# XSLF Cookbook

**by Yegor Kozlov**

## 1. XSLF Cookbook

This page offers a short introduction into the XSLF API. More examples can be found in the XSLF Examples in the POI SVN repository.

> **Note:**
> Please note that XSLF is still in early development and is a subject to incompatible changes in a future release.

## 1.1. Index of Features

- Create a new presentation
- Read an existing presentation
- Create a slide with a predefined layout
- Delete slide
- Re-order slides
- Change slide size
- Read shapes
- Add image
- Read images contained in a presentation
- Format text
- Hyperlinks
- Convert .pptx slides into images
- Merge multiple presentations together

## 1.2. Cookbok

### 1.2.1. New Presentation

The following code creates a new .pptx slide show and adds a blank slide to it:

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

```
//add first slide
XSLFSlide blankSlide = ppt.createSlide();
```

### 1.2.2. Read an existing presentation and append a slide to it

```
XMLSlideShow ppt = new XMLSlideShow(new FileInputStream("slideshow.pptx"));

//append a new slide to the end
XSLFSlide blankSlide = ppt.createSlide();
```

### 1.2.3. Create a new slide from a predefined slide layout

```
XMLSlideShow ppt = new XMLSlideShow(new FileInputStream("slideshow.pptx"));

// first see what slide layouts are available :
System.out.println("Available slide layouts:");
for(XSLFSlideMaster master : ppt.getSlideMasters()){
    for(XSLFSlideLayout layout : master.getSlideLayouts()){
        System.out.println(layout.getType());
    }
}

// blank slide
XSLFSlide blankSlide = ppt.createSlide();

// there can be multiple masters each referencing a number of layouts
// for demonstration purposes we use the first (default) slide master
XSLFSlideMaster defaultMaster = ppt.getSlideMasters()[0];

// title slide
XSLFSlideLayout titleLayout = defaultMaster.getLayout(SlideLayout.TITLE);
// fill the placeholders
XSLFSlide slide1 = ppt.createSlide(titleLayout);
XSLFTextShape title1 = slide1.getPlaceholder(0);
title1.setText("First Title");

// title and content
XSLFSlideLayout titleBodyLayout = defaultMaster.getLayout(SlideLayout.TITLE_AND_CON
XSLFSlide slide2 = ppt.createSlide(titleBodyLayout);

XSLFTextShape title2 = slide2.getPlaceholder(0);
title2.setText("Second Title");

XSLFTextShape body2 = slide2.getPlaceholder(1);
body2.clearText(); // unset any existing text
body2.addNewTextParagraph().addNewTextRun().setText("First paragraph");
body2.addNewTextParagraph().addNewTextRun().setText("Second paragraph");
body2.addNewTextParagraph().addNewTextRun().setText("Third paragraph");
```

### 1.2.4. Delete slide

```
XMLSlideShow ppt = new XMLSlideShow(new FileInputStream("slideshow.pptx"));

ppt.removeSlide(0); // 0-based index of a slide to be removed
```

### 1.2.5. Re-order slides

```
XMLSlideShow ppt = new XMLSlideShow(new FileInputStream("slideshow.pptx"));
XSLFSlide[] slides = ppt.getSlides();

XSLFSlide thirdSlide = slides[2];
ppt.setSlideOrder(thirdSlide, 0); // move the third slide to the beginning
```

### 1.2.6. How to retrieve or change slide size

```
XMLSlideShow ppt = new XMLSlideShow();
//retrieve page size. Coordinates are expressed in points (72 dpi)
java.awt.Dimension pgsize = ppt.getPageSize();
int pgx = pgsize.width; //slide width in points
int pgy = pgsize.height; //slide height in points

//set new page size
ppt.setPageSize(new java.awt.Dimension(1024, 768));
```

### 1.2.7. How to read shapes contained in a particular slide

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

```
XMLSlideShow ppt = new XMLSlideShow(new FileInputStream("slideshow.pptx"));
//get slides
XSLFSlide[] slide = ppt.getSlides();
for (int i = 0; i < slide.length; i++){
  XSLFShape[] 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.geom.Rectangle2D anchor = sh[j].getAnchor();

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

```
        //work with a shape that can hold text
    } else if (sh[j] instanceof XSLFPictureShape){
      XSLFPictureShape shape = (XSLFPictureShape)sh[j];
      //work with Picture
    }
  }
}
```

### 1.2.8. Add Image to Slide

```
    XMLSlideShow ppt = new XMLSlideShow();
    XSLFSlide slide = ppt.createSlide();

    byte[] pictureData = IOUtils.toByteArray(new FileInputStream("image.png"));

    int idx = ppt.addPicture(pictureData, XSLFPictureData.PICTURE_TYPE_PNG);
    XSLFPictureShape pic = slide.createPicture(idx);
```

### 1.2.9. Read Images contained within a presentation

```
    XMLSlideShow ppt = new XMLSlideShow(new FileInputStream("slideshow.pptx"));
    for(XSLFPictureData data : ppt.getAllPictures()){
        byte[] bytes = data.getData();
        String fileName = data.getFileName();

    }
```

### 1.2.10. Basic text formatting

```
  XMLSlideShow ppt = new XMLSlideShow();
  XSLFSlide slide = ppt.createSlide();

  XSLFTextBox shape = slide.createTextBox();
  XSLFTextParagraph p = shape.addNewTextParagraph();

  XSLFTextRun r1 = p.addNewTextRun();
  r1.setText("The");
  r1.setFontColor(Color.blue);
  r1.setFontSize(24);

  XSLFTextRun r2 = p.addNewTextRun();
  r2.setText(" quick");
  r2.setFontColor(Color.red);
  r2.setBold(true);

  XSLFTextRun r3 = p.addNewTextRun();
  r3.setText(" brown");
  r3.setFontSize(12);
```

```
    r3.setItalic(true);
    r3.setStrikethrough(true);

    XSLFTextRun r4 = p.addNewTextRun();
    r4.setText(" fox");
    r4.setUnderline(true);
```

### 1.2.11. How to read hyperlinks from a slide show

```
    XMLSlideShow ppt = new XMLSlideShow();
    XSLFSlide slide = ppt.createSlide();

    // assign a hyperlink to a text run
    XSLFTextBox shape = slide.createTextBox();
    XSLFTextRun r = shape.addNewTextParagraph().addNewTextRun();
    r.setText("Apache POI");
    XSLFHyperlink link = r.createHyperlink();
    link.setAddress("http://poi.apache.org");
```

### 1.2.12. PPTX2PNG is an application that converts each slide of a .pptx slideshow into a PNG image

```
Usage: PPTX2PNG [options] <pptx file>
Options:
    -scale <float>    scale factor (default is 1.0)
    -slide <integer> 1-based index of a slide to render. Default is to render all slide
```

How it works:

The XSLFSlide object implements a draw(Graphics2D graphics) method that recursively paints all shapes in the slide into the supplied graphics canvas:

```
            slide.draw(graphics);
```

where graphics is a class implementing java.awt.Graphics2D. In PPTX2PNG the graphic canvas is derived from java.awt.image.BufferedImage, i.e. the destination is an image in memory, but in general case you can pass any compliant implementation of java.awt.Graphics2D. The PPTX2SVG example demonstrates how to use Apache Batik to convert .pptx slides into SVG format.

### 1.2.13. Merge multiple presentations together

```
    XMLSlideShow ppt = new XMLSlideShow();
    String[] inputs = {"presentations1.pptx", "presentation2.pptx"};
    for(String arg : inputs){
```

```
        FileInputStream is = new FileInputStream(arg);
        XMLSlideShow src = new XMLSlideShow(is);
        is.close();

        for(XSLFSlide srcSlide : src.getSlides()){
            ppt.createSlide().importContent(srcSlide);
        }
    }

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