

设为首页 收藏本站



caishaoyi | 设置 | 消息 | 提醒 | 退出

积分: 359 | 用户组: 中级会员



论坛

快速导航

请输入搜索内容

帖子

热搜: 活动 交友 discuz

论坛 软件部 软件技术分享 浅析FileExplore （V3.2）

发帖 回复

返回列表

查看: 19 | 回复: 2

dengzhengui



1 主题

40 帖子

190 积分

注册会员



积分 190

发消息

浅析FileExplore （V3.2） [复制链接]

发表于 前天 11:15 | 只看该作者

楼主 电梯直达

本帖最后由 dengzhengui 于 2013-11-30 12:05 编辑

如果想要比较全面的分析理解文件管理器，你可以在阅读文件管理器源码之前稍微了解一些相关知识。

1）、了解java基础知识；

2）、了解应用程序的几大基本组件。

特别是 Activity和 Intent

Activity: <http://developer.android.com/reference/android/app/Activity.html>

Intent: <http://developer.android.com/reference/android/content/Intent.html>

了解下应用的几大基本组成。基本组件介绍: <http://wenku.baidu.com/view/51a01438376baf1ffc4fad68.html>

3）、了解android io操作数据的基本方法:

<http://developer.android.com/reference/java/io/File.html>

<http://developer.android.com/reference/java/io/BufferedInputStream.html>

<http://developer.android.com/reference/java/io/BufferedOutputStream.html>

文件管理器的源码在android4.2\device\softwinner\fiber-common\prebuild\packages\FileExplore

源码中的主要类作用说明:

Main.java 界面UI显示。

Settings.java 设置界面 可以设置文本颜色 排序类型，图片预览等

TypeFilter.java 根据文件后缀名 判断文件类型

ThumbnailCreator.java 创建文件显示缩略图

FileManager.java 文件操作部分。

EventHandler.java 事件响应 部分。

AudioPlayblack.java 音乐播放界面

WirelessManager.java wifi设置界面 菜单中没有添加 已注释掉

ApplicationBackup.java 显示第三方应用，提供一个备份功能

HelpManager.java 帮助菜单

源码中

Main.java

FileManager.java

EventHandler.java

这三个类是主要的，其他类都是一些工具类，现在我们主要来分析这3个类。

一 .Main.java

UI界面显示界面UI 继承了 ListActivity， ListActivity是自带ListView的Activity。

如果你了解 Activity 的生命周期，你就应该直奔onCreate函数。在onCreate 函数中设置了Activity 的显示界面布局 setContentView(R.layout.main);

```
01. mSettings = getSharedPreferences(PREFS_NAME, 0);
02. boolean hide = mSettings.getBoolean(PREFS_HIDDEN, false);
03. boolean thumb = mSettings.getBoolean(PREFS_THUMBNAIL, true);
04. int space = mSettings.getInt(PREFS_STORAGE, View.VISIBLE);
05. int color = mSettings.getInt(S_COLOR, -1);
06. int sort = mSettings.getInt(_SORT, 1);
```

复制代码

本程序使用了SharedPreferences的轻型数据存储 首先读取了设置中的一些偏好设置 颜色，排序方式 等，稍后初始化列表显示的时候会使用到这些值。

```
01. mHandler = new EventHandler(Main.this, this, mFileMag,mCatalist);
02.     mHandler.setTextColor(color);
03.     mHandler.setShowThumbnails(thumb);
04.     mTable = mHandler.new TableRow();
05.
06.     /*sets the ListAdapter for our ListActivity and
07.      *gives our EventHandler class the same adapter
08.      */
09.     mHandler.setListAdapter(mTable);
10.     setListAdapter(mTable);
11.
```

[复制代码](#)

TableRow是EventHandler类中的一个内部类。
调用setListAdapter 传入一个Adapter，用来绑定ListActivity中的ListView，只有通过Adapter才可以把列表中的数据映射到ListView中,当我们需要显示内容丰富的ListView时。就需要自定义布局，此处自定义布局ArrayAdapter。TableRow就是继承自ArrayAdapter并且重写了getView方法。getView中设置了ListView 中每条数据的布局显示，
通过文件后缀名判断是什么类型的文件（TypeFilter.java中根据文件后缀名 判断文件类型），然后就显示什么图片。图片文件就显示缩略图。

```
01. listView().setOnItemLongClickListener(mHandler);
02.
03.     /* register context menu for our list view */
04.     registerForContextMenu(getListView());
```

[复制代码](#)

registerForContextMenu(getListView()); 为listView注册长按监听，显示上下文菜单。客户长按某个文件后会调用onCreateContextMenu 函数，显示长按菜单，供客户选择操作。
客户点击某项操作后，会调用onContextItemSelected 方法，根据客户选择使用showDialog（id）方法弹出相应的对话框。当你使用showDialog（id）方法时会调用回调函数onCreateDialog（ID）来显示创建对话框。

```
01. Bundle bundle = getIntent().getExtras();
02.     if (bundle != null){
```

[复制代码](#)

getIntent()取得上一个activity启动的intent，getExtras()得到intent所附带的额外数据。
if (bundle != null){
 如果是其他程序启动的文件管理器，判断传过来的路径，然后在改变界面显示的默认路径。
}else{
 如果不是，设置默认路径为机器内部flash路径。
}

```
01. /* setup buttons */
02.     int[] img_button_id = {R.id.home_flash_button,
03.                             R.id.home_sdcard_button,R.id.home_usbhost_button,
04.                             R.id.back_button,
05.                             R.id.manage_button, R.id.multiselect_button,
06.                             R.id.image_button,R.id.movie_button};
07.
08.     int[] button_id = {R.id.hidden_paste, R.id.hidden_copy, R.id.hidden_attach,
09.                        R.id.hidden_delete, R.id.hidden_move};
10.
11.     ImageButton[] bimg = new ImageButton[img_button_id.length];
12.     Button[] bt = new Button[button_id.length];
13.
14.     for(int i = 0; i < img_button_id.length; i++) {
15.         bimg[i] = (ImageButton)findViewById(img_button_id[i]);
16.         bimg[i].setOnClickListener(mHandler);
17.
18.         if(i < 5) {
19.             bt[i] = (Button)findViewById(button_id[i]);
20.             bt[i].setOnClickListener(mHandler);
21.         }
22.     }
```

[复制代码](#)

如私人在需要帮助的时候 可以点击按钮联系 方便联系在EventHandler中实现

例如在土介面添加的按键，并且按键说明，并让说明在eventhandler中实现。

```
01. mReceiver = new BroadcastReceiver() {
02.     .....
03.     IntentFilter filter = new IntentFilter();
04.         filter.addAction(Intent.ACTION_MEDIA_MOUNTED);
05.         filter.addAction(Intent.ACTION_MEDIA_REMOVED);
06.         filter.addAction(Intent.ACTION_MEDIA_BAD_REMOVAL);
07.         filter.addDataScheme("file");
08.         registerReceiver(mReceiver, filter);
复制代码
```

在onCreate 函数的最后，动态注册了sd卡，U盘等插入拔出的广播监听器，当有sd卡等插入时，接收插入广播，更新页面文件显示等。

```
01. @Override
02.     public void onItemClick(AdapterView parent, View view, int position, long id) {
03.         .....
复制代码
```

设置listview中的item 的点击事件，实现对文件的打开或查看，针对不同的文件类型，进行不同的操作，通过Intent setDataAndType 打开各种文件，如 文本类型的文件，则会通过Intent进行过滤type为“text/plain“调用系统自带的文件查看器，进行浏览；音乐类型的文件，则会通过Intent进行过滤type为“audio/*“调用系统自带的音乐播放器，进行播放，若用户安装有多种播放器，则会让用户进行选择 等。

```
private void selectFileType_dialog() {
当不是常见的文件类型时，会弹出对话框让客户选择用什么类型的程序打开此文件。
```

#在这里快速回复#

快速回复



回复

举报

dengzhengui



1

主题

40

帖子

190

积分

注册会员

积分

190

发消息

 楼主 | 发表于 前天 11:41 | 只看该作者

沙发

本帖最后由 dengzhengui 于 2013-11-30 11:55 编辑

onOptionsItemSelected(Menu menu) 创建右下角选项菜单，提供新建目录，搜索，设置，和退出功能。只会调用一次，他只会在Menu显示之前去调用一次，之后就不会再去调用。

@Override
public boolean onPrepareOptionsMenu(Menu menu) {
判断是否为根目录，然后显示或者隐藏 新建目录和搜索。
onPrepareOptionsMenu 是每次在display Menu之前，都会去调用，只要按一次Menu按键，就会调用一次。所以可以在这里动态的改变menu。

在onKeyDown 函数中设置了虚拟按键和实体按键监听，是根目录时，判断按两次退出键退出应用，防止误操作。

二 . FileManager.java 和 EventHandler.java是本程序的核心部分。
FileManager文件管理模块为核心模块，这个类处理所有的文件和文件夹操作，如复制，粘贴，压缩，重命名，搜索等。这个类已经把所有的方法做成一个一个模块了 此类也没有在本应用做任何图形界面的操作。
也就是说，如果你以后写应用时需要做什么样的文件操作，你稍微修改下就可以直接使用这个类。我们写程序时，为了避免程序的冗杂，减少各个类之间的依赖性，就需要经常使用低耦合和高内聚的设计理念。
FileManager中涉及了较多的IO数据，文件操作，与逻辑判断，每个方法前面都有一定的注释说明，而且方法名也都比较通俗易懂，大家可自行翻阅资料，结合google baidu学习。

EventHandler 事件处理类，implements了 OnClickListener, OnItemLongClickListener
通过调用FileManager类里面的方法，使复制，压缩等一些比较耗时的操作在本类的后台线程中执行。然后更新ui显示界面。

onClick(View v)

设置主activity 顶部的按钮监听，点击切换预览，返回键，sdcard键，flash键等

private void display_dialog(int type) {

弹出工具箱dialog，应用进程信息和应用备份选项。Wifi设置已被注释掉，需要添加只需去掉方法里面的注释，和AndroidManifest.xml里面关于wifi的注释。

public class TableRow extends ArrayAdapter<String> {

自定义ArrayAdapter，设置了显示界面文件的显示布局，上文提到过。

EventHandler中有一个内部类，在这里另起线程执行比较耗时的操作。是本程序的核心。

当一个程序第一次启动时，Android会同时启动一个对应的主线程(Main Thread)，主线程主要负责处理与UI相关的事件，如：用户的按键事件，用户接触屏幕的事件以及屏幕绘图事件，并把相关的事件分发到对应的组件进行处理。

所以主线程通常又被叫做UI线程。运行在UI线程里的任何方法都尽可能少做事情，一些比较耗时的操作不能在UI线程里面做，如复制，解压等一些操作，不要阻塞UI线程，因为当应用程序在5秒内没有响应输入的事件（例如，按键按下，屏幕触摸）时，

应用程序会显示ANR（应用程序无响应）。因此需要另起线程执行比较耗时的操作，这样会很好的提高用户体验，这对一个应用程序是非常重要的。

Android 1.5之后提供了一个工具类：AsyncTask，它使创建需要与用户界面交互的长时间运行的任务变得更简单。当你需要实现一些很复杂的操作并需要频繁地更新UI时，你可以使用它。步骤如下：

1) 继承AsyncTask

```
01. private class BackgroundWork extends AsyncTask<String, Void, ArrayList<String>> {
    复制代码
```

2) 实现AsyncTask中定义的下面一个或几个方法

①onPreExecute()

该方法将在执行实际的后台操作前被UI thread调用。可以在该方法中做一些准备工作，如在界面上显示一个进度条。

```
01. /**
02.      * This is done on the EDT thread. this is called before
03.      * doInBackground is called
04.      */
05.
06. @Override
07. protected void onPreExecute() {
08.     /* add by chenjd,[email]chenjd@allwinnertech.com[/email],20120506
09.      * lock standby when it is in file operation*/
10.     PowerManager pm = (PowerManager)
11.     mContext.getSystemService(Context.POWER_SERVICE);
12.     wl = pm.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK, WAKE_LOCK);
13.     wl.acquire();
14.
15.     switch(type) {
16.         case SEARCH_TYPE:
17.             pr_dialog = ProgressDialog.show(mContext, "Searching",
18.             "Searching current file system...",
19.             true, true);
20.             break;
21.
22.         case COPY_TYPE:
23.             pr_dialog = ProgressDialog.show(mContext, "Copying",
24.             "Copying file...",
25.             true, false);
26.             break;
```

这里申请了WakeLock，即使手机进入睡眠模式，也可以让进程继续执行文件操作，然后 弹出dialog对话框，显示正在搜索或者正在复制等.....

②doInBackground(Params...)

将在onPreExecute 方法执行后马上执行，该方法运行在后台线程中。这里将主要负责执行那些很耗时的后台计算工作。可以调用 publishProgress方法来更新实时的任务进度。该方法是抽象方法，子类必须实现。

01. @Override

02. protected ArrayList<String> doInBackground(String... params) {

复制代码

这里调用了FileManager中的方法，主要执行复制，搜索，解压等一些比较耗时的操作

③ **onProgressUpdate(Progress...)**

在publishProgress方法被调用后，UI thread将调用这个方法从而在界面上展示任务的进展情况，例如通过一个进度条进行展示。

这里没有复写这个方法，因为没有用进度条等需要更新ui的显示。

④ **onPostExecute(Result)**

在doInBackground执行完成后，onPostExecute 方法将被UI thread调用，后台的计算结果将通过该方法传递到UI thread.

01. @Override

02. protected void onPostExecute(final ArrayList<String> file) {

03. final CharSequence[] names;

04. int len = file != null ? file.size() : 0;

05.

06. /* add by chenjd,chenjd@allwinnertech.com,20120506

07. * unlock standby

08. */

09. if(wl != null)

10. wl.release();

复制代码

文件操作已经执行完，需要释放WakeLock


然后更新界面UI显示，提示操作完成，刷新目录,弹出搜索结果对话框等

文件管理器源码分析到这里就差不多了，如果你想要更加全面的理解，就需要你自己去查看学习了。谢谢。

最后，共享一个网上的文件管理器源码，请看附件FileManager.rar 。

里面的中文注释比代码还多，大家有时间，有兴趣可以看看，如果你还是看不懂，一定是你打开的方式不对，换个姿势重新打开。

这个文件管理器有一些bug，大家有兴趣也可以解解看，优化一下。

 [FileManager.rar](#)

1.52 MB, 下载次数: 0

回复

支持

反对

举报

caishaoyi



1

111

359

主题 | 帖子 | 积分

中级会员

积分 359

QQ交谈

发表于 前天 12:12 | 只看该作者

板凳

沙发！！！

SIGNATURE

本人已死，有事烧纸～ 小事招魂，大事挖坟～[img]http://192.168.1.240/scbbs/static/image/smiley/grapema

回复

编辑

发帖 ▾

回复

◀ 返回列表



B

A

高级模式

发表回复

☐ 回帖后跳转到最后一页

本版积分规则