



ID	KIJ001370	Creation date	June 11, 2009
Platform	S60 3rd Edition, FP2, S60 5th Edition	Devices	Nokia N96, Nokia 5730 XpressMusic, Nokia 5800 XpressMusic
Category	Java ME	Subcategory	Multimedia

Keywords (APIs, classes, methods, functions): Display.setCurrent(Displayable)

Description

When streaming video on Canvas and switching from that Canvas to another Displayable without stopping the streaming, the switching is not done correctly in certain devices. Either the Canvas area for streamed video appears on the switched Displayable or, when switched back to the Canvas, the switched Displayable area appears inside of the Canvas area for streamed video.

Alternatively, for some devices, video on Canvas disappears completely and part of the other Displayable will appear within the video streaming area.

How to reproduce

To test this issue, implement a test MIDlet using the following code:

```
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;
import javax.microedition.midlet.MIDlet;
import java.io.*;

public class VidMIDlet extends MIDlet implements CommandListener, PlayerListener
{
    Display display;
    Form mainForm;
    Form statusForm;
    VidCanvas canvas;

    VideoControl videoControl;
    Command exitCmd = new Command("Exit", Command.EXIT, 1);
    Command streamCmd = new Command("Play", Command.SCREEN, 2);
}
```

KIJ001370_-_Switching_Displayable_and_repainting_Canvas_result_in_incorrect_behaviour_while_streaming_video

```
Command backMainCmd= new Command("Back", Command.SCREEN, 2);
Command backCanvasCmd = new Command("Back to Video", Command.SCREEN, 2);
Command statusCmd = new Command("Status", Command.SCREEN, 2);

Player player;

String url = "Place your test URL here";

String info;
long mediaTime;

public VidMIDlet()
{
    display = Display.getDisplay(this);
    mainForm = new Form("Displayable test");
    mainForm.addCommand(streamCmd);
    mainForm.addCommand(exitCmd);
    mainForm.setCommandListener(this);
    display.setCurrent(mainForm);
}

public void startApp()
{
}

private void startVidStream(String urlString)
{
    String streamUrl = urlString;

    info = "";
    videoControl = null;
    canvas = new VidCanvas();
    canvas.setCommandListener(this);
    canvas.addCommand(statusCmd);
    canvas.addCommand(backMainCmd);
    canvas.addCommand(exitCmd);
    canvas.setFullScreenMode(true);
    display.setCurrent(canvas);

    try
    {
        player = Manager.createPlayer(streamUrl);
        player.addPlayerListener(this);
        info += "URL: " + streamUrl + "\n";
        info += "Video player created \n";
        player.realize();
        info += "Video player realized \n";
        player.prefetch();
        info += "Video player prefetched\n";

        videoControl = (VideoControl)player.getControl("VideoControl");
        if(videoControl != null)
        {
            videoControl.initDisplayMode(VideoControl.USE_DIRECT_VIDEO, canvas);
            videoControl.setDisplayLocation(30, 30);
            videoControl.setVisible(true);
            try
            {
                {
                    videoControl.setDisplaySize(canvas.getWidth() / 2, canvas.getHeight() / 2);
                }
            }
            catch(MediaException me)
            {
            }
        }
    }
}
```

```

        info += "setDisplaySize not supported\n";
    }
} else
{
    info += "VideoControl not supported\n";
}

try
{
    player.start();
}
catch(Exception e)
{
    info += "Player start Exception: " + e.getMessage() + "\n";
}

    canvas.repaint();

}
catch(Error e)
{
    info += "Error: " + e.getMessage() + "\n";
}
catch(Exception e)
{
    info += " - Exception: " + e.getMessage() + "\n";
}
}

public void statusForm()
{
    statusForm = new Form("Status");
    statusForm.setTitle("Error");
    statusForm.addCommand(exitCmd);
    statusForm.addCommand(backCanvasCmd);
    statusForm.setCommandListener(this);
    statusForm.append(info);
    display.setCurrent(statusForm);
}

public void commandAction(Command command, Displayable displayable)
{
    if(command == exitCmd)
        notifyDestroyed();

    else if(command == streamCmd)
    {
        startVidStream(url);
    }
    else if(command == statusCmd)
        statusForm();

    else if(command == backCanvasCmd)
    {
        // player.close(); //closing Player seems to do the trick
        canvas.setCommandListener(this);
        display.setCurrent(canvas);
    }

    else if(command == backMainCmd)
    {
        player.deallocate();
    }
}

```

```

        player.close();
        player = null;
        display.setCurrent(mainForm);
    }
}

public void pauseApp()
{
}

public void destroyApp(boolean unconditional)
{
    try
    {
        canvas = null;
        player.deallocate();
        player.close();
        player = null;
    }
    catch(Exception ex)
    {
        info += ex.getMessage();
        player = null;
    }
}

public void playerUpdate(Player player1, String s, Object obj)
{
}
}

```

Code for test Canvas:

```

import javax.microedition.lcdui.*;

public class VidCanvas extends Canvas
{
    public VidCanvas()
    {
    }

    public void paint(Graphics g)
    {
        g.setGrayScale(255);
        g.drawString("Test canvas. Select status by pressing LSK.", 10, 10, 10);

        g.setGrayScale(0);
        g.drawRect(0, 0, getWidth(), getHeight());
    }
}

```

After installing the MIDlet, test this issue as follows:

1. Press **Play** with the LSK.

2. Once the video stream starts, press the LSK to prompt the Options list and select **Status**. For some of the affected devices, the video stream area on the Canvas will be displayed on top of the **Status** form.

In some devices the video will not be played at all and part of the **Status** form appears under the Canvas stream area.

3. Return to the video. In some of the affected devices part of the **Status** form will be displayed on the video play area.

Solution

This issue is expected to be fixed in future software versions for the affected devices starting from S60 3rd Edition, Feature Pack 2 and S60 5th Edition.