型号作假

**1. 需求说明**

要求能够随时设置手机的型号、品牌、蓝牙名称、WIFI热点名称、MTP名称、PTP名称和Windows系统资源管理器中磁盘的名称。

**2. 修改文件：**



**3. 型号作假设置应用**

ModelSetting

**4. 公共部分修改：**

（1）在device/alibaba/xunhu\_public/FeatureOption中添加如下代码：

XUNHU\_QTY\_MODEL\_SETTING

（2）在device/alibaba/xunhu\_public/FeatureOption.txt中添加如下代码：

[XUNHU\_QTY\_MODEL\_SETTING]---->[客制化型号、品牌、BT、WIFI AP、MTP、PTP、USB、Device Manager Name]---->[默认关闭]

[XUNHU\_QTY\_CUSTOM\_MODEL]---->[同时设置model、bt、wifi ap、usb、mtp、ptp、dm的默认名称为该宏的值，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]---->[默认为no，即不设置]

[XUNHU\_QTY\_MODEL]---->[设置手机型号默认名称，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]--->[默认为no，即不设置]

[XUNHU\_QTY\_BRAND]---->[设置手机品牌默认名称，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]--->[默认为no，即不设置]

[XUNHU\_QTY\_BT]---->[设置蓝牙默认名称，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]--->[默认为no，即不设置]

[XUNHU\_QTY\_WIFI\_AP]---->[设置wifi热点默认名称，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]--->[默认为no，即不设置]

[XUNHU\_QTY\_USB]---->[设置USB默认名称，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]--->[默认为no，即不设置]

[XUNHU\_QTY\_MTP]---->[设置MTP默认名称，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]--->[默认为no，即不设置]

[XUNHU\_QTY\_PTP]---->[设置PTP默认名称，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]--->[默认为no，即不设置]

[XUNHU\_QTY\_DM]---->[设置Windows系统资源管理器中手机磁盘默认名称，这个宏依赖于XUNHU\_QTY\_MODEL\_SETTING，如果没有打开XUNHU\_QTY\_MODEL\_SETTING，则表示只设置一次，不能再改变]--->[默认为no，即不设置]

(3)在/device/alibaba/xunhu\_public/ProjectConfig.mk文件中添加如下代码：

XUNHU\_AUTO\_ADD\_GLOBAL\_DEFINE\_BY\_NAME = … XUNHU\_QTY\_MODEL\_SETTING

XUNHU\_QTY\_MODEL\_SETTING=no

XUNHU\_QTY\_CUSTOM\_MODEL=no

XUNHU\_QTY\_MODEL=no

XUNHU\_QTY\_BRAND=no

XUNHU\_QTY\_BT=no

XUNHU\_QTY\_WIFI\_AP=no

XUNHU\_QTY\_USB=no

XUNHU\_QTY\_MTP=no

XUNHU\_QTY\_PTP=no

XUNHU\_QTY\_DM=no

(4)在device/alibaba/common/resources/custom/engineer.xml文件中添加如下代码：

<!-- Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty -->

<!-- Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称 -->

<customer>

<commandId>\*#8185#</commandId>

<secretCode>android\_secret\_code://8185</secretCode>

</customer>

<customer>

<commandId>\*#8186#</commandId>

<secretCode>android\_secret\_code://8186</secretCode>

</customer>

<customer>

<commandId>\*#352137#</commandId>

<secretCode>android\_secret\_code://352137</secretCode>

</customer>

<!-- Xunhu: end custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty →

（5）在device/mediatek/build/build/libs/custom.mk文件中添加如下代码：

#Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

#Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称

ifneq ($(strip $(XUNHU\_QTY\_CUSTOM\_MODEL)), no)

XUNHU\_QTY\_MODEL := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_BT := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_WIFI\_AP := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_USB := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_MTP := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_PTP := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_DM := $(XUNHU\_QTY\_CUSTOM\_MODEL)

endif

#&&}}

（6）在/device/mediatek/mt6735/device.mk文件中添加如下代码：

#Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

#Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称

ifeq ($(strip $(XUNHU\_QTY\_MODEL\_SETTING)), yes)

PRODUCT\_PACKAGES += ModelSetting

endif

ifneq ($(strip $(XUNHU\_QTY\_CUSTOM\_MODEL)), no)

XUNHU\_QTY\_MODEL := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_BT := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_WIFI\_AP := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_USB := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_MTP := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_PTP := $(strip $(XUNHU\_QTY\_CUSTOM\_MODEL))

XUNHU\_QTY\_DM := $(XUNHU\_QTY\_CUSTOM\_MODEL)

endif

$(shell mkdir -p out/Autogen\_out/custom\_model)

$(shell echo "#ifdef XUNHU\_QTY\_USB" > out/Autogen\_out/custom\_model/teksun\_custom\_model.h)

$(shell echo "#undef XUNHU\_QTY\_USB" >> out/Autogen\_out/custom\_model/teksun\_custom\_model.h)

$(shell echo "#endif" >> out/Autogen\_out/custom\_model/teksun\_custom\_model.h)

$(shell echo "#ifdef XUNHU\_QTY\_DM" >> out/Autogen\_out/custom\_model/teksun\_custom\_model.h)

$(shell echo "#undef XUNHU\_QTY\_DM" >> out/Autogen\_out/custom\_model/teksun\_custom\_model.h)

$(shell echo "#endif" >> out/Autogen\_out/custom\_model/teksun\_custom\_model.h)

ifneq ($(strip $(XUNHU\_QTY\_USB)),no)

$(shell echo "#define XUNHU\_QTY\_USB \"$(strip $(XUNHU\_QTY\_USB))\"" >> out/Autogen\_out/custom\_model/teksun\_custom\_model.h)

endif

ifneq ($(strip $(XUNHU\_QTY\_DM)),no)

$(shell echo "#define XUNHU\_QTY\_DM \"$(strip $(XUNHU\_QTY\_DM))\"" >> out/Autogen\_out/custom\_model/teksun\_custom\_model.h)

endif

#&&}}

**5. 客制化型号和品牌：**

在system/core/init/property\_service.cpp文件中更新作假型号值：

（1）获取NV中的型号名称：

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(更新型号和品牌属性)

#if defined(XUNHU\_QTY\_MODEL\_SETTING)

static char nv\_string[32];

char\* read\_string\_from\_nv(int flag)

{

int fd,n;

char \*buffer = NULL;

int retry=0;

char PN[128];//Length must be no shorter than the path below!!!

sprintf(PN,"/dev/block/platform/mtk-msdc.0/11230000.msdc0/by-name/proinfo");//path may vary from case to case

fd= open(PN,O\_RDONLY);

while ((retry < 3)&&(fd < 0)){

sleep(10);

retry++;

fd= open(PN,O\_RDONLY);

}

if (fd < 0)

{

return nv\_string;

}

buffer = (char\*)malloc(2048);

if(buffer == NULL)

{

close(fd);

return nv\_string;

}

memset(buffer,0,2048);

n=read(fd,buffer,2048);

if(n != 2048){

close(fd);

return nv\_string;

}

close(fd);

if (flag == 0) {

strncpy(nv\_string, &buffer[1024 + 570], 32);

} else if (flag == 1) {

strncpy(nv\_string, &buffer[1024 + 602], 32);

}

ERROR("qty, initModelAndBrand=>nv\_string: %s.", nv\_string);

free(buffer);

return nv\_string;

}

#endif

///&&}}

（2）在设置作假型号属性值：

static void load\_properties(char \*data, const char \*filter)

{

...

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(更新型号和品牌属性)

//property\_set(key, value);

#if defined(XUNHU\_QTY\_MODEL\_SETTING)

if (strcmp(key,"persist.sys.xunhu\_model") == 0) {

strcpy(nv\_string ,read\_string\_from\_nv(0));

if (strlen(nv\_string) > 0) {

property\_set(key, nv\_string);

} else {

property\_set(key, value);

}

} else if (strcmp(key, "persist.sys.xunhu\_brand") == 0) {

strcpy(nv\_string ,read\_string\_from\_nv(1));

if (strlen(nv\_string) > 0) {

property\_set(key, nv\_string);

} else {

property\_set(key, value);

}

} else {

property\_set(key, value);

}

#else

property\_set(key, value);

#endif

///&&}}

...

}

(3)在system/core/libcutils/properties.c中替换掉ro.product.model和ro.product.brand属性的值：

int property\_get(const char \*key, char \*value, const char \*default\_value)

{

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改品牌和型号)

/\*

int len;

len = \_\_system\_property\_get(key, value);

\*/

int len = 0;

#if defined(XUNHU\_QTY\_MODEL\_SETTING)

if (strcmp(key,"ro.product.model") == 0) {

len = \_\_system\_property\_get("persist.sys.xunhu\_model", value);

} else if (strcmp(key, "ro.product.brand") == 0) {

len = \_\_system\_property\_get("persist.sys.xunhu\_brand", value);

} else if (strcmp(key, "ro.product.model\_origin") == 0) {

len = \_\_system\_property\_get("ro.product.model", value);

} else if (strcmp(key, "ro.product.brand\_origin") == 0) {

len = \_\_system\_property\_get("ro.product.brand", value);

}

ALOGV("mbbwaumpd, property\_get(top)=>key: %s, value: %s, len: %d", key, value, len);

if (len > 0 && strcmp(value, "unknown") != 0) {

return len;

} else {

len = \_\_system\_property\_get(key, value);

}

ALOGV("mbbwaumpd, property\_get(bottom)=>key: %s, value: %s, len: %d", key, value, len);

#else

len = \_\_system\_property\_get(key, value);

#endif

///&&}}

if(len > 0) {

return len;

}

if(default\_value) {

len = strlen(default\_value);

if (len >= PROPERTY\_VALUE\_MAX) {

len = PROPERTY\_VALUE\_MAX - 1;

}

memcpy(value, default\_value, len);

value[len] = '\0';

}

return len;

}

3. 客制化蓝牙名称：

修改system/bt/btif/src/btif\_dm.c文件：

void btif\_dm\_read\_energy\_info()

{

… …

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改蓝牙名称)

#if defined(XUNHU\_QTY\_CUSTOM\_DEVICES\_MODEL\_NAME)

char prop\_model[PROPERTY\_VALUE\_MAX];

int max\_len = sizeof(btif\_default\_local\_name) - 1;

property\_get("persist.sys.custom\_dev\_model", prop\_model, "");

BTIF\_TRACE\_DEBUG("cdmn, prop\_model: %s, length: %d", prop\_model, strlen(prop\_model));

if (strlen(prop\_model) > 0) {

strncpy(btif\_default\_local\_name, prop\_model, max\_len);

btif\_default\_local\_name[max\_len] = '\0';

}

#elif defined(XUNHU\_QTY\_MODEL\_SETTING)

char bt\_name[PROPERTY\_VALUE\_MAX];

int max\_len = sizeof(btif\_default\_local\_name) - 1;

property\_get("persist.sys.xunhu\_bt", bt\_name, "");

BTIF\_TRACE\_DEBUG("mbbwaumpd, bt\_name: %s, length: %d", bt\_name, strlen(bt\_name));

if (strlen(bt\_name) > 0) {

strncpy(btif\_default\_local\_name, bt\_name, max\_len);

btif\_default\_local\_name[max\_len] = '\0';

}

#endif

///&&}}

BTIF\_TRACE\_DEBUG("%s: default name=%s", \_\_FUNCTION\_\_, btif\_default\_local\_name);

return btif\_default\_local\_name;

}

4.客制化WIFI热点名称：

（1）修改frameworks/opt/net/wifi/service/java/com/android/server/wifi/WifiApConfigStore.java文件：

private void setDefaultApConfiguration() {

WifiConfiguration config = new WifiConfiguration();

IWifiFwkExt wifiFwkExt = MPlugin.createInstance(IWifiFwkExt.class.getName(), mContext);

if (SystemProperties.get("ro.mtk\_bsp\_package").equals("1")) {

if (wifiFwkExt != null) {

config.SSID = wifiFwkExt.getApDefaultSsid();

} else {

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改WIFI热点名称)

//config.SSID = mContext.getString(R.string.wifi\_tether\_configure\_ssid\_default);

if (XunhuOption.XUNHU\_QTY\_CUSTOM\_DEVICES\_MODEL\_NAME) {

String ssid = SystemProperties.get("persist.sys.custom\_dev\_model");

if (!TextUtils.isEmpty(ssid)) {

config.SSID = ssid;

} else {

config.SSID = mContext.getString(R.string.wifi\_tether\_configure\_ssid\_default);

}

Log.d("cdmn", "[WifiApConfigStore]setDefaultApConfiguration(1)=>ssid: " + ssid + " config: " + config.SSID);

} else if (XunhuOption.XUNHU\_QTY\_MODEL\_SETTING) {

String ssid = SystemProperties.get("persist.sys.xunhu\_wifiap");

if (!TextUtils.isEmpty(ssid)) {

config.SSID = ssid;

} else {

config.SSID = mContext.getString(R.string.wifi\_tether\_configure\_ssid\_default);

}

Log.d("mbbwaumpd", "[WifiApConfigStore]setDefaultApConfiguration(1)=>ssid: " + ssid + " config: " + config.SSID);

} else {

config.SSID = mContext.getString(R.string.wifi\_tether\_configure\_ssid\_default);

}

}

///&&}}

} else {

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改WIFI热点名称)

/\*

config.SSID = com.mediatek.custom.CustomProperties.getString(

com.mediatek.custom.CustomProperties.MODULE\_WLAN,

com.mediatek.custom.CustomProperties.SSID,

mContext.getString(R.string.wifi\_tether\_configure\_ssid\_default));

\*/

if (XunhuOption.XUNHU\_QTY\_CUSTOM\_DEVICES\_MODEL\_NAME) {

String ssid = SystemProperties.get("persist.sys.custom\_dev\_model");

if (!TextUtils.isEmpty(ssid)) {

config.SSID = ssid;

} else {

config.SSID = com.mediatek.custom.CustomProperties.getString(

com.mediatek.custom.CustomProperties.MODULE\_WLAN,

com.mediatek.custom.CustomProperties.SSID,

mContext.getString(R.string.wifi\_tether\_configure\_ssid\_default));

}

Log.d("cdmn", "[WifiApConfigStore]setDefaultApConfiguration(2)=>ssid: " + ssid + " config: " + config.SSID);

} else if (XunhuOption.XUNHU\_QTY\_MODEL\_SETTING) {

String ssid = SystemProperties.get("persist.sys.xunhu\_wifiap");

if (!TextUtils.isEmpty(ssid)) {

config.SSID = ssid;

} else {

config.SSID = com.mediatek.custom.CustomProperties.getString(

com.mediatek.custom.CustomProperties.MODULE\_WLAN,

com.mediatek.custom.CustomProperties.SSID,

mContext.getString(R.string.wifi\_tether\_configure\_ssid\_default));

}

Log.d("mbbwaumpd", "[WifiApConfigStore]setDefaultApConfiguration(2)=>ssid: " + ssid + " config: " + config.SSID);

} else {

config.SSID = com.mediatek.custom.CustomProperties.getString(

com.mediatek.custom.CustomProperties.MODULE\_WLAN,

com.mediatek.custom.CustomProperties.SSID,

mContext.getString(R.string.wifi\_tether\_configure\_ssid\_default));

}

///&&}}

if (wifiFwkExt != null && wifiFwkExt.needRandomSsid()) {

Random random = new Random(SystemClock.elapsedRealtime());

config.SSID = config.SSID + random.nextInt(1000);

Log.d(TAG, "setDefaultApConfiguration, SSID:" + config.SSID);

}

}

config.allowedKeyManagement.set(KeyMgmt.WPA2\_PSK);

String randomUUID = UUID.randomUUID().toString();

//first 12 chars from xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx

config.preSharedKey = randomUUID.substring(0, 8) + randomUUID.substring(9,13);

sendMessage(WifiStateMachine.CMD\_SET\_AP\_CONFIG, config);

}

（2）以及替换掉所有使用 R.string.wifi\_tether\_configure\_ssid\_default字符串的地方，主要是Settings应用。

4. 客制化MTP名称：

修改frameworks/base/media/java/android/mtp/MtpDatabase.java文件：

private int getDeviceProperty(int property, long[] outIntValue, char[] outStringValue) {

Log.d(TAG, "getDeviceProperty property = 0x" + Integer.toHexString(property));

switch (property) {

case MtpConstants.DEVICE\_PROPERTY\_SYNCHRONIZATION\_PARTNER:

case MtpConstants.DEVICE\_PROPERTY\_DEVICE\_FRIENDLY\_NAME:

// writable string properties kept in shared preferences

String value = mDeviceProperties.getString(Integer.toString(property), "");

int length = value.length();

if (length > 255) {

length = 255;

}

value.getChars(0, length, outStringValue, 0);

outStringValue[length] = 0;

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改MTP在PC"我的电脑"中显示的label名称)

if (XunhuOption.XUNHU\_QTY\_CUSTOM\_DEVICES\_MODEL\_NAME) {

Log.d("cdmn", "[MtpDatabase]getDeviceProperty=>value: " + value + " length: " + length);

String deviceName = SystemProperties.get("persist.sys.custom\_dev\_model");

if (!TextUtils.isEmpty(deviceName)) {

int lengthDeviceName = deviceName.length();

if (lengthDeviceName > 255) {

lengthDeviceName = 255;

}

if(lengthDeviceName > 0) {

deviceName.getChars(0, lengthDeviceName, outStringValue, 0);

outStringValue[lengthDeviceName] = 0;

Log.d("cdmn", "[MtpDatabase]getDeviceProperty=>deviceName = " + deviceName + ", lengthDeviceName = " + lengthDeviceName);

} else {

Log.d("cdmn", "[MtpDatabase]getDeviceProperty=>lengthDeviceName = " + lengthDeviceName);

}

} else {

deviceName = SystemProperties.get("ro.product.model", "");

int lengthDeviceName = deviceName.length();

if (lengthDeviceName > 255) {

lengthDeviceName = 255;

}

if(lengthDeviceName > 0) {

deviceName.getChars(0, lengthDeviceName, outStringValue, 0);

outStringValue[lengthDeviceName] = 0;

Log.d("cdmn", "[MtpDatabase]getDeviceProperty=>deviceName = " + deviceName + ", lengthDeviceName = " + lengthDeviceName);

} else {

Log.d("cdmn", "[MtpDatabase]getDeviceProperty=>lengthDeviceName = " + lengthDeviceName);

}

}

} else if (XunhuOption.XUNHU\_QTY\_MODEL\_SETTING) {

String usbConfig = SystemProperties.get("sys.usb.config");

String mtpName = SystemProperties.get("persist.sys.xunhu\_mtp");

if (!TextUtils.isEmpty(usbConfig) && usbConfig.contains("ptp")) {

mtpName = SystemProperties.get("persist.sys.xunhu\_ptp");

}

if (!TextUtils.isEmpty(mtpName)) {

int nameLength = mtpName.length();

if (nameLength > 255) {

nameLength = 255;

}

mtpName.getChars(0, nameLength, outStringValue, 0);

outStringValue[nameLength] = 0;

Log.d("mbbwaumpd", "[MtpDatabase]getDeviceProperty=>mtpName = " + mtpName + ", nameLength = " + nameLength);

}

}

///&&}

return MtpConstants.RESPONSE\_OK;

case MtpConstants.DEVICE\_PROPERTY\_IMAGE\_SIZE:

// use screen size as max image size

Display display = ((WindowManager)mContext.getSystemService(

Context.WINDOW\_SERVICE)).getDefaultDisplay();

int width = display.getMaximumSizeDimension();

int height = display.getMaximumSizeDimension();

String imageSize = Integer.toString(width) + "x" + Integer.toString(height);

imageSize.getChars(0, imageSize.length(), outStringValue, 0);

outStringValue[imageSize.length()] = 0;

return MtpConstants.RESPONSE\_OK;

// DEVICE\_PROPERTY\_BATTERY\_LEVEL is implemented in the JNI code

default:

return MtpConstants.RESPONSE\_DEVICE\_PROP\_NOT\_SUPPORTED;

}

}

5. 客制化PTP名称：

修改frameworks/av/media/mtp/MtpServer.cpp文件：

MtpResponseCode MtpServer::doGetDeviceInfo() {

MtpStringBuffer string;

char prop\_value[PROPERTY\_VALUE\_MAX];

MtpObjectFormatList\* playbackFormats = mDatabase->getSupportedPlaybackFormats();

MtpObjectFormatList\* captureFormats = mDatabase->getSupportedCaptureFormats();

MtpDevicePropertyList\* deviceProperties = mDatabase->getSupportedDeviceProperties();

// fill in device info

mData.putUInt16(MTP\_STANDARD\_VERSION);

if (mPtp) {

mData.putUInt32(0);

} else {

// MTP Vendor Extension ID

mData.putUInt32(6);

}

mData.putUInt16(MTP\_STANDARD\_VERSION);

if (mPtp) {

// no extensions

string.set("");

} else {

// MTP extensions

string.set("microsoft.com: 1.0; android.com: 1.0;");

}

mData.putString(string); // MTP Extensions

mData.putUInt16(0); //Functional Mode

mData.putAUInt16(kSupportedOperationCodes,

sizeof(kSupportedOperationCodes) / sizeof(uint16\_t)); // Operations Supported

mData.putAUInt16(kSupportedEventCodes,

sizeof(kSupportedEventCodes) / sizeof(uint16\_t)); // Events Supported

mData.putAUInt16(deviceProperties); // Device Properties Supported

mData.putAUInt16(captureFormats); // Capture Formats

mData.putAUInt16(playbackFormats); // Playback Formats

property\_get("ro.product.manufacturer", prop\_value, "unknown manufacturer");

string.set(prop\_value);

mData.putString(string); // Manufacturer

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(PTP在PC"我的电脑"中显示的label名称)

//property\_get("ro.product.model", prop\_value, "MTP Device");

#if defined(XUNHU\_QTY\_CUSTOM\_DEVICES\_MODEL\_NAME)

property\_get("persist.sys.custom\_dev\_model", prop\_value, "MTP Device");

ALOGE("cdmn, [MtpServer]doGetDeviceInfo=>prop\_value length: %d", strlen(prop\_value));

if(strlen(prop\_value) <= 0) {

property\_get("ro.product.model", prop\_value, "MTP Device");

}

#elif defined(XUNHU\_QTY\_MODEL\_SETTING)

char usb\_config[PROPERTY\_VALUE\_MAX];

property\_get("sys.usb.config", usb\_config, "");

if (strlen(usb\_config) > 0 && strstr(usb\_config, "mtp")) {

property\_get("persist.sys.xunhu\_mtp", prop\_value, "");

} else {

property\_get("persist.sys.xunhu\_ptp", prop\_value, "");

}

ALOGE("mbbwaumpd, [MtpServer]doGetDeviceInfo=>prop\_value length: %d", strlen(prop\_value));

if(strlen(prop\_value) <= 0) {

property\_get("ro.product.model", prop\_value, "MTP Device");

}

#else

property\_get("ro.product.model", prop\_value, "MTP Device");

#endif

ALOGE("mbbwaumpd, [MtpServer]doGetDeviceInfo=>prop\_value: %s", prop\_value);

///&&}}

string.set(prop\_value);

mData.putString(string); // Model

string.set("1.0");

mData.putString(string); // Device Version

property\_get("ro.serialno", prop\_value, "????????");

string.set(prop\_value);

mData.putString(string); // Serial Number

delete playbackFormats;

delete captureFormats;

delete deviceProperties;

return MTP\_RESPONSE\_OK;

}

7. 客制化USB和资源管理器磁盘名称：

公共部分：

（1）将usb名称和资源管理器磁盘名称写入/proc/cmdline节点中：

a. 在/vendor/mediatek/proprietary/bootloader/lk/platform/mt6735/load\_image.c文件中添加如下代码：

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改USB和资源管理器名称)

#ifdef XUNHU\_QTY\_MODEL\_SETTING

static char manufacturer\_temp[256];

static char product\_temp[256];

static char usb\_dm[32];

// flags => 0 : usb 1: devices manager

char\* get\_name\_from\_nv(int flags) {

char part\_name[]="proinfo";

unsigned char addr[2048]={0};

long len;

#ifdef MTK\_EMMC\_SUPPORT

unsigned long long start\_addr;

#else

unsigned long start\_addr;

#endif

part\_t \*part;

part\_dev\_t \*dev;

dev = mt\_part\_get\_device();

if (!dev)

{

return 0;

}

part = mt\_part\_get\_partition(part\_name);

if (!part)

{

return 0;

}

#ifdef MTK\_EMMC\_SUPPORT

start\_addr = (u64)part->start\_sect \* BLK\_SIZE;

#else

start\_addr = part->startblk \* BLK\_SIZE;

#endif

#ifdef MTK\_EMMC\_SUPPORT

#ifdef MTK\_NEW\_COMBO\_EMMC\_SUPPORT

len = dev->read(dev, start\_addr, (uchar\*)addr, 2048, part->part\_id);

#else

len = dev->read(dev, start\_addr, (uchar\*)addr, 2048);

#endif

#else

len = dev->read(dev, start\_addr, (uchar\*)addr, 2048);

#endif

if (len < 0)

{

return 0;

}

if (flags == 0) {

strncpy(usb\_dm, &addr[1024+698], 32);

} else if (flags == 1) {

strncpy(usb\_dm, &addr[1024+796], 32);

} else {

return 0;

}

return usb\_dm;

}

#endif

///&&}}

b. 修改/vendor/mediatek/proprietary/bootable/bootloader/lk/app/mt\_boot/mt\_boot.c文件：

添加如下代码：

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改USB和资源管理器名称)

#ifdef XUNHU\_QTY\_MODEL\_SETTING

extern char\* get\_name\_from\_nv(int);

#endif

///&&}}

修改如下函数：

int boot\_linux\_from\_storage(void)

{

…...

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改USB和资源管理器名称)

#ifdef XUNHU\_QTY\_MODEL\_SETTING

char temp[42];

snprintf(temp, 42, "usb=%s-usb", get\_name\_from\_nv(0));

cmdline\_append(temp);

snprintf(temp, 42, "dm=%s-dm", get\_name\_from\_nv(1));

cmdline\_append(temp);

#endif

///&&}}

…...

}

(2)修改build/core/Makefile文件：

#Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

#Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称

$(hide) CUSTOM\_MODEM="$(CUSTOM\_MODEM)" \

XUNHU\_QTY\_MODEL="$(XUNHU\_QTY\_MODEL)" \

XUNHU\_QTY\_BRAND="$(XUNHU\_QTY\_BRAND)" \

XUNHU\_QTY\_BT="$(XUNHU\_QTY\_BT)" \

XUNHU\_QTY\_WIFI\_AP="$(XUNHU\_QTY\_WIFI\_AP)" \

XUNHU\_QTY\_USB="$(XUNHU\_QTY\_USB)" \

XUNHU\_QTY\_MTP="$(XUNHU\_QTY\_MTP)" \

XUNHU\_QTY\_PTP="$(XUNHU\_QTY\_PTP)" \

XUNHU\_QTY\_DM="$(XUNHU\_QTY\_DM)" \

bash $(GITINFO\_SH) $@ >> $@

#xunhu:add by lww end

（3）在/build/tools/gitinfo.sh文件中添加如下代码：

#Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

#Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称

if [[ "$XUNHU\_QTY\_MODEL" != "no" ]] ; then

echo "persist.sys.xunhu\_model=$XUNHU\_QTY\_MODEL"

echo "persist.sys.xunhu\_model\_b=$XUNHU\_QTY\_MODEL"

fi

if [[ "$XUNHU\_QTY\_BRAND" != "no" ]] ; then

echo "persist.sys.xunhu\_brand=$XUNHU\_QTY\_BRAND"

echo "persist.sys.xunhu\_brand\_b=$XUNHU\_QTY\_BRAND"

fi

if [[ "$XUNHU\_QTY\_BT" != "no" ]] ; then

echo "persist.sys.xunhu\_bt=$XUNHU\_QTY\_BT"

echo "persist.sys.xunhu\_bt\_b=$XUNHU\_QTY\_BT"

fi

if [[ "$XUNHU\_QTY\_WIFI\_AP" != "no" ]] ; then

echo "persist.sys.xunhu\_wifiap=$XUNHU\_QTY\_WIFI\_AP"

echo "persist.sys.xunhu\_wifiap\_b=$XUNHU\_QTY\_WIFI\_AP"

fi

if [[ "$XUNHU\_QTY\_USB" != "no" ]] ; then

echo "persist.sys.xunhu\_usb=$XUNHU\_QTY\_USB"

echo "persist.sys.xunhu\_usb\_b=$XUNHU\_QTY\_USB"

fi

if [[ "$XUNHU\_QTY\_MTP" != "no" ]] ; then

echo "persist.sys.xunhu\_mtp=$XUNHU\_QTY\_MTP"

echo "persist.sys.xunhu\_mtp\_b=$XUNHU\_QTY\_MTP"

fi

if [[ "$XUNHU\_QTY\_PTP" != "no" ]] ; then

echo "persist.sys.xunhu\_ptp=$XUNHU\_QTY\_PTP"

echo "persist.sys.xunhu\_ptp\_b=$XUNHU\_QTY\_PTP"

fi

if [[ "$XUNHU\_QTY\_DM" != "no" ]] ; then

echo "persist.sys.xunhu\_dm=$XUNHU\_QTY\_DM"

echo "persist.sys.xunhu\_dm\_b=$XUNHU\_QTY\_DM"

fi

#&&}}

（4）在/kernel-3.18/Makefile文件中添加如下代码：

#Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

#Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称

USERINCLUDE += -I$(srctree)/../out/Autogen\_out/custom\_model

#&&}}

USB名称部分：

（1）修改/kernel-3.18/drivers/usb/gadget/android.c文件：

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称

#include "teksun\_custom\_model.h"

#include <linux/string.h>

#include <linux/ctype.h>

///&&}}

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改USB名称)

#ifdef XUNHU\_QTY\_MODEL\_SETTING

extern char \* saved\_command\_line;

extern int is\_space\_string(char\*);

static char usb\_name[32];

static char manufacturer\_temp[256];

static char product\_temp[256];

char\* get\_usb\_name\_from\_cmdline(void) {

int i;

int start\_index = -1;

int end\_index = -1;

for (i=0; i< strlen(saved\_command\_line); i++) {

if (strncmp(saved\_command\_line+i, "usb=", strlen("usb="))==0) {

start\_index = i;

}

if (strncmp(saved\_command\_line+i, "-usb", strlen("-usb"))==0) {

end\_index = i;

break;

}

}

pr\_info("mbbwaumpd, [android.c]get\_usb\_name\_from\_cmdline=>cmdline: %s\n", saved\_command\_line);

pr\_info("mbbwaumpd, [android.c]get\_usb\_name\_from\_cmdline=>start: %d, end: %d\n", start\_index, end\_index);

if (start\_index != -1 && end\_index != -1 && start\_index + strlen("usb=") != end\_index) {

strncpy(usb\_name, saved\_command\_line + (start\_index + strlen("usb=")), end\_index - (start\_index + strlen("usb=")));

pr\_info("mbbwaumpd, [android.c]get\_usb\_name\_from\_cmdline=>is\_space\_str: %d.\n", is\_space\_string(usb\_name));

if (is\_space\_string(usb\_name) == 1) {

memset(usb\_name, 0,sizeof(usb\_name));

}

} else {

#ifdef XUNHU\_QTY\_USB

strncpy(usb\_name, XUNHU\_QTY\_USB, sizeof(usb\_name) - 1);

#else

memset(usb\_name, 0,sizeof(usb\_name));

#endif

}

return usb\_name;

}

#endif

///&&}}

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改USB名称)

#if defined(XUNHU\_QTY\_CUSTOM\_DEVICES\_MODEL\_NAME)

DESCRIPTOR\_STRING\_ATTR(iManufacturer, product\_string\_temp)

DESCRIPTOR\_STRING\_ATTR(iProduct, product\_string\_temp)

#elif defined(XUNHU\_QTY\_MODEL\_SETTING)

DESCRIPTOR\_STRING\_ATTR(iManufacturer, manufacturer\_temp)

DESCRIPTOR\_STRING\_ATTR(iProduct, product\_temp)

#else

DESCRIPTOR\_STRING\_ATTR(iManufacturer, manufacturer\_string)

DESCRIPTOR\_STRING\_ATTR(iProduct, product\_string)

#endif

///&&}}

static int android\_bind(struct usb\_composite\_dev \*cdev)

{

…...

/\* Default strings - should be updated by userspace \*/

///Xunhu: custom model, brand, bt, wifi ap, usb, mtp, ptp, devices manager at 2016-10-28 12:17:12 by qty{{&&

///Description: 客制化型号、品牌、蓝牙、wifi热点、usb、mtp、ptp和devices manager名称(修改USB名称)

/\*

strncpy(manufacturer\_string, MANUFACTURER\_STRING, sizeof(manufacturer\_string) - 1);

strncpy(product\_string, PRODUCT\_STRING, sizeof(product\_string) - 1);

\*/

#ifdef XUNHU\_QTY\_MODEL\_SETTING

get\_usb\_name\_from\_cmdline();

if (strlen(usb\_name) > 0) {

memset(manufacturer\_string,0,sizeof(manufacturer\_string));

strncpy(product\_string, usb\_name, sizeof(product\_string) - 1);

} else {

if (strlen(manufacturer\_temp) > 0) {

strncpy(manufacturer\_string, manufacturer\_temp, sizeof(manufacturer\_string) - 1);

} else {

strncpy(manufacturer\_string, MANUFACTURER\_STRING, sizeof(manufacturer\_string) - 1);

}

if (strlen(product\_temp) > 0) {

strncpy(product\_string, product\_temp, sizeof(product\_string) - 1);

} else {

strncpy(product\_string, PRODUCT\_STRING, sizeof(product\_string) - 1);

}

}

#else

strncpy(manufacturer\_string, MANUFACTURER\_STRING, sizeof(manufacturer\_string) - 1);

strncpy(product\_string, PRODUCT\_STRING, sizeof(product\_string) - 1);

#endif

///&&}}

…...

}