviceModel

set lblmaker .Text to PhoneInfo1 .DeviceManufacturer

set lblinternal .Text to join round call MemoryInfo1 .InternalStorageAvailable / 1000000
" MB / "
round call MemoryInfo1 .InternalStorageTotal / 1000000
" MB Available "

set lblexternal .Text to join round call MemoryInfo1 .InternalStorageUsed / 1000000
" MB / "
round call MemoryInfo1 .ExternalStorageAvailable / 1000000
" MB Available "

if PhoneInfo1 .AndroidAPI < 23

```

if PhoneInfo1.AndroidAPI ≤ 7
then set lblandroid.Text to "Android 2.1 or Under"
else if PhoneInfo1.AndroidAPI = 8
then set lblandroid.Text to "Android 2.2"
else if PhoneInfo1.AndroidAPI ≤ 10
then set lblandroid.Text to "Android 2.3 ~ 2.3.7"
else if PhoneInfo1.AndroidAPI ≤ 13
then set lblandroid.Text to "Android 3.0 ~ 3.2"
else if PhoneInfo1.AndroidAPI ≤ 15
then set lblandroid.Text to "Android 4.0 ~ 4.0.4"
else if PhoneInfo1.AndroidAPI ≤ 18
then set lblandroid.Text to "Android 4.1 ~ 4.3"
else if PhoneInfo1.AndroidAPI ≤ 20
then set lblandroid.Text to "Android 4.4"
else if PhoneInfo1.AndroidAPI ≤ 22
then set lblandroid.Text to "Android 5.0 ~ 5.1"
else if PhoneInfo1.AndroidAPI ≤ 23
then set lblandroid.Text to "Android 6.0"
else if PhoneInfo1.AndroidAPI ≤ 25
then set lblandroid.Text to "Android 7.0 ~ 7.1.2"
else if PhoneInfo1.AndroidAPI ≤ 27
then set lblandroid.Text to "Android 8.0 ~ 8.1"

```

```

when Specific.BackPressed
do close screen

```

```

when Specific.Initialize
do
  set lbl_kernel.Text to SpecialTools1.KernelVersion
  set lbl_buildnumber.Text to SpecialTools1.BuildNumber
  set lbl_baseband.Text to SpecialTools1.BasebandVersion
  set lbl_bootloader.Text to SpecialTools1.BootloaderVersion
  set lbl_serial.Text to SpecialTools1.SerialNumber
  set lbl_country.Text to SpecialTools1.CountryCode
  set lbl_lang.Text to SpecialTools1.LanguageCode

```

Mobile Forensic Toolkit

System Information

Show

Device Name: SM-G950N

Manufacturer: samsung

Root Access: Not Rooted

SDK Version: 26

RAM Available: 651MB / 3452MB

Internal Memory: 31533MB / 58260MB Available

External Memory: 26728MB / 31512MB Available

IP Address: 118.44.43.158

Android Version: Android 8.0 ~ 8.1

Network Operator: SKTelecom (45005)

SIM ID: 8982051408415624703 (45005)

More

Network Forensics

Hash

Specific

Kernel Version: 4.4.111-14502969-QB200

Build Number: R16NW.G950NKSU3CRJ1

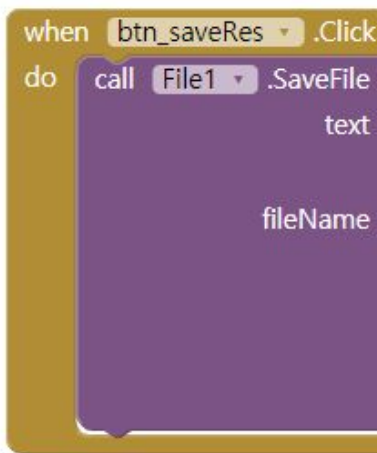
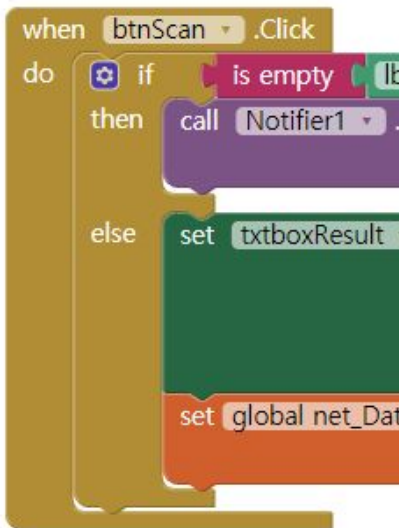
Baseband Version: G950NKOU3CRJ1

Bootloader Version: G950NKSU3CRJ1

Serial Number: ce10171abab0001e04

Country Code: KR

Language Code: ko



TXT	NetworkScannin...2018093604.txt	11월 23일 오후 9:36	150 B
TXT	NetworkScannin...2018100641.txt	11월 23일 오후 10:06	150 B
TXT	NetworkScannin...2018105808.txt	11월 23일 오후 10:58	150 B
TXT	NetworkScannin...2018111526.txt	11월 23일 오후 11:15	32 B
TXT	NetworkScannin...2018024536.txt	11월 24일 오전 2:45	32 B
TXT	NetworkScannin...2018104715.txt	11월 24일 오후 10:47	56 B
TXT	NetworkScannin...2018084316.txt	11월 25일 오후 8:43	32 B

NetworkForensics

Network Data

IP Address: 0.0.0.0

MAC Address: 08:AE:D6:5B:51:93

Connected Wi-Fi: NOT CONNECTED TO WI-FI

MAC of AP: NOT CONNECTED TO WI-FI

Host Scanning

IP Band: 210.115.255 to 20

Scan

Save Result

Scanned Result:

210.115.255.1
210.115.255.4
210.115.255.11
210.115.255.12
210.115.255.13
210.115.255.14
210.115.255.15
210.115.255.16



해쉬 기능 구동 영상

<https://youtu.be/yF4ff1qiKoA>

네트워크 포렌식 구동 영상

<https://www.youtube.com/watch?v=JxvV8Mkx12E&feature=youtu.be>



NetworkForensics

Network Data

IP Address: 0.0.0.0

MAC Address: 08:AE:D6:5B:51:93

Connected Wi-Fi: NOT CONNECTED TO WI-FI

MAC of AP: NOT CONNECTED TO WI-FI

Host Scanning

IP Band: 210.115.255 to 20

Scan

Save Result

Scanned Result:

210.115.255.1
210.115.255.4
210.115.255.11
210.115.255.12
210.115.255.13
210.115.255.14
210.115.255.15
210.115.255.16



2ricecake@gmail.com
lotsofstars1995@gmail.com