NvRam Agent HIDL

Doc No: CS6000-BC4E-CGD-V1.0EN
Version: V1.0
Release date: 2018-01-02
Classification: internal

Keywords

Customization Guide

MediaTek Inc.

Postal address

No. 1, Dusing 1st Rd. , Hsinchu Science Park, Hsinchu City, Taiwan 30078

MTK support office address

No. 1, Dusing 1st Rd. , Hsinchu Science Park, Hsinchu City, Taiwan 30078

Internet

http://www.mediatek.com/

Document Revision History

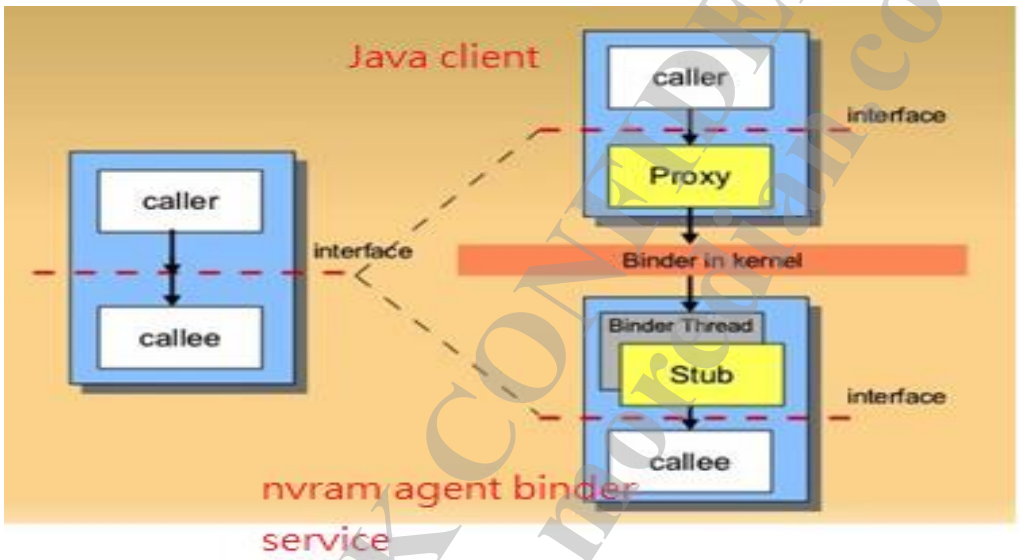
Revision	Date	Author	Description
V1.0	2017-01-02	**Henry huang**	Nvram agent hidl interface

Table of Contents

1 The introduction of Nvram agnet hidl Interface

Purpose

1. Lots of cross partition IPC such as binder are not allowed in Android O, and nvram agent binder could not be used.
2. Add a nvram hidl interface for accessing nvram in java layer



Two nvram api for read/write nvram:

1. String readFileByName(String filename, int size)
2. byte writeFileByNamevec(String filename, int size, java.util.ArrayList<Byte> data)

2 How to access Nvram in java layer from hidl interface

readFileByName example in alps/vendor/mediatek/proprietary/packages/apps/CdsInfo / src/com/mediatek/connectivity/CdsWifiInfoActivity.java

1. include vendor.mediatek.hardware.nvram-V1.0-java-static in Android.mk

```
include $(CLEAR_VARS)

LOCAL_AIDL_INCLUDES := $(LOCAL_PATH)/src/java
LOCAL_SRC_FILES := $(call all-java-files-under, src/java) \
  $(call all-Iaidl-files-under, src/java) \
  $(call all-logtags-files-under, src/java) \
  $(call all-proto-files-under, proto)

LOCAL_JAVA_LIBRARIES := voip-common ims-common
LOCAL_STATIC_JAVA_LIBRARIES := android.hardware.radio-V1.0-java-static \
    android.hardware.radio.deprecated-V1.0-java-static \
    vendor.mediatek.hardware.nvram-V1.0-java-static
LOCAL_MODULE_TAGS := optional
LOCAL_MODULE := telephony-common
LOCAL_PROTOC_OPTIMIZE_TYPE := nano
LOCAL_PROTO_JAVA_OUTPUT_PARAMS := store_unknown_fields=true,enum_style=java

LOCAL_JARJAR_RULES := $(LOCAL_PATH)/jarjar-rules.txt
LOCAL_ADDITIONAL_DEPENDENCIES := $(LOCAL_PATH)/Android.mk $(LOCAL_PATH)/jarjar-rules.txt

ifeq ($(EMMA_INSTRUMENT_FRAMEWORK),true)
LOCAL_EMMA_INSTRUMENT := true
endif
```

2. import nvram interface

```
import vendor.mediatek.hardware.nvram.V1_0.INvram;
```

3. get nvram agent service, then use readFileByName

```
private String getMacAddrFromNvram() {
    StringBuffer nvramBuf = new StringBuffer();
    try {
        int i = 0;
        String buff = null;
        INvram agent = INvram.getService();
        if (agent == null) {
            mToast.setText("No support MAC address writing due to NVRAM");
            mToast.show();
            Log.e(TAG, "NvRAMAgent is null");
            return "";
        }
        try {
            buff = agent.readFileByName(
                    MAC_ADDRESS_FILENAME, MAC_ADDRESS_OFFSET + MAC_ADDRESS_DIGITS);
        } catch (Exception e) {
            e.printStackTrace();
            return "";
        }
        Log.i(TAG, "Raw data:" + buff);
        if (buff.length() < 2 * (MAC_ADDRESS_OFFSET + MAC_ADDRESS_DIGITS)) {
            mToast.setText("The foramt of NVRAM is not correct");
            mToast.show();
            return "";
        }
```

writeFileByNamevec example in alps/vendor/mediatek/proprietary/packages/apps/CdsInfo / src/com/mediatek/connectivity/CdsWifiInfoActivity.java

```
String buff = null;
try {
    buff = agent.readFileByName(
            MAC_ADDRESS_FILENAME, MAC_ADDRESS_OFFSET + MAC_ADDRESS_DIGITS);
} catch (Exception e) {
    e.printStackTrace();
    return;
}

// Remove \0 in the end
byte[] buffArr = HexDump.hexStringToByteArray(
                    buff.substring(0, buff.length() - 1));

for (i = 0; i < MAC_ADDRESS_DIGITS; i ++) {
    buffArr[i + 4] = macAddr[i];
}

ArrayList<Byte> dataArray = new ArrayList<Byte>(
                    MAC_ADDRESS_OFFSET + MAC_ADDRESS_DIGITS);

for (i = 0; i < MAC_ADDRESS_OFFSET + MAC_ADDRESS_DIGITS; i++) {
    dataArray.add(i, new Byte(buffArr[i]));
}

int flag = 0;
try {
    flag = agent.writeFileByNamevec(MAC_ADDRESS_FILENAME,
            MAC_ADDRESS_OFFSET + MAC_ADDRESS_DIGITS, dataArray);
} catch (Exception e) {
    e.printStackTrace();
    mToast.setText(e.getMessage() + ":" + e.getCause());
```

To be a client of nvram agent hidl, we need to add app selinux policy

1. example in device\mediatek\sepolicy\basic\non_plat\system_app.te

```
# Purpose :[CdsInfo] read/ write WI-FI MAC address by NVRAM API
# Package Name: com.mediatek.connectivity
hal_client_domain(system_app, hal_nvramagent);
```