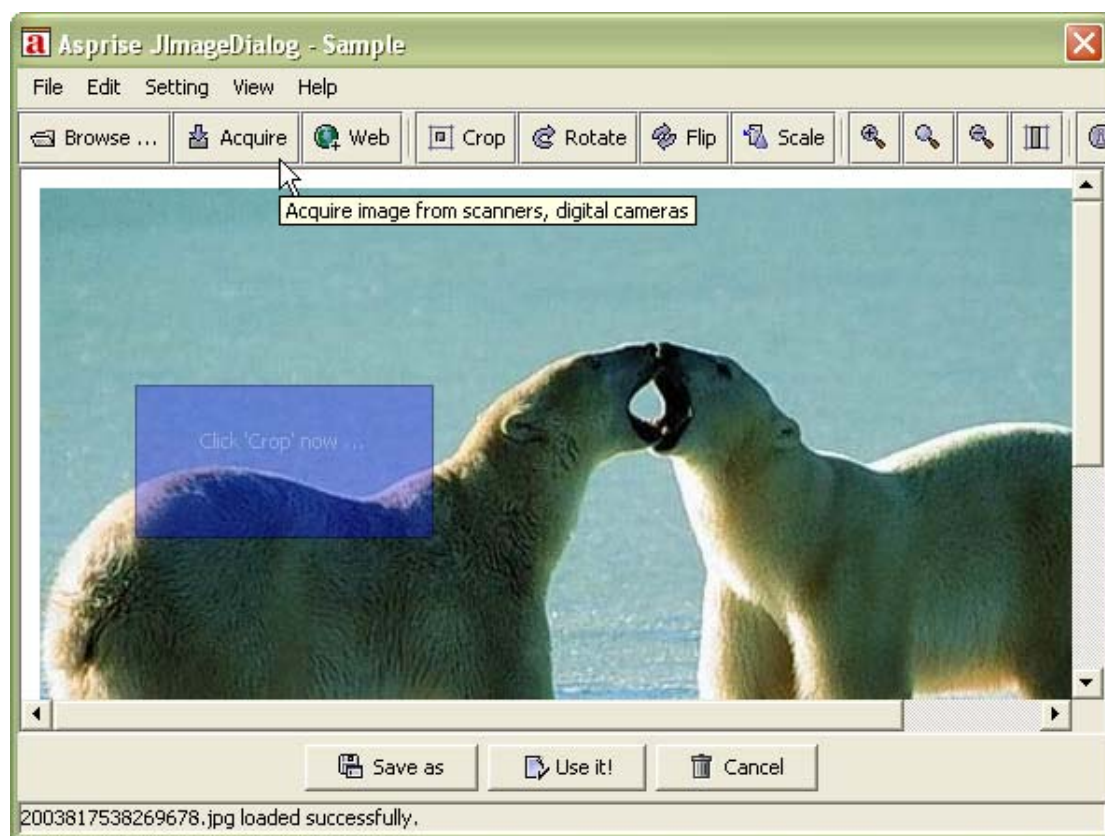


7 Image Acquisition Components

The image acquisition UI components are not part of Asprise OCR library. However, based on our customers' experience, if you need to build a front-end for OCR, they are invaluable and could save you a lot of time. Otherwise, you may skip this chapter.

7.1 JImageDialog

JImageDialog is an image acquisition UI component that allows the user to load images and to perform basic image editing tasks. If you are developing some applications that require the user to select/edit/input images, then JImageDialog will make your life extremely easy – and more importantly, the user experience will be improved dramatically.



Let say you want to build an album application, the user is required to supply photos

(i.e. images). You put a button on your panel. When the user click the button, JImageDialog is brought up – now the user can select existing pictures files from his or her computer or acquire images from digital cameras or scanners. And the user can edit images before putting it into the album.

7.1.1 Advantages

- ◆ Multiple image sources supported: local computer, digital cameras, scanners and the web;
- ◆ Multiple image formats: read and write BMP, PNG, JPG, GIF, PCT, PSD and many other formats;
- ◆ Platform/Virtual machine independent: Any platform, any Java virtual machine (version 1.3 or above);
- ◆ Powerful features: rotation, flipping, scaling, clipping, etc.
- ◆ User friendly as well as developer friendly

The user can load images from local computer or the web, he or she can also acquire images from digital cameras and scanners. After the image has been loaded, the user can rotate, clip, flip, and scale the image. The image has been loaded and edited, the user can save the image or select the image - which will be used in your applications.

7.1.2 Sample Uses

7.1.2.1 Modal (synchronous) mode

```
1. JImageDialog dialog = new JImageDialog(frame, "Sample", true); //  
   Modal dialog  
2. BufferedImage image = dialog.showDialog();  
3. ...
```

Line 1 constructs the image dialog.

Line 2 brings up the image dialog and waiting for user's selection/acquisition.

Besides using JImageDialog in synchronous mode, you can also use it in:

7.1.2.2 Asynchronous mode

```
4. public class JImageDialogSample extends JPanel implements
    JImageDialogListener {
5.     ...
6.     BufferedImage image;
7.
8.     // Displays selected image if any.
9.     public void paintComponent(Graphics g) {
10.         super.paintComponent(g); // Paint background.
11.         if(image != null)
12.             g.drawImage(image, 0, 0, null);
13.     }
14.
15. // Sets image and refreshes the panel.
16.     public void setImage(BufferedImage image) {
17.         this.image = image;
18.         setPreferredSize(getPreferredSize());
19.         revalidate();
20.         repaint();
21.     }
22.
23. // Methods in JImageDialogListener
24. // When the user presses cancel button, this method will be called.
25. public void onCancel() {
26.     setImage(null);
27. }
28.
29. // When the user presses the selection button, will be invoked.
30. public void onImageSet(BufferedImage image) {
31.     setImage(image);
32. }
33. }
34.
35. ...
36. JImageDialogSample imagePanel = new JImageDialogSample();
37.
38. JImageDialog dialog = new JImageDialog();
39. dialog.addImageDialogListener(imagePanel);
40. dialog.showDialog();
```

Line 1-30 implements a *JImageDialogListener*.

Line 33 constructs the listener.

Line 35 constructs the dialog.
 Line 36 registers the listener the dialog
 Line 37 brings up the dialog

When the user acquires an image and selects it, JimageDialog's listeners will be notified. In this case, *imagePanel.onImageSet(BufferedImage image)* will be called and thus the panel will display the selected image. If the user cancels the selection, *onCancel()* will be called instead.

Sample application: *com.asprise.util.ui.JImageDialogSample*

7.1.3 Supported Image Formats

The following table shows image formats supported by JImageDialog:

Formats	File extensions	READ	WRITE
Adobe Photoshop	*.psd	Y	Y
Bitmap, Windows/OS2	*.bmp, *.dib	Y	Y
Cursor	*.cur	Y	
Graphics Interchange Format	*.gif	Y	
Icon	*.ico	Y	
JPEG	*.jpg, *.jpeg	Y	Y
Macintosh PICT Format	*.pict, *.pct	Y	Y
PCX Format	*.pcx	Y	Y
Portable Network Graphics	*.png	Y	Y
Sun Raster Format	*.ras	Y	
Tag Image File Format	*.tif, *.tiff	Y	
Targa	*.tga	Y	Y
X Bitmap	*.xbm	Y	Y
X PixMap	*.xpm	Y	Y

On any Java platforms (version 1.3 or above), JImageDialog supports the above formats (using its own library to read/write image files). JImageDialog intelligently selects the best way to read or write files – e.g. on Java 1.4, it may invoke *ImageIO* to see whether a file can be read or written; if the *ImageIO* can do the job then JImageDialog will let it do; otherwise, JImageDialog will use its own library to access the file.

Note: You can only read/write image files from the JImageDialog UI component with unlicensed image acquisition UI component package. If you want to access image files

from your Java code and/or to perform other advanced operations, you need to obtain an affordable license from LAB Asprise!.

7.1.4 Compatibility

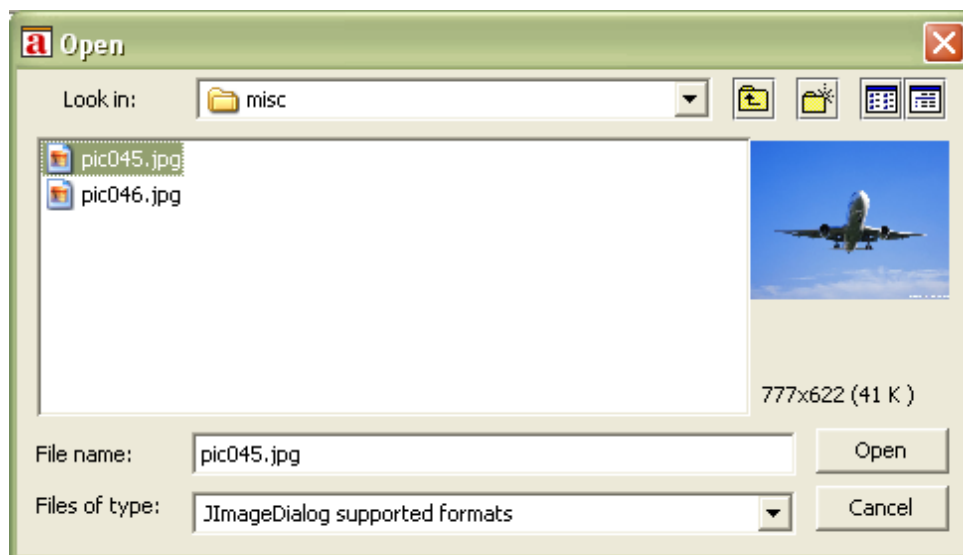
All Java runtimes with version 1.3 or above.

7.1.5 Software Packaging and Distribution

Mandatory: `jid.jar`, `JTwain.jar`

7.2 JImageFileChooser

JImageFileChooser is an extended *JFileChooser* that supports image preview and image information extraction.



When the user clicks an image file, its preview and associated information will be displayed to assist the user to select the proper image.

7.2.1 Sample Use

```
41. JFileChooser fc = new JImageFileChooser(lastDirectory);
42. fc.addChoosableFileFilter(JImageFileChooser.getImageFileFilter()
    );
43. int returnVal = fc.showOpenDialog(frame);
44. ...
```

Line 1 creates the image file chooser;

Line 2 set the file filter.

You can use it as normal JFileChooser, and it improves the user experience greatly.

7.2.2 Supported Image Formats

Please refer to Supported Image Formats in JImageDialog section.

Note: You can only preview image files from the JImageFileChooser UI component with unlicensed image acquisition UI component package. If you want to read/write image files from your Java code with the package and/or to perform other advanced operations, you need to obtain an affordable license from LAB Asprise!.

7.2.3 Compatibility

All operating systems;

All Java runtimes with version 1.2 or above.

7.2.4 Software Packaging and Distribution

Mandatory: **jid.jar**

8 Support and professional services

8.1 Support Web Site

<http://www.asprise.com/product/jid>

8.2 Basic Support

Our team provides basic support for general Asprise UI Components developers. Email your technical questions to support@asprise.com

Advice: You are strongly recommended to subscribe our premium support service in order to get your problems solved quickly.

8.3 Premium Support Services

Free fixed period of premium support services subscription comes with every license purchased. You may optionally extend premium support services after your subscription expires. For more details, visit the order page.

8.4 Professional Services

Our team are ready to help you to develop various applications, components. Please send your query to info@asprise.com