

Selected tray : Drawer 2

Do you want to print a test page? [y/n]



linux)

I published on github a small project (MediaTrayTest _) that allows to interactively list the printer trays and ask to the printer to print an empty page using the selected tray.

As many of you know, this should work without any problem for Windows platforms but it's possible that it won't work using Linux / Unix platforms. It's a known issue of the JRE, you can read more at the end of the article).

Here what happen when you execute the Main.class (you have to run it using java Main in the Terminal/Console, it cannot work with an ide):

```
Write the printer name or press enter for [Network_printer_example]

IPP Printer: Network_printer_example

0: Auto - sun.print.CustomMediaTray

1: Stack Bypass - sun.print.CustomMediaTray

2: Drawer 1 - sun.print.CustomMediaTray

3: Drawer 2 - sun.print.CustomMediaTray

4: Drawer 3 - sun.print.CustomMediaTray

5: Drawer 4 - sun.print.CustomMediaTray

6: Side Paper Deck - sun.print.CustomMediaTray

Select tray target id:
```



The code is sell explaining and commented.

```
public static void main(String args[]) {
       // get default printer
       PrintService defaultPrintService = PrintServiceLookup.lookupDefaultPrintService()
       // suggest the use of the default printer
       System.out.println("Write the printer name or press enter for [" + defaultPrintSe
       Console console = System.console();
       // read from the console the name of the printer
       String printerName = console.readLine();
       // if there is no input, use the default printer
       if (printerName == null || printerName.equals("")) {
           printerName = defaultPrintService.getName();
       }
       // the printer is selected
       AttributeSet aset = new HashAttributeSet();
       aset.add(new PrinterName(printerName, null));
       // selection of all print services
       PrintService[] services = PrintServiceLookup.lookupPrintServices(null, aset);
       // we store all the tray in a hashmap
       Map<Integer, Media> trayMap = new HashMap<Integer, Media>(10);
       // we chose something compatible with the printable interface
       DocFlavor flavor = DocFlavor.SERVICE FORMATTED.PRINTABLE;
       for (PrintService service : services) {
           System.out.println(service);
           // we retrieve all the supported attributes of type Media
           // we can receive MediaTray, MediaSizeName, ...
           Object o = service.getSupportedAttributeValues(Media.class, flavor, null);
           if (o != null && o.getClass().isArray()) {
                for (Media media : (Media[]) o) {
                    // we collect the MediaTray available
                    if (media instanceof MediaTray) {
                        System.out.println(media.getValue() + " : " + media + " - " + media
                        trayMap.put(media.getValue(), media);
```



System. Out. princing Serect tray target in

```
String mediaId = console.readLine();
   MediaTray selectedTray = (MediaTray) trayMap.get(Integer.valueOf(mediaId));
    System.out.println("Selected tray : " + selectedTray.toString());
    System.out.println("Do you want to print a test page? [y/n]");
    String printPage = console.readLine();
    if (printPage.equalsIgnoreCase("Y")) {
        // we have to add the MediaTray selected as attribute
        PrintRequestAttributeSet attributes = new HashPrintRequestAttributeSet();
        attributes.add(selectedTray);
        // we create the printer job, it print a specified document with a set of job
        DocPrintJob job = services[0].createPrintJob();
        try {
            System.out.println("Trying to print an empty page on : " + selectedTray.
            // we create a document that implements the printable interface
            Doc doc = new SimpleDoc(new PrintableDemo(), DocFlavor.SERVICE FORMATTED
            // we print using the selected attributes (paper tray)
            job.print(doc, attributes);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
static class PrintableDemo implements Printable {
    @Override
    public int print(Graphics pg, PageFormat pf, int pageNum) {
        // we print an empty page
        if (pageNum >= 1)
            return Printable.NO SUCH PAGE;
        pg.drawString("", 10, 10);
        return Printable.PAGE_EXISTS;
```





You can find the following open bugs for the selection of the paper tray with Linux:

https://bugs.openjdk.java.net/browse/JDK-6357887

https://bugs.openjdk.java.net/browse/JDK-7107175

https://bugs.openjdk.java.net/browse/JDK-7107138

We are working to find a solution and override some JRE (rt.jar) classes to use the paper tray in Linux/Unix.

The problem is in sun.print.PSPrinterJob.java. The print attributes passed are not read at all.

java.print.UnixPrintJob.java read the attributes and it transforms them in a cups string in the print method (there are some errors in this method too), this can give you an idea about how correct PSPrinterJob.

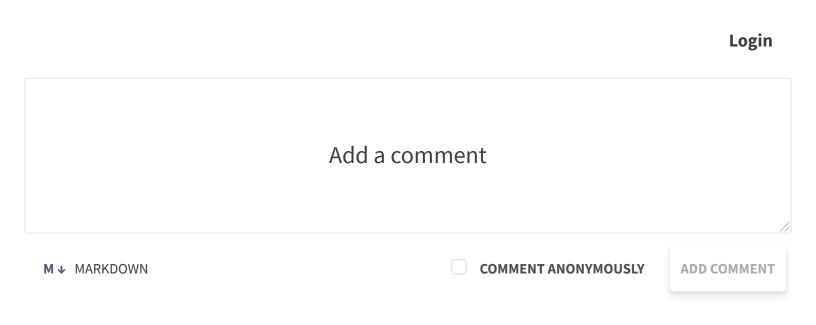
Adding this code to PSPrinterJob solved our problem.

```
if (super.attributes != null){
   Attribute[]attrs = super.attributes.toArray();
   CustomMediaTray customTray = null;
   for (int i=0; i<attrs.length; i++) {</pre>
        Attribute attr = attrs[i];
        Class category = attr.getCategory();
        if (category == Media.class) {
            if (attr instanceof CustomMediaTray) {
                customTray = (CustomMediaTray)attr;
                String choice = customTray.getChoiceName();
                if (choice != null) {
                    if (mOptions == null){
                        mOptions = "";
                mOptions += " media="+choice;
            }
       }
```



Author

Marco Molteni



Powered by **Commento**