

DB (DATA BASE): 顾名思义, DB即数据库, 在DPF中, 底层提供了一个对盘数据的管理, 而实现这种管理的途径就是借助于数据库(DB), 本文主要讲述的是如何去使用这个管理接口, 即如何去使用DB。

DB可以根据不同的检索方式形成不同排序方式的视图, 比如单以Name作为检索方式的则会形成按照字母顺序来排序的文件视图, 如以Tag为检索方式的则会形成以tag大小为顺序排序的视图, 应用上是可以在一个视图上再去建立另外一个视图, 因此新视图就会遵循双重视图的排序标准。

本文主要是把在photo模块里使用DB所遇到的问题做个小结, 因此在行文中主要会以实际遇到的问题及使用的步骤为序来进行。于此可分为以下几步:

- 一. 打开数据库及视图的建立
- 二. DB的访问

## 一. 数据库的打开 (建立)

对于使用了DB的photo模块, 鉴于它的所有操作都需要借助于DB来完成, 因此进入此模块的第一步操作就是打开DB, 建立视图。

打开DB:

```
DBhandle = MediaLib.openDB(MediaLib.ML_MTYPE_PHOTO, DbName);
```

如果打开成功了就可以创建基本视图:

```
BasicViewHandle= MediaLib.openView(DBhandle,sorttype,"",MediaLib.ML_VM_PURE_SORT,0);
```

如果打开DB失败了则创建DB后再创建基本视图, 成功后后续的操作才能正常进行。

```
IsTaskCreate = MediaLib.createDB("/mnt/udisk/",storetype,"jpg bmp jpeg",DbName,MediaLib.ML_MTYPE_PHOTO);
```

*在创建DB操作执行后需要不断的去检测是否创建DB成功, 一直要等到成功后才能进行创建视图等其他操作。*

**注意:** 在打开DB之前实际上还需要添加一步操作——检测DB所管理的存储介质里的数据是否被改变了, 如被改变了则需要更新数据库, 最直接的解决办法就是发现数据一旦被更改了就直接删除DB, 这样程序在走正常流程时会出现打开DB失败(此时, 打开DB肯定是失败的, 因此程序会走打开DB失败流程——也即先创建DB, 然后进行DB的其他操作)。

检测办法: 调用函数 **MediaLib.check\_udisk\_dirty()**; 返回值0—表示存储介质数据没被PC通过usb连接DPF后改变, 返回1则表示改变了。因此可如下操作:

```
if(MediaLib.check_udisk_dirty())
{
    MediaLib.destroyDB(MediaLib.ML_MTYPE_PHOTO,DbName); //执行DB删除操作
}
```

完整的流程程序流程如下:

```
if(MediaLib.check_udisk_dirty())
{
    MediaLib.destroyDB(MediaLib.ML_MTYPE_PHOTO,DbName); //执行DB删除操作
}
DBhandle = MediaLib.openDB(MediaLib.ML_MTYPE_PHOTO, DbName);
if(DBhandle==0) //DB打开失败
{
    IsTaskCreate = MediaLib.createDB("/mnt/udisk/",storetype,"jpg bmp jpeg",DbName,MediaLib.ML_MTYPE_PHOTO);
    //等待检测DB创建成功后再执行创建视图等其他操作
}
else
{
    BasicViewHandle= MediaLib.openView(DBhandle,sorttype,"",MediaLib.ML_VM_PURE_SORT,0);
    //执行DB的其他操作
}
//检测DB是否创建成功的代码——通过检测MediaLib.getDBCreateStatus()的值来判断
如果检测创建成功了则开始执行正常的DB操作流程
//往下就是数据操作流程
注: var sorttype:Number = MediaLib.ML_ST_FILE_NAME;
```

## 二. DB的访问

1. 以Tag为操作对象的操作形式:

```
if(TagViewHandle!=0)
{
    MediaLib.closeView(TagViewHandle);
    TagViewHandle=0;
}
sorttype=MediaLib.ML_ST_USER_TAG1;
TagViewHandle=MediaLib.openView(DBhandle,sorttype,String(sourceAlbumTag),MediaLib.ML_VM_FITLER_BYVALUE_EQUAL,BasicViewHandle);
//在基本视图BasicViewHandle上建立以Tag=sourceAlbumTag为序的新视图, 此视图满足: 在此视图上的图片全部是tag值为sourceAlbumTag的且是按照字母顺序排序的。
注意: 在此建立的新视图, 不注意的话会出现错误: 因为tag用了两个字节即16位分别表示不同的tag值, 更详细的请仔细查看MediaLib的相关文档及底层代码
新视图的建立要根据此视图的tag值是否为0来区别对待:
tag为0的album视图:
TagViewHandle = MediaLib.openView(DBhandle, sorttype, String(tempTagvalue), MediaLib.ML_VM_FITLER_BYVALUE_EQUAL, BasicViewHandle);
tag非0的album视图:
TagViewHandle =
MediaLib.openView(DBhandle, sorttype, String(tempTagvalue), MediaLib.ML_VM_FITLER_BYVALUE_BITEQUAL, BasicViewHandle);
```

打tag:

```
MediaLib.modifyItem(TagViewHandle,picseArray[0],sorttype,String(tempTagValue),MediaLib.ML_MODIFY_MODE_SETBITS);
```

即: 把tag值为sourceAlbumTag的选定图片picseArray[0] (此是在新视图中的选定图片的序号)的tag值增加一个新值tempTagValue, 也即加上新的tag, 但需要注意的是加上新tag(打tag)的时候其他的tag值依然存在, 这样就会有同一张图片存在多个tag值的情况, 也即同一张图片可以出现在不同的album中。

**删除tag:** (可以认为是删除某个album中的某张图片)

删除tag是针对某一个album中的某张或某几张图片操作的, 上边介绍过的, 同一张图片可以存有多个不同的tag, 因此此处删除tag也仅仅是删除图片中的某一个tag, 对于其他的tag值是不会受到影响的, 从宏观相册概念上来说, 删除某一个相册的图片并不会删除到其他相册的图片的, 不然那就是BUG了 **MediaLib.modifyItem(TagViewHandle,i,sorttype,String(tempTagvalue),MediaLib.ML\_MODIFY\_MODE\_CLRBITS);**

解释:

从tag值为tempTagvalue的视图(TagViewHandle为视图的handle, 此视图中所有图片都有一个相同的tag值tempTagvalue)中删除掉第i张图片, 也即删除掉某个相册中的第i张图片。

**注意点1:**

如果现在删除的图片位于tag值为0的album中（实际上是没被打过TAG的图片都存在这个album中），那删除的方式就不一样了遇到这种情况就等于直接删除掉存储介质上的图片，同时也需要删除掉DB中对应的图片内容。

```
if(FileSystem.FS_ERR_OK == FileSystem.removeFile(urlArray[i])) //判断介质上图片是否删除成功
{
    MediaLib.delItem(TagViewHandle,i+(pageNum-1)*perNum);///在DB中删除掉对应的图片，如不删除就会出现进入对应的album时看到
    //图片是黑的情况，而这情况实际上就是DB根据索引及路径取不到对应的图片--因为介质上的图片已经被删除了
}
```

#### 注意点2:

对于有DB存在的DPF，如果没有进入album（意味着没打开DB）而直接在mainmenu中进入slideshow的情况，为了兼容有DB和无DB的情况，从mainmenu进入slideshow一概是不使用DB，而是采用传统的filelist的方式去操作数据的。

为了保证数据的一致性，在有DB存在的情况下，直接从mainmenu进入slideshow.swf需要先打开DB（不管打开成功与否都会直接进行下边的数据操作，而不会去创建DB的，这里打开DB目的是为了保证DB中的数据和介质中实际的数据保持一致性，如果打不开则表示此DB已经因为某种原因遭到破坏，因此在进入album模块的时候它会去重新建立DB的）。

在有DB的DPF中，且是从mainmenu进入slideshow中，然后进行数据的删除操作的，此时需要进行如下操作方可保证数据的一致，否则会出现出现在slideshow中删除图片后，但DB中依然存在此图片的数据，于是在进入到对应的album中会出现对应图片（实际上在介质中已经被删除了）是黑的情况。

- 1.先通过findItem在DB中找到名为FileList.getLongnameNew(photoFileHandle,PhotoIndex)所对应图片的序号tempIndex。
- 2.删除掉介质上的图片 FileList.deleteFileNew(photoFileHandle, PhotoIndex)
- 3.在成功打开的DB中删除掉序号为tempIndex的图片

```
Var photoName=FileList.getLongnameNew(photoFileHandle,PhotoIndex);
tempIndex =
MediaLib.findItem(BasicViewHandle,MediaLib.ML_ST_FILE_NAME,photoName,MediaLib.ML_SEARCH_MODE_SEQUENCE);
if(tempIndex != -1)//表示成功找到索引
{
    MediaLib.delItem(BasicViewHandle,tempIndex);
    MediaLib.updateDB(DBhandle);
}
```

注意1，以上两步操作作用的handle都是BasicViewHandle，这个不能不同，否则会删错的

注意2，以上步骤1和2不可颠倒，否则一旦删除了介质上的图片后再去找相同序号的图片，此时所找到的图片name实际上是下一张图片的name。