

[FAQ04906][USB名称修改系列] 第13项-如何修改USB存储在PC" 我的电脑" 中显示的label名称

[Description]

如何修改USB存储在PC" 我的电脑" 中显示的label名称



[Keyword]

USB 磁盘名称 存储设备名称 存储模式

[Solution]

修改步骤：

主要是在format时去指定 “-L” 的参数值，并给出label name。具体修改涉及三个文件：

KK之前的版本：

1. /system/vold/Fat.cpp

添加一个新的format函数，第三个参数为bool isInternalSd：

```
Fat::format(const char *fsPath, unsigned int numSectors, bool isInternalSd)
```

修改新增format函数的实现：

```
.....
#ifdef MTK_FORMAT_NOT_PARAM_CLUSTER
#ifndef MTK_FAT_ON_NAND //add for fat on nand 1/3
args[1] = "-0";
args[2] = "android";
close(fd);
if(numSectors)
{
char tmp[32];
snprintf(tmp, sizeof(tmp), "%u", numSectors);
const char *size = tmp;
args[3] = "-s";
args[4] = size;
args[5] = fsPath;
args[6]= NULL;
rc = logwrap(7, args, 1);
}
else
{
if(isInternalSd)
{
args[3] = "-L";
args[4] = "YOUR LABEL NAME"; // 修改内置T卡的label，注意长度不能超过11个字符
args[5] = fsPath;
args[6]= NULL;
SLOGD("[LabelTest1]%s %s %s %s %s %s", args[0], args[1], args[2], args[3],
```

```

args[4], args[5]);
rc = logwrap(7, args, 1);
}
else
{
args[3] = "-L";
args[4] = "YOUR LABEL NAME"; // 修改外置T卡的label，注意长度不能超过11个字符
args[5] = fsPath;
args[6] = NULL;
SLOGD("[LabelTest2]%s %s %s %s %s %s", args[0], args[1], args[2], args[3],
args[4], args[5]);
rc = logwrap(7, args, 1);
}
}
#else//add for faton nand starts here 2/3
args[1] = "-0";
args[2] = "android";
if (numSectors) {
char tmp[32];
snprintf(tmp, sizeof(tmp), "%u", numSectors);
const char *size = tmp;
args[3] = "-L";
args[4] = "YOUR LABEL NAME";//add your lable name need <11 char
args[5] = "-s";
args[6] = size;
args[7] = fsPath;
args[8] = NULL;
SLOGD("%s %s %s %s %s %s %s %s", args[0], args[1], args[2], args[3],
args[4], args[5], args[6], args[7]);
rc = logwrap(9, args, 1);
} else {
//对于是MTK_FAT_ON_NAND的版本，IsEmmcStorage()会always fail,因此需要其他的方法
//来辨别是内卡还是外卡，如这个函数的第一个参数*fsPath,格式是
//dev/block/vold/%d:%d,可以打印出来确定内卡的%d的值，在以后是不会变的。
args[3] = "-L";
args[4] = "YOUR LABEL NAME";//add your lable name need <11 char
args[5] = fsPath;
args[6]= NULL;
SLOGD("[LabelTest1]%s %s %s %s %s %s", args[0], args[1], args[2], args[3],
args[4], args[5]);
rc = logwrap(7, args, 1);
}
#endif //add for faton nand ends here 3/3

#else
.....
if(numSectors)

```

```

{
char tmp[32];
snprintf(tmp, sizeof(tmp), "%u", numSectors);
const char *size = tmp;
args[7] = "-s";
args[8] = size;
args[9] = fsPath;
args[10]= NULL;
rc = logwrap(11, args, 1);
}
else
{
if(isInternalSd)
{
args[7] = "-L";
args[8] = "YOUR LABEL NAME"; // 修改内置T卡的label，注意长度不能超过11个字符
args[9] = fsPath;
args[10]= NULL;
SLOGD("[LabelTest3]%s %s %s %s %s %s %s %s", args[0], args[1], args[2],
args[3], args[4], args[5], args[8], args[9]);
rc = logwrap(11, args, 1);
}
else
{
args[7] = "-L";
args[8] = "YOUR LABEL NAME"; // 修改外置T卡的label，注意长度不能超过11个字符
args[9] = fsPath;
args[10] = NULL;
SLOGD("[LabelTest4]%s %s %s %s %s %s %s %s", args[0], args[1], args[2],
args[3], args[4], args[5], args[8], args[9]);
rc = logwrap(11, args, 1);
}
}
#endif
.....

```

2. /system/vold/Fat.h

添加新增format函数的定义

3. /system/vold/Volume.cpp

在调用Fat::format函数的地方，增加第三个参数IsEmmcStorage()

Fat::format(devicePath, 0, IsEmmcStorage())

KK及以后的版本

KK上的修改同之前版本类似，实现上的细小差别可以参考下面（以内置SD为例）

1. /system/vold/Fat.cpp

对于新增函数：

```

int Fat::format(const char *fsPath, unsigned int numSectors, bool wipe,
bool forceFat32, bool isInternalSd){

```

```

..... //同int Fat::format(const char *fsPath, unsigned int numSectors,
bool wipe, bool forceFat32)函数的实现
#ifdef MTK_FORMAT_NOT_PARAM_CLUSTER
args[1] = "-0";
args[2] = "android";
close(fd);
if (numSectors) {
char tmp[32];
snprintf(tmp, sizeof(tmp), "%u", numSectors);
const char *size = tmp;
args[3] = "-s";
args[4] = size;
args[5] = fsPath;
if (forceFat32){
args[5] = "-F";
args[6] = "32";
args[7] = fsPath;
SLOGD("%s %s %s %s %s %s %s %s", args[0], args[1], args[2], args[3],
args[4], args[5], args[6], args[7]);
rc = android_fork_execvp(8, (char **)args, &status, false, true);
}
else {
SLOGD("%s %s %s %s %s %s", args[0], args[1], args[2], args[3], args[4],
args[5]);
rc = android_fork_execvp(6, (char **)args, &status,
false, true);
}
} else {
args[3] = fsPath;
if (forceFat32){
args[3] = "-F";
args[4] = "32";
args[5] = fsPath;
SLOGD("%s %s %s %s %s %s", args[0], args[1], args[2], args[3], args[4],
args[5]);
rc = android_fork_execvp(6, (char **)args, &status, false, true);
}
else
{
if(isInternalSd){
args[3] = "-L";
args[4] = "wq_Inlab"; // 修改外置T卡的label，注意长度不能超过11个字符
args[5] = fsPath;
rc = android_fork_execvp(6, (char **)args, &status, false, true);
}
else{
rc = android_fork_execvp(4, (char **)args, &status, false, true);
}
}
}

```

```

}
}
..... // 同int Fat::format(const char *fsPath, unsigned int numSectors,
bool wipe, bool forceFat32)函数的实现
2. /system/vold/Fat.h
添加新增format函数的定义
3. /system/vold/Volume.cpp
修改实质同之前版本，实现上，将下列判断和调用用新增函数来替换。
if (Fat::format(otgNodePath, 0, wipe, isForceFat32)) ==>> if
(Fat::format(otgNodePath, 0, wipe, isForceFat32, IsEmmcStorage()))
if (Fat::format(devicePath, 0, wipe, isForceFat32)) ==>> if
(Fat::format(devicePath, 0, wipe, isForceFat32, IsEmmcStorage()))
if (Fat::format(devicePath, 0, false)) ==>> if (Fat::format(devicePath, 0,
false, false, IsEmmcStorage()))

```

注意：

1. VolumeManger.cpp里面调用Fat::format()的地方不需要修改
2. 下载image时，需要进行格式化下载
3. 要修改外置T卡的盘符，需要在手机上格式化SD卡，盘符设置才会起效（请注意）
4. 需要打开大容量存储(即连接电脑之后，打开UMS功能)，修改的盘符才会显示出来。

5. 支持大小写切换：Modify the function mklablel () in file
/system/core/toolbox/newfs_msdos.c.

```

static void
mklablel(u_int8_t *dest, const char *src)
{
    int c, i;

    for (i = 0; i < 11; i++) {
        //c = *src ? toupper(*src++) : ' ';
        c = *src ? (*src++) : ' ';
        *dest++ = !i && c == '\xe5' ? 5 : c;
    }
}

```

【另外需要注意的是】：

若软件有内置的fat_sparse.img，那么用上面的修改方法不会成功，因为用软件fat_sparse.img的情况下，第一次开机并不会format内置t card。需要用以下方法修改：在制作fat.img时，使用下面这个命令：

```
mkfs.vfat -n volume_name -v -C fat.img [block-count]
```

可以参考DMS(DCC)文档：

SW > 3G-4G > Smart Phone > Standard Package > MT6582 > Tools >
Make_And_DL_FAT_Image.ppt

注意需要先关闭share_sdcard的feature。