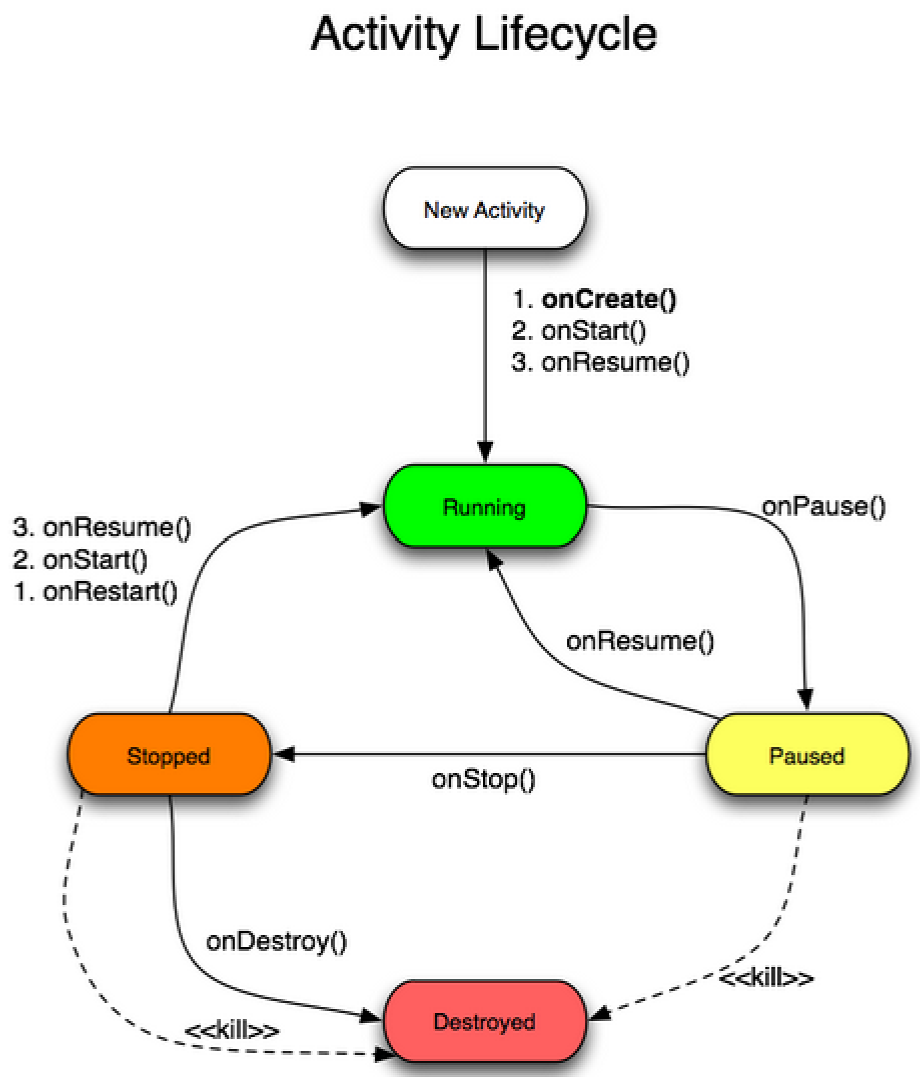
망고100 보드로 놀아보자-19

Android Ethernet 분석

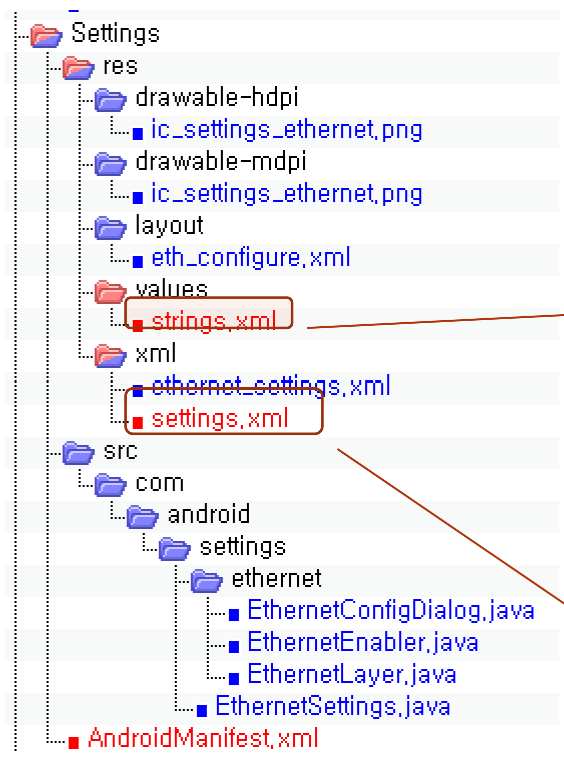
http://cafe.naver.com/embeddedcrazyboys

http://www.mangoboard.com

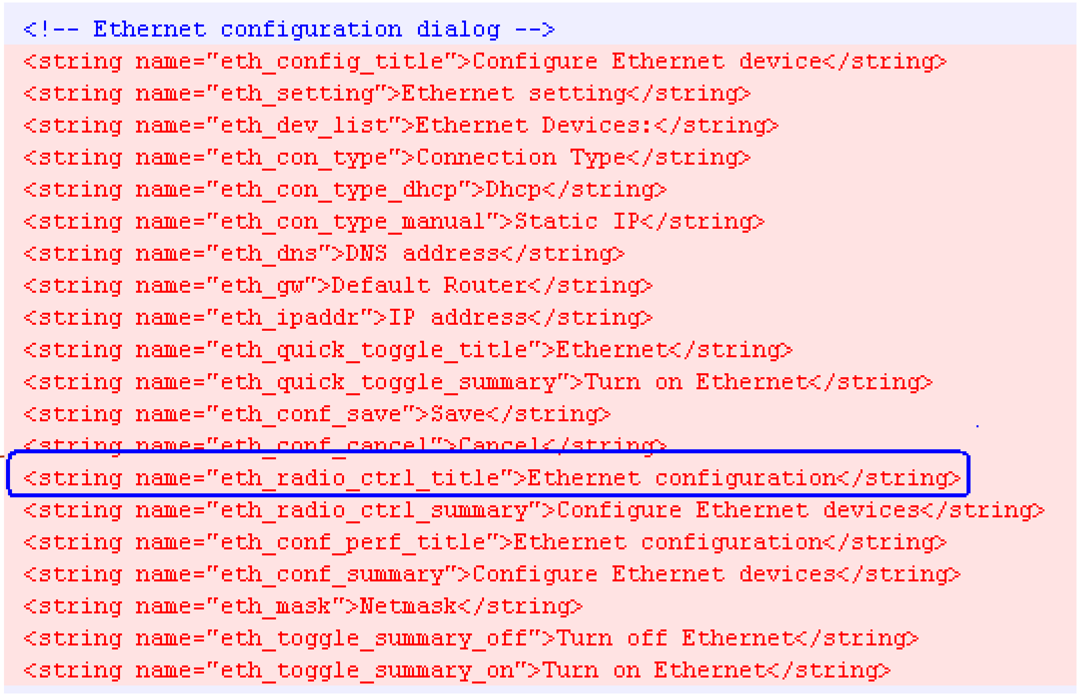
Android Activity Lifecycle



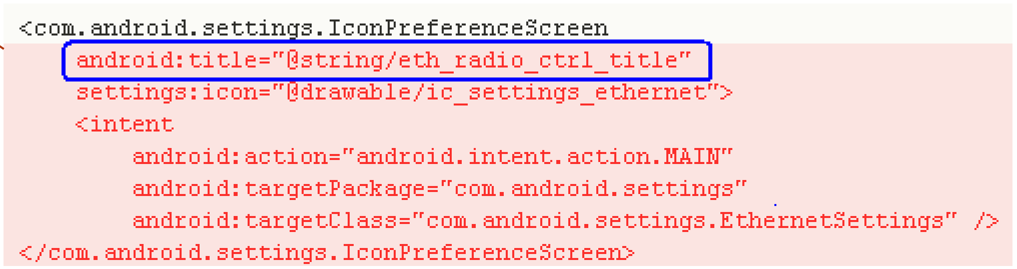
안드로이드 Ethernet Config 추가



```
Settings
  res
    drawable-hdpi
      ic_settings_ethernet.png
    drawable-mdpi
      ic_settings_ethernet.png
    layout
      eth_configure.xml
    values
      strings.xml
    xml
      ethernet_settings.xml
      settings.xml
  src
    com
      android
        settings
          ethernet
            EthernetConfigDialog.java
            EthernetEnabler.java
            EthernetLayer.java
          EthernetSettings.java
  AndroidManifest.xml
```

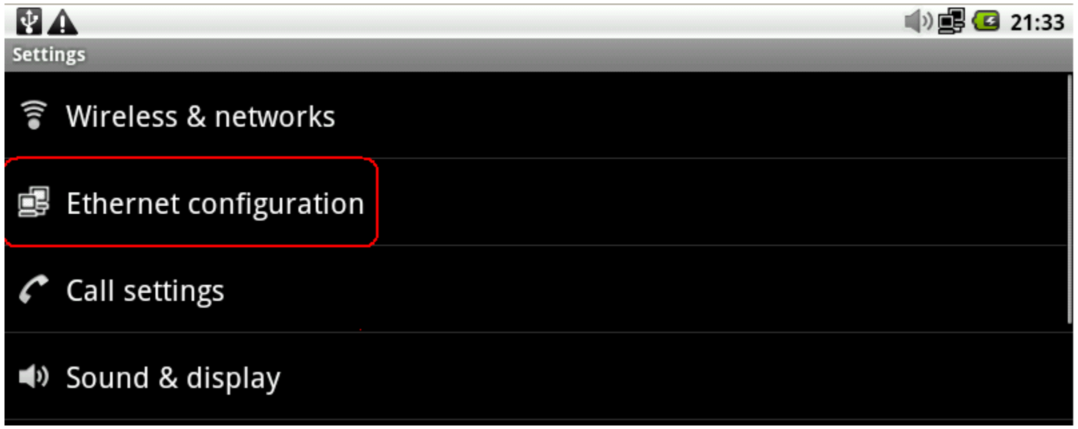


```xml
<!-- Ethernet configuration dialog -->
<string name="eth_config_title">Configure Ethernet device</string>
<string name="eth_setting">Ethernet setting</string>
<string name="eth_dev_list">Ethernet Devices:</string>
<string name="eth_con_type">Connection Type</string>
<string name="eth_con_type_dhcp">Dhcp</string>
<string name="eth_con_type_manual">Static IP</string>
<string name="eth_dns">DNS address</string>
<string name="eth_gw">Default Router</string>
<string name="eth_ipaddr">IP address</string>
<string name="eth_quick_toggle_title">Ethernet</string>
<string name="eth_quick_toggle_summary">Turn on Ethernet</string>
<string name="eth_conf_save">Save</string>
<string name="eth_conf_cancel">Cancel</string>
<string name="eth_radio_ctrl_title">Ethernet configuration</string>
<string name="eth_radio_ctrl_summary">Configure Ethernet devices</string>
<string name="eth_conf_perf_title">Ethernet configuration</string>
<string name="eth_conf_summary">Configure Ethernet devices</string>
<string name="eth_mask">Netmask</string>
<string name="eth_toggle_summary_off">Turn off Ethernet</string>
<string name="eth_toggle_summary_on">Turn on Ethernet</string>
```



```xml
<com.android.settings.IconPreferenceScreen
    android:title="@string/eth_radio_ctrl_title"
    settings:icon="@drawable/ic_settings_ethernet">
    <intent
        android:action="android.intent.action.MAIN"
        android:targetPackage="com.android.settings"
        android:targetClass="com.android.settings.EthernetSettings" />
</com.android.settings.IconPreferenceScreen>
```

안드로이드 Ethernet Config 추가



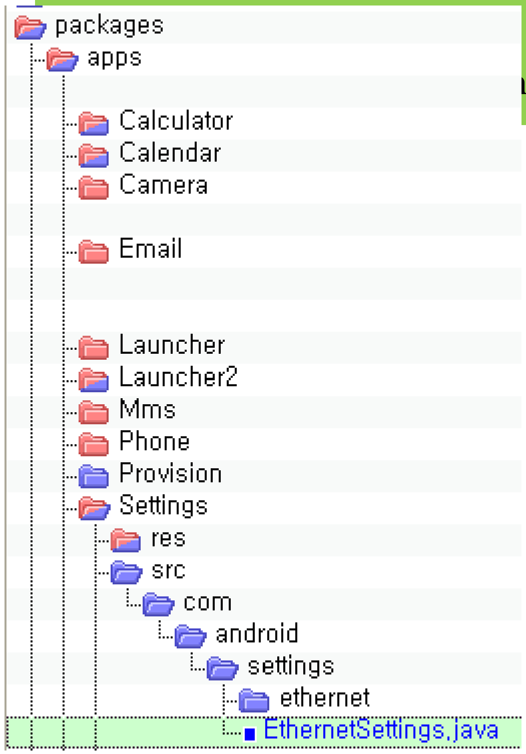
```
<com.android.settings.IconPreferenceScreen
    android:title="@string/eth_radio_ctrl_title"
    settings:icon="@drawable/ic_settings_ethernet">
    <intent
        android:action="android.intent.action.MAIN"
        android:targetPackage="com.android.settings"
        android:targetClass="com.android.settings.EthernetSettings" />
</com.android.settings.IconPreferenceScreen>
```

Setting.xml 과 IconPreferenceScreen

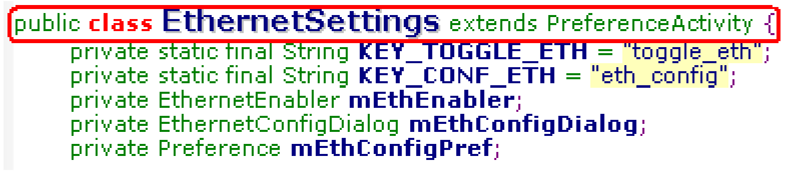
```
<com.android.settings.IconPreferenceScreen
        android:title="@string/eth_radio_ctrl_title"
        settings:icon="@drawable/ic_settings_ethernet">
        <intent
            android:action="android.intent.action.MAIN"
            android:targetPackage="com.android.settings"
            android:targetClass="com.android.settings.EthernetSettings" />
    </com.android.settings.IconPreferenceScreen>
```

XML과 EthernetSettings class관계

```
<com.android.settings.IconPreferenceScreen
        android:title="@string/eth_radio_ctrl_title"
        settings:icon="@drawable/ic_settings_ethernet">
        <intent
            android:action="android.intent.action.MAIN"
            android:targetPackage="com.android.settings"
            android:targetClass="com.android.settings.EthernetSettings"

                    gs.IconPreferenceScreen>
```



```
packages
└ apps
   ├ Calculator
   ├ Calendar
   ├ Camera
   ├ Email
   ├ Launcher
   ├ Launcher2
   ├ Mms
   ├ Phone
   ├ Provision
   └ Settings
      ├ res
      └ src
         └ com
            └ android
               └ settings
                  ├ ethernet
                  └ EthernetSettings.java
```



```
public class EthernetSettings extends PreferenceActivity {
    private static final String KEY_TOGGLE_ETH = "toggle_eth";
    private static final String KEY_CONF_ETH = "eth_config";
    private EthernetEnabler mEthEnabler;
    private EthernetConfigDialog mEthConfigDialog;
    private Preference mEthConfigPref;
```

Ethernet Service 초기화

```
(EthernetManager) getSystemService(ETH_SERVICE),
./packages/apps/Settings/src/com/android/settings/EthernetSettings.java
```

```
}else if (ETH_SERVICE.equals(name)) {
        return getEthernetManager();
./frameworks/base/core/java/android/app/ApplicationContext.java
```

```
private EthernetManager getEthernetManager()
    {
        synchronized (sSync) {
            if (sEthManager == null) {
                IBinder b = ServiceManager.getService(ETH_SERVICE);
                IEthernetManager service = IEthernetManager.Stub.asInterface(b);
                sEthManager = new EthernetManager(service, mMainThread.getHandler());
./frameworks/base/core/java/android/app/ApplicationContext.java
```

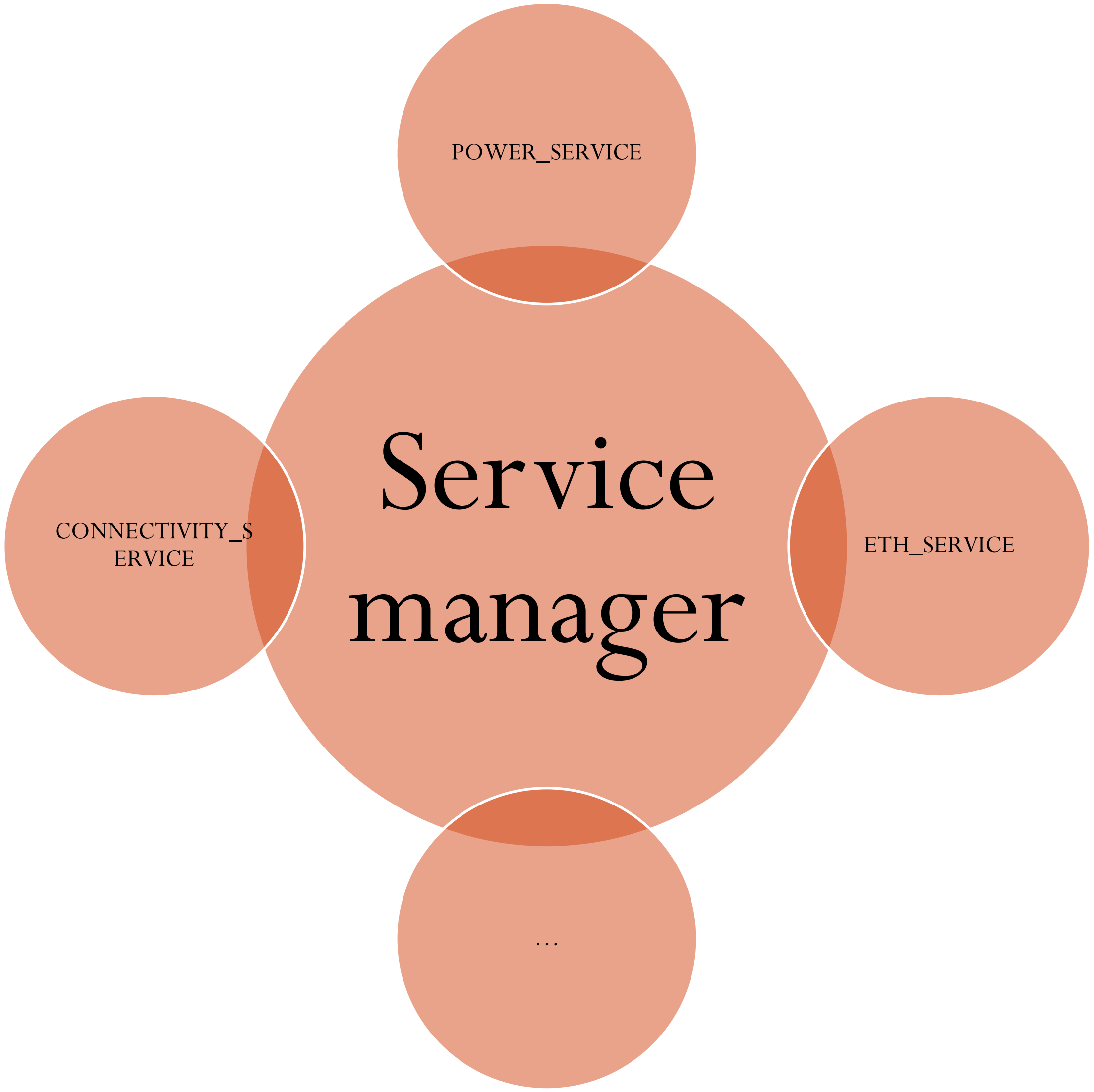
Service Manager Ethernet 등록

```
private EthernetManager getEthernetManager()
  {
      synchronized (sSync) {
          if (sEthManager == null) {
              IBinder b = ServiceManager.getService(ETH_SERVICE);
              IEthernetManager service = IEthernetManager.Stub.asInterface(b);
              sEthManager = new EthernetManager(service, mMainThread.getHandler());
./frameworks/base/core/java/android/app/ApplicationContext.java
```

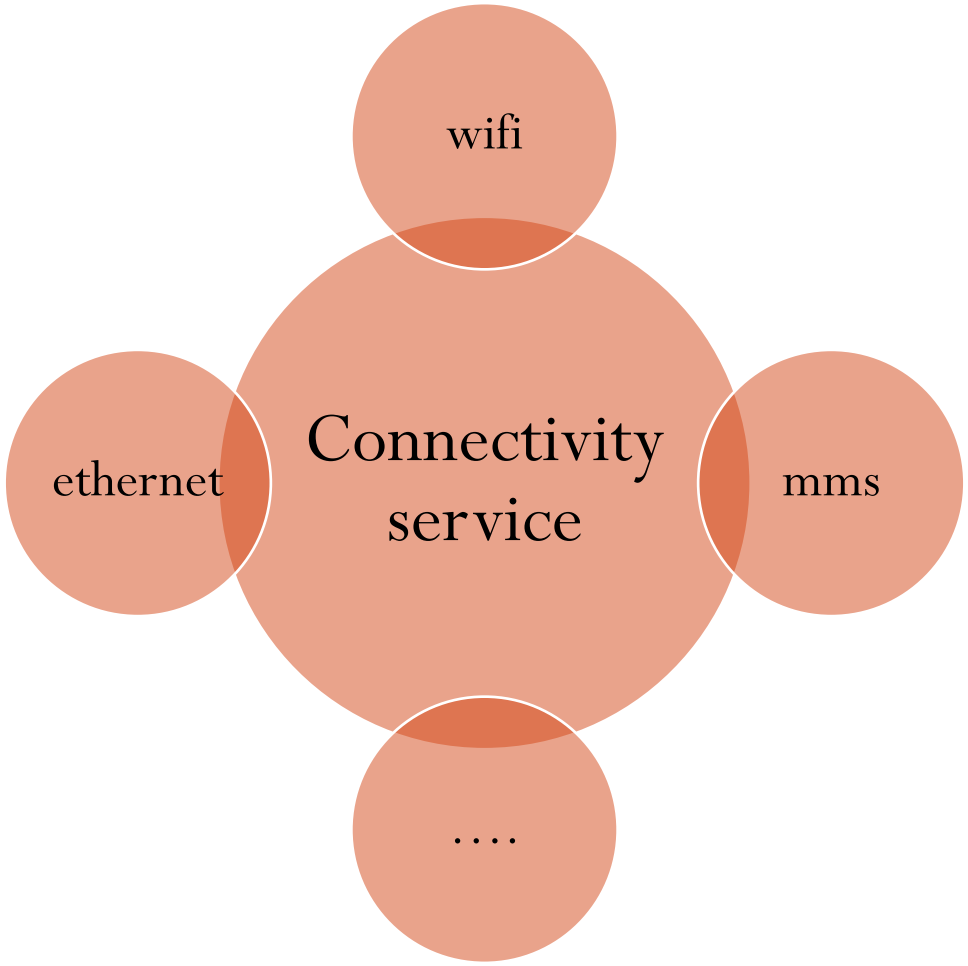
Connectivity service에서 등록

```
private ConnectivityService(Context context) {
….
ServiceManager.addService(Context.ETH_SERVICE, ethService);
./frameworks/base/services/java/com/android/server/ConnectivityService.java
```

Service Manager 역할



Connectivity Service



Ethernet service (부팅 시)

```
private ConnectivityService(Context context) {
        if (DBG) Log.v(TAG, "ConnectivityService starting up");
```

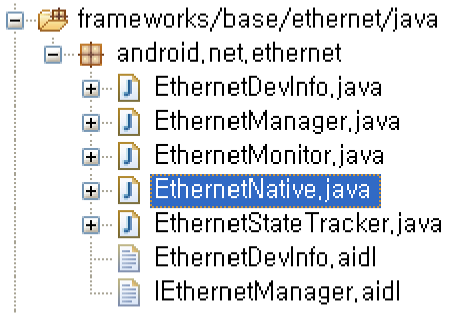
```
if (DBG) Log.v(TAG, "Starting Ethernet Service");
     mEthernetStateTracker = new EthernetStateTracker(context,mHandler);
     EthernetService ethService = new EthernetService(context,
                                     mEthernetStateTracker);
     ServiceManager.addService(Context.ETH_SERVICE, ethService);
     mNetTrackers[ConnectivityManager.TYPE_ETH] = mEthernetStateTracker;
```

mEthernetStateTracker = new EthernetStateTracker(context,mHandler);

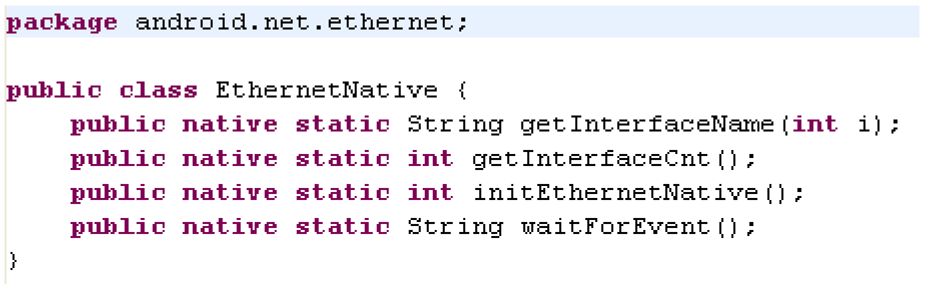
Etherne

super는 자손클래스에서 조상클래스로부터 상속받은 멤버를 참조하는데 사용되는 참조변수

```
public EthernetStateTracker(Context context, Handler target) {
		super(context, target, ConnectivityManager.TYPE_ETH, 0, "ETH", "");
Log.i(TAG,"Starts...");
		if(EthernetNative.initEthernetNative() != 0 )
		{
			Log.e(TAG,"Can not init ethernet device layers");
			return;
		}
```



```
frameworks/base/ethernet/java
  android.net.ethernet
    EthernetDevInfo.java
    EthernetManager.java
    EthernetMonitor.java
    EthernetNative.java
    EthernetStateTracker.java
    EthernetDevInfo.aidl
    IEthernetManager.aidl
```



```
package android.net.ethernet;

public class EthernetNative {
    public native static String getInterfaceName(int i);
    public native static int getInterfaceCnt();
    public native static int initEthernetNative();
    public native static String waitForEvent();
}
```

Ethernet service (부팅 시)

```
static JNINativeMethod gEthernetMethods[] = {
    {"waitForEvent", "()Ljava/lang/String;“,      (void *)android_net_ethernet_waitForEvent},
    {"getInterfaceName", "(I)Ljava/lang/String;“, void )android_net_ethernet_getInterfaceName},
    {"initEthernetNative", "()I“,      (void *)android_net_ethernet_initEthernetNative},
    {"getInterfaceCnt","()I“,      (void *)android_net_ethernet_getInterfaceCnt}
  };
```

```
 static jint android_net_ethernet_initEthernetNative(JNIEnv *env,
                                               jobject clazz)
    {
if ((ret = netlink_init_interfaces_list()) < 0) {
```

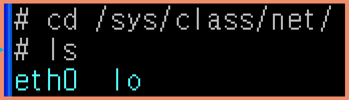
Ethernet service (부팅 시)

static int netlink_init_interfaces_list(void) {

..

if ((netdir = opendir(SYSFS_CLASS_NET)) != NULL) {

while((de = readdir(netdir))!=NULL) {

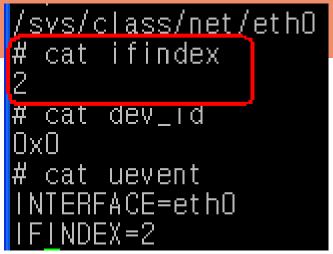


```
# cd /sys/class/net/
# ls
eth0   lo
```

static const char SYSFS_CLASS_NET[]   =
"/sys/class/net";

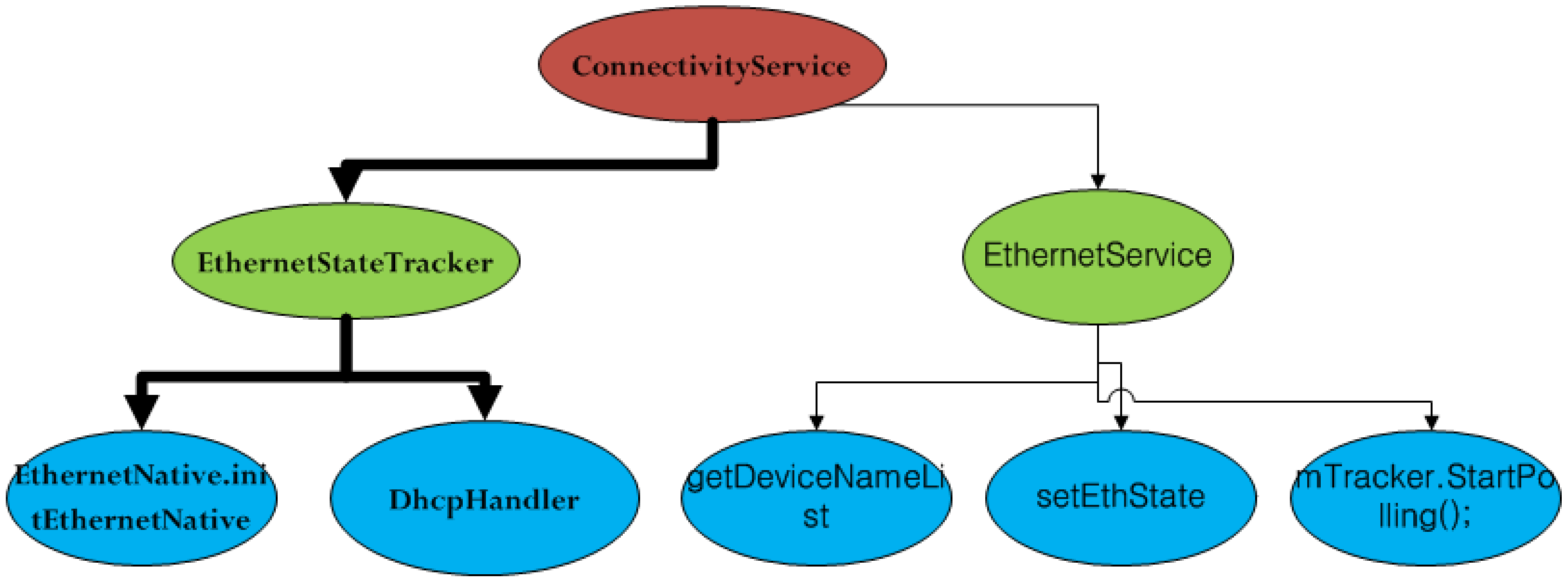
snprintf(path, SYSFS_PATH_MAX,"%s/%s/ifindex",SYSFS_CLASS_NET,de->d_name);

if ((ifidx = fopen(path,"r")) != NULL) {



```
/sys/class/net/eth0
# cat ifindex
2
# cat dev_id
0x0
# cat uevent
INTERFACE=eth0
IFINDEX=2
```

Ethernet service (부팅 시)



Ethernet Service 등록 Flow

```
 private ConnectivityService(Context context) {
…
if (DBG) Log.v(TAG, "Starting Ethernet Service");
        mEthernetStateTracker = new EthernetStateTracker(context,mHandler);
        EthernetService ethService = new EthernetService(context,
                                              mEthernetStateTracker);
```

```
public EthernetService(Context context, EthernetStateTracker Tracker){
                        mTracker = Tracker;
                        mContext = context;
                        isEthEnabled = getPersistedState();
                        Log.i(TAG,"Ethernet dev enabled " + isEthEnabled );
                        getDeviceNameList();
                        setEthState(isEthEnabled);
                        Log.i(TAG, "Trigger the ethernet monitor");
                        mTracker.StartPolling();
          }
```

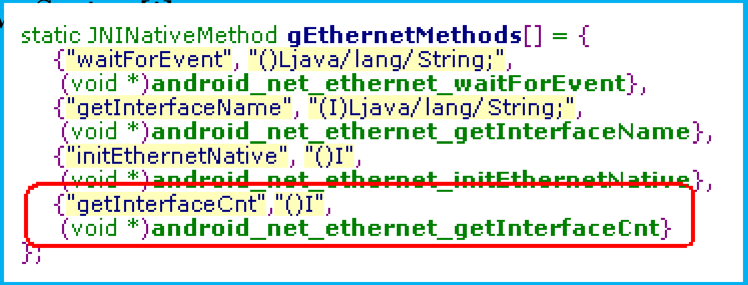
Ethernet Service 등록 Flow(계속)

```
public EthernetService(Context context, EthernetStateTracker Tracker){
…
                    getDeviceNameList();
          }
```

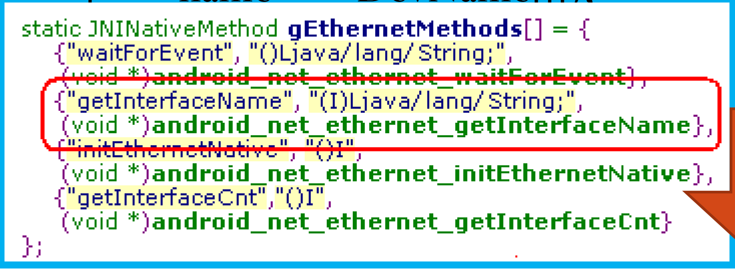
```
public String[] getDeviceNameList() {
                    if (scanEthDevice() > 0 )
                              return DevName;
                    else
                              return null;
          }
```

Ethernet Service 등록 Flow(계속)

```
private int scanEthDevice() {
                int i = 0,j;
                if ((i = EthernetNative.getInterfaceCnt()) != 0) {
                        Log.i(TAG, "total found "+i+ " net devices");
                        DevName = new
                }
                else
                        return i;

                for (j = 0; j < i; j++) {
                        DevName[j] = EthernetNative.getInterfaceName(j);
                        if (DevName[j] == null)
                                break;
                        Log.i(TAG,"device " + j + " name " + DevName[j]);
                }

                return i;
        }
```



```
static JNINativeMethod gEthernetMethods[] = {
    {"waitForEvent", "()Ljava/lang/String;",
     (void *)android_net_ethernet_waitForEvent},
    {"getInterfaceName", "(I)Ljava/lang/String;",
     (void *)android_net_ethernet_getInterfaceName},
    {"initEthernetNative", "()I",
     (void *)android_net_ethernet_initEthernetNative},
    {"getInterfaceCnt","()I",
     (void *)android_net_ethernet_getInterfaceCnt}
};
```



```
static JNINativeMethod gEthernetMethods[] = {
    {"waitForEvent", "()Ljava/lang/String;",
     (void *)android_net_ethernet_waitForEvent},
    {"getInterfaceName", "(I)Ljava/lang/String;",
     (void *)android_net_ethernet_getInterfaceName},
    {"initEthernetNative", "()I",
     (void *)android_net_ethernet_initEthernetNative},
    {"getInterfaceCnt","()I",
     (void *)android_net_ethernet_getInterfaceCnt}
};
```

Ethernet Service 등록 Flow(계속)

```
static jint android_net_ethernet_getInterfaceCnt() {
        return total_int;
   }
```

```
static jint android_net_ethernet_initEthernetNative(JNIEnv *env,
                                                     jobject clazz)
```

```
netlink_init_interfaces_list()
```

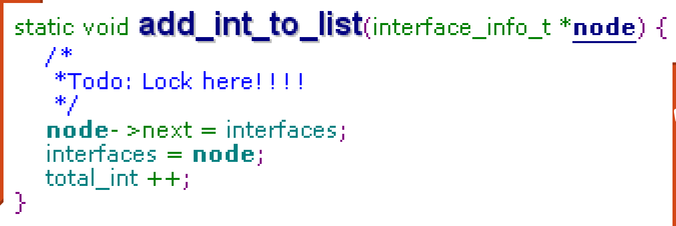
```
LOGI("interface %s:%d found",intfinfo->name,intfinfo->i);
          add_int_to_list(intfinfo);//에서 total_int ++
```

Ethernet Service 등록 Flow(계속)

```
static jstring android_net_ethernet_getInterfaceName(JNIEnv *env,
jobject clazz,                                      jint index)   {
    info= interfaces;
    if (total_int != 0 && index <= (total_int -1)) {
      while (info != NULL) {
        if (index == i) {
          LOGI("Found :%s",info->name);
          return env->NewStringUTF(info->name);
```

```
static jint android_net_ethernet_initEthernetNative(JNIEnv *env,
                              jobject clazz)
```

netlink_init_interfaces_list()



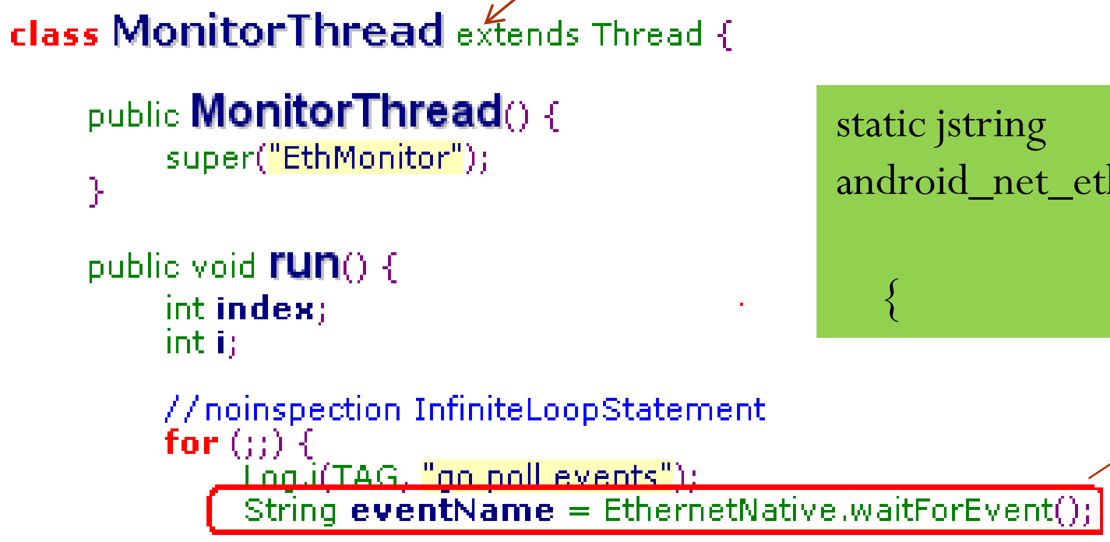
```
static void add_int_to_list(interface_info_t *node) {
   /*
    *Todo: Lock here!!!!
    */
   node->next = interfaces;
   interfaces = node;
   total_int ++;
}
```

'interface %s:%d found",intfinfo->name,intfinfo->i);
add_int_to_list(intfinfo);//에서 Node 생성

Ethernet Event 처리 쓰레드 생성

```
public EthernetService(Context context, EthernetStateTracker Tracker){
        Log.i(TAG, "Trigger the ethernet monitor");
        mTracker.StartPolling();
```

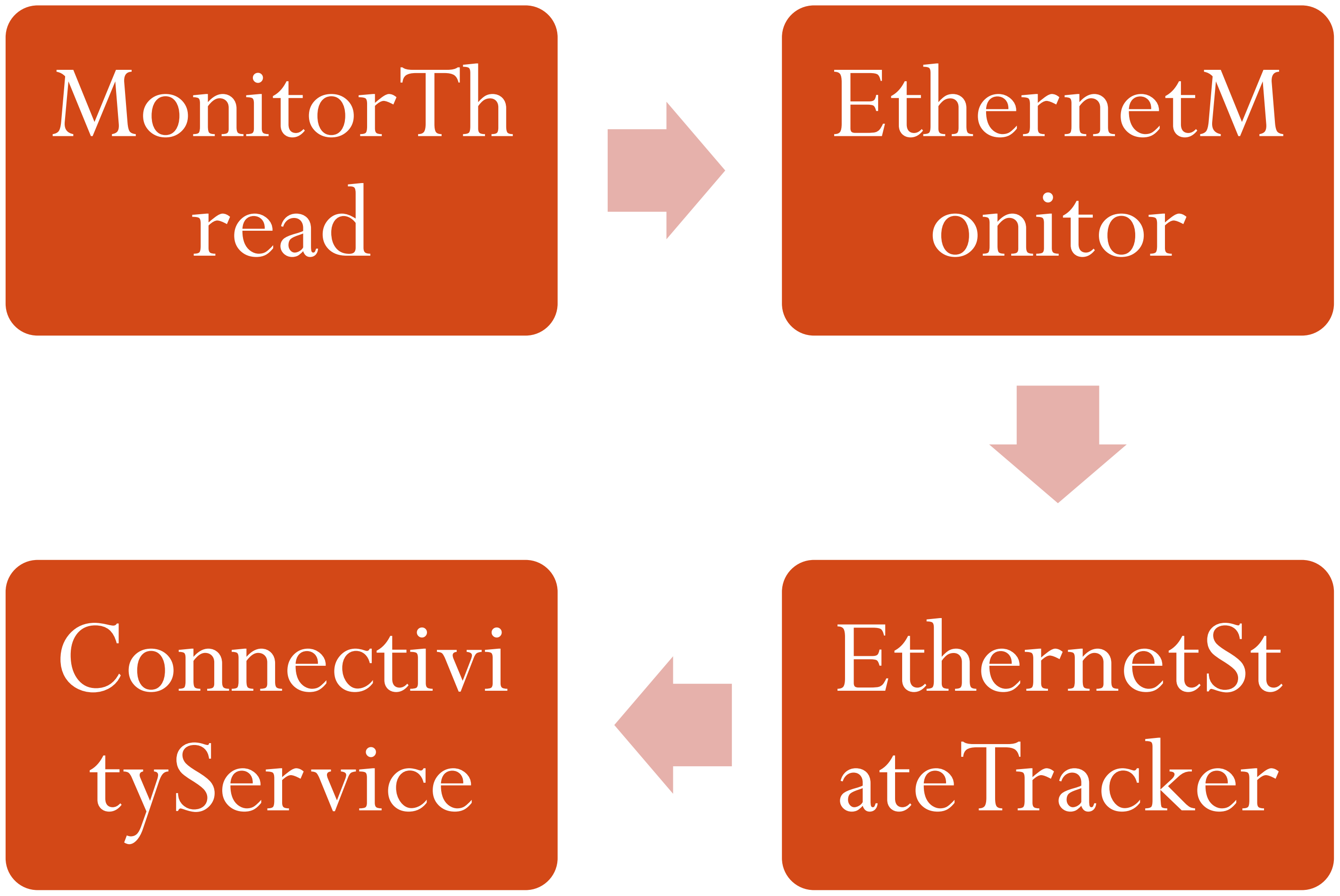
쓰레드 생성



```
class MonitorThread extends Thread {

    public MonitorThread() {
        super("EthMonitor");
    }

    public void run() {
        int index;
        int i;

        //noinspection InfiniteLoopStatement
        for (;;) {
            Log.i(TAG, "go poll events");
            String eventName = EthernetNative.waitForEvent();
```

```
static jstring
android_net_ethernet_waitForEvent(JNIEnv *env,
                                  jobject clazz)
  {
```

이더넷이 연결된 경우(DHCP)



이더넷이 연결된 경우(DHCP)

```
static jstring android_net_ethernet_waitForEvent(JNIEnv *env,
                                        jobject clazz)
    {
if((len = recvmsg(nl_socket_poll, &msg, 0))>= 0) {
```

```
MonitorThread.run
.
} else if (cmd == NEW_LINK) {
event = PHYUP;
handleEvent(events[i],event);
```

이더넷이 연결된 경우(DHCP)

```
void handleEvent(String ifname,int event) {
switch (event) {
case PHYUP:
 mTracker.notifyPhyConnected(ifname););
            break;
```

```
public void notifyPhyConnected(String ifname) {
            if synchronized(this) {
            this.sendEmptyMessage(EVENT_HW_PHYCONNECTED);
```

MonitorThread

이더넷이 연결된 경우(DHCP)

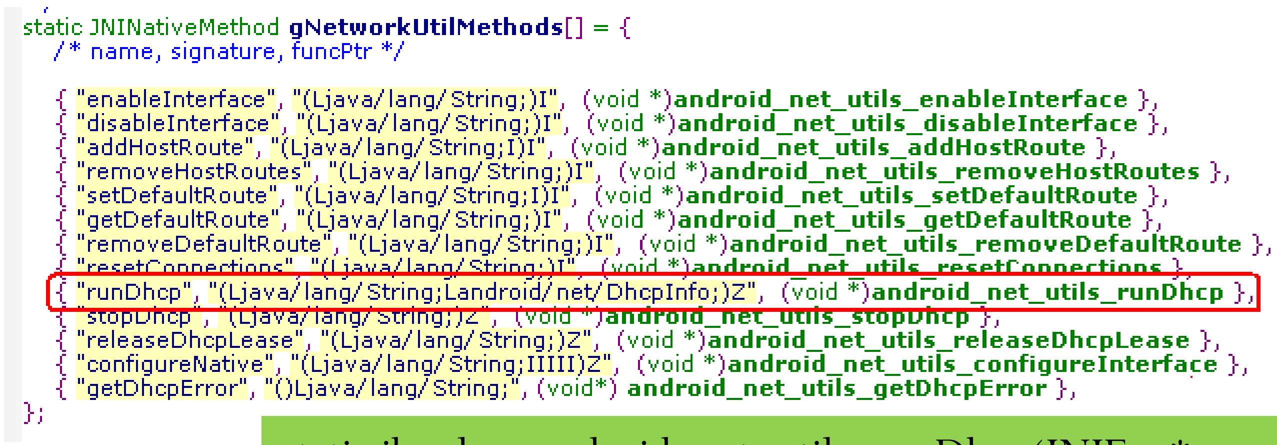
```
public void handleMessage(Message msg) {
..
case EVENT_HW_PHYCONNECTED:
try {
        configureInterface(info);
}
```

```
private boolean configureInterface(EthernetDevInfo info) throws
UnknownHostException {
if (info.getConnectMode().equals(EthernetDevInfo.ETH_CONN_MODE_DHCP))
{
 mDhcpTarget.sendEmptyMessage(EVENT_DHCP_START);
```

DHCP
Handler

이더넷이 연결된 경우(DHCP)

```
public void handleMessage(Message msg) {
..
switch (msg.what) {
case EVENT_DHCP_START:
if (NetworkUtils.runDhcp(mInterfaceName, mDhcpInfo)) {}
```



```
static JNINativeMethod gNetworkUtilMethods[] = {
    /* name, signature, funcPtr */

    { "enableInterface", "(Ljava/lang/String;)I",  (void *)android_net_utils_enableInterface },
    { "disableInterface", "(Ljava/lang/String;)I",  (void *)android_net_utils_disableInterface },
    { "addHostRoute", "(Ljava/lang/String;I)I",  (void *)android_net_utils_addHostRoute },
    { "removeHostRoutes", "(Ljava/lang/String;)I",  (void *)android_net_utils_removeHostRoutes },
    { "setDefaultRoute", "(Ljava/lang/String;I)I",  (void *)android_net_utils_setDefaultRoute },
    { "getDefaultRoute", "(Ljava/lang/String;)I",  (void *)android_net_utils_getDefaultRoute },
    { "removeDefaultRoute", "(Ljava/lang/String;)I",  (void *)android_net_utils_removeDefaultRoute },
    { "resetConnections", "(Ljava/lang/String;)I",  (void *)android_net_utils_resetConnections },
    { "runDhcp", "(Ljava/lang/String;Landroid/net/DhcpInfo;)Z",  (void *)android_net_utils_runDhcp },
    { "stopDhcp", "(Ljava/lang/String;)Z",  (void *)android_net_utils_stopDhcp },
    { "releaseDhcpLease", "(Ljava/lang/String;)Z",  (void *)android_net_utils_releaseDhcpLease },
    { "configureNative", "(Ljava/lang/String;IIIII)Z",  (void *)android_net_utils_configureInterface },
    { "getDhcpError", "()Ljava/lang/String;", (void*) android_net_utils_getDhcpError },
};
```

```
static jboolean android_net_utils_runDhcp(JNIEnv* env, jobject clazz, jstring
ifname, jobject info)
{
result = ::dhcp_do_request(nameStr, &ipaddr, &gateway, &mask,
                    &dns1, &dns2, &server, &lease);
```