

In this chapter, you will learn how to add an image to a PPT and how to read an image from it.

Adding Image

You can add images to a presentation using the **createPicture** method of **XSLFSlide**. This method accepts image in the form of byte array format. Therefore, you have to create a byte array of the image that is to be added to the presentation.

Follow the given procedure to add an image to a presentation. Create an empty slideshow using **XMLSlideShow** as shown below:

```
XMLSlideShow ppt = new XMLSlideShow();
```

Create an empty presentation in it using **createSlide**.

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

Read the image file that is to be added and convert it into byte array using **IOUtils.toByteArray** of the **IOUtils** class as shown below:

```
//reading an image
File image=new File("C://POIPPT//boy.jpg");

//converting it into a byte array
byte[] picture = IOUtils.toByteArray(new FileInputStream(image));
```

Add the image to the presentation using **addPicture**. This method accepts two variables: byte array format of the image that is to be added and the static variable representing the file format of the image. The usage of the **addPicture** method is shown below:

```
int idx = ppt.addPicture(picture, XSLFPictureData.PICTURE_TYPE_PNG);
```

Embed the image to the slide using **createPicture** as shown below:

```
XSLFPictureShape pic = slide.createPicture(idx);
```

Given below is the complete program to add an image to the slide in a presentation:

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

import org.apache.poi.util.IOUtils;
import org.apache.poi.xslf.usermodel.XMLSlideShow;
import org.apache.poi.xslf.usermodel.XSLFPictureData;
import org.apache.poi.xslf.usermodel.XSLFPictureShape;
import org.apache.poi.xslf.usermodel.XSLFSlide;

public class AddingImage {

    public static void main(String args[]) throws IOException{

        //creating a presentation
        XMLSlideShow ppt = new XMLSlideShow();

        //creating a slide in it
        XSLFSlide slide = ppt.createSlide();
```

```

//reading an image
File image=new File("C://POIPTT//boy.jpg");

//converting it into a byte array
byte[] picture = IOUtils.toByteArray(new FileInputStream(image));

//adding the image to the presentation
int idx = ppt.addPicture(picture, XSLFPictureData.PICTURE_TYPE_PNG);

//creating a slide with given picture on it
XSLFPictureShape pic = slide.createPicture(idx);

//creating a file object
File file=new File("addingimage.pptx");
FileOutputStream out = new FileOutputStream(file);

//saving the changes to a file
ppt.write(out)
System.out.println("image added successfully");
out.close();
}
}

```

Save the above Java code as **AddingImage.java**, and then compile and execute it from the command prompt as follows:

```

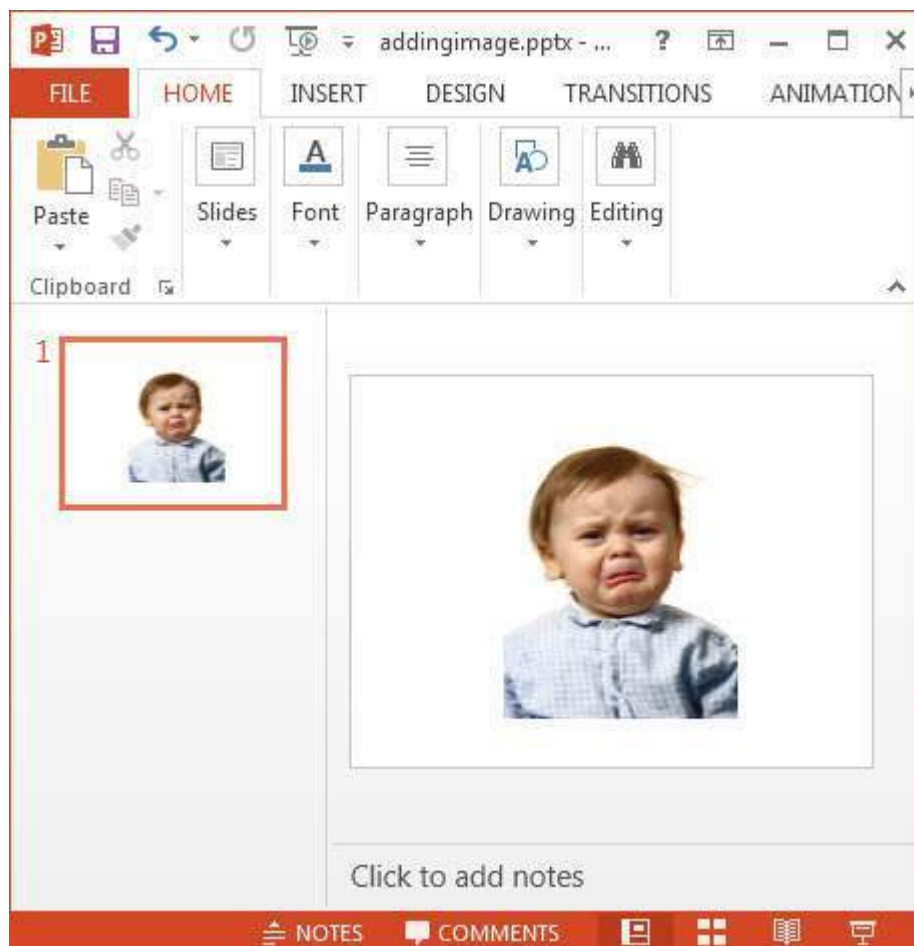
$javac AddingImage.java
$java AddingImage

```

It will compile and execute to generate the following output:

```
reordering of the slides is done
```

The presentation with the newly added slide with image appears as follows:



Reading Image

You can get the data of all the pictures using the **getAllPictures** method of the **XMLSlideShow** class. The following program reads the images from a presentation:

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

import org.apache.poi.xslf.usermodel.XMLSlideShow;
import org.apache.poi.xslf.usermodel.XSLFPictureData;

public class Readingimage {

    public static void main(String args[]) throws IOException{

        //open an existing presentation
        File file = new File("addingimage.pptx");
        XMLSlideShow ppt = new XMLSlideShow(new FileInputStream(file));

        //reading all the pictures in the presentation
        for(XSLFPictureData data : ppt.getAllPictures()){

            byte[] bytes = data.getData();
            String fileName = data.getFileName();
            int pictureFormat = data.getPictureType();
            System.out.println("picture name: " + fileName);
            System.out.println("picture format: " + pictureFormat);
        }

        //saving the changes to a file
        FileOutputStream out = new FileOutputStream(file);
        ppt.write(out);
        out.close();
    }
}
```

Save the above Java code as **Readingimage.java**, and then compile and execute it from the command prompt as follows:

```
$javac Readingimage.java
$java Readingimage
```

It will compile and execute to generate the following output:

```
picture name: image1.png
picture format: 6
```

```
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```