

```

1 // File Index_To_Rgb.java
2
3 import ij.ImagePlus;
4 import ij.plugin.filter.PlugInFilter;
5 import ij.process.ColorProcessor;
6 import ij.process.ImageProcessor;
7 import java.awt.image.IndexColorModel;
8
9 public class Index_To_Rgb implements PlugInFilter {
10     static final int R = 0, G = 1, B = 2;
11
12     public int setup(String arg, ImagePlus im) {
13         return DOES_8C + NO_CHANGES; //does not alter original image
14     }
15
16     public void run(ImageProcessor ip) {
17         int w = ip.getWidth();
18         int h = ip.getHeight();
19
20         //retrieve the color table (palette) for R,G,B
21         IndexColorModel icm =
22             (IndexColorModel) ip.getColorModel();
23         int mapSize = icm.getMapSize();
24         byte[] Rmap = new byte[mapSize]; icm.getReds(Rmap);
25         byte[] Gmap = new byte[mapSize]; icm.getGreens(Gmap);
26         byte[] Bmap = new byte[mapSize]; icm.getBlues(Bmap);
27
28         //create new 24-bit RGB image
29         ColorProcessor cp = new ColorProcessor(w,h);
30         int[] RGB = new int[3];
31         for (int v = 0; v < h; v++) {
32             for (int u = 0; u < w; u++) {
33                 int idx = ip.getPixel(u, v);
34                 RGB[R] = 0xFF & Rmap[idx]; // treat maps as
35                 RGB[G] = 0xFF & Gmap[idx]; // UNSIGNED byte!
36                 RGB[B] = 0xFF & Bmap[idx];
37                 cp.putPixel(u, v, RGB); // putPixel() instead of set()
38             }
39         }
40         ImagePlus cimg = new ImagePlus("RGB Image",cp);
41         cimg.show();
42     }
43
44 } // end of class Index_To_Rgb

```

Program 8.4 Converting an indexed image to a true color RGB image (ImageJ plugin).

```

1 int tIdx = 2; // index of transparent color
2 IndexColorModel icm2 = new
3     IndexColorModel(pixBits, mapSize, Rmap, Gmap, Bmap, tIdx);
4 ip.setColorModel(icm2);

```