

# 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](#)
- [Tables](#)
- [How to remove shapes](#)
- [How to retrieve embedded OLE objects](#)
- [How to retrieve embedded sounds](#)
- [How to create shapes of arbitrary geometry](#)
- [Shapes and Graphics2D](#)
- [How to convert slides into images](#)
- [Headers / Footers](#)
- [How to embed movies](#)

### 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 pyg = 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 following code 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.setFill(Color.red);
slide.addShape(sh1);

//Trapezoid
AutoShape sh2 = new AutoShape(ShapeTypes.Trapezoid);
sh2.setAnchor(new java.awt.Rectangle(150, 150, 100, 200));
sh2.setFill(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

```
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();
    }
}
}
```

### 1.2.12. How to create tables

```
//table data
String[][] data = {
    {"INPUT FILE", "NUMBER OF RECORDS"},
    {"Item File", "11,559"},
    {"Vendor File", "300"},
    {"Purchase History File", "10,000"},
    {"Total # of requisitions", "10,200,038"}
};

SlideShow ppt = new SlideShow();

Slide slide = ppt.createSlide();
//create a table of 5 rows and 2 columns
Table table = new Table(5, 2);
for (int i = 0; i < data.length; i++) {
    for (int j = 0; j < data[i].length; j++) {
        TableCell cell = table.getCell(i, j);
        cell.setText(data[i][j]);

        RichTextRun rt = cell.getTextRun().getRichTextRuns()[0];
        rt.setFontName("Arial");
        rt.setFontSize(10);

        cell.setVerticalAlignment(TextBox.AnchorMiddle);
        cell.setHorizontalAlignment(TextBox.AlignCenter);
    }
}

//set table borders
Line border = table.createBorder();
border.setLineColor(Color.black);
border.setLineWidth(1.0);
table.setAllBorders(border);

//set width of the 1st column
table.setColumnWidth(0, 300);
//set width of the 2nd column
table.setColumnWidth(1, 150);

slide.addShape(table);
table.moveTo(100, 100);

FileOutputStream out = new FileOutputStream("hslf-table.ppt");
ppt.write(out);
out.close();
```



### **1.2.13. How to remove shapes from a slide**

```
Shape[] shape = slide.getShapes();
for (int i = 0; i < shape.length; i++) {

    //remove the shape
    boolean ok = slide.removeShape(shape[i]);
    if(ok){
        //the shape was removed. Do something.
    }
}
```

### **1.2.14. How to retrieve embedded OLE objects**

```
Shape[] shape = slide.getShapes();
for (int i = 0; i < shape.length; i++) {
    if (shape[i] instanceof OLEShape) {
        OLEShape ole = (OLEShape) shape[i];
        ObjectData data = ole.getObjectData();
        String name = ole.getInstanceName();
        if ("Worksheet".equals(name)) {
            HSSFWorkbook wb = new HSSFWorkbook(data.getData());
        } else if ("Document".equals(name)) {
            HWPFDocument doc = new HWPFDocument(data.getData());
        }
    }
}
```

### **1.2.15. How to retrieve embedded sounds**

```
FileInputStream is = new FileInputStream(args[0]);
SlideShow ppt = new SlideShow(is);
is.close();

SoundData[] sound = ppt.getSoundData();
for (int i = 0; i < sound.length; i++) {
    //save *WAV sounds on disk
    if(sound[i].getSoundType().equals(".WAV")){
        FileOutputStream out = new FileOutputStream(sound[i].getSoundName());
        out.write(sound[i].getData());
        out.close();
    }
}
```

### 1.2.16. How to create shapes of arbitrary geometry

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

java.awt.geom.GeneralPath path = new java.awt.geom.GeneralPath();
path.moveTo(100, 100);
path.lineTo(200, 100);
path.curveTo(50, 45, 134, 22, 78, 133);
path.curveTo(10, 45, 134, 56, 78, 100);
path.lineTo(100, 200);
path.closePath();

Freeform shape = new Freeform();
shape.setPath(path);
slide.addShape(shape);
```

### 1.2.17. How to draw into a slide using Graphics2D

**Note:**

Current implementation of the PowerPoint Graphics2D driver is not fully compliant with the java.awt.Graphics2D specification. Some features like clipping, drawing of images are not yet supported.

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

//draw a simple bar graph
//bar chart data. The first value is the bar color, the second is the width
Object[] def = new Object[]{
    Color.yellow, new Integer(100),
    Color.green, new Integer(150),
    Color.gray, new Integer(75),
    Color.red, new Integer(200),
};

//all objects are drawn into a shape group so we need to create one

ShapeGroup group = new ShapeGroup();
//define position of the drawing in the slide
Rectangle bounds = new java.awt.Rectangle(200, 100, 350, 300);
//if you want to draw in the entire slide area then define the anchor as follow
//Dimension pgsz = ppt.getPageSize();
//java.awt.Rectangle bounds = new java.awt.Rectangle(0, 0, pgsz.width, pgsz.height);

group.setAnchor(bounds);
slide.addShape(group);
```

```
//draw a simple bar chart
Graphics2D graphics = new PPGraphics2D(group);
int x = bounds.x + 50, y = bounds.y + 50;
graphics.setFont(new Font("Arial", Font.BOLD, 10));
for (int i = 0, idx = 1; i < def.length; i+=2, idx++) {
    graphics.setColor(Color.black);
    int width = ((Integer)def[i+1]).intValue();
    graphics.drawString("Q" + idx, x-20, y+20);
    graphics.drawString(width + "%", x + width + 10, y + 20);
    graphics.setColor((Color)def[i]);
    graphics.fill(new Rectangle(x, y, width, 30));
    y += 40;
}
graphics.setColor(Color.black);
graphics.setFont(new Font("Arial", Font.BOLD, 14));
graphics.draw(bounds);
graphics.drawString("Performance", x + 70, y + 40);

FileOutputStream out = new FileOutputStream("hslf-graphics2d.ppt");
ppt.write(out);
out.close();
```

### 1.2.18. Export PowerPoint slides into java.awt.Graphics2D

HSLF provides a way to export slides into images. You can capture slides into java.awt.Graphics2D object (or any other) and serialize it into a PNG or JPEG format. Please note, although HSLF attempts to render slides as close to PowerPoint as possible, the output may look differently from PowerPoint due to the following reasons:

- Java2D renders fonts differently vs PowerPoint. There are always some differences in the way the font glyphs are painted
- HSLF uses java.awt.font.LineBreakMeasurer to break text into lines. PowerPoint may do it in a different way.
- If a font from the presentation is not available, then the JDK default font will be used.

Current Limitations:

- Some types of shapes are not yet supported (WordArt, complex auto-shapes)
- Only Bitmap images (PNG, JPEG, DIB) can be rendered in Java

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

Dimension pgsz = ppt.getPageSize();

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

```
        BufferedImage img = new BufferedImage(pgsz.width, pgsh.height, BufferedImage.TYPE_INT_RGB);
        Graphics2D graphics = img.createGraphics();
        //clear the drawing area
        graphics.setPaint(Color.white);
        graphics.fill(new Rectangle2D.Float(0, 0, pgsw, pgh));

        //render
        slide[i].draw(graphics);

        //save the output
        FileOutputStream out = new FileOutputStream("slide-" + (i+1) + ".png");
        javax.imageio.ImageIO.write(img, "png", out);
        out.close();
    }
}
```

### **1.2.19. How to extract Headers / Footers from an existing presentation**

```
FileInputStream is = new FileInputStream("slideshow.ppt");
SlideShow ppt = new SlideShow(is);
is.close();
Slide[] slides = ppt.getSlides();

//presentation-scope headers / footers
HeadersFooters hdd = ppt.getSlideHeadersFooters();
if(hdd.isFooterVisible()) {
    String footerText = hdd.getFooterText();
}

//per-slide headers / footers
for (int i=0; i < slides.length; i++){
    HeadersFooters hdd2 = slides[i].getHeadersFooters();
    if(hdd2.isFooterVisible()) {
        String footerText = hdd2.getFooterText();
    }
    if(hdd2.isUserDateVisible()) {
        String customDate = hdd2.getDateTimeText();
    }
    if(hdd2.isSlideNumberVisible()){
        int slideNum = slides[i].getSlideNumber();
    }
}
}
```

### **1.2.20. How to set Headers / Footers**

```
SlideShow ppt = new SlideShow();
```

```
//presentation-scope headers / footers
HeadersFooters hdd = ppt.getSlideHeadersFooters();
hdd.setSlideNumberVisible(true);
hdd.setFootersText("Created by POI-HSLF");
```

### **1.2.21. How to embed movies**

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

//UNC or local Returns UNC or local path to a video file
String moviePath = "card.mpg";
int movieIdx = ppt.addMovie(moviePath, MovieShape.MOVIE_MPEG);

String thumbnailPath = "card.png";
int thumbnailIdx = ppt.addPicture(new File(thumbnailPath), Picture.PNG);
MovieShape shape = new MovieShape(movieIdx, thumbnailIdx);
shape.setAnchor(new Rectangle2D.Float(300,225,120,90));
slide.addShape(shape);

FileOutputStream out = new FileOutputStream("hslf-movie.ppt");
ppt.write(out);
out.close();
```