

# **BlackBerry 照相编程**

作者：俞伟

# 目录

BlackBerry 拍照流程介绍 .....	2
获取支持的图片规格.....	2
显示并确定照片设置.....	5
启动摄像头准备拍照.....	5
拍照.....	7

BlackBerry SDK 5.0 以及之前的版本通过 MMAPAPI 实现拍照功能，主要的功能包包括：

- javax.microedition.media
- javax.microedition.media.control
- javax.microedition.amms.control.camera

拍照应用包括如下几部分：

- 获取黑莓手机相机支持信息，照片大小，格式，照片质量
- 显示并确定照片设置
- 打开相机设备，设置自动对焦，显示摄像头图像
- 拍照

## 获取支持的图片规格

照片规格通过 4 种属性来定义：照片宽度，照片高度，照片格式，和照片质量。为了方便定义照片规格，需要一个照片规格类，比如叫 `EncodingProperties`，代码如下：

```
/**
 * 定义照片规格，通过函数getFullEncoding()为拍照函数VideoControl.getSnapshot()提供
 * 照片定义
 */
public final class EncodingProperties
{
    /** 照片格式 */
    private String _format;

    /** 照片宽 */
    private String _width;

    /** 照片高 */
    private String _height;

    /** 照片质量 */
    private String _quality;

    public void setFormat(String format)
    {
        _format = format;
    }

    public void setWidth(String width)
    {
        _width = width;
    }
}
```

```

public void setHeight(String height)
{
    _height = height;
}

public void setQuality(String quality)
{
    _quality = quality;
}

/**
 * VideoControl.getSnapshot(EncodingProperties.getFullEncoding())
 */
public String getFullEncoding()
{
    StringBuffer fullEncoding = new StringBuffer();

    fullEncoding.append("encoding=");
    fullEncoding.append(_format);

    fullEncoding.append("&width=");
    fullEncoding.append(_width);

    fullEncoding.append("&height=");
    fullEncoding.append(_height);

    fullEncoding.append("&quality=");
    fullEncoding.append(_quality);

    return fullEncoding.toString();
}
}

```

照片规格类定义完毕，下面要获取照片规格，代码如下：

```

private EncodingProperties[] getEncodingList(){
    try
    {
        // 最终需要的照片规格
        EncodingProperties[] encodings = null;

        // 获取照片规格
        String encodingString = System.getProperty("video.snapshot.encodings");

        // 初步解析照片规格
        String[] properties = StringUtils.stringToKeywords(encodingString);

        Vector encodingList = new Vector();

        String encoding = "encoding";
        String width = "width";
        String height = "height";
        String quality = "quality";
    }
}

```

```

EncodingProperties temp = null;

// 开始解析照片规格并组装成我们定义的照片规格
for(int i = 0; i < properties.length ; ++i)
{
    if( properties[i].equals(encoding))
    {
        if(temp != null && temp.isComplete())
        {
            // Add a new encoding to the list if it has been
            // properly set.
            encodingList.addElement( temp );
        }
        temp = new EncodingProperties();

        // Set the new encoding's format
        ++i;
        temp.setFormat(properties[i]);
    }
    else if( properties[i].equals(width))
    {
        // Set the new encoding's width
        ++i;
        temp.setWidth(properties[i]);
    }
    else if( properties[i].equals(height))
    {
        // Set the new encoding's height
        ++i;
        temp.setHeight(properties[i]);
    }
    else if( properties[i].equals(quality))
    {
        // Set the new encoding's quality
        ++i;
        temp.setQuality(properties[i]);
    }
}

// If there is a leftover complete encoding, add it.
if(temp != null && temp.isComplete())
{
    encodingList.addElement( temp );
}

// Convert the Vector to an array for later use
encodings = new EncodingProperties[ encodingList.size() ];
encodingList.copyInto( (Object[])encodings );
}
catch (Exception e)
{
    // Something is wrong, indicate that there are no encoding options
    encodings = null;
}
// 返回需要的照片规格
return encodings;
}

```

## 显示并确定照片设置

我们已经有了 `EncodingProperties`，可以重写 `toString()` 函数获取照片规格并用 `Radio Button` 的格式显示出来，如下图所示：

- 2048 x 1536 jpeg (superfine)
- 2048 x 1536 jpeg (normal)
- 2048 x 1536 jpeg (fine)
- 1024 x 768 jpeg (normal)
- 1024 x 768 jpeg (fine)
- 1024 x 768 jpeg (superfine)
- 480 x 360 jpeg (normal)
- 480 x 360 jpeg (fine)
- 480 x 360 jpeg (superfine)

这是 BlackBerry 9700 所支持的照片规格。

## 启动摄像头准备拍照

首先要初始化摄像头，然后启动摄像头，设置自动对焦，再把展现摄像的 `Field` 添加到屏幕上。

```
/**
 * 初始化Player, VideoControl 和 VideoField
 */
private void initializeCamera()
{
    try
    {
        // 注意这里的参数是 "capture://video", Player作为摄像用
        Player player = Manager.createPlayer( "capture://video" );

        // 预备摄像
        player.realize();
    }
}
```

```

// 获取 VideoControl
_videoControl = (VideoControl)player.getControl( "VideoControl" );

// 获取 FocusControl
_focusControl = (FocusControl)
player.getControl("javax.microedition.amms.control.camera.FocusControl");

if (_videoControl != null)
{
    // 获取显示Video的UI组件: Field
    _videoField = (Field) _videoControl.initDisplayMode
        (VideoControl.USE_GUI_PRIMITIVE,
         "net.rim.device.api.ui.Field");
    _videoControl.setDisplayFullScreen(true);
    _videoControl.setVisible(true);
}

if (_focusControl != null)
{
    _focusControl.setFocus(FocusControl.AUTO);
}

// 启动摄像头
player.start();

// 添加到屏幕
add(_videoField);
}
catch(Exception e)
{
    System.out.println("ERROR " + e.getClass() + ": " + e.getMessage());
}
}

```

两个关键的类分别是 VideoControl 和 FocusControl。VideoControl 控制在哪个 UI 组件上显示摄像头捕捉到的 Video。这里指定是在 Field 上显示 Video，显示模式为全屏。FocusControl 控制对焦，可以设置自动对焦或者自定对焦。

## 拍照

拍照要设定照片规格，在使用 VideoControl 拍照，代码如下：

```
/**
 * 按指定的照片规格拍照
 */
public Bitmap takePicture(){
    try{
        Bitmap image = null;
        String encoding = null;

        if( _encodings != null )
        {
            // 指定照片规格
            encoding = _encodings[_indexOfEncoding].getFullEncoding();
        }

        // 指定照片规格，使用VideoControl拍照
        // 照片规格范例: “encoding=jpeg&width=1024&height=768&quality=normal”
        byte[] raw = _videoControl.getSnapshot( encoding );

        // 组装成图片
        image = Bitmap.createBitmapFromBytes( raw, 0, -1, 1 );
    }
    catch(Exception e)
    {
        PingAn.errorDialog("ERROR " + e.getClass() + ": " + e.getMessage());
    }

    return image;
}
```

这段代码指定照片规格并使用 VideoControl 拍照。照片规格范例：

```
String encoding = "encoding=jpeg&width=1024&height=768&quality=normal";
```

使用照片规格拍照：

```
byte[] raw = _videoControl.getSnapshot(encoding);
```

获取的图片二进制数据流可以被组装成图片。