

TraverseFile

北京小 C

教程介绍:

今天我们来介绍如何遍历 PlayBook 上的文件。虽然 PlayBook 支持 AIR，但是却不是完全支持某些文件命令。比如:

`browseForDirectory(title:String):void`

显示一个目录选择器对话框，用户可从中选择一个目录。

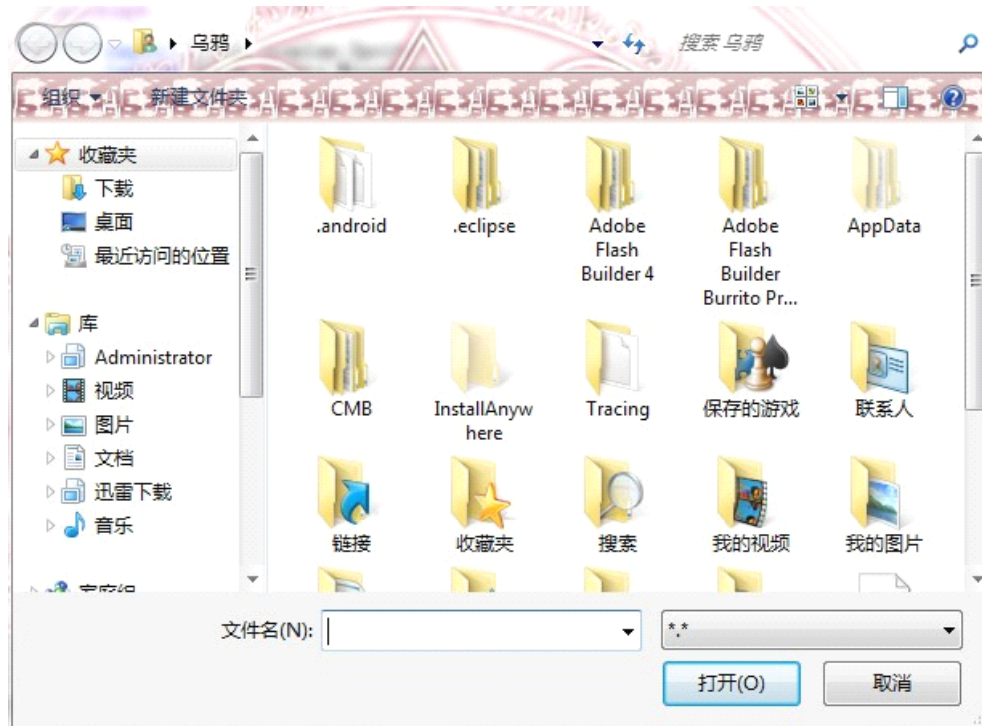
`browseForOpen(title:String, typeFilter:Array = null):void`

显示“打开文件”对话框，用户可从中选择要打开的文件。

`browseForOpenMultiple(title:String, typeFilter:Array = null):void`

显示“打开文件”对话框，用户可从中选择一个或多个要打开的文件。

既然不能弹出框让用户选择需要的文件。如下，此不能在 PlayBook 上工作。



那么我们应该如何让用户来选择呢。或者我们又该如何来指导其完整路径呢。

教程说明：绘制一个简易的资源管理器。只能查看资源，打开文件夹，进入上一级。

学习流程：

1、首先我们需要一个导航条来实现，返回上一层，进入几个文件夹的选项。此为 UI 代码如下：

//导航条设置

```
private function button_do():void{

    var up:LabelButton=new LabelButton();
    up.x=10;
    up.y=1;
    up.label="上一级";
    up.scaleX=0.5;
    up.scaleY=0.7;
    up.name="up";

    var app:LabelButton=new LabelButton();
    app.x=111.5;
    app.y=1;
    app.label="程序目录";
    app.scaleX=0.5;
    app.scaleY=0.7;
    app.name="app";

    var des:LabelButton=new LabelButton();
    des.x=213;
    des.y=1;
    des.label="储存目录";
    des.scaleX=0.5;
    des.scaleY=0.7;
    des.name="sppS";

    var doc:LabelButton=new LabelButton();
    doc.x=314.5;
    doc.y=1;
    doc.label="用户目录";
    doc.scaleX=0.5;
    doc.scaleY=0.7;
    doc.name="user";

    title.x=416;
    title.y=2;
    title.mouseEnabled=false;
    title.width=stage.stageWidth-title.x;
```

```

var tf:TextFormat=new TextFormat();
tf.size=20;
title.defaultTextFormat=tf;
title.text="";

container_button.graphics.beginFill(0xcccccc);
container_button.graphics.drawRect(0,0,1024,30);
container_button.graphics.endFill();

container_button.addChild(up);
container_button.addChild(app);
container_button.addChild(des);
container_button.addChild(doc);
container_button.addChild(title);

container_button.addEventListener(MouseEvent.CLICK,button_click);
}

```

2、接下来我们为这几个按钮点击后响应事件。

在这里有几个只读的静态命令。用于返回固定的文件夹。

File.applicationDirectory 包含应用程序已安装文件的文件夹。

File.applicationStorageDirectory 应用程序的专用存储目录。

File.userDirectory 用户目录。

代码如下：

```

//按钮点击
private function button_click(evt:MouseEvent):void{
    switch (evt.target.name) {
        case "up":
            dir_up_do();
            break;
        case "app":
            file_dir(File.applicationDirectory.nativePath);
            break;
        case "sppS":
            file_dir(File.applicationStorageDirectory.nativePath);
            break;
        case "user":
            file_dir(File.userDirectory.nativePath);
            break;
        default:
            break;
    }
}
}

```

3、接下来我们遍历文件夹，并绘制文件列表。

这里做了一个文件夹判断，

当为文件夹时，绘制 OPEN 按钮，并允许打开。

不为文件夹时，绘制 FILE 按钮，但不允许打开。

更简单的写法可以将中间的界面绘制部分提取出来。

getDirectoryListing 将返回与此 File 对象表示的目录中的文件和目录对应的 File 对象的数组。

代码如下：

```
//遍历文件列表
```

```
private function file_dir(path:String):void{  
    clear_line();
```

```
  
    now_path=path;  
    title.text=now_path;
```

```
  
    var filesdir:File=new File(path);  
    var files:Array=filesdir.getDirectoryListing();
```

```
  
    var len:int=files.length;
```

```
    for (var i:uint = 0; i < len; i++) {
```

```
        if(files[i].isDirectory==true) {
```

```
            //为目录时绘制
```

```
            var dir_lab:LabelButton=new LabelButton();  
            dir_lab.x=24;  
            dir_lab.width=90;  
            dir_lab.scaleY=0.5;  
            dir_lab.y=40+i*dir_lab.height+5;  
            dir_lab.label="open";  
            dir_lab.name=files[i].nativePath;  
            trace(files[i].nativePath);  
            container_list.addChild(dir_lab);
```

```
  
            var text_line:TextField=new TextField();  
            text_line.mouseEnabled=false;  
            text_line.x=120;  
            text_line.y=dir_lab.y+2;  
            text_line.width=880;  
            text_line.text=dir_lab.name;  
            container_list.addChild(text_line);
```

```
  
            dir_lab.addEventListener(MouseEvent.CLICK, dir_click);
```

```

    }else{
        var txt_lab:LabelButton=new LabelButton();
        txt_lab.x=24;
        txt_lab.width=90;
        txt_lab.scaleY=0.5;
        txt_lab.y=40+i*txt_lab.height+5;
        txt_lab.label="file";
        txt_lab.name=files[i].nativePath;
        trace(files[i].nativePath);
        container_list.addChild(txt_lab);

        var text_line:TextField=new TextField();
        text_line.mouseEnabled=false;
        text_line.x=120;
        text_line.y=txt_lab.y+2;
        text_line.width=880;
        text_line.text=txt_lab.name;
        container_list.addChild(text_line);
    }
}
}

```

4、打开目录，因为我们将完整地址传递到 name 中。所以我们直接用 name 来表示。
代码如下：

```

//目录打开
private function dir_click(evt:MouseEvent):void{
    file_dir(evt.target.name);
}

```

5、返回上一级。

代码如下：

```

//返回上一级
private function dir_up_do():void{
    if(now_path!=null){
        var file:File=new File(now_path);
        if(file.parent != null)
        {
            file_dir(file.parent.nativePath);
        }
    }
}

```

6、清空显示列表，每次遍历文件列表前都要清空上一次的显示列表。

代码如下：

```
//清空line
private function clear_line():void{
    while(container_list.numChildren!=0){
        container_list.removeChildAt(0);
    }
    container_list.y=0;
}
```

7、其他部分

代码如下：

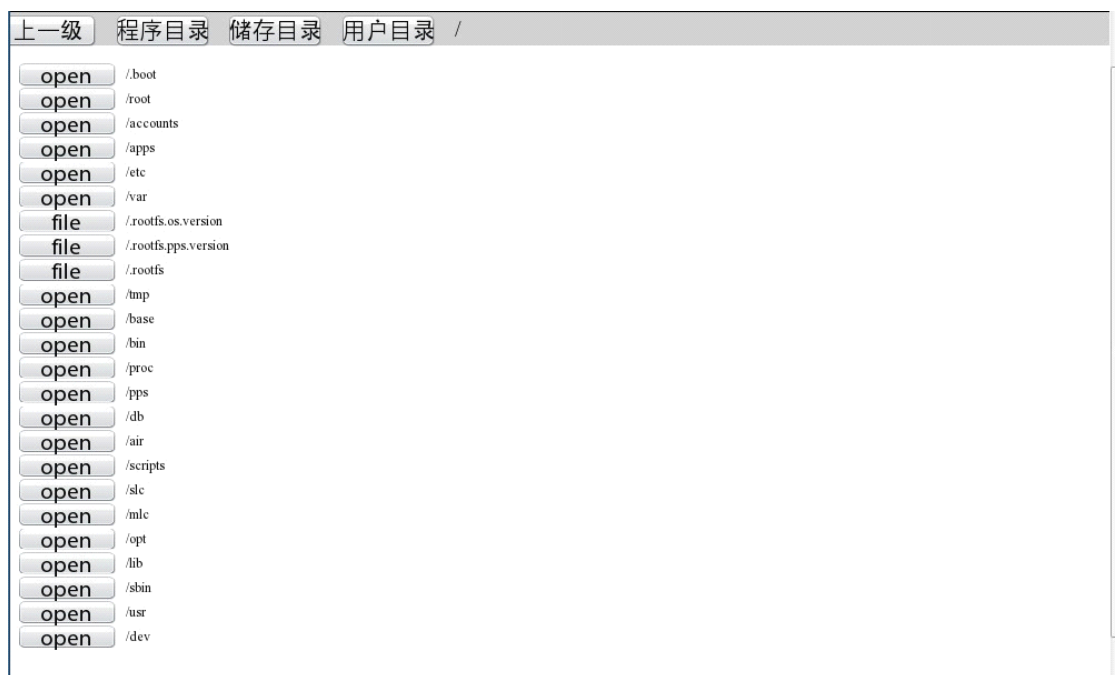
```
package
{
    import flash.display.Sprite;
    import flash.events.MouseEvent;
    import flash.filesystem.*;
    import flash.text.*;

    import qnx.ui.buttons.LabelButton;

    [SWF(width='1024', height='600', backgroundColor='#FFFFFF', frameRate='30')]
    public class TraverseFile extends Sprite
    {
        private var container_list:Sprite=new Sprite;
        private var container_button:Sprite=new Sprite;
        private var now_path:String;
        private var title:TextField=new TextField();

        public function TraverseFile()
        {
            button_do();
            this.addChild(container_list);
            this.addChild(container_button);
        }
    }
}
```

8、测试下



界面看上去有点怪异。

不过没关系，相信您可以配合 FLASH 绘制出你想要的界面。

您可以绘制自己的资源管理器，然后放进您的程序。

让用户选择打开其选择的文件。

或者您只是需要遍历文件的完整地址。

但这都是必须的过程。