

Busy Developers' Guide to HSLF drawing layer

by Yegor Kozlov

1. Busy Developers' Guide to HSLF drawing layer

1.1. Index of Features

- [How to create a new presentation and add new slides to it](#)
- [How to retrieve or change slide size](#)
- [How to get shapes contained in a particular slide](#)
- [Drawing a shape on a slide](#)
- [How to work with pictures](#)
- [How to set slide title](#)
- [How to work with slide/shape background](#)
- [How to create bulleted lists](#)
- [Hyperlinks](#)

1.2. Features

1.2.1. New Presentation

```
//create a new empty slide show
SlideShow ppt = new SlideShow();

//add first slide
Slide s1 = ppt.createSlide();

//add second slide
Slide s2 = ppt.createSlide();

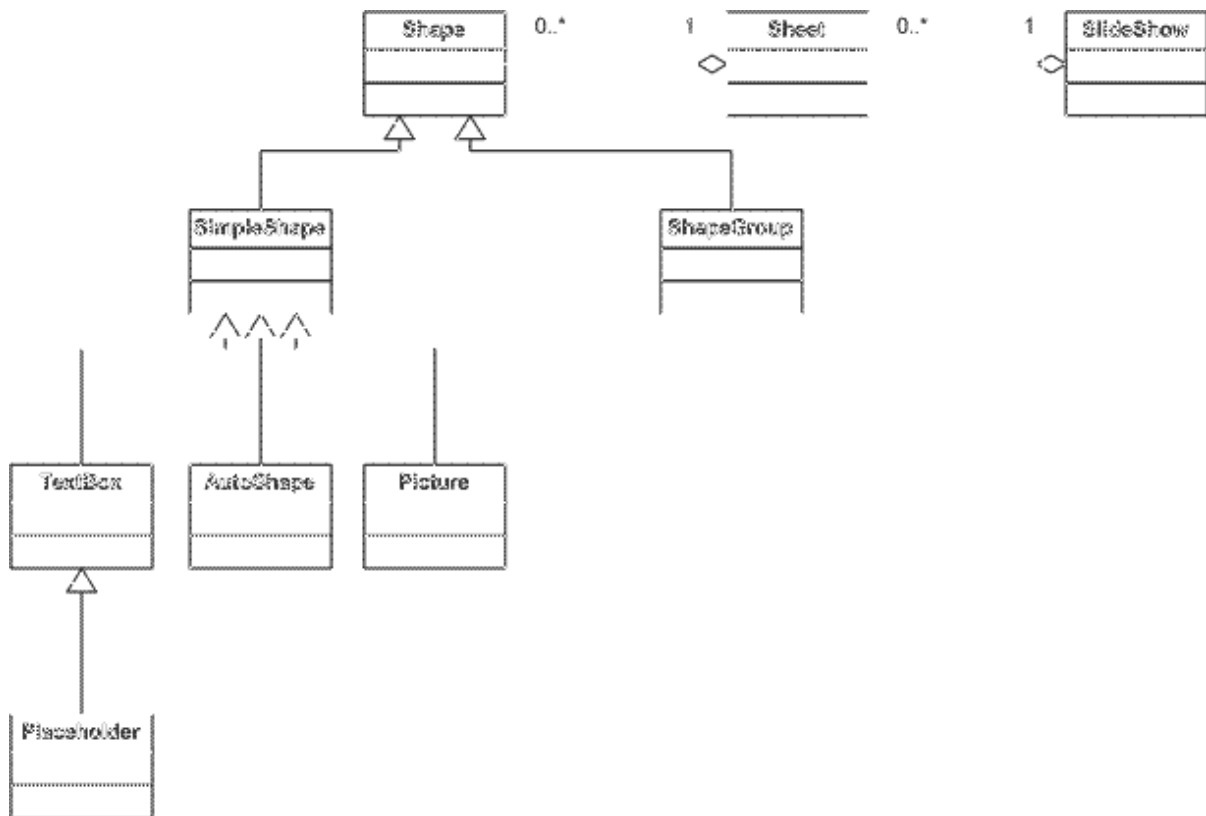
//save changes in a file
FileOutputStream out = new FileOutputStream("slideshow.ppt");
ppt.write(out);
out.close();
```

1.2.2. How to retrieve or change slide size

```
SlideShow ppt = new SlideShow(new HSLFSlideShow("slideshow.ppt"));  
//retrieve page size. Coordinates are expressed in points (72 dpi)  
java.awt.Dimension pgsz = ppt.getPageSize();  
int pgx = pgsz.width; //slide width  
int pgy = pgsz.height; //slide height  
  
//set new page size  
ppt.setPageSize(new java.awt.Dimension(1024, 768));  
//save changes  
FileOutputStream out = new FileOutputStream("slideshow.ppt");  
ppt.write(out);  
out.close();
```

1.2.3. How to get shapes contained in a particular slide

The superclass of all shapes in HSLF is the Shape class - the elemental object that composes a drawing. The following picture shows the class tree of HSLF shapes:



Class Tree of HSLF Shapes

The following fragment demonstrates how to iterate over shapes for each slide.

```
SlideShow ppt = new SlideShow(new HSLFSlideShow("slideshow.ppt"));
//get slides
Slide[] slide = ppt.getSlides();
for (int i = 0; i < slide.length; i++){
    Shape[] sh = slide[i].getShapes();
    for (int j = 0; j < sh.length; j++){
        //name of the shape
        String name = sh[j].getShapeName();

        //shapes's anchor which defines the position of this shape in the slide
        java.awt.Rectangle anchor = sh[j].getAnchor();

        if (sh[j] instanceof Line){
            Line line = (Line)sh[j];
            //work with Line
        } else if (sh[j] instanceof AutoShape){
            AutoShape shape = (AutoShape)sh[j];
            //work with AutoShape
        } else if (sh[j] instanceof TextBox){
            TextBox shape = (TextBox)sh[j];
            //work with TextBox
        } else if (sh[j] instanceof Picture){
            Picture shape = (Picture)sh[j];
            //work with Picture
        }
    }
}
```

1.2.4. Drawing a shape on a slide

Note:

To work with graphic objects HSLF uses Java2D classes that may throw exceptions if graphical environment is not available. In case if graphical environment is not available, you must tell Java that you are running in headless mode and set the following system property: `java.awt.headless=true` (either via `-Djava.awt.headless=true` startup parameter or via `System.setProperty("java.awt.headless", "true")`).

When you add a shape, you usually specify the dimensions of the shape and the position of the upper left corner of the bounding box for the shape relative to the upper left corner of the slide. Distances in the drawing layer are measured in points (72 points = 1 inch).

```
SlideShow ppt = new SlideShow();

Slide slide = ppt.createSlide();

//Line shape
```

```
Line line = new Line();
line.setAnchor(new java.awt.Rectangle(50, 50, 100, 20));
line.setLineColor(new Color(0, 128, 0));
line.setLineStyle(Line.LINE_DOUBLE);
slide.addShape(line);

//TextBox
TextBox txt = new TextBox();
txt.setText("Hello, World!");
txt.setAnchor(new java.awt.Rectangle(300, 100, 300, 50));

//use RichTextRun to work with the text format
RichTextRun rt = txt.getTextRun().getRichTextRuns()[0];
rt.setFontSize(32);
rt.setFontName("Arial");
rt.setBold(true);
rt.setItalic(true);
rt.setUnderlined(true);
rt.setFontColor(Color.red);
rt.setAlignment(TextBox.AlignRight);

slide.addShape(txt);

//Autoshape
//32-point star
AutoShape sh1 = new AutoShape(ShapeTypes.Star32);
sh1.setAnchor(new java.awt.Rectangle(50, 50, 100, 200));
sh1.setFillColor(Color.red);
slide.addShape(sh1);

//Trapezoid
AutoShape sh2 = new AutoShape(ShapeTypes.Trapezoid);
sh2.setAnchor(new java.awt.Rectangle(150, 150, 100, 200));
sh2.setFillColor(Color.blue);
slide.addShape(sh2);

FileOutputStream out = new FileOutputStream("slideshow.ppt");
ppt.write(out);
out.close();
```

1.2.5. How to work with pictures

Currently, HSLF API supports the following types of pictures:

- Windows Metafiles (WMF)
- Enhanced Metafiles (EMF)
- JPEG Interchange Format
- Portable Network Graphics (PNG)
- Macintosh PICT

Busy Developers' Guide to HSLF drawing layer

```
SlideShow ppt = new SlideShow(new HSLFSlideShow("slideshow.ppt"));

//extract all pictures contained in the presentation
PictureData[] pdata = ppt.getPictureData();
for (int i = 0; i < pdata.length; i++){
    PictureData pict = pdata[i];

    // picture data
    byte[] data = pict.getData();

    int type = pict.getType();
    String ext;
    switch (type){
        case Picture.JPEG: ext=".jpg"; break;
        case Picture.PNG: ext=".png"; break;
        case Picture.WMF: ext=".wmf"; break;
        case Picture.EMF: ext=".emf"; break;
        case Picture.PICT: ext=".pict"; break;
        default: continue;
    }
    FileOutputStream out = new FileOutputStream("pict_"+i + ext);
    out.write(data);
    out.close();
}

// add a new picture to this slideshow and insert it in a new slide
int idx = ppt.addPicture(new File("clock.jpg"), Picture.JPEG);

Picture pict = new Picture(idx);

//set image position in the slide
pict.setAnchor(new java.awt.Rectangle(100, 100, 300, 200));

Slide slide = ppt.createSlide();
slide.addShape(pict);

//now retrieve pictures contained in the first slide and save them on disk
slide = ppt.getSlides()[0];
Shape[] sh = slide.getShapes();
for (int i = 0; i < sh.length; i++){
    if (sh[i] instanceof Picture){
        Picture pict = (Picture)sh[i];
        PictureData pictData = pict.getPictureData();
        byte[] data = pictData.getData();
        int type = pictData.getType();
        if (type == Picture.JPEG){
            FileOutputStream out = new FileOutputStream("slide0_"+i+".jpg");
            out.write(data);
            out.close();
        } else if (type == Picture.PNG){
            FileOutputStream out = new FileOutputStream("slide0_"+i+".png");
            out.write(data);
            out.close();
        }
    }
}
```

```
    }  
  }  
}  
  
FileOutputStream out = new FileOutputStream("slideshow.ppt");  
ppt.write(out);  
out.close();
```

1.2.6. How to set slide title

```
SlideShow ppt = new SlideShow();  
Slide slide = ppt.createSlide();  
TextBox title = slide.addTitle();  
title.setText("Hello, World!");  
  
//save changes  
FileOutputStream out = new FileOutputStream("slideshow.ppt");  
ppt.write(out);  
out.close();
```

Below is the equivalent code in PowerPoint VBA:

```
Set myDocument = ActivePresentation.Slides(1)  
myDocument.Shapes.AddTitle.TextFrame.TextRange.Text = "Hello, World!"
```

1.2.7. How to modify background of a slide master

```
SlideShow ppt = new SlideShow();  
SlideMaster master = ppt.getSlidesMasters()[0];  
  
Fill fill = master.getBackground().getFill();  
int idx = ppt.addPicture(new File("background.png"), Picture.PNG);  
fill.setFillType(Fill.FILL_PICTURE);  
fill.setPictureData(idx);
```

1.2.8. How to modify background of a slide

```
SlideShow ppt = new SlideShow();  
Slide slide = ppt.createSlide();  
  
//This slide has its own background.  
//Without this line it will use master's background.  
slide.setFollowMasterBackground(false);  
Fill fill = slide.getBackground().getFill();  
int idx = ppt.addPicture(new File("background.png"), Picture.PNG);  
fill.setFillType(Fill.FILL_PATTERN);
```

```
fill.setPictureData(idx);
```

1.2.9. How to modify background of a shape

```
SlideShow ppt = new SlideShow();
Slide slide = ppt.createSlide();

Shape shape = new AutoShape(ShapeTypes.Rectangle);
shape.setAnchor(new java.awt.Rectangle(100, 100, 200, 200));
Fill fill = shape.getFill();
fill.setFillType(Fill.FILL_SHADE);
fill.setBackgroundColor(Color.red);
fill.setForegroundColor(Color.green);

slide.addShape(shape);
```

1.2.10. How to create bulleted lists

```
SlideShow ppt = new SlideShow();
Slide slide = ppt.createSlide();

TextBox shape = new TextBox();
RichTextRun rt = shape.getTextRun().getRichTextRuns()[0];
shape.setText(
    "January\r" +
    "February\r" +
    "March\r" +
    "April");
rt.setFontSize(42);
rt.setBullet(true);
rt.setBulletOffset(0); //bullet offset
rt.setTextOffset(50); //text offset (should be greater than bullet offset)
rt.setBulletChar('\u263A'); //bullet character
slide.addShape(shape);

shape.setAnchor(new java.awt.Rectangle(50, 50, 500, 300)); //position of the text box
slide.addShape(shape);

FileOutputStream out = new FileOutputStream("bullets.ppt");
ppt.write(out);
out.close();
```

1.2.11. How to read hyperlinks from a slide show

```
FileInputStream is = new FileInputStream("slideshow.ppt");
SlideShow ppt = new SlideShow(is);
is.close();
```

```
Slide[] slide = ppt.getSlides();
for (int j = 0; j < slide.length; j++) {

    //read hyperlinks from the text runs
    TextRun[] txt = slide[j].getTextRuns();
    for (int k = 0; k < txt.length; k++) {
        String text = txt[k].getText();
        Hyperlink[] links = txt[k].getHyperlinks();
        if(links != null) for (int l = 0; l < links.length; l++) {
            Hyperlink link = links[l];
            String title = link.getTitle();
            String address = link.getAddress();
            String substring = text.substring(link.getStartIndex(), link.getEndIndex());
        }
    }

    //in PowerPoint you can assign a hyperlink to a shape without text,
    //for example to a Line object. The code below demonstrates how to
    //read such hyperlinks
    Shape[] sh = slide[j].getShapes();
    for (int k = 0; k < sh.length; k++) {
        Hyperlink link = sh[k].getHyperlink();
        if(link != null) {
            String title = link.getTitle();
            String address = link.getAddress();
        }
    }
}
```