

DEVELOPER'S GUIDE TO ASPRISE TIFF WRITER/READER LIBRARY

Multi-platforms

Last updated on September 16, 2005

ALL RIGHTS RESERVED BY LAB ASPRISE! © 1998, 2005.

Table of Contents

1	INTRODUCTION	3
2	TIFF WRITER	4
2.1	METHOD: TIFFWRITER.CREATETIFFFROMIMAGES.....	4
2.2	METHOD: TIFFWRITER.CREATETIFFFROMIMAGES.....	4
2.3	TIFFWRITER.MAIN	5
3	TIFF READER	6
4	LICENSE SCHEMES	7

1 Introduction

Asprise offers TIFF writer and reader library as valued add-on to our flagship products – Asprise OCR & JTwain. With this library, you can create TIFF files from separate images easily and vice versa.

The evaluation kit contains the following files:

File/Dir	Remarks
javadoc	Contains the Javadoc API for Asprise TIFF Writer/Reader classes
aspriseTIFF.jar	Contains TIFF Writer/Reader related classes;
helpScanned.gif	Sample image 1
Scanned-text-200dpi.jpg	Sample image 2
demo-write.bat/sh	Demo program. Double click to run it – this demo creates output.tiff from the two sample images.
demo-read.bat/sh	Demo program. Double click to run it – this demo extract the pages from output.tiff (generated by demo-write) into the current directory – 0.jpg & 1.jpg.

Trying them will give you some feelings about the TIFF writer and reader.

2 TIFF Writer

Main class: `com.asprise.util.tiff.TIFFWriter`

2.1 Method:

TIFFWriter.createTIFFFromImages

```
public static void createTIFFFromImages(BufferedImage[] images,  
java.io.File file) throws java.io.IOException
```

This is the simplest method to create a TIFF file from multiple images. The images must be supplied as *BufferedImage*. If you got only *java.awt.Image*, you can use the utility method: *TIFFWriter.getBufferedImageFromImage(Image img)* to convert it to *BufferedImage*. If your JVM does not support *ImageIO*, you can use the *com.asprise.util.tiff.ImageReader* to load various formats (gif, JPG, bmp) into *BufferedImage*. If you need more controls on TIFF creation, see next function.

2.2 Method:

TIFFWriter.createTIFFFromImages

```
public static void createTIFFFromImages(BufferedImage[] images, int  
conversion, int compression, java.io.File file) throws IOException
```

This method creates a TIFF file from the specified images. Before the images are put into the TIFF, you can optionally convert them to gray (*TIFF_CONVERSION_TO_GRAY*) or black-white (*TIFF_CONVERSION_TO_BLACK_WHITE*) to reduce the file size. Also, you can set the compression algorithm for TIFF.

Parameters:

- ◆ **images** -- list of images to be put into the TIFF file.
- ◆ **conversion** -- converts the images into the specified format before putting into the TIFF.
Valid values:
 - TIFF_CONVERSION_NONE - Do not convert images.
 - TIFF_CONVERSION_TO_BLACK_WHITE - Converts images into black white.
 - TIFF_CONVERSION_TO_GRAY - Converts images into gray.
 - All the constants are defined in the same class: *TIFFWriter*
- ◆ **compression** -- compression algorithm used for TIFF.
Valid values:
 - TIFF_COMPRESSION_NONE - Indicates no compression.
 - TIFF_COMPRESSION_GROUP3_1D - CCITT run length encoding.
 - TIFF_COMPRESSION_GROUP3_2D - CCITT T.4 compression for bilevel only.
 - TIFF_COMPRESSION_GROUP4 - CCITT T.6 compression for bilevel only.
 - TIFF_COMPRESSION_DEFLATE - Zip in TIFF compression (lossless).
- ◆ **file** -- the TIFF file to be written to.

In rare cases, you might encounter color inversion problem – black background & white text. If this is undesired, try to set *TIFFWriter.reverseColorDuringConversion* to true.

2.3 TIFFWriter.main

TIFFWriter.main is a simply console utility tool to create TIFF file from image files.

```
Usage: java com.asprise.util.tiff.TIFFWriter [tiff file to be created]
[image conversion option, set -1 to use default] [TIFF compression option,
set -1 to use default] [path of image1] [path of image2] ...
```

3 TIFF Reader

Main class: **com.asprise.util.tiff.TIFFReader**

The following code demos how to use TIFF reader:

```
1. TIFFReader reader = new TIFFReader("C:\\my.tiff"); // loads the TIFF
   file.
2.
3. for(int i=0; i<reader.countPages(); i++) {
4.     System.out.println("Page #" + i);
5.     RenderedImage image = reader.getPage(i); // extract page
6.     ImageIO.write(image, "jpg", new File("page" + I + ".jpg"));
7. }
```

The above code reads a TIFF file, extracts each page and saves each into a JPG file.

Methods used:

public TIFFReader(java.io.File tiffFile) throws java.io.IOException

Creates a TIFF reader on the specified file.

public int countPages()

Returns the total number pages in the TIFF file.

public RenderedImage getPage(int index) throws IOException

Returns an image that contains the decoded contents of the specified page of the TIFF.

0 refer to the first page, 1 to the second, and so on.

4 License Schemes

License	Terms	Price
Writer Only	Royalty-free distribution	USD 268/developer
Reader Only	Royalty-free distribution	USD 268/developer
Writer & Reader	Royalty-free distribution	USD 498/developer
	Royalty-free distribution + Site License	USD 998/site
	Royalty-free distribution + Site License + Full Source Code	USD 1498/site

Visit <http://asprise.com/product/javatiff> for order instructions.