How to capture an image at given interval using MMAPI

This article demonstrates how to capture an image at given interval using MMAPI. In this article, the given code snippet takes a snapshot at the interval of 5 seconds.

VideoMIDlet.java

```
package CamDemo;
import java.io.IOException;
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;
import javax.microedition.midlet.MIDlet;
import javax.microedition.media.control.VideoControl;
import java.util.*;

public class VideoMIDlet extends MIDlet implements CommandListener {

    // Class to capture the snapshot
    public class Video extends Thread {
 SnapTimerTask st;
 public Video(SnapTimerTask st) {
  this.st = st;
 }

 public void run() {
         captureVideo();
        }

 public void captureVideo() {
     try {
       byte[] raw = videoControl.getSnapshot(null);
       Image image = Image.createImage(raw, 0, raw.length);
       form.append(image);
       display.setCurrent(form);

       player.close();
       player = null;
       videoControl = null;
       showCamera();
       } catch (MediaException me) { }
 }
    };

    // Task to initiate the task to take the snapshot
    class SnapTimerTask extends TimerTask {
 public final void run() {
       video = new Video(this);
       video.start();
 }
    };

    private Display display;
    private Form form;
    private Command exit,back,camera,cancel;
    private Player player;
    private VideoControl videoControl;
    private Video video;
    private Timer timer;
    private SnapTimerTask task;


    public VideoMIDlet() {
        exit = new Command("Exit", Command.EXIT, 0);
        camera = new Command("Camera", Command.SCREEN, 0);
        back = new Command("Back", Command.BACK, 0);

        cancel = new Command("Cancel", Command.STOP, 0);

        form = new Form("Capture Video");
        form.addCommand(camera);
        form.setCommandListener(this);
    }

    public void startApp() {
        display = Display.getDisplay(this);
        display.setCurrent(form);
    }

    public void pauseApp() {}

    public void destroyApp(boolean unconditional) {}

    public void commandAction(Command c, Displayable s) {
        if (c == exit) {
            destroyApp(true);
            notifyDestroyed();
        }
 else if (c == camera) {
            showCamera();
        }
 else if (c == back) {
     timer.cancel();
            display.setCurrent(form);
 }
 else if(c == cancel)
     timer.cancel();
    }

    public void showCamera() {
        try {
     timer = new Timer();
```


(C) Copyright Nokia 2013. All rights reserved.

```
      task = new SnapTimerTask();
      timer.schedule(task,5000); // 5000 ms
             player = Manager.createPlayer("capture://video");
             player.realize();

             videoControl = (VideoControl)player.getControl("VideoControl");
             Canvas canvas = new VideoCanvas(this, videoControl);
             canvas.addCommand(back);
             canvas.addCommand(exit);
        canvas.addCommand(cancel);
             canvas.setCommandListener(this);
             display.setCurrent(canvas);
             player.start();
         }
 catch (IOException ioe) {}
 catch (MediaException me) {}
     }
}
```

VideoCanvas.java

The following code snippet shows Canvas for video.

```
package CamDemo;
import javax.microedition.lcdui.*;
import javax.microedition.media.MediaException;
import javax.microedition.media.control.VideoControl;

public class VideoCanvas extends Canvas {
    private VideoMIDlet midlet;

    public VideoCanvas(VideoMIDlet midlet, VideoControl videoControl) {
        int width = getWidth();
        int height = getHeight();
        this.midlet = midlet;

        videoControl.initDisplayMode(VideoControl.USE_DIRECT_VIDEO, this);
        try {
            videoControl.setDisplayLocation(2, 2);
            videoControl.setDisplaySize(width - 4, height - 4);
        } catch (MediaException me) {}
        videoControl.setVisible(true);
    }

    public void paint(Graphics g) {
        int width = getWidth();
        int height = getHeight();

        g.setColor(0x00ff00);
        g.drawRect(0, 0, width - 1, height - 1);
        g.drawRect(1, 1, width - 3, height - 3);
    }
}
```

Further Reading

- TimerTask class in J2ME

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.