IMAGE HANDLER - LEADTOOLS Demonstration Programs

Summary of LEADTOOLS Features
Summary of All Supported Image File Formats
Introduction To Image Processing with LEADTOOLS
LEADTOOLS Products
About LEAD Technologies, Inc.
User Interface for Annotations

NOTE: These programs let you get acquainted with the features of LEADTOOLS. **They will quit working on Decmber 31, 1997.**

Thank you for evaluating LEADTOOLS Image Handler, version 8.0. This version is actually a sneak preview of version 8.0. This diskette will allow you to evaluate several new features and improvements that will be released in version 8.0, including LEAD's fastest CCITT G3 and G4 loading functions, and LEAD's fastest document image processing functions.

In addition to the speed enhancements, version 8.0 of LEADTOOLS will also include more than 100 special display, text, and shape multimedia effects. Programmers will be able to use and combine these effects to create dynamic applications for the Internet, Intranet, and desktop application markets.

The Features and File Formats in this help file represent the currently shipping code in LEADTOOLS version 8.0.

Thank you again for considering LEADTOOLS for your imaging toolkit. If you have any questions or comments, please contact LEAD Technologies at (704) 332-5532 or at sales@leadtools.com.

These programs demonstrate many of the features of the LEADTOOLS imaging toolkits. The toolkits include 16- and 32-bit ActiveXs (OCXs), a 16-bit VBX, and 16- and 32-bit DLLs, which are available in various bundled combinations described in LEADTOOLS Products.

Each IMAGE HANDLER diskette contains either the WIN16 or WIN32 demonstration programs.

These programs are taken directly from the example programs for the LEADTOOLS API. The various toolkits include these and many other examples. All of the examples have source code that you can copy and use in your own application. (The examples are designed to assist developers, and are not intended as end-user applications.) For a summary of the examples in various toolkits, refer to Example Programs.

Some of the features in these demonstration programs, such as annotations and region processing, are available only in the Express editions. (With the release of version 8.0, region processing will be available in Pro versions.) Also, some significant features, such as data binding, are not demonstrated in these programs. To review all of the available features in various toolkits, refer to Summary of LEADTOOLS Features.

- The following demonstration programs are for WIN32 platforms. They use the LEADTOOLS WIN32 DLLs, and therefore, will only run under Win32s, NT, or Win95.

WIN32 Demonstration shows most of the features of the LEADTOOLS API in a 32-bit environment. Demonstrated features include file format support, image processing, region processing, and display options.

WIN32 Special Effects demonstrates the LEADTOOLS special effects, including paint effects, transitions, three-dimensional text, and three-dimensional shapes.

WIN32 Annotations demonstrates the high-level annotation features of the LEADTOOLS API in a 32-bit environment. Most of the demonstrated features are automated and require very little programming. Refer to User Interface for Annotations.

- The following demonstration programs are for WIN16 platforms. They use the LEADTOOLS WIN16 DLLs, and therefore, will run under Windows 3.x, or under NT or Win95 as 16-bit applications.

WIN16 Demonstration shows most of the features of the LEADTOOLS API in a 16-bit environment. Demonstrated features include file format support, image processing, region processing, and display options.

WIN16 Special Effects demonstrates the LEADTOOLS special effects, including paint effects, transitions, three-dimensional text, and three-dimensional shapes.

WIN16 Annotations demonstrates the high-level annotation features of the LEADTOOLS API in a 16-bit environment. Most of the demonstrated features are automated and require very little programming. User Interface for Annotations.

All of the demonstration programs were compiled with the *LEADTOOLS Pro Express* product. To achieve the same capabilities and performance in your code, you should purchase LEADTOOLS Pro Express.

About LEAD Technologies, Inc.

LEAD is the developer and publisher of the award winning LEADTOOLS imaging toolkits, and has been developing and supplying imaging toolkits to programmers for over six years. LEAD produces code that is mature, stable, easy to use, and optimized for high performance. After six years of working with thousands of programmers, LEAD understands that issues such as documentation, function samples, example source code, and timely technical support, are very important to a developer. These non-object code components can make the difference when meeting a deadline, or when trying to win Best of Show.

LEADTOOLS is market tested and proven, and has been shipped to millions of customers through the integration into products by Microsoft, Corel, Delrina, Xerox, Hewlett Packard, and many other satisfied customers. Whether a programmer is creating a web page, a document imaging system, a paint/drawing program, or an imaging database that will run on a desktop, or over an Intranet or the Internet, LEADTOOLS is the best choice in imaging toolkits.

General Information

Technical Support: As a registered user, free technical support is available. You must register your product to receive technical support. Technical support is available 9:00 AM through 6:00 PM Eastern Standard Time.

MONEY BACK GUARANTEE ON ALL PRODUCTS: In the U.S. and Canada LEAD offers an unconditional 30 day money-back guarantee on all products. Try our toolkit - risk free - and we guarantee that you will select LEAD as your partner in development.

For more Information, please contact:

Sales.@Leadtools.com
Home page: http://www.leadtools.com

LEAD Technologies, Inc.
900 Baxter Street
Charlotte, NC 28204

Main: (704) 332-5532
Sales: (800)-637-4699

Technical support: (704) 372-9681 or Support@Leadtools.com
Fax: (704)372-8161
BBS: (704) 334-9045
CompuServe: 71333,2237 or GO LEADTECH

Example Programs

These are descriptions of the example programs that ship with the LEADTOOLS toolkits.

- API Example Programs
- ActiveX (OCX) Example Programs
- VBX Example Programs

API Example Programs

These are descriptions of the DLL example programs that ship with the LEADTOOLS toolkits.

ANNOTATE	**Annotations** demonstrates the annotation capabilities of the express toolkit.
CLRSPACE	**Color Space** loads and displays the image file specified on the command line, then uses the *L_ConvertColorSpace* function to convert from RGB to YUV, CMYK, and other formats, redisplaying the image after all conversions are complete.
COLOR	**Color Resolution** loads the 24-bit image file specified on the command line and calls *L_ColorResBitmap* to optimize and dither the image to 8 bits per pixel.
COMPCB	**Compress Callback** loads the image file specified on the command line, then displays the image, compresses the bitmap to file using a callback function, clears the bitmap, loads the compressed image, and displays the newly loaded image.
DELPHI	**Delphi Demo** demonstrates the use of LEADTOOLS with the Borland Delphi development system.
DEMO	**Main Demo** is the main example program. It uses pull-down menus to demonstrate most of the LEADTOOLS features in one application.
DIBDDB	**DIBDDB Demo** demonstrates the functions that translate to and from Windows device independent bitmaps (DIBs) and device dependent bitmaps (DDBs).
DRAW	**Draw Demo** uses pull-down menus to show some of the drawing capabilities available with LEADTOOLS.
EZFUNC	**Simple Load** uses simple LEADTOOLS functions to load an image file specified on the command line and display the bitmap in a window.
EZREGC	**Reg. Class Load** uses the L_BITMAPCLASS registered class to load an image file specified on the command line and display the bitmap in a window.
FEATURE1	**Some Basic Features** loads and displays the 24-bit image file specified on the command line, then makes the following changes, redisplaying the image after each change: optimize to 8 bits per pixel, flip, lighten, resize, and rotate.
FEATURE2	**More Basic Features** loads and displays the 24-bit image file specified on the command line, then makes the following changes, redisplaying the image after each change: dither to 8 bits per pixel, reverse, grayscale, resize, and fill with one color.
FEATURE3	**A Few More Features** loads and displays the image file specified on the command line, copies the image data to a different bitmap, halftones the original image, restores the image by copying the handle of the new bitmap, then redisplays the image.
FEEDLOAD	**Feed Load** demonstrates how to use *L_FeedLoad* and related functions by simulating the receipt of a transmitted image.
GETROW	**Get Row** loads the 24-bit image file specified on the command line, then rotates the image 90 degrees using *L_GetBitmapRow* and *L_PutBitmapRowCol*, and redisplays the image.
INFO	**Info** displays image information from the file specified on the command line.
LOADCB	**Load Callback** loads the image file specified on the command line, using a callback function to display the image as it is loaded.
LOADSAVE	**Load and Save** uses pull-down menus to let the user load an image file, display the image in a window, and save the image to a specified file name, format, and bits per pixel. The options are implemented using a Windows common dialog box.
MEMORY	**Load Memory** demonstrates the ability load a file from memory. It allocates memory, loads the image file specified on the command line into the allocated memory, then loads the image from memory to a bitmap using *L_LoadBitmapMemory* and displays the image.
MEMORYCB	**Load Memory Callback** demonstrates the ability use a callback function when loading a file from memory. It allocates memory, loads the image file specified on the command line

	into the allocated memory, then loads the image from memory to a bitmap using *L_LoadMemory* and a callback function, and displays the image.
MFCDEMO	**MFC Demo** demonstrates the use of LEADTOOLS with the Microsoft Foundation Class (MFC). This example includes a C++ wrapper for the LEADTOOLS API.
MFCFX	**Special Effects MFC Demo** demonstrates the LEADTOOLS special effects, including paint effects, transitions, three-dimensional text, and three-dimensional shapes.
OFFSET	**Load Save Offset** demonstrates the ability to load an image file that is embedded in a larger file. It does so by using the *L_LoadFileOffset* function, which lets you specify the offset and the size of the embedded file. This example first loads the normal image file specified on the command line. It embeds the image in a second file, which is specified in the second command line argument. It then loads and displays the embedded file.
PAINTDC	**Paint DC** loads the image file specified on the command line, and displays the image. The user can then choose a menu option to modify the source and display rectangles, and redisplay the image to see the changes.
PAINTEFF	**Paint Effect Demo** uses pull-down menus to demonstrate the various transitional painting effects that are available with LEADTOOLS.
REDIRECT	**Redirect I/O** demonstrates how to use the *L_RedirectIO* function. It redirects the standard I/O functions so that they read and seek into the allocated memory rather than a file.
REGCLASS	**Registered Class Demo** demonstrates most of the capabilities of L_BITMAPCLASS registered class.
RESIZE	**Resize** loads the image file specified on the command line, resizing the image while loading it. It then places the resized image over the original and displays the combined image.
SAVECB	**Save Callback** loads the image file specified on the command line, then displays the image and saves it to the current directory using a callback and status window.
SAVEMEM	**Save to Memory** demonstrates the ability to save a file in memory. It lets the user load an image file, display the image in a window, and save the image to a file in memory.
SCRLZOOM	**Scroll & Zoom Demo** uses pull-down menus to let the user load an image file, display the image in a window, and use zooming and scrolling features. In doing so, it also demonstrates the ability of the *L_PaintDC* function to automatically dither a 16-, 24-, or 32-bit image when displaying on a 4- or 8-bit device.
STAMPGET	**Get Stamp** reads a thumbnail (stamp) image stored in a compressed file that the user selects; then displays the image in a window. The only valid input file is a LEAD or JPEG file that is compressed with a stamp image. (Use the STAMPPUT example first to create an image file that contains a thumbnail.)
STAMPPUT	**Put Stamp** loads the image file specified on the command line, calls *L_ColorResBitmap* (if necessary) to obtain a 24 bit image, calls *L_CompressBitmapWithStamp* to compress the image to STAMP.CMP (a LEAD CMP compressed file format), then loads the stamp to a bitmap and displays the stamp.
THREAD	**Multi-thread** reads and displays all files from the directory specified on the command line. It creates a new thread to load images while the main thread paints the last image that was loaded. When the last image is loaded, the example uses the last image to cycle through the paint effects. The multi-thread feature works only on WIN32 systems.
UNDERLAY	**Underlay** demonstrates the *L_UnderlayBitmap* function. It loads two images specified on the command line and displays the first one. It then combines the images using the second one as an underlying image, and displays the combined image.

ActiveX (OCX) Example Programs

These are descriptions of the ActiveX (OCX) example programs that ship with the LEADTOOLS toolkits.

Internet

- EXAMPLES\OCX\INTERNET contains sample HTML files that use Java and VB Script to demonstrate the net-aware capabilities of the ActiveX.

Visual Basic

- EXAMPLES\OCX\VB\DEMO contains a sample application that demonstrates most of the ActiveX capabilities in Visual Basic 4.0. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.
- EXAMPLES\OCX\VB\ANNOTATE contains a sample application that demonstrates the annotation capabilities. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.
- EXAMPLES\OCX\VB\BIND contains a sample database application implemented using data binding. For WIN32 systems, this example references the IMAGES\LEADPIC.MDB database. For WIN16 systems, it references the IMAGES\LDPIC16.MDB database. To run the sample application, you can run the EXE file in the LEAD80\BIN directory, specifying the correct database as a command-line argument.
- (32-bit editions only) EXAMPLES\OCX\VB\DB contains sample project code for an ODBC database example in Visual Basic 4.0. To use the DB demo, you must first do the following:
    1. Specify an ODBC data source named LEADACCESS that references the IMAGES\LEADPIC.MDB database.
    2. Open the project in Visual Basic and modify the Dpeople data control's DatabaseName property to reference the IMAGES\LEADPIC.MDB database.
- EXAMPLES\OCX\VB\DRAW uses the LEADTOOLS ActiveX to load an image, to demonstrate graphics functions, and to save the modified image. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.
- EXAMPLES\OCX\VB\FXDEMO demonstrates the LEADTOOLS special effects, including paint effects, transitions, three-dimensional text, and three-dimensional shapes. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.
- EXAMPLES\OCX\VB\MAGNIFY uses two LEAD controls. The first control displays the unmagnified image. The second control, when enabled, can be dragged across the first one to magnify the covered portion of the image. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.

Visual C++

- EXAMPLES\OCX\MFC\DEMO contains sample project code for Visual C++ 1.0 and later (16- or 32-bit). To run the sample application, you can run the EXE file in the LEAD80\BIN directory.
- (16-bit editions only) EXAMPLES\OCX\MFC\ANNOTATE demonstrates the LEADTOOLS annotation capabilities. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.
- (16-bit editions only) EXAMPLES\OCX\MFC\DRAW uses the LEADTOOLS ActiveX to load an image, to demonstrate graphics functions, and to save the modified image. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.
- (16-bit editions only) EXAMPLES\OCX\MFC\MAGNIFY uses two LEAD controls. The first control displays the unmagnified image. The second control, when enabled, can be dragged across the first one to magnify the covered portion of the image. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.

Visual C++ 4.0

- (32-bit editions only) EXAMPLES\OCX\MFC40\DEMO contains sample project code for Visual C++ 4.0. You can then open the project in Visual C++; then compile and run it.
- (32-bit editions only) EXAMPLES\OCX\MFC40\ANNOTATE demonstrates the LEADTOOLS annotation capabilities. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.

■ (32-bit editions only) EXAMPLES\OCX\MFC40\DB contains sample project code for a database example in Visual C++ 4.0. To use the DB demo, you must first do the following:

1. Specify an ODBC data source named LEADACCESS that references the IMAGES\LEADPIC.MDB database.
2. Open the project in Visual C++ and modify the GetDefaultDBName function to reference the IMAGES\LEADPIC.MDB database.

■ (32-bit editions only) EXAMPLES\OCX\MFC40\DRAW uses the LEADTOOLS ActiveX to load an image, to demonstrate graphics functions, and to save the modified image. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.

■ (32-bit editions only) EXAMPLES\OCX\MFC40\FXDEMO demonstrates the LEADTOOLS special effects, including paint effects, transitions, three-dimensional text, and three-dimensional shapes. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.

■ (32-bit editions only) EXAMPLES\OCX\MFC40\MAGNIFY uses two LEAD controls. The first control displays the unmagnified image. The second control, when enabled, can be dragged across the first one to magnify the covered portion of the image. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.

Visual FoxPro

■ (32-bit editions only) EXAMPLES\OCX\FOX contains sample project code for Visual FoxPro 3.0. To run the sample application from FoxPro, use the *Program* > *Do* option, and select the \EXAMPLES\OCX\FOX\DEMO.APP application.

To implement the database capabilities in the DEMO.APP application, specify the current (\FOX) directory as an ODBC data source. Then when you run the application, use the *File* > *Open Database* option to open the ODBC data source.

Access 7.0

Note: You can convert the Access 7.0 examples to Access 97. In the database example, the maximize button is disabled because of problems with the the SetMenuItem method in Access 97.

■ (32-bit editions only) EXAMPLES\OCX\ACCESS7\DEMO contains sample project code for Microsoft Access 7.0. To run the sample application from Access, open the sample database, then open the MAIN form. To avoid ODBC setup issues, this demo does not add images to a database.

■ (32-bit editions only) EXAMPLES\OCX\ACCESS7\DB contains sample project code for a database example in Access 7.0. To use the DB demo, you must do the following:

1. Specify an ODBC data source named LEADACCESS that references the IMAGES\LEADPIC.MDB database.
2. Open the project in Access 7.0 and modify Query1 to reference the IMAGES\LEADPIC.MDB database.
3. Run the sample application from Access by opening the MAIN form.

Access 2.0

■ (16-bit editions only) EXAMPLES\OCX\ACCESS contains sample project code for Microsoft Access 2.0. To run the sample application from Access, open the MAIN form.

Delphi 2.0

■ EXAMPLES\OCX\DELPHI\DEMO contains sample project code for Delphi 2.0. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.

■ EXAMPLES\OCX\DELPHI\DRAW uses the LEADTOOLS ActiveX to load an image, to demonstrate graphics functions, and to save the modified image. To run the sample application, you can run the EXE file in the LEAD80\BIN directory.

VBX Example Programs

These are descriptions of the VBX example programs that ship with the LEADTOOLS toolkits.

Main Demo

VBXDEMO.EXE is a demonstration program that uses the LEADTOOLS VBX for displaying and manipulating images. Most of the properties of the LEAD Control are demonstrated at some point in this program. The project that created this .EXE is **DEMO.MAK**, which is in the DEMO subdirectory.

Generally, the menu options reflect the programming capabilities, which you can find by searching in the online help. Options on the *Preferences* menu do the following:
Dithering sets the PaintDither property.

Bitonal Scaling sets the BitonalScaling property.

Paint With Fixed Palette sets the PaintPalette property.

Paint While Load sets the PaintWhileLoad property.

Show Open Options implements a dialogue box for opening a file. The dialogue box lets you select the bits per pixel to load and the page number (for multi-page files). It also lets you access file information using the InfoFile property.

Show Progress sets the EnableStatus property, and uses the status event to show completion percentages, as described in Detecting Events.

Draw Example

VBXDRAW.EXE is a demonstration program that uses the LEADTOOLS VBX to load an image, to demonstrate graphics functions, and to save the modified image. This program calls several Windows GDI functions using the bitmap as a drawing surface. The project that created this .EXE is **DRAW.MAK**, which is in the DRAW subdirectory.

Database Example

VBXDBASE.EXE is a demonstration program that uses the LEAD Visual Basic Control for displaying, adding, and updating images in an ACCESS database. The project that created this .EXE is **DATABASE.MAK**, which is in the DATABASE subdirectory.

A command line argument specifying the full name of the database is required. The command line argument can be set from the project by selecting the following sequence of options from the main VB Window after opening the project: Options, Project, Command Line Arguments.

If you are running the example .EXE, the full name of the database should follow the .EXE name on the command line. One database is provided by LEAD. It is an .MDB file in the IMAGES subdirectory of the main installation directory.

Magnify Example

VBXMAGN.EXE is a demonstration program that uses two LEAD Visual Basic Controls. The first control displays the unmagnified image. The second control, when enabled, can be dragged across the first one to magnify the covered portion of the image. The project that created this .EXE is **MAGNIFY.MAK**, which is in the MAGNIFY subdirectory.

Delphi Example

DPDEMO.EXE demonstrates the use of the LEADTOOLS VBX with Borland Delphi. The project that created this .EXE is **DPDEMO.DPR**, which is in the DELPHI subdirectory.

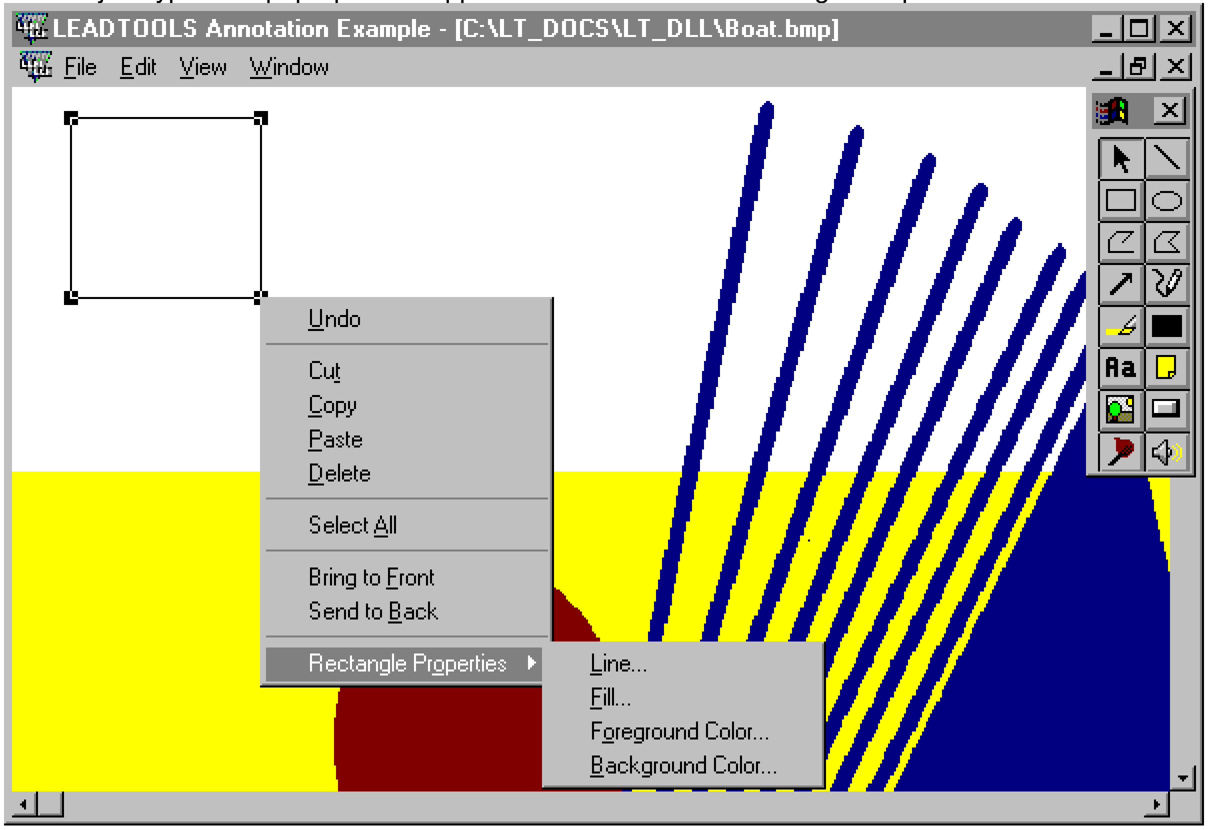
License Agreement

This is a copy of the license agreement that appears when you install IMAGE HANDLER.

User Interface for Annotations

The *Edit* > *User Mode* option lets you select design or run mode for annotations.

In design mode, you can select tools using the *Edit* > *Tool* option and draw annotation objects over the image in the window. You can then use the right mouse button to get a pop-up menu that is tailored to each object type. The pop-up menu appears as shown in the following example:



The popup menu is context sensitive. It lists only the properties that can be changed on the selected object or objects. If you use the popup menu to change a property, the value that you specify is applied to the current selection, and it becomes the default for the next object.

The following topics describe the automated annotation features for particular objects and tools:

- Setting Defaults
- Selection Pointer Tool
- Line Tool
- Rectangle Tool
- Ellipse Tool
- Polyline Tool
- Polygon Tool
- Pointer Tool
- Freehand Line Tool
- Highlight Tool
- Redaction Tool
- Text Tool
- Note Tool
- Stamp Tool
- Hot spot Tool
- Button Tool
- Audio Clip Tool

Setting Defaults

Note: This topic is for Express editions only.

When using annotations in design mode, you can change the default properties by clicking with the right mouse button on an empty spot (a place on the image without any annotations). The popup menu lets you do the following:

- Undo the last user action.
- Select all annotation objects in the container.
- Delete all selected objects.
- Change the default line width and style properties. See Line Dialog Box for Annotations.
- Change the default fill mode (transparent, translucent, or opaque) and the fill pattern. See Fill Dialog Box for Annotations.
- Change the polygonal fill mode, which controls which areas are filled when lines cross. See Fill Dialog Box for Annotations.
- Change the default foreground and background colors. See Color Dialog Box for Annotations.
- Change the font selection and font characteristics. See Font Dialog Box for Annotations.

The default background color does not affect highlight, redaction, and note objects.

Selection Pointer Tool

Note: This topic is for Express editions only.

The selection pointer is the mouse pointer for selecting and manipulating annotation objects in design mode.

With the selection pointer, you can select one or more objects as follows:

- You can select a single object by clicking on the object.
- You can click and drag to form a rectangular selection area that will select all objects that are completely contained within it.
- You can hold down the shift key and click on several objects one-at-a-time to select them all.

When an object is selected, handles appear on the corners of its bounding rectangle. You can click on the body of the object and drag it to another position. You can click on a handle and drag to resize the object. If more than one object is selected, all of them are moved or resized.

In addition, you can use this pointer to rotate an object. If you hold down the shift key, click on a handle, and drag the mouse, the object rotates. If more than one object is selected, all of them are rotated, using the center of the group as the center of rotation. You cannot rotate a note, button, or audio clip. If you rotate a group that includes one of these objects, the object will move to the new position, but will retain its orientation.

The popup menu is also affected if more than one object is selected. In that case, when you click the right mouse button, the popup menu lets you change the properties of all the selected objects, and it lets you delete all the selected objects.

Line Tool

Note: This topic is for Express editions only.

The line annotation object is a simple line. In design mode, the line is started on a mouse-down event and completed on a mouse-up event.

You can then click on the object with the right mouse button to do the following:
- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the line width and style properties. See Line Dialog Box for Annotations.
- Change the foreground color. See Color Dialog Box for Annotations.

Rectangle Tool

Note: This topic is for Express editions only.

The rectangle annotation object is a simple rectangle. In design mode, you can click and drag to specify the rectangle in the current window.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the line width and style properties. See Line Dialog Box for Annotations.
- Change the fill mode (transparent, translucent, or opaque) and the fill pattern. See Fill Dialog Box for Annotations.
- Change the foreground and background colors. See Color Dialog Box for Annotations.

Ellipse Tool

Note: This topic is for Express editions only.

The ellipse annotation object is a simple ellipse. In design mode, the you click and drag to specify the ellipse in the current window.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the line width and style properties. See Line Dialog Box for Annotations.
- Change the fill mode (transparent, translucent, or opaque) and the fill pattern. See Fill Dialog Box for Annotations.
- Change the foreground and background colors. See Color Dialog Box for Annotations.

Polyline Tool

Note: This topic is for Express editions only.

The polyline annotation object is an array of points that create a sequence of joined lines. In design mode, each line segment is formed with a mouse click, and the object is completed on a double click.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the line width and style properties. See Line Dialog Box for Annotations.
- Change the foreground color. See Color Dialog Box for Annotations.

Polygon Tool

Note: This topic is for Express editions only.

The polygon annotation object is an array of points that define the vertices of a polygon. In design mode, each line segment is formed with a mouse click, and the object is completed on a double click, which closes the polygon with a line from the last point to the first point.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the line width and style properties. See Line Dialog Box for Annotations.
- Change the fill mode (transparent, translucent, or opaque) and the fill pattern. See Fill Dialog Box for Annotations.
- Change the polygonal fill mode, which controls which areas are filled when lines cross. See Fill Dialog Box for Annotations.
- Change the foreground and background colors. See Color Dialog Box for Annotations.

Pointer Tool

Note: This topic is for Express editions only.

The pointer annotation object is defined by two points, with the an arrow at the first point. In design mode, you can click and drag to specify the points in the current window.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the line width and style properties. See Line Dialog Box for Annotations.
- Change the foreground color. See Color Dialog Box for Annotations.

■ Freehand Line Tool

Note: This topic is for Express editions only.

The freehand line annotation object is an array of points that create a sequence of joined lines. In design mode, each line segment is formed with a mouse move event when the left button is down, and the object is completed on a mouse up event.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the line width and style properties. See Line Dialog Box for Annotations.
- Change the foreground color. See Color Dialog Box for Annotations.

Highlight Tool

Note: This topic is for Express editions only.

The highlight annotation object is a rectangle with default properties that are appropriate for highlighting an area. In design mode, you can click and drag to specify the rectangle in the current window.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the background color. See Color Dialog Box for Annotations.

Redaction Tool

Note: This topic is for Express editions only.

The redaction annotation object is a rectangle with default properties that are appropriate for blacking out an area. In design mode, you can click and drag to specify the rectangle in the current window.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the background color. See Color Dialog Box for Annotations.

Text Tool

Note: This topic is for Express editions only.

The text annotation object is a rectangle containing a character string with font properties. The text will wrap on word breaks within the object's specified rectangle. In design mode, you can click and drag to specify the rectangle in the current window. A dialog box then lets you enter the text.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the fill mode (transparent, translucent, or opaque). See Fill Dialog Box for Annotations.
- Change the background color. See Color Dialog Box for Annotations.
- Change the font selection and font characteristics. See Font Dialog Box for Annotations.
- Edit the text. See Text Dialog Box for Annotations.

You cannot flip or reverse a text object, but you can rotate it. If you flip or reverse a container that includes a text object, the text will move to the new position, but will retain its orientation.

Note Tool

Note: This topic is for Express editions only.

The note annotation object is a rectangle with a shadow border and color background (yellow by default). It contains a character string with font properties. The text will wrap on word breaks within the object's specified rectangle. In design mode, you can click and drag to specify the rectangle in the current window. A dialog box then lets the user enter the text.

You can then click on the object with the right mouse button to do the following:
- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the background color. See Color Dialog Box for Annotations.
- Change the font selection and font characteristics. See Font Dialog Box for Annotations.
- Edit the text .See Text Dialog Box for Annotations.

You cannot flip, reverse, or rotate a note. If you flip, reverse, or rotate a container that includes a note, the note will move to the new position, but will retain its orientation.

Stamp Tool

Note: This topic is for Express editions only.

The stamp annotation object is a bitmap image. In design mode, you can click and drag to specify a rectangle. The bitmap is scaled to fit the rectangle.

You can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Select an image file to load into the stamp. See Stamp Bitmap Dialog Box for Annotations.

On a device that displays 256 colors or less, you should use a fixed palette when displaying bitmaps. Otherwise, a palette shift will occur on stamp objects.

You cannot flip, reverse, or rotate a stamp object. If you flip, reverse, or rotate a container that includes a stamp, the object will move to the new position, but will retain its orientation.

Button Tool

Note: This topic is for Express editions only.

The button annotation object is an image of a standard Windows button that can be activated. In design mode, you can click and drag to specify the size of the button in the current window. A dialog box then lets you enter the text for the button. In run mode, clicking on the button triggers an AnnClicked event. You can add code to process the event. (The example program merely pops up a message box.)

In design mode, you can then click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Change the foreground color. See Color Dialog Box for Annotations.
- Edit the text. See Text Dialog Box for Annotations.

You cannot flip, reverse, or rotate a button. If you flip, reverse, or rotate a container that includes a button, the button will move to the new position, but will retain its orientation.

Hot Spot Tool

Note: This topic is for Express editions only.

The hot spot annotation object is a rectangle that can be activated for an application-defined purpose. In design mode, the rectangle contains a scaled image. You can click and drag to specify the rectangle. In run mode, the object is transparent, but the mouse pointer changes to a hand when it is over the object. In run mode, clicking on the button triggers an AnnClicked event. You can add code to process the event. (The example program merely pops up a message box.)

In design mode, you can click on the object with the right mouse button to do the following:

- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.

You cannot flip, reverse, or rotate a hot spot. If you flip, reverse, or rotate a container that includes a hot spot, the object will move to the new position, but will retain its orientation.

Audio Clip Tool

Note: This topic is for Express editions only.

The audio clip annotation object is a rectangle containing a scaled image that can be activated to play a WAV file. In design mode, you can click and drag to specify the rectangle in the current window. A dialog box then lets you specify the WAV file. In run mode, clicking on the icon plays the WAV file.

In design mode, you can click on the object with the right mouse button to do the following:

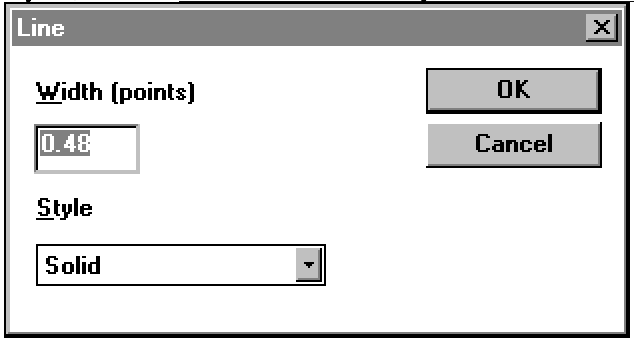
- Undo the last user action.
- Cut or copy the object to the clipboard.
- Delete the object.
- Bring the object to the front or send it to the back.
- Select a WAV file to load into the object. See Audio File Dialog Box for Annotations.

You cannot flip, reverse, or rotate an audio clip. If you flip, reverse, or rotate a container that includes an audio clip, the object will move to the new position, but will retain its orientation.

Line Dialog Box for Annotations

Note: This topic is for Express editions only.

The following dialog box appears in automation mode when you change the line properties of an annotation object. The line width is based on the associated bitmap's physical resolution (DPI), so that it matches the scale of the original scanned image. Setting the width to a small number will not make the line disappear. On any output device, lines will always be at least one pixel wide. To see the possible line styles, refer to Illustration of Line Styles for Annotations.



Width (points)

0.48

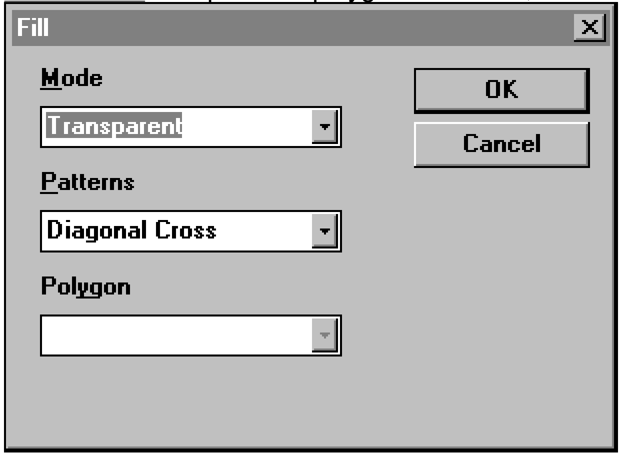
Style

Solid

Fill Dialog Box for Annotations

Note: This topic is for Express editions only.

The following dialog box appears in automation mode when you change the fill properties of an annotation object. To see the possible modes and patterns, refer to Illustration of Fill Options for Annotations. For possible polygon fill modes, refer to Illustration of the Polygon Fill Mode for Annotations.



Mode: Transparent

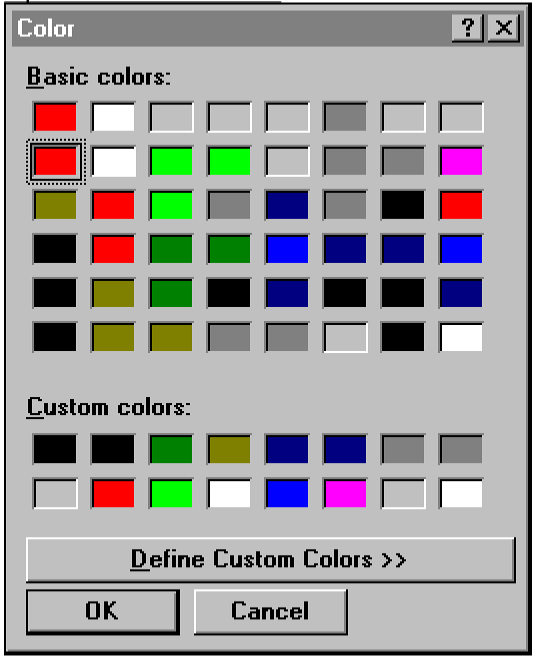
Patterns: Diagonal Cross

Polygon:

Color Dialog Box for Annotations

Note: This topic is for Express editions only.

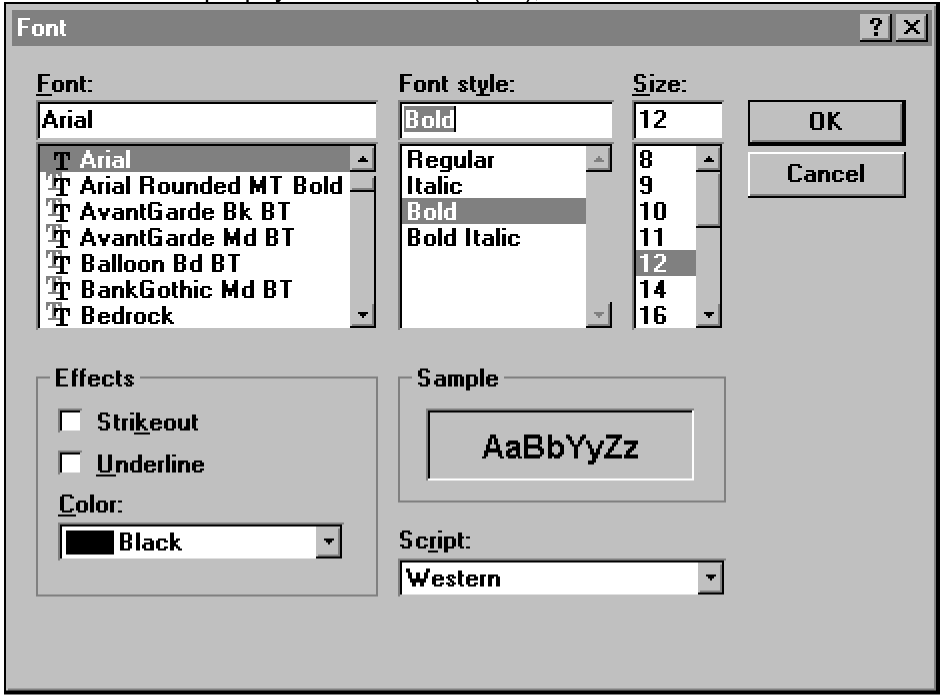
The following dialog box appears in automation mode when you change the foreground or background color property of an annotation object. It is implemented using the Windows common dialog box. To see how the foreground and background colors affect the appearance of an object, refer to Illustration of Fill Options for Annotations.



Font Dialog Box for Annotations

Note: This topic is for Express editions only.

The following dialog box appears in automation mode when you change the font properties of an annotation object. It is implemented using the Windows common dialog box. The font size is based on the associated bitmap's physical resolution (DPI), so that it matches the scale of the original scanned image.



Font: Arial

Font style: Bold

Size: 12

Strikeout: [ ]

Underline: [ ]

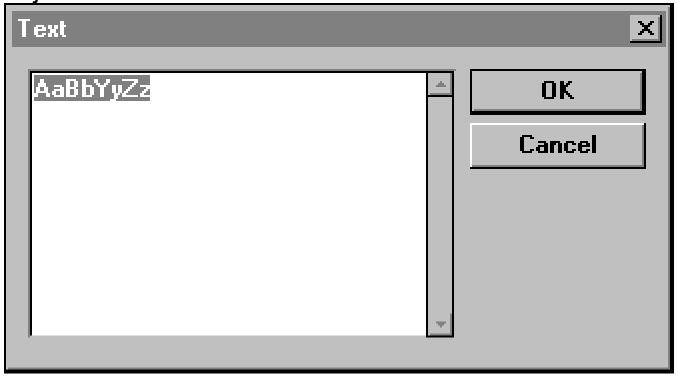
Color: Black

Script: Western

Text Dialog Box for Annotations

Note: This topic is for Express editions only.

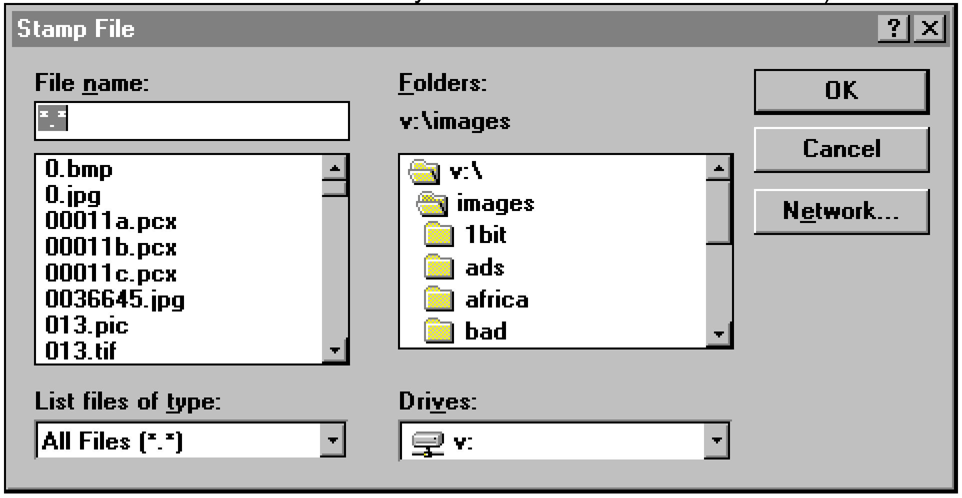
The following dialog box appears in automation mode when you change the text property of an annotation object. Word wrap is implemented in the dialog box and in the rectangle that defines the annotation object.



Stamp Bitmap Dialog Box for Annotations

Note: This topic is for Express editions only.

The following dialog box appears in automation mode when you update the bitmap property of a stamp annotation object. It is implemented using the Windows common dialog box. (The style of this dialog box is different on WIN16 and WIN32 systems. This is the WIN16 version.)



Audio File Dialog Box for Annotations

Note: This topic is for Express editions only.

The following dialog box appears in automation mode when you specify the WAV file for an audio clip annotation object. It is implemented using the Windows common dialog box. (The style of this dialog box is different on WIN16 and WIN32 systems. This is the WIN16 version.)

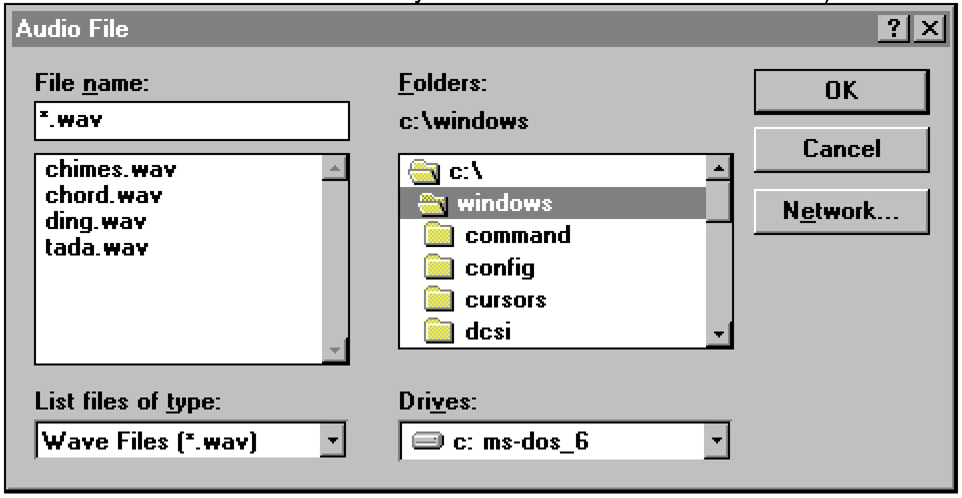


Illustration of Line Styles for Annotations

Note: This topic is for Express editions only.

The following rectangle annotation objects illustrate the available line styles. Note that   Windows displays all lines greater than 1 pixel as solid.

Back row:	Solid	Dash	Dot
Front row:	Dash-dot	Dash-dot-dot	Null

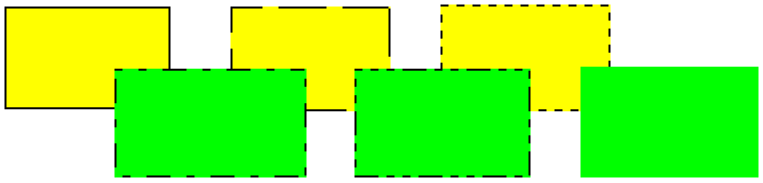


Illustration of Fill Options for Annotations

Note: This topic is for Express editions only.

The following rectangle annotation objects illustrate the available fill modes and patterns. The foreground and background colors help to illustrate the effects.

Back row:

Fill mode: Opaque	Fill mode: Opaque	Fill mode: Opaque	Fill mode: Opaque
Fill pattern: Solid	Fill pattern: Cross	Fill pattern: Backward diagonal	Fill pattern: Vertical
Fore color: Black	Fore color: Black	Fore color: Black	Fore color: Black
Back color: Yellow	Back color: Yellow	Back color: Yellow	Back color: Yellow

Front row:

Fill mode: Translucent	Fill mode: Transparent	Fill mode: Opaque
Fill pattern: Forward diagonal	Fill pattern: Diagonal cross	Fill pattern: Horizontal
Fore color: Black	Fore color: Black	Fore color: Red
Back color: Green	Back color: Green	Back color: Green

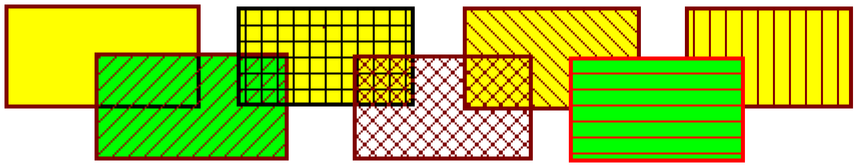
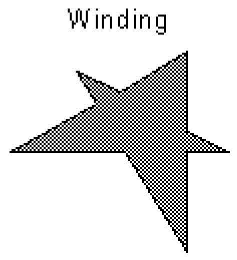


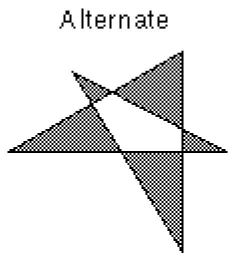
Illustration of the Polygon Fill Mode for Annotations

Note: This topic is for Express editions only.

The following polygon annotation objects illustrate the available polygon fill modes.

Winding	The area inside the resulting exterior lines is filled.
Alternate	The filled area includes the area between odd-numbered and even-numbered polygon sides on each scan line.





Summary of LEADTOOLS Features

This is a summary of LEADTOOLS features for all components:

- DLL functions and Windows messages in the API
- ActiveX (OCX) methods, properties, and events
- VBX properties and events

You can use this help file to explore the full range of possibilities for implementing LEADTOOLS capabilities.

LEADTOOLS Products

General Features

Working with the Internet

Working with Files

Working with Image Data

Working with the Images in Bitmaps

Working with Palettes

Working with Displays

Working with Special Effects, Shapes, and Text

Working with Regions

Working with Annotations [Pro Express Only]

Working with Databases

Working with Other Input and Output

Working with Events and Callbacks

Working with Raw Image Data [DLL Only]

Working with Image Descriptions and Other Information

General Features

LEADTOOLS is a family of comprehensive imaging tools designed to help programmers integrate black & white, grayscale and color images into their applications as quickly as possible.   Designed to be the most comprehensive imaging technology available in a development toolkit, LEADTOOLS is a collection of almost 300 functions, properties and methods, with imaging technology in 12 general categories: scanning, color conversion, display, multi-media special effects, annotation, image processing, animation, compression, image format import/export filters, Internet imaging, database imaging, and printing.

The following are general features of the LEADTOOLS toolkits:

■ **Scanning**. LEADTOOLS supports both 16 & 32 bit TWAIN interfaces for driving the most popular scanning devices.   Developers have the option of using the default TWAIN interface of the datasource, or they can choose to by-pass the user interface to create their own.   Additionally, both native and buffered RAM transfer modes are supported. Refer to Working with Other Input and Output.

■ **Color Conversion**.   LEADTOOLS makes it possible to expand or reduce an image's color depth with multiple dithering methods using various palette options.   Programmers can choose from 8 different dithering methods including Floyd Stein, Burkes, Ordered dithering, Color reduction, Stucki, Stephens, Sierra and Jarvis.   LEADTOOLS offers flexible palette options including LEAD's fixed palette, an auto-optimized palette, or a user defined palette.   Images can be separated and/or reconstructed to/from individual RGB, CMYK, CMY, HSV, YUV and HSL color planes.   LEADTOOLS also supports grayscaling and halftoning with rotated screens. Refer to Working with Image Data.

■ **Display**.   LEADTOOLS provides optimized rendering of any image to all display devices with auto-dithering and scaling.   The fixed palette option will eliminate the palette shift nightmare of a 256 or less color device, yet maintain the maximum image quality.   Easy to use, ultra fast paint, zoom, scroll, crop, and pan will make the imaging portion of the programmer's application fly.   The 256 shade ScaleToGray (interpolated) and FavorBlack viewing options will enhance the quality and readability of the display of 1-bit document images. (The ScaleToGray and FavorBlack functions are only available in the Pro Express). Refer to Working with Displays.

■ **Multi-Media Special Effects.**   LEADTOOLS now provides special effects for developing multimedia projects and/or presentation applications.   Use 113 Paint Effects, 64 Dissolve Effects, and 24 Transition Effects with delays, grain sizes, pattern brushes (up to 64 passes), a colored wand and a transparent color. The combinations are unlimited.   Other Special Effects include: 3D Shapes, 3D Rotated Text, 3D Frames, Gradient and Pattern Filled shapes, and much more.   Refer to Working with Special Effects, Shapes, and Text.

■ **Annotations**.   LEADTOOLS' annotations can be added to both black and white, and color images. When using annotations, the developer can select from text, highlights, sticky notes, ellipses, free-hand scribbling, redaction (blackout), hot-spots, audio, bitmap and text stamps, polygons and many more. Programmers can choose to keep the annotation as a separate object layer, or the annotation can be burned or embedded into the image.   The developer can choose between low level access to the annotations, or a fully automated mode with a ready to use floating, customizable toolbar.   (The annotation code is only available in the Pro Express). Refer to Working with Annotations.

■ **Image Processing**.   LEADTOOLS has over 50 image processing functions, some of which are typically found only in high performance end user programs such as PhotoShop.   The image processing functions are separated into in four (4) categories: transforms, filters, drawing and region of interest. **Transforms**.   LEADTOOLS' transforms include resize, resample (interpolated resize), rotate (.01 degree), flip, invert, reverse, crop, underlay, shear, transpose, fill, auto deskew, and combine bitmap (with mathematical and Boolean operations).   **Filters**.   LEADTOOLS' filters include sharpen, blur, brighten, darken, invert, hue and saturation, intensity, contrast, gamma correction, histogram equalize, edge detect, line detect, emboss, mosaic, posterize, median and noise filters, spatial filter (which can be pre-defined such as gradient, laplacian, sobel, prewitt, shift and difference, line segment, or they can be customized), auto deskew, despeckle, and more. For a complete list, refer to Working with the Images in Bitmaps. **Drawing**.   LEADTOOLS provides ActiveX methods for drawing and gives programmers access to Microsoft Windows GDI functions (such as TextOut, BitBlt, Ellipse, and Rectangle), to draw directly to the bitmap's surface.   **Region of Interest**.   The region of interest code makes it possible to process only a specific portion of an image rather than the entire bitmap.   Regions can be comprised of any combination of rectangles, ellipses, rounded-rectangles, freehand shapes, polygons, transparent color and more. Refer to Working with Regions.

- **Animation**. LEADTOOLS gives the programmer the ability to read and display AVI and animated GIF files, and work with arrays of images to convert still images into animated GIFs. LEADTOOLS allows loading of a multipage file into a list of bitmaps and playing an animation sequence using the list. With the paint-while-load feature a programmer can play the animation while the list is being loaded.
- **Compression**. LEADTOOLS offers more compression options than any other toolkit on the market, including JPEG, CCITT G3/G4, Huffman, Arithmatic, RLE, LZW, and LEAD's own proprietary CMP. LEADTOOLS Pro Express delivers the fastest software only JPEG algorithms available. The LEAD CMP formats results in smaller file sizes and maintains better image quality than industry standard compression formats.
- **Image Format Import/Export filters**. LEADTOOLS provides support for over 40 of the most popular raster image formats (including practically all known "flavors") at different pixel depths (1-32 bits per pixel) and compression options. A partial list includes DICOM (DIC), BMP, CALS, DIB, EPS, FAX (raw), GIF, IMG, IOCA, JPEG (JPG, JFIF, JTIF), LEAD CMP, MAC, MSP, PCD, PCT, PCX, PNG, PSD, RAS, TGA, TIFF 6.0 (LEADTOOLS contains the most comprehensive TIFF support available), Kodak FlashPix, VDA, VST, WINFAX, MSFAX (AWD), WMF, WPG. LEAD is constantly adding the support for new file formats, and "flavor" support to formats already supported. (A separate license from Unisys is required to use the GIF and TIFF LZW code). Refer to Summary of All Supported Image File Formats and Working with Files.
- **Internet Imaging**. LEADTOOLS features a Net Aware ActiveX and a Netscape plug-in for Internet applications, including a Bitmap Datapath allowing images to be read from anywhere on the Internet, LEADTOOLS includes support for Progressive JPEG, Progressive CMP, and support for GIF interlace, transparency, animation, and embedded text. The GIF code has also been expanded to read and write the rare 2,3,5,6 and 7 bits per pixel formats. A FeedLoad function has been created to allow image data to be displayed as it is being transmitted across the net, providing Internet programmers with the fastest way to begin painting the client's screen. Lastly, examples for VB script and Java script are included to assist the programmer in incorporating images into their Internet applications. Refer to Working with the Internet.
- **Database Imaging.** LEADTOOLS has specific features designed for the imaging database developer, such as VB data binding, 32-bit ODBC, a customized OLE 2.0 in-place server, Load/Save memory, and Load/Save file offset. For details, refer to Working with Databases.
- **Printing**. LEADTOOLS performs all image processing necessary to print directly to any Windows supported printer, with the ability to print text and multiple images on the same page. Refer to Working with Other Input and Output.
- **High-level and low-level functions.** LEADTOOLS lets you choose the level of control you want. You can implement most LEADTOOLS features using high-level functions. With the DLLs, you can go to the lowest level, redirecting the standard input and output, and processing image data line-by-line or pixel-by-pixel.
- **Modular design.** LEADTOOLS provides need-specific DLLs that let you limit the size of your application by including only the applicable DLLs. For example, each supported file format has its own DLL; so you can leave out the formats that you do not wish to support.
- **Multi-thread support.** LEADTOOLS fully supports the development of multi-threaded applications for Windows NT and Windows 95.
- **Code examples.** The LEADTOOLS toolkits have many code examples in the online documentation and in working demonstration programs. You can copy code from the example programs and paste it directly into your application.

LEADTOOLS is offered as a DLL (32 & 16), an ActiveX (32 & 16), and a VBX. LEADTOOLS saves programmers time by providing source code examples and sample applications, which can be directly added to applications. LEAD's mature code is backed up by a 30 day money-back guarantee (US & Canada only) and FREE technical support is available via phone, fax, BBS, Internet or CompuServe. For the most current and complete technical details, and comprehensive lists of supported file formats, image processing filters, functions, methods, and properties, please visit LEAD's web site at http://www.leadtools.com, or call (704) 332-5532.

Working with the Internet

LEADTOOLS lets you do the following when working with the Internet:

- [ActiveX only] Use the ActiveX control's net-aware capabilities to load a bitmap asynchronously over the Internet. You can do this by specifying the URL in a single property.

  ActiveX: BitmapDataPath property, ReadyState property, ReadyStateChange event
- [ActiveX only] Use the LEADTOOLS Netscape Plug-In with Netscape Navigator or Microsoft Internet Explorer to load, display, and save any of the image formats that LEADTOOLS supports.
- [DLL and ActiveX only] Control how data is fed to a load routine.

  DLL: L_StartFeedLoad, L_FeedLoad, L_StopFeedLoad

  ActiveX: StartFeedLoad method, FeedLoad method, StopFeedLoad method
- [DLL and ActiveX only] Handle progressive reading and writing of JPEG and LEAD CMP files.

  DLL: L_GetExtFileOption, L_SetExtFileOption

  ActiveX: ProgressivePasses property
- [DLL and ActiveX only] Handle transparency, interlaced formatting, and text in GIF files.

  DLL: L_GetExtFileOption, L_SetExtFileOption

  ActiveX: BitmapEnableTransparency property, AnimationBackColor property, AnimationBitsPerPixel property, AnimationPalette property, AnimationWidth property, AnimationHeight property
- [DLL and ActiveX only] Create and maintain lists of bitmaps that are used for animation.

  DLL: L_CreateBitmapList, L_DeleteBitmapListItems, L_DestroyBitmapList, L_CopyBitmapList, L_GetBitmapListCount, L_GetBitmapListItem, L_InsertBitmapListItem, L_RemoveBitmapListItem, L_SetBitmapListItem, L_LoadBitmapList, L_SaveBitmapList, L_ColorResBitmapList, L_TranslateBitmapColor,

  ActiveX: BitmapList property, BitmapListCount property, InsertBitmapListItem method, BitmapListIndex property, DeleteBitmapListItems method, ColorResList method, BitmapDelay property, BitmapDisposalMethod property, BitmapEnableTransparency property, BitmapTransparentColor property
- [DLL and ActiveX only] Create and play animations, read and write animated GIF files, and read AVI files.

  DLL: L_LoadBitmapList, L_CreatePlayback, L_DestroyPlayback, L_ProcessPlayback, L_GetPlaybackState, L_GetPlaybackUpdateRect, L_ClearPlaybackUpdateRect, L_GetPlaybackIndex, L_SetPlaybackIndex, L_GetPlaybackDelay, L_CancelPlaybackWait, L_AppendPlayback, L_ValidatePlaybackLines, L_GetExtFileOption, L_SetExtFileOption

  ActiveX: InfoAnimation property, InfoAnimationHeight property, InfoAnimationWidth property, AnimationLoop property, AnimationEnable property, AutoAnimate property, AutoAnimationLoop property, Animate event, AnimationBackColor property, AnimationBitsPerPixel property, AnimationPalette property, AnimationWidth property, AnimationHeight property, BitmapLeft property, BitmapTop property
- Read and write most other image file formats that are transmitted on the Internet. Refer to Summary of All Supported Image File Formats.

Working with Files

LEADTOOLS lets you do the following when working with files on disk and in memory:

- Get information about the image in a file, and set information about how to handle the input.

  DLL: L_FileInfo, L_FileInfoMemory, L_ReadFileComment, L_GetPCDResolution, L_SetPCDResolution, L_EnumDimensions, L_GetDimension, L_SetDimension, L_GetWMFResolution, L_SetWMFResolution, L_SetLoadInfoCallback

  ActiveX: GetFileInfo method, InfoBits property, InfoCompress property, InfoFormat property, InfoHeight property, InfoPage property, InfoPCDRes property, InfoSizeDisk property, InfoSizeMem property, InfoWidth property, PCDRes property, PaintWhileLoad property, LoadInfo event, LoadInfoBits property, LoadInfoFlags property, LoadInfoFormat property, LoadInfoHeight property, LoadInfoOffset property, LoadInfoWidth property, LoadInfoXRes property, LoadInfoYRes property, EnumDimensions method, DimensionCount property, DimensionHeight property, DimensionWidth property, SetDimension method, GetDimension method, LoadUseViewPerspective property

  VBX: InfoPage, InfoFile, InfoBits, InfoCompress, InfoFormat, InfoHeight, InfoPCDResolution, InfoScaleHeight, InfoScaleWidth, InfoSizeDisk, InfoSizeMem, InfoWidth, PaintWhileLoad, FilePage, PCDResolution
- Set information about how to save an image to a file.

  DLL: L_GetComment, L_SetComment, L_GetExtFileOption, L_SetExtFileOption

  ActiveX: BitmapXRes property, BitmapYRes property, SaveInterlaced property

  VBX: BitmapXRes, BitmapYRes, SaveBits, SaveFormat, SaveQFactor, SaveMulti
- [DLL and ActiveX only] Get and set alpha-channel data in 16- and 32-bit files.

  DLL: L_GetBitmapAlpha, L_SetBitmapAlpha.

  ActiveX: BitmapAlpha property
- Load an image from a file into memory as a bitmap, or save a bitmap to a file. The image can be one of several in a multipage file. In a JFIF or LEAD CMP file, it can be either the main image or a thumbnail stored in the same file.

  DLL: L_LoadBitmap, L_LoadFile, L_LoadFileOffset, L_GetCompressFileStamp, L_LoadBitmapList, L_SaveBitmap, , L_SaveBitmapList, L_SaveFile, L_SaveFileOffset, L_CompressBitmapWithStamp

  Windows messages: L_BM_LOAD, L_BM_SAVE

  ActiveX: Load method, LoadStamp method, FilePage event, Save method, SaveWithStamp method

  VBX: File, LoadStamp, SaveFile, SaveBits, SaveFormat, SaveQFactor, SaveStamp, SaveStampBits, SaveStampHeight, SaveStampWidth
- Convert a file from one format to another.

  DLL: The load and save functions, plus L_FileConvert

  ActiveX: Implemented with load and save.

  VBX: Implemented with load and save.
- [DLL and ActiveX only] Load image files from memory and save them in memory. This lets you take advantage of file compression and memory speed, and it lets you exchange compressed data with other systems, such as database systems.

  DLL: L_FileInfoMemory, L_LoadBitmapMemory, L_LoadMemory, L_SaveBitmapMemory, L_DecompressBitmapMemory, L_DecompressMemory

  ActiveX: GetMemoryInfo method, LoadMemory method, SaveMemory method
- [DLL and ActiveX only] Control how data is fed to a load routine. This is useful when receiving images

over the Internet or when getting chunks of data from a database.

DLL: L_StartFeedLoad, L_FeedLoad, L_StopFeedLoad

ActiveX: StartFeedLoad method, FeedLoad method, StopFeedLoad method

- [DLL and ActiveX only] Read and write comments in all of the file formats that support comments, including TIFF, GIF, DICOM, and Exif files.

  DLL: L_GetComment, L_SetComment, L_ReadFileComment.

- [DLL and ActiveX only] Create your own TIFF tags (in addition to the standard comment tags).

  DLL: L_SetTag, L_GetTag, L_ReadFileTag, L_ReadFileTagMemory.

Working with Image Data

LEADTOOLS uses a structure called a bitmap handle to manage image data in memory. The bitmap handle has several fields, one of which is an internal pointer to the bitmap (the actual image data). In the DLL toolkit, your code references the bitmap handle; in the ActiveX or VBX your code references the LEAD control. (To access the image data with non-LEADTOOLS functions, you must copy or convert the bitmap to another kind of object, such as a DIB or DDB.)

To work with bitmaps and bitmap handles, you can do the following:

- Manipulate the bitmap storage.

  DLL: L_ClearBitmap, L_FreeBitmap, L_AccessBitmap, L_ReleaseBitmap

  Windows messages: L_BM_CLEAR, L_BM_FREE

  ActiveX: Bitmap property

  VBX: Bitmap
- Load a bitmap from a file or other input source. (Refer to Working with Files and Working with Other Input and Output.)
- Obtain or update the information in a bitmap handle. (For palette information, refer to Working with Palettes.)

  DLL: L_IsGrayScaleBitmap, and access the bitmap handle structure directly.

  Windows messages: L_BM_INFO, L_BM_GETBITMAP, L_BM_SETBITMAP

  ActiveX: BitmapBits property, BitmapHeight property, BitmapWidth property, BitmapXRes property, BitmapYRes property, IsGrayScale property

  VBX: BitmapXRes, BitmapYRes, BitmapHeight, BitmapScaleHeight, BitmapScaleWidth, BitmapWidth
- [DLL and ActiveX only] Implement transparency, and save a mask for transparency in a bitmap's alpha channel.

  DLL: L_CreateMaskFromBitmapRgn, L_SetBitmapRgnFromMask, L_GetBitmapAlpha, L_SetBitmapAlpha

  ActiveX: UseTransparentColor property, TransparentColor property, Transparent property, BitmapAlpha property, SetRgnColor method, GetRgnHandle method, SetRgnHandle method, PaintRgnOnly property, BitmapAlpha property
- Create a bitmap from scratch.

  DLL: L_CreateBitmap, L_InitBitmap, L_AllocateBitmap

  ActiveX: BitmapXRes property, BitmapYRes property, CreateBitmap method

  VBX: BitmapXRes, BitmapYRes, CreateBitmap, CreateBits, CreateHeight, CreateWidth
- Copy a bitmap to or from the clipboard.

  DLL: L_ClipboardReady, L_CopyFromClipboard, L_CopyToClipboard, L_CopyToClipboardExt

  Windows messages: L_BM_COPY, L_BM_PASTE

  ActiveX: Copy method, Paste method

  VBX: Copy, Paste
- Make a copy of a bitmap handle, or its bitmap, or both.

  DLL: L_CopyBitmap, L_CopyBitmapData, L_CopyBitmapHandle, L_CopyBitmapRect

  Windows messages: L_BM_COPYBITMAP

  ActiveX: Bitmap property

  VBX: Bitmap

- Convert or copy a bitmap to or from a device independent bitmap (DIB) or a device dependent bitmap (DDB).

  DLL: L_ConvertFromDDB, L_ConvertFromDIB, L_ConvertToDDB, L_ConvertToDIB, L_ChangeFromDDB, L_ChangeFromDIB, L_ChangeToDDB, L_ChangeToDIB

  ActiveX: GetDDB method, GetDIB method, SetDDB method, SetDIB method

  VBX: DDB, DIB
- [ActiveX only] Copy a bitmap to or from a LEAD OLE Server object.

  ActiveX: GetServerBitmap method, SetServerBitmap method
- [In Visual Basic only] Copy a bitmap to or from a picture object.

  ActiveX: GetPicture method, SetPicture method

  VBX: PictureCreate, Picture
- Resize a bitmap, with or without resampling.

  DLL: L_ResampleBitmap, L_ResizeBitmap, L_SizeBitmap

  Windows messages: L_BM_RESIZE, L_BM_RESAMPLE

  ActiveX: Size method

  VBX: ResizeType, ResizeHeight, ResizeWidth, Resize
- Trim a bitmap.

  DLL: L_CopyBitmapRect

  ActiveX: Trim method

  VBX: TrimHeight, TrimLeft, TrimTop, TrimWidth, Trim
- Change the bits per pixel or color order of a bitmap (with several possible dithering methods used for color reduction).

  DLL: L_ColorResBitmap, L_ColorResBitmapList, L_DefaultDithering

  Windows messages: L_BM_COLORRES

  ActiveX: ColorRes method

  VBX: BitChgDither, BitChgOrder, BitChgPalette, BitChgUserPal, BitmapBits
- Specify a bitmap's palette. (Refer to Working with Palettes.)
- Change a bitmap to grayscale.

  DLL: L_GrayScaleBitmap

  Windows messages: L_BM_GRAYSCALE

  ActiveX: Grayscale method

  VBX: GrayScale
- Create and merge color separations. A color separation is a set of 8-bit grayscale images, one for each plane of the color-space model. The supported color-space models are RGB, CMYK, CMY, HSV, and HLS.

  DLL: L_ColorSeparateBitmap, L_ColorMergeBitmap

  ActiveX: ColorSeparate method, ColorMerge method, ColorPlanes property

  VBX: ColorSeparate, ColorPlanes, ColorMerge
- [DLL and ActiveX only] Get the number of colors in a bitmap.

  DLL: L_GetBitmapColorCount

  ActiveX: GetColorCount method
- Modify the appearance of a bitmap image, as described in Working with the Images in Bitmaps.
- Display a bitmap image, with options for dithering, positioning, and zooming, as described in Working

with Displays.

Working with the Images in Bitmaps

Most of the LEADTOOLS functions that change the appearance of an image act on the bitmap (not just the display). This is an important distinction, because the changes become permanent when the bitmap is saved to a file.

To change the appearance of the image in a bitmap, you can do the following:

- Define the bitmap as a display surface where you can use Windows graphics device interface (GDI) functions for drawing or adding text.

  DLL: L_CreateLeadDC, L_DeleteLeadDC

  ActiveX: GetBitmapDC method, ReleaseBitmapDC method

  VBX: BitmapDC
- [Express editions only] Automatically straighten (deskew) a 1-bit image.

  DLL: L_DeskewBitmap

  Windows messages: L_BM_DESKEW

  ActiveX: Deskew method

  VBX: Deskew
- [Express editions only] Remove specks (despeckle) a 1-bit image.

  DLL: L_DespeckleBitmap

  Windows messages: L_BM_DESPECKLE

  ActiveX: Despeckle method

  VBX: Despeckle
- Automatically trim a bitmap to remove blank space around the edges.

  DLL: L_AutoTrimBitmap, L_GetAutoTrimRect, L_TrimBitmap.

  ActiveX: AutoTrim method
- Rotate the image. (Angles can be precise to 100th of a degree.)

  DLL: L_RotateBitmap, L_RotateBitmapFine

  Windows messages: L_BM_ROTATE

  ActiveX: Rotate method

  VBX: Rotate, RotateFine
- [Express only] Do fast rotation in 90-degree increments.

  DLL: L_RotateBitmapViewPerspective, L_PointFromBitmap, L_PointToBitmap, L_RectFromBitmap, L_RectToBitmap, L_ChangeBitmapViewPerspective

  ActiveX: FastRotate method
- Shear the image in the fashion of a parallelogram. (Angles are precise to 100th of a degree.)

  DLL: L_ShearBitmap

  Windows messages: L_BM_SHEAR

  ActiveX: Shear method

  VBX: ShearX, ShearY
- Change the orientation by flipping the image horizontally or vertically.

  DLL: L_FlipBitmap, L_ReverseBitmap

  Windows messages: L_BM_FLIP, L_BM_REVERSE

ActiveX: Flip method, Reverse method

VBX: Flip, Reverse

- Adjust colors and intensities as follows:
  - -- Change brightness using a flat scale.
  - -- Change brightness using gamma correction.
  - -- Change contrast.
  - -- Stretch the range of intensities.
  - -- Remap intensities using a lookup table.
  - -- Invert colors.
  - -- Change hue and saturation.
  - -- Histogram equalize.
  - -- Fill with a specified color.
  - -- [DLL and ActiveX only] Get and put colors of individual pixels.

  DLL: L_ChangeBitmapIntensity, L_GammaCorrectBitmap, L_ChangeBitmapContrast, L_HistoContrastBitmap, L_StretchBitmapIntensity, L_RemapBitmapIntensity, L_InvertBitmap, L_ChangeBitmapHue, L_ChangeBitmapSaturation, L_HistoEqualizeBitmap, L_FillBitmap, L_GetPixelColor, L_PutPixelColor

  Windows messages: L_BM_INTENSITY, L_BM_GAMMACORRECT, L_BM_CONTRAST, L_BM_INVERT, L_BM_HUE, L_BM_SATURATION, L_BM_HISTOEQ, L_BM_FILL

  ActiveX: Intensity method, GammaCorrect method, Contrast method, HistoContrast method, StretchIntensity method, RemapIntensity method, RemapTable property, Invert method, Hue method, Saturation method, HistoEqualize method, Fill method, Pixel property

  VBX: Intensity, GammaCorrect, Contrast, HistoContrast, StretchIntensity, RemapTable, RemapChannel, RemapIntensity, Invert, Hue, Saturation, HistoEq, Fill
- Apply the following kinds of conversions:
  - -- Halftone for display or printing.
  - -- Sharpen or blur.
  - -- Posterize, specifying the number of color planes.
  - -- Mosaic, specifying the tile size.
  - -- Emboss, specifying the lighting direction.
  - -- Soften an image using an average filter.
  - -- Reduce noise using a median filter.
  - -- Add noise in any or all color planes.
  - -- Apply an oil-painting effect.
  - -- Apply a solarization effect.

  DLL: L_HalfToneBitmap, L_HalftoneBitmapExt, L_SharpenBitmap, L_PosterizeBitmap, L_MosaicBitmap, L_EmbossBitmap, L_AverageFilterBitmap, L_MedianFilterBitmap, L_AddBitmapNoise, L_OilifyBitmap, L_SolarizeBitmap

  Windows messages: L_BM_HALFTONE, L_BM_SHARPEN, L_BM_POSTERIZE, L_BM_MOSAIC, L_BM_AVERAGE, L_BM_MEDIAN, L_BM_ADDNOISE

  ActiveX: Halftone method, Sharpen method, Posterize method, Mosaic method, Emboss method, Average method, Median method, AddNoise method, Oilify method, Solarize method

  VBX: HalftoneAngle, HalfTone, Sharpness, Posterize, Mosaic, FltEmbossDepth, FltEmboss, Average, Median, AddNoiseChannel, AddNoise
- Apply filters to do the following:
  - -- Detect ranges of intensity.
  - -- Detect edges using gradient or Laplacian edge detection.
  - -- Detect lines using Sobel, Prewitt, shift and difference, or line segment detection.

DLL: L_IntensityDetectBitmap, L_SpatialFilterBitmap

Windows messages: L_BM_INTENSITYDETECT, L_BM_SPATIALFLT

ActiveX: IntensityDetect method, SpatialFilter method

VBX: IntensityDetectHigh, IntensityDetectLow, IntensityDetect, FltGradient, FltLaplace, FltLineSeg, FltPrewitt, FltShiftDiff, FltSobel

- [Express editions only] Apply morphological (binary) filters to erode or dilate black objects.

  DLL: L_BinaryFilterBitmap, L_MaxFilterBitmap, L_MinFilterBitmap

  Windows messages: L_BM_BINARYFLT

  ActiveX: BinaryFilter method, MaxFilter method, MinFilter method

  VBX: FltDilation, FltErosion

- [DLL only] Implement your own spatial filters.

  DLL: L_SpatialFilterBitmap

- Combine images using boolean and arithmetic operators, and color masking. (This is useful for combining filtered images with originals.)

  DLL: L_CombineBitmap

  ActiveX: Combine method

  VBX: CombineDstLeft, CombineDstTop, CombineHeight, CombineOp, CombineSrcLeft, CombineSrcTop, CombineWidth, Combine

- Combine two images so that one appears to be an underlying texture of the other.

  DLL: L_UnderlayBitmap

  ActiveX: Underlay method

  VBX: UnderlayType, Underlay

- [DLL and ActiveX only] Use low-level functions to change parts of an image as follows:

  -- Get and put rows of image data.

  -- Get and put parts of rows.

  -- Get and put the colors of individual pixels.

  DLL: L_GetBitmapRow, L_PutBitmapRow, L_GetBitmapRowCol, L_PutBitmapRowCol, L_GetPixelColor, L_PutPixelColor, L_GetBitmapRowCompressed, L_PutBitmapRowCompressed, L_CompressRows, L_ExpandRows

  ActiveX: Pixel property

- Process a region within a bitmap, as described in Working with Regions.

Working with Palettes

Images that are less than 16 bits per pixel require a palette, which is an array of color values. The value of each pixel in one of these images is an index into the palette. (Images that are 16 bits per pixel or higher store the color values directly in the image, as pixel values.)

The number of colors in a palette is limited to the highest possible index value. For example, if an image is 8 bits per pixel, it can have a palette of 256 colors, which is the number of different colors that can be referenced by using the pixel values as 8-bit indexes.

If an image file is less than 16 bits per pixel, the file contains a palette, and when you load the image into a bitmap, the palette is also loaded into the bitmap handle.

Whenever you reduce a bitmap to less than 16 bits per pixel, a palette must be specified or created. You can do any of the following:

- Let LEADTOOLS create a palette that is optimized for the particular bitmap or list of bitmaps. (As an option, this palette can include the system identity palette.)

  DLL: The load functions, plus L_ColorResBitmap, L_ColorResBitmapList

  Windows messages: L_BM_COLORRES

  ActiveX: ColorRes method, UserPalette property

  VBX: BitChgPalette, BitmapBits
- Specify a fixed palette, which is the same for all images that use it. The palette can be LEAD's fixed palette or the Netscape fixed palette.

  DLL: The load functions, plus L_GetFixedPalette, L_ColorResBitmap, L_GetDisplayMode, L_SetDisplayMode

  Windows messages: L_BM_COLORRES

  ActiveX: ColorRes method, PaintPalette property

  VBX: BitChgPalette, BitmapBits, PaintPalette
- Specify a palette of any size that you have created. In creating your own palette, you can let LEADTOOLS do part of the work. For example, you can fill in part of the palette, and let LEADTOOLS fill the rest with optimized colors.

  DLL: L_CreatePaintPalette

  ActiveX: UserPalette property

  VBX: BitChgPalette, BitChgUserPal, BitmapBits
- Modify the colors in the bitmap palette.

  DLL: L_GetBitmapColors, L_PutBitmapColors, L_TranslateBitmapColor

  Windows messages: L_BM_GETPALETTE

  ActiveX: BitmapPalette property

  VBX: BitmapPalette
- [DLL and ActiveX only] Copy a palette from a device context.

  DLL: L_GetFixedPalette, L_DupPalette

  ActiveX: GetPalette method
- [DLL only] Use a table to increase the efficiency of the palette that you define.

  DLL: L_CreateUserMatchTable, L_SetUserMatchTable, L_FreeUserMatchTable

Working with Displays

When painting images on any video device, you can do the following:

- Get and set properties that control the behavior of the LEAD control.

  Windows messages: L_BM_SETSTYLE

  ActiveX: Standard control properties, plus Appearance property, OLEDropAllowed property, Transparent property, PaintSizeMode property, PaintSizeUseDPI property, PaintZoomFactor property, AutoSize property, AutoRepaint property, AutoScroll property, AutoSetRects property, BackColor property, BackErase property, BorderStyle property, ForePalette property, MouseIcon property, MousePointer property, Enabled property, hWnd property

  VBX: Align, AutoRepaint, AutoScroll, BackColor, BackErase, BorderStyle, DragIcon, DragMode, Enabled, MousePointer, hWnd, HelpContextID, Index, Name, Parent, PropertySet, TabIndex, TabStop, Tag, Visible
- [DLL and ActiveX only] Control the brightness, contrast, and gamma correction when painting (without affecting the bitmap).

  DLL: L_GetPaintContrast, L_SetPaintContrast, L_GetPaintGamma, L_SetPaintGamma, L_GetPaintIntensity, L_SetPaintIntensity

  ActiveX: PaintGamma property, PaintIntensity property, PaintContrast property
- Get the client area of the LEAD control.

  DLL: Implemented using Windows functions.

  ActiveX: ScaleMode property, ScaleHeight property, ScaleLeft property, ScaleTop property, ScaleWidth property

  VBX: Height, Left, Top, Width
- Define the client area as a display surface where you can use Windows graphics device interface (GDI) functions for drawing or adding text.

  DLL: Implemented using Windows functions.

  ActiveX: GetClientDC method, ReleaseClientDC method

  VBX: hDC
- [ActiveX only] Display a rubberband to mark an area.

  ActiveX: AutoRubberBand property, RubberBand event, SetRubberBandRect method, RubberBandVisible property, RubberBandHeight property, RubberBandLeft property, RubberBandTop property, RubberBandWidth property
- [ActiveX only] Zoom in on a selected area.

  ActiveX: ZoomToRect method
- Force repainting of the image in a LEAD control.

  DLL: Implemented using Windows functions.

  Windows messages: L_BM_BITMAPCHANGED

  ActiveX: ForceRepaint method

  VBX: Repaint
- Scale and position the image. This lets you zoom in or out, fit the image to a window, and position the image in a window.

  DLL: L_PaintDC, L_PaintDCBuffer, L_PaintDCEffect

  Windows messages: L_BM_SETIMAGERECT, L_BM_SETZOOM, L_BM_FOCUSEDZOOM, L_BM_SETBITMAPCLIP

  ActiveX: SetDstRect method, SetSrcRect method, DstHeight property, DstLeft property, DstTop property, DstWidth property, SrcHeight property, SrcLeft property, SrcTop property, SrcWidth property

VBX: ImageHeight, ImageLeft, ImageTop, ImageWidth

- Limit the area to be painted by specifying the source and destination cropping areas.

DLL: L_PaintDC, L_PaintDCBuffer, L_PaintDCEffect

ActiveX: SetDstClipRect method, SetSrcClipRect method, DstClipHeight property, DstClipLeft property, DstClipTop property, DstClipWidth property, SrcClipHeight property, SrcClipLeft property, SrcClipTop property, SrcClipWidth property

VBX: CropHeight, CropLeft, CropTop, CropWidth, Crop

- Apply speciale effects, commonly used in slide presentations, as described in Working with Special Effects, Shapes, and Text.
- Apply ROP codes, which determine how to interact with the existing image on the screen.

DLL: L_PaintDC, L_PaintDCBuffer, L_PaintDCEffect

ActiveX: PaintROP3 property

VBX: PaintRop3

- [DLL only] Use a buffer as the source to paint from.

DLL: L_PaintDCBuffer

- [DLL only] Specify an option for fast painting (without device error checking).

DLL: L_GetDisplayMode, L_SetDisplayMode

If the display mode is 256 colors or less, you can do the following:

- Use the palette in the bitmap handle or use a fixed palette. You can specify LEAD's fixed palette or the Netscape fixed palette.

DLL: L_GetDisplayMode, L_SetDisplayMode

ActiveX: PaintPalette property

VBX: PaintPalette

- Specify the dithering method, for images that have more bits per pixel than the current video mode. The following are possible options:

  -- Normal dithering (using error diffusion).
  -- Ordered dithering, which is faster but less accurate than normal.
  -- [DLL only] No dithering, which relies on the display device for color reduction.

DLL: L_GetDisplayMode, L_SetDisplayMode

ActiveX: PaintDither property

VBX: PaintDither

For displaying 1-bit (black-and-white) images, you can do the following to enhance the quality of display:

- [Express editions only] Specify a scale-to-gray option, which increases the clarity of the 1-bit images when they are scaled (zoomed in or zoomed out).

DLL: L_GetDisplayMode, L_SetDisplayMode

ActiveX: BitonalScaling property

VBX: PaintBwAsGray, BitonalScaling

- [Express editions only] Specify a favor-black option, which prevents loss of details, such as fine lines, when an image is scaled down (zoomed out).

DLL: L_GetDisplayMode, L_SetDisplayMode

ActiveX: BitonalScaling property

VBX: BitonalScaling

Working with Special Effects, Shapes, and Text

You can apply hundreds of different paint effects, with control over the speed and granularity of each effect. You can also draw plain or three-dimensional text and graphic shapes on the screen or on the bitmap. Specifically, you can do the following:

- Specify an effect that is applied when painting the image from a bitmap.

  DLL: L_EfxEffectBlt, L_EfxPaintBitmap, L_PaintDCEffect

  ActiveX: PaintEffect property, PaintMaxPasses property, PaintPass property, PaintNotificationDelay property, PaintNotification event, WandColor property, WandSize property

  VBX: Effect

- [DLL and ActiveX only] Specify a transition between images, which is an effect that paints a pattern or gradient color (instead of the image from a bitmap).

  DLL: L_EfxPaintTransition

  ActiveX: TransitionStyle property, TransitionEffect property, PaintMaxPasses property, PaintPass property, PaintNotificationDelay property, PaintNotification event, WandColor property, WandSize property

- [DLL and ActiveX only] Control the speed, granularity, multiple paint passes, transparent color, and wand usage when painting images or transitions.

  DLL: L_EfxEffectBlt, L_EfxPaintBitmap, L_EfxPaintTransition

  ActiveX: EffectDelay property, EffectGrain property, UseTransparentColor property, TransparentColor property

- [DLL and ActiveX only] Specify a transparent color and paint a control with a transparent background.

  DLL: L_EfxEffectBlt, L_EfxPaintBitmap, L_EfxPaintTransition

  ActiveX: UseTransparentColor property, TransparentColor property, Transparent property

When drawing shapes or text objects on the screen or on the bitmap, you can do the following:

- [DLL and ActiveX only] Paint a rectangular area that contains a pattern or gradient color. This is useful for creating a bitmap that can be used as the background of a shape object or the foreground of a text object.

  DLL: L_EfxGradientFillRect, L_EfxPatternFillRect

  ActiveX: Accomplished by drawing a rectangular shape on a bitmap.

- [DLL and ActiveX only] Draw a shape with a three-dimensional appearance. The shape can be one of 17 variations of rectangles, triangles, circles, and so forth, or it can be created from an existing Windows region.

  DLL: L_EfxDraw3dShape

  ActiveX: DrawPersistence property, DrawShape method, ShapeLeft property, ShapeTop property, ShapeWidth property, ShapeHeight property, ShapeBackgroundStyle property, ShapeBorderColor property, ShapeBorderStyle property, ShapeBorderThickness property, ShapeInnerBandHiliteColor property, ShapeInnerBandShadowColor property, ShapeInnerBandStyle property, ShapeInnerBandThickness property, ShapeOuterBandHiliteColor property, ShapeOuterBandShadowColor property, ShapeOuterBandStyle property, ShapeOuterBandThickness property, ShadowColor property, ShadowXDepth property, ShadowYDepth property, GradientStyle property, GradientStartColor property, GradientEndColor property, GradientSteps property, PatternStyle property, PatternBackColor property, PatternForeColor property

- [DLL and ActiveX only] Draw a text object with a three-dimensional appearance. The text can be rotated with precision to a tenth of a degree.

  DLL: L_EfxDraw3dText, L_EfxDrawRotated3dText

ActiveX: DrawPersistence property, DrawText method, DrawTextOut method, TextLeft property, TextTop property, TextWidth property, TextHeight property, TextAlign property, Font property, DrawFontColor property, TextStyle property, TextWordWrap property, TextAngle property, TextHiliteColor property, DrawTextHeight method, DrawTextWidth method, ShadowColor property, ShadowXDepth property, ShadowYDepth property, GradientStyle property, GradientStartColor property, GradientEndColor property, GradientSteps property, PatternStyle property, PatternBackColor property, PatternForeColor property

- [DLL and ActiveX only] Specify a bitmap to use as the background of a shape, or as the foreground of a text object.

  DLL: L_EfxDraw3dShape, L_EfxDraw3dText, L_EfxDrawRotated3dText

  ActiveX: DrawShape method, DrawText method, BackgroundImageLeft property, BackgroundImageTop property, BackgroundImageWidth property, BackgroundImageHeight property

- [DLL and ActiveX only] Draw a frame with a three-dimensional appearance. (In the ActiveX the frame is applied as a border of the control.)

  DLL: L_EfxDrawFrame

  ActiveX: FrameThickness property, FrameColor property, FrameInnerStyle property, FrameInnerThickness property, FrameInnerHiliteColor property, FrameInnerShadowColor property, FrameOuterStyle property, FrameOuterThickness property, FrameOuterHiliteColor property, FrameOuterShadowColor property

- Draw a simple Windows-style line, rectangle, or ellipse, with standard options for the draw-mode, pen, and fill.

  DLL: Implemented with Windows GDI functions.

  ActiveX: DrawPersistence property, DrawLine method, DrawEllipse method, DrawRectangle method, DrawFillColor property, DrawFillStyle property, DrawMode property, DrawPenColor property, DrawPenStyle property, DrawPenWidth property

  VBX: BitmapDC with Windows GDI functions

Working with Regions

You can create and maintain shapes, position a shape on a bitmap to define a region to be processed, and use image-processing functions to modify the region. Specifically, you can do the following.

- [ActiveX only] Automate the creation and display of a region.

  ActiveX: RgnMarkingMode property, RgnChange event, RgnFrameType property
- [DLL and ActiveX only] Copy and paste a region using a floating bitmap.

  DLL: Implemented using lower-level functions.

  ActiveX: Floater property, FloaterVisible property, IsPtInFloater method, SetFloaterDstRect method, SetFloaterDstClipRect method, FloaterDstHeight property, FloaterDstLeft property, FloaterDstTop property, FloaterDstWidth property, FloaterDstClipHeight property, FloaterDstClipLeft property, FloaterDstClipTop property, FloaterDstClipWidth property, FloaterHeight property, FloaterWidth property, GetFloaterHandle method
- [DLL and ActiveX only] Create a shape using a rectangle, ellipse, polygon, or specified color.

  DLL: L_SetBitmapRgnColor, L_SetBitmapRgnEllipse, L_SetBitmapRgnPolygon, L_SetBitmapRgnRect, L_SetBitmapRgnRoundRect

  ActiveX: SetRgnEllipse method, SetRgnRect method, SetRgnRoundRect method, SetRgnColor method, PolygonSize property, PolygonX property, PolygonY property, SetRgnPolygon method
- [DLL and ActiveX only] Size and position the region on the bitmap.

  DLL: L_GetBitmapRgnBounds, L_OffsetBitmapRgn

  ActiveX: RgnHeight property, RgnLeft property, RgnTop property, RgnWidth property, OffsetRgn method
- [DLL and ActiveX only] Manage the bitmap's region.

  DLL: L_FrameBitmapRgn, L_BitmapHasRgn, L_FreeBitmapRgn, L_GetBitmapRgnArea, L_IsPtInBitmapRgn

  ActiveX: HasRgn property, IsPtInRgn method, FreeRgn method, GetRgnArea method, RepaintRect method, RgnFrameType property
- [DLL and ActiveX only] Copy to and from standard Windows regions.

  DLL: L_GetBitmapRgnHandle, L_SetBitmapRgnHandle

  ActiveX: GetRgnHandle method, GetFloaterHandle method, SetRgnHandle method, DeleteRgnHandle method
- [DLL and ActiveX only] Copy a region to or from a mask image.

  DLL: L_CreateMaskFromBitmapRgn, L_SetBitmapRgnFromMask

  ActiveX: GetRgnHandle method, SetRgnHandle method, DeleteRgnHandle method
- [DLL and ActiveX only] Paint only the bitmap's region.

  DLL: L_PaintRgnDC, L_PaintRgnDCBuffer, L_PaintRgnDCEffect

  ActiveX: PaintRgnOnly property
- [DLL and ActiveX only] Process the region using any of the appropriate functions described in Working with the Images in Bitmaps.

Working with Annotations [Express Editions Only]

You can create a wide range annotation objects, group them, store them in files, apply them to bitmaps, size and position them, and rotate them. Specifically, you can do the following.

- [DLL and ActiveX only] Create any of the following objects: line, arrow, rectangle, ellipse, polygon, freehand line, text, highlighter, hot spot, note, stamp, redaction, button, or audio clip.

  DLL: L_AnnDraw, L_AnnCreate, L_AnnCreateContainer, L_AnnCreateItem, L_AnnCopy, L_AnnDestroy, L_AnnResize, L_AnnSetVisible, L_AnnGetVisible, L_AnnSetSelected, L_AnnGetSelected, L_AnnSetWnd, L_AnnGetWnd, L_AnnGetType, L_AnnGetActiveState, L_AnnSetActiveState, L_AnnSetText, L_AnnGetText, L_AnnGetTextLen, L_AnnSetFontName, L_AnnGetFontName, L_AnnGetFontNameLen, L_AnnSetFontSize, L_AnnGetFontSize, L_AnnSetFontUnderline, L_AnnGetFontUnderline, L_AnnSetFontStrikeThrough, L_AnnGetFontStrikeThrough, L_AnnSetFontItalic, L_AnnGetFontItalic, L_AnnSetFontBold, L_AnnGetFontBold, L_AnnSetForeColor, L_AnnGetForeColor, L_AnnSetBackColor, L_AnnGetBackColor, L_AnnSetLineWidth, L_AnnGetLineWidth, L_AnnSetLineStyle, L_AnnGetLineStyle, L_AnnSetFillPattern, L_AnnGetFillPattern, L_AnnSetFillMode, L_AnnGetFillMode, L_AnnGetPolyFillMode, L_AnnSetPolyFillMode, L_AnnSetBitmap, L_AnnGetBitmap, L_AnnSetTag, L_AnnGetTag

  ActiveX: Automated support, plus AnnSetTag method, AnnGetTag method, AnnGetType method, AnnContainer property
- [DLL and ActiveX only] Implement automated annotation support.

  DLL: L_AnnCreateContainer, L_AnnCreate, L_AnnSetAutoContainer, L_AnnGetAutoContainer, L_AnnSetDpiX, L_AnnSetDpiY, L_AnnGetDpiX, L_AnnGetDpiY, L_AnnSetTool, L_AnnGetTool, L_AnnGetUserMode, L_AnnSetUserMode, L_AnnSetUndoDepth, L_AnnUndo, L_AnnCreateToolBar, L_AnnGetToolBarChecked, L_AnnSetToolBarChecked, L_AnnGetToolBarButtonVisible, L_AnnSetToolBarButtonVisible, L_AnnSelectRect, L_AnnGetSelectCount, L_AnnSelectPoint, L_AnnHitTest, L_AnnDefine, L_AnnGetBoundingRect, L_AnnGetSelectRect, L_AnnGetAutoMenuEnable, L_AnnSetAutoMenuEnable, L_AnnGetAutoDrawEnable, L_AnnSetAutoDrawEnable

  ActiveX: AnnUserMode property, AnnTool property, AnnClicked event, AnnCreate event, AnnDestroy event, AnnDrawn event, AnnChange event, AnnAutomation property, AnnAutoMenuEnable property, AnnAutoDrawEnable property, AnnGetSelectCount method, AnnGetSelectList method, AnnSelect event, AnnAutoRotate property
- [DLL and ActiveX only] Modify text strings used in automated annotation support.

  DLL: L_AnnGetAutoText, L_AnnGetAutoTextLen, L_AnnSetAutoText

  ActiveX: AnnGetAutoText method, AnnSetAutoText method
- [DLL and ActiveX only] Scale and position a group to fit the displayed bitmap.

  DLL: L_AnnSetScalarX, L_AnnSetScalarY, L_AnnGetScalarX, L_AnnGetScalarY, L_AnnSetOffsetX, L_AnnSetOffsetY, L_AnnGetOffsetX

  ActiveX: Implemented with automated annotation.
- [DLL and ActiveX only] Group and ungroup objects.

  DLL: L_AnnInsert, L_AnnRemove, L_AnnGetContainer, L_AnnGetTopContainer, L_AnnEnumerate, L_AnnGetItem

  ActiveX: Implemented with automated annotation.
- [DLL and ActiveX only] Apply an object or group to a bitmap.

  DLL: L_AnnRealize

  ActiveX: AnnRealize method
- [DLL and ActiveX only] Size and position an object or group.

DLL: L_AnnBringToFront, L_AnnSendToBack, L_AnnSetRect, L_AnnGetRect, L_AnnSetPoints, L_AnnGetPointCount, L_AnnGetPoints

ActiveX: Automated annotation, plus AnnGetPointCount method, AnnGetPointX method, AnnGetPointY method

- [DLL and ActiveX only] Rotate an object or group.

  DLL: L_AnnRotate, L_AnnMove

  ActiveX: Automated annotation, plus AnnRotate method, AnnAutoRotate property, AnnMove method
- [DLL and ActiveX only] Flip, reverse, or shear an object or group.

  DLL: L_AnnFlip, L_AnnReverse, L_AnnSetPoints

  ActiveX: Automated annotation, plus AnnFlip method, AnnReverse method
- [DLL and ActiveX only] Save an object or group on the Windows clipboard.

  DLL: L_AnnCopyFromClipboard, L_AnnCopyToClipboard, L_AnnCutToClipboard, L_AnnClipboardReady

  ActiveX: AnnCopy method, AnnPasteReady property, AnnPaste method
- [DLL and ActiveX only] Print an object or group.

  DLL: L_AnnPrint

  ActiveX: Implemented with automated annotation.
- [DLL and ActiveX only] Save an object or group in a file.

  DLL: L_AnnSave, L_AnnSaveMemory, L_AnnSaveOffset, L_AnnLoad, L_AnnLoadMemory, L_AnnLoadOffset

  ActiveX: AnnLoad method, AnnSave method, AnnDataPath property, AnnLoadMemory method, AnnSaveMemory method
- [DLL only] Save an object or group as a file in memory. (This is useful for database support.)

  DLL: L_AnnSaveMemory, L_AnnLoadMemory
- [ActiveX only] Save an object or group in a database using Visual Basic data binding.

  ActiveX: Implemented with automated annotation.

Working with Databases

You can maintain images in a database, using any valid file format to store an image in a long binary field. You can do so as follows:

- [ActiveX32 only] Use Open Database Connectivity (ODBC).

  ActiveX: dbOpen method, dbClose method, dbMove method, dbMoveFirst method, dbMoveLast method, dbMoveNext method, dbMovePrev method, dbLoadBits property, dbAddNew method, dbEdit method, dbLockingMode property, dbUpdate method, dbDelete method, dbRequery method, dbCanAppend property, dbCanRestart property, dbCanScroll property, dbCanUpdate property, dbCurrentRecord property, dbEditMode property, dbIsBOF property, dbIsDeleted property, dbIsEOF property, dbIsOpen property, dbRecordCount property
- [ActiveX and VBX with Visual Basic only] Use data binding.

  ActiveX: DataSource property, DataField property, DataSaveBits property, DataSaveFormat property, DataSaveQuality property, DataLoadBits property, DataChanged property

  VBX: DataChanged, DataField, DataSource
- [DLL and ActiveX only] Use functions to read files from memory and write files to memory.

  DLL: L_FileInfoMemory, L_LoadBitmapMemory, L_LoadMemory, L_SaveBitmapMemory, L_DecompressBitmapMemory, L_DecompressMemory, L_StartFeedLoad, L_FeedLoad, L_StopFeedLoad, L_AnnSaveMemory, L_AnnLoadMemory

  ActiveX: GetMemoryInfo method, LoadMemory method, SaveMemory method, StartFeedLoad method, FeedLoad method, StopFeedLoad method

Working with Other Input and Output

For additional input and output, you can do the following:

- Load a bitmap from a TWAIN device. TWAIN is an industry standard interface used with scanners, digital cameras, and frame grabbers. LEADTOOLS supports device selection and multipage scanning.

  DLL: L_TwainAcquire, L_TwainAcquireExt, L_TwainEnumSources, L_TwainSelect, L_TwainSetProps

  Windows messages: L_BM_TWAINSELECT, L_BM_TWAINACQUIRE

  ActiveX: TwainAcquire method, TwainSelect method, EnableTwainEvent property, TwainPage event, TwainFlags property, TwainEnumSources method, TwainSourceCount property, TwainSourceList property, TwainAppAuthor property, TwainAppFamily property, TwainAppName property, TwainBits property, TwainFrameHeight property, TwainFrameLeft property, TwainFrameTop property, TwainFrameWidth property, TwainMaxPages property, TwainPixelType property, TwainRes property, TwainSourceName property, TwainRealize method

  VBX: EnableTwainMulti, TwainAcquire, TwainSelect, TwainFlags, TwainImage event
- Print one or more bitmaps on a page, positioning, scaling, and cropping the images as necessary.

  DLL: L_PrintBitmap, L_PrintBitmapExt, L_PrintBitmapFast

  Windows messages: L_BM_PRINT

  ActiveX: Render method, PrintStart method, PrintEnd method

  VBX: PrintHeight, PrintLeft, PrintTop, PrintWidth, PrintCrop, PrintCropHeight, PrintCropLeft, PrintCropTop, PrintCropWidth, PrintBitmap
- Capture an image from a screen.

  DLL: L_ScreenCaptureBitmap

  ActiveX: Capture method

  VBX: CaptureHeight, CaptureLeft, CaptureTop, CaptureWidth, Capture
- [VBX only] Share data with other applications using DDE links.

  VBX: LinkItem, LinkMode, LinkTimeout, LinkTopic
- [DLL only] Redirect the standard input and output functions so that you can replace the low-level file I/O with your own input and output streams.

  DLL: L_RedirectIO

Working with Events and Callbacks

To detect user interrupts or add other processing to the LEADTOOLS internal loops, you can write code for events in the ActiveX and VBX, and for callback functions in the DLL. Specifically, you can do the following:

- Provide a generic function for detecting user interrupts and updating a status bar.

  DLL: L_SetStatusCallBack

  ActiveX: EnableProgressEvent property, ProgressStatus event

  VBX: EnableStatus, Status event
- Handle multipage scanning.

  DLL: L_TwainAcquireExt

  ActiveX: EnableTwainEvent property, TwainPage event

  VBX properties and events: EnableTwainMulti, TwainImage event
- [DLL only] Provide other function-specific callbacks.

  DLL: L_ColorResBitmap, L_DecompressMemory, L_LoadFile, L_LoadFileOffset, L_LoadMemory, L_SaveFile, L_SaveFile, L_SaveFileOffset, L_SetLoadInfoCallback, L_StartCompressBuffer, L_TwainEnumSources
- [ActiveX only] Detect changes in regions.

  ActiveX: RgnChange event
- [ActiveX only] Turn on or turn off the generation of error events.

  ActiveX: EnableMethodErrors property
- Detect standard events.

  DLL: Implemented using Windows messages.

  Windows messages: L_BN_CHANGE, L_BN_CLICKED, L_BN_DBLCLK, L_BN_HSCROLL, L_BN_KILLFOCUS, L_BN_PAINTED, L_BN_SETFOCUS, L_BN_UPDATE, L_BN_VSCROLL

  ActiveX: Change event, Click event, DblClick event, KeyDown event, KeyPress event, KeyUp event, MouseDown event, MouseMove event, MouseUp event, Paint event, Resize event, Scroll event

  VBX events: Change, Click, DblClick, DragDrop, DragOver, GotFocus, KeyDown, KeyPress, KeyUp, LinkChange, LinkClose, LinkError, LinkNotify, LinkOpen, LostFocus, MouseDown, MouseMove, MouseUp, Paint, Resize, Scroll

Working with Raw Image Data [DLL Only]

LEADTOOLS provides low-level functions that you can use to process image data in a buffer. You can use these functions to create your own image processing functions. For example, you can write a callback function that paints an image while it is loading by processing and painting one line at a time.

To work with raw image data, you can do the following:

- Paint a buffer to a device context.

  DLL: L_PaintDCBuffer
- Change the bits per pixel.

  DLL: L_ConvertBuffer, L_StartDithering, L_DitherLine, L_StopDithering
- Change the color order.

  DLL: L_ConvertBuffer
- Compress or decompress the data when working with LEAD CMP or JPEG files.

  DLL: L_StartCompressBuffer, L_CompressBuffer, L_EndCompressBuffer, L_DecompressMemory
- Process the data for buffered resizing of the bitmap.

  DLL: L_StartResize, L_Resize, L_StopResize
- Change the color-space model to or from any of the following: RGB, YUV, CMYK, CMY, YIQ, HSV, or HLS.

  DLL: L_ConvertColorSpace
- Modify the allocated height of a bitmap for mid-stream adjustments.

  DLL: L_ChangeBitmapHeight

Working with Image Descriptions and Other Information

LEADTOOLS lets you get information as follows:

- You can get the information you need about an image in a file, such as its format, size, bits per pixel, compression method, size on disk, and size in memory. Refer to Working with Files.
- You can get information about a bitmap in memory, such as its width, height, bits per pixel, memory allocation, and palette usage. Refer to Working with Image Data.
- You can create a histogram that maps the frequency of occurrence of each color in a bitmap.

  DLL: L_GetBitmapHistogram

  ActiveX: HistogramTable property, GetHistogram method

  VBX: Histogram, HistogramAcquire, HistogramChannel
- You can get information about the current version of your toolkit.

  DLL: L_VersionInfo

  ActiveX: AboutBox method, VersionDate property, VersionLevel property, VersionMajor property, VersionMinor property, VersionProduct property, VersionTime property
- You can get and set information about optional features (such as GIF or TIFF LZW file support or Express capabilities) that you are licensed to use.

  DLL: L_IsSupportLocked, L_UnlockSupport

  ActiveX: IsSupportLocked method, UnlockSupport method

  VBX: SupportType, IsSupportLocked, UnlockSupport

Express Editions

Refer to the **LEADTOOLS Pro Express** product description for a summary of features and performance enhancements that are available only in that product.

LZW License Instructions

Warning

Use of this software for providing LZW capability for any purpose is not authorized unless user first enters into a license agreement with Unisys under U.S. Patent No. 4,558,302 and foreign counterparts. For information concerning licensing, please contact:

East coast:	Unisys Corporation, Welch Licensing Department - C1SW19 Mark Starr, Staff VP and General Patent and Technology Counsel Township Line & Union Meeting Roads P.O. Box 500 Blue Bell, PA 19424 Phone: (215) 986-4411 FAX: (215) 986-3090
West coast:	Unisys Corporation Nathan Cass, West Coast Licensing Department 2049 Century Park East, Suite 310 Los Angeles, CA 90067 Phone: (310) 284-9631 FAX: (310) 284-9634

To implement LZW capability, you must do the following:

1. Obtain an LZW patent license or a written waiver from Unisys Corporation.
2. FAX the signature page of your LZW license to LEAD at (704) 372-8161. In the FAX, describe which LEADTOOLS product you have, and tell LEAD how to send you the LZW keyword (E-mail, FAX, or phone).
3. Use the LZW keyword with a LEADTOOLS unlock-support function to implement LZW support in your application.

Introduction To Image Processing with LEADTOOLS

The following topics provide general information about image processing, and where appropriate, they describe the related LEADTOOLS capabilities. These topics build up concepts and terminology; so if you are new to image processing, it is best to read them in order.

Bits Per Pixel and Related Ideas

Color Resolution and Dithering

Palette Handling

Image Manipulation and Analysis

Image Display

Bitmaps in Memory and in Files

DIBs, DDBs, and the Clipboard

Printing

TWAIN Input

Database Interaction

Data Transfer

Animation

Introduction: Bits Per Pixel and Related Ideas

The terminology for image formats can be confusing because there are often several ways of describing the same format. This topic explains what the terms mean.

If an image is 24 bits per pixel, it is also called a 24-bit image, a true color image, or a 16M color image. Sixteen million is roughly the number of different colors that can be represented by 24 bits, where there are 8 bits for each of the red, green, and blue (RGB) values.

A 32-bit image is a specialized true-color format used in image files, where the extra byte carries information that is either converted or ignored when the file is loaded. The extra byte is used for an additional color plane in CMYK files, which are specialized files for color printing. In that case, LEADTOOLS, by default, converts the values to 24-bit RGB values when loading the image. The additional byte may also be used for an Alpha channel, which carries extra information such as a transparency indicator.

If an image is 16 bits per pixel, it is also called a 16-bit image, a high color image, or a 32K color image. Thirty-two thousand is roughly the number of different colors that can be represented by 16 bits, where there are 5 bits for each of the red, green, and blue values. (Devices that specify 64K color support are also referring to 16-bit images, but they are counting the left-over bit.)

If an image is 8 bits per pixel, it is also called an 8-bit image or a 256-color image. Two hundred fifty-six is the number of different colors that can be achieved by using the image data as 8-bit indexes to an array of colors called a palette.

If an image is 4 bits per pixel, it is also called a 4-bit image or a 16-color image. Sixteen is the number of different colors that can be achieved by using the image data as 4-bit indexes to a palette.

If an image is 1 bit per pixel, it is also called a 1-bit image, a black and white image, a 2-color image, or a bitonal image. Two is the number of different colors that can be achieved by using the image data as 1-bit indexes to a palette. The palette can contain colors other than black and white, although black and white are most common.

If an image is *grayscale*, its red, green, and blue values are all the same, and the values are incremented from the lowest to the highest. For example, an 8-bit grayscale image has 256 shades of gray, with values from 0 to 255. LEADTOOLS supports 4- and 8-bit grayscale.

Most of this same terminology is applied to video cards. For example, an 8-bit card is one that is capable of displaying 256 colors.

Introduction: Color Resolution and Dithering

Color resolution (also called color depth) refers to the number of possible colors in an image, as determined by the bits-per-pixel. If an image in your computer is loaded from a FAX scanner or received as a FAX transmission, it is a black-and-white (1-bit) image. If it is loaded from a color scanner or a JPEG file, it can have 16 million colors. If it is loaded from a GIF file, it is likely to have 256 colors. There are many possibilities, ranging from two colors to 16 million.

LEADTOOLS lets you manipulate an image and display it on any Windows-compatible device, regardless of the image's color resolution. Therefore, in most cases you can load, display, modify, and save an image without ever changing its color resolution. Nevertheless, in some cases you may need to increase or decrease the color resolution. Here are some examples:

- Suppose you need to load a large 24-bit image on a computer that does not have much memory. You can conserve memory by loading it as an 8-bit image. Then, if you were using an 8-bit display device, you would not see any difference in the quality of the image. Of course, there would be some loss of quality on a 24-bit display device, and the loss would be permanent if you then saved the image in the same file.
- Suppose you want to save an image in a different file format. If the new file format does not support the original color resolution, you can specify a different color resolution when you save the image.
- Suppose you want to combine one 8-bit image with another one. To combine two images, they must have the same color resolution, and if they are less than 16 bits per pixel, they must use the same palette. The simplest solution is to convert both to 24 bits per pixel before combining them. Then, if necessary, you can reduce the color resolution of the combined image.

Whenever you reduce an image's color resolution to 8 bits per pixel or less, a *dithering* method comes into play. One alternative is to use a nearest-color match (no dithering), which means that the color of each pixel is changed to the palette color that most closely matches it. If the original image contains subtle color details, the result of a nearest-color match may have large blotches of color that are not very pleasing.

Dithering methods create the appearance of more subtle shades by mixing in pixels of different colors. This is similar to the way newspaper pictures produce the appearance of shades of gray, even though the only actual colors are black and white.

Ordered dithering is the fastest method. It is the default dithering method when painting to a display device that is 256 colors or less. Ordered dithering takes advantage of the fact that the colors in most palettes are ordered so that similar shades are next to each other in the palette. This method avoids blotches of color by adding to or subtracting from the nearest-color value of each pixel to ensure that adjacent pixels do not have exactly the same color. (If the colors in the palette are not ordered, the results are not good.)

All of the other dithering methods in LEADTOOLS use *error-diffusion* algorithms. In error diffusion*,* the *error* is the difference between the original pixel color and the nearest match, and *diffusion* of this error is what the algorithm accomplishes. *Floyd-Steinberg* is a high-quality, fast error-diffusion method. It is the default method that LEADTOOLS uses if you reduce the color resolution when loading or saving an image, or if you specify error diffusion as the dithering method when painting to a display device that is 256 colors or less.

If you use a LEADTOOLS function to change the color resolution of a bitmap in memory, you can choose from a list of possible dithering methods. All of the alternative error-diffusion methods are slower than Floyd-Steinberg, and the quality may or may not be better, depending on the original image, and depending on whether the resulting image is to be displayed or printed. Choosing an alternative dithering method is a subjective, trial-and-error decision. *Stevenson and Arce* dithering, the slowest of the alternatives, is most likely to produce a higher quality result.

Introduction: Palette Handling

Any image that is 8 bits per pixel or less must have a palette. The palette is an array that contains the colors for displaying the image. The pixel values are indexes into the array. An image can have its own unique palette or it can share a common fixed palette. In either case, every bitmap in memory that is 8 bits per pixel or less has its own copy of a palette (even if it is a common fixed palette).

One way for a bitmap to have a unique palette is to load an image from a file that is 8 bits per pixel or less. In that case, the file contains a palette for the image, and you can assume that the palette is unique. By default, LEADTOOLS loads the palette that is stored in the file and associates it with the bitmap in memory.

Another way for a bitmap to get a unique palette is to create the palette. For example, if you reduce a 24-bit image to 8 bits per pixel, you can let LEADTOOLS create an optimized palette for the image.

By using a unique, optimized palette, you get the best possible image quality, but there is a drawback. When a display device uses a palette (for example, when it is in 256-color mode), it can only use one palette at a time to display colors. Therefore, if you are displaying more than one image at the same time, the current image may look perfect, while all of the others look bad, because they are mapped to the wrong palette. This is called *palette shift*.

Using a fixed palette solves these palette-shift problems. The best approach is to specify the fixed palette as a display option. Then, LEADTOOLS remaps each image to the fixed palette when it is displayed, without disturbing the individual image's palette. (When displaying 16-, 24-, or 32-bit images on a device with 256 colors or less, LEADTOOLS always uses the fixed palette.)

Alternatively, you can use a color resolution function to specify a fixed palette for each individual image. But then, when you save the image to a file, the fixed palette is saved in the file, and any unique palette information is lost.

Introduction: Image Manipulation and Analysis

Most LEADTOOLS functions for image manipulation and analysis act on the bitmap in memory. It is important to keep this in mind, because the changes become permanent when you save the bitmap in a file. Sometimes you want the changes to be permanent, and sometimes you do not. It all depends on the reason for making the changes.

Sometimes, you start with an image that has some undesirable qualities. For example, if the original image is too flat, you may want to use one of the contrast enhancement functions to make permanent improvements.

At other times, you may need to make temporary changes for use on a particular monitor or printer. Using the ActiveX control or DLL functions, you can set the brightness, contrast, and gamma correction for the display of all images, without affecting the bitmaps. But for other changes, such as creating a halftone for printing, a good strategy is to modify a copy of the image, and give the user the ability to undo changes. The undo strategy also makes sense with a manipulation, such as a mosaic effect, that intentionally distorts an image.

Some image manipulations, such as edge-detection filters, are commonly used to analyze an image, rather than to improve it. For example, in an industrial quality-assurance application, defects in a machine part may be easier to detect if an image of the part is processed with an edge-detection filter. Similarly, the creation of a *histogram* (an array that charts the frequency of color usage in an image) can be of value in an analytical situation.

Edge detection and histogram creation can also be used as steps in improving an image. For example, creating a histogram is a necessary step, internally, in some of the contrast-improvement functions. Also, an image created with an edge detection filter can be combined with the original image to harden or soften the original's edges.

Introduction: Image Display

When you display an image, the following things come into play:

- Sizing and positioning the display.
- Adapting the image to the color resolution of the display device.
- Making other refinements that affect the display but not the bitmap in memory.

The LEADTOOLS *paint* functions let you specify which part of an image to display, how big to make the display, and where to position the origin (for scrolling and zooming). In the ActiveX and VBX, the origin defaults to 0,0 and the size defaults to the bitmap size. This is arbitrary, and even with automatic scroll bars, it is unlikely to be what a user wants to see. Instead, the typical user either wants to see the whole image or wants to select a portion of the image to view.

Fortunately, LEADTOOLS makes it easy to show the user exactly what he wants to see. LEADTOOLS lets you control the size and position of the display without resizing or trimming the bitmap. In fact, it usually is best to think of the bitmap size as a measure of resolution, rather than a measure of size. The larger the bitmap, the more you can zoom in without losing detail.

As for adapting to the color resolution of the display device, if the color resolution of your device is greater than 256 colors, no adaptation is necessary; images are displayed using their own color values. Of course, if a 24-bit image is displayed on a 16-bit device, there is some loss, but there are still no programming issues, since the only remedy is to change the display mode.

If the color resolution of your display device is 256 colors or less, you have to make some programming decisions regarding dithering and palette selection. When displaying 16-, 24-, or 32-bit images, you can determine whether LEADTOOLS uses ordered dithering or error diffusion to handle the color reduction. (LEADTOOLS uses a fixed palette for these images.) When displaying 8-, 4-, or 1-bit images, you must decide whether to use a fixed palette or the individual image's palette. Using the fixed palette avoids palette shifting when more than one image is displayed at the same time. Using the image's original palette produces the best possible display of the image's colors.

Other LEADTOOLS functions let you determine how the new display interacts with whatever is already on the display device. For example, you can use Windows ROP codes to combine the displayed images. Also, when displaying 1-bit images, you can set scale-to-gray or favor-black options that can improve the readability of scaled images (images that are zoomed in or out to fit the available space).

Introduction: Bitmaps in Memory and in Files

Generally, the size of a bitmap in memory is determined by its dimensions and its color resolution. For example, to calculate the number of bytes for a 500-by-500 pixel, 24-bit image, you multiply 500 * 500 * 3. (Multiply by 3 because there are 3 bytes in 24 bits.) For an 8-bit image, it would be 500 * 500, plus the size of its palette, which is about 1K. There is an exception for 1-bit files, which can be loaded in a compressed format using the LEADTOOLS Pro Express toolkit.

For any color image, whether the color values are in a palette or in the image data, the color of each pixel in the bitmap consists of red, green, and blue values. They always use the RGB *color-space model,* which means that red, green, and blue values are combined to represent a color.

The size of the same color image when it is stored in a file is often much smaller, and the color-space model is not always RGB. For example, JPEG and LEAD CMP files both use a YUV color-space model, where Y is a *luminance* value, and the U and V values are *chrominance* values. In this color-space model, brightness information is stored in the luminance value, and color information is stored in the chrominance values. JPEG and LEAD CMP achieve data compression through sampling techniques that can affect the accuracy of the YUV values. This kind of compression is sometimes referred to as *lossy* compression, because you can manipulate the compression parameters to choose between greater compression or greater accuracy. (YCbCr, used in TIFF images, is the same color-space model as YUV.)

Another color-space model used in image files is CMYK (designed for color printing using cyan, magenta, yellow, and blackness). Other compression techniques include RLE (run-length encoding) and LZW (Lempel-Ziv and Welch). Both of these techniques preserve the accuracy of the image data, and they achieve compression solely by identifying repetitions of data. Some grayscale and 1-bit image file formats also use RLE and LZW compression techniques.

The image file formats are not only used for storage in the computer's file system, but are also used for database storage and for transmission of images. Furthermore, some applications keep frequently-used images in memory in a compressed file format. Thus, they can take advantage of memory speed when loading bitmaps from these files.

Introduction: DIBs, DDBs, and the Clipboard

A *DIB* (device independent bitmap) is a standard Windows representation of a bitmap. It is similar to the LEADTOOLS representation of a bitmap in that it includes palette information. A *DDB* (device dependent bitmap) is a hardware-specific representation that does not include palette information. Sometimes, you may need to manipulate a bitmap using Windows functions that work with DIBs or DDBs. For example, you may need to interact with an existing application that uses those functions. For that purpose, LEADTOOLS functions let you convert or copy to and from DIBs or DDBs. Of course, when converting to and from a DDB, you must preserve the palette information separately.

The c*lipboard* is a standard Windows object for copying data from one application to another. LEADTOOLS lets you copy images to the clipboard and paste images from the clipboard. When copying to the clipboard, most LEADTOOLS functions clear the existing data from the clipboard, and copy image data to it in the form of a DIB, a DDB, and a palette (if one is needed). If that is not what you want, you can use a LEADTOOLS DLL function to specify exactly which actions to take.

Introduction: Printing

LEADTOOLS printing functions are designed to work with Windows printing functions. Normally, you use Windows to select a *device context* (DC) and start the print job. LEADTOOLS functions let you scale and position one or more images in the printer DC. You use Windows functions to get the available page dimensions. You also use Windows functions to eject each page and to end the print job.

In most cases, you can scale and position an image on a printer DC with the same flexibility as when you display an image on a computer screen. You can let the printer handle any dithering or halftoning. However, if you halftone a bitmap, yourself, before sending it to a printer, you should resize the bitmap to the desired height and width, then use the halftone function, then send the image to the printer without scaling it. (You get bad results if you scale a halftoned image.)

In C, C++, Visual Basic, and Delphi, you can use the actual Windows printing functions, or their equivalents in the development system. However, printing with Microsoft FoxPro and older versions of Access is more limited, because they do not give you direct access to the printer DC. The LEADTOOLS ActiveX handles this problem by providing methods to start and stop the print job.

With Access 7.0, you can also add code to a report's Print event to include ActiveX images. Or, more simply, you can embed the LEAD OLE Server, which is provided as a separate toolkit component.

Introduction: TWAIN Input

TWAIN is a standard for image input devices, such as scanners and digital cameras. Getting an image from a TWAIN device can be very simple. First, there is a standard facility for selecting the input device. It is similar to selecting a default printer, and you can skip this step if you know that the default is right. Then, each TWAIN device has its own user interface, which you invoke to *acquire* an image.

Some complexity is introduced only when acquiring multiple images or when bypassing the user interface. If you acquire multiple images, you must code a callback function or event procedure to handle each image. If you bypass the user interface, you must set some standard TWAIN parameters to provide the information that the TWAIN driver would normally get from the user interface. The tricky part is that some TWAIN drivers implement the standard more fully than others. If you do not know in advance how the TWAIN device will behave, it is a good idea to add logic to your program to test the success of setting the TWAIN parameters. LEADTOOLS provides functions for all of this.

Introduction: Database Interaction

With LEADTOOLS, you can use a variety of strategies to maintain images in a database. This variety is necessary because of the uneven support for storing images using different development and database systems.

The simplest database support is with Visual Basic, where you can bind a LEAD control (ActiveX or VBX) to a data control. You can associate the LEAD control with a long binary field in the database, and let the data control handle all of the database maintenance and navigation.

For other systems with ActiveX support, you may be able to use ODBC (Open Database Connectivity). ODBC is a standard interface for accessing local and remote databases. To use it, you define a data source that is implemented with an ODBC driver; then you access the data source in the same way you would a local database. LEADTOOLS provides ActiveX methods and properties that let you directly access image data through an ODBC data source. There are a couple of limitations in this approach. First, since the ActiveX accesses the image data directly and does not access any other data in the record, you must deal with the problem of identifying the image. To solve this problem, you can use another means of selecting records; then synchronize the ActiveX's view, so that it is always accessing the same record. (The ActiveX tutorials describe how to do this.) The second limitation is with ODBC drivers. Most of the available ODBC drivers do not properly support access to long binary fields. (The ACCESS 7.0 and FoxPro 2.x drivers have been tested successfully and are described in the ActiveX tutorials.)

A more flexible approach on lower-level development systems, such as C, C++, and Delphi, is to use the ability of LEADTOOLS to save a file in memory. Once the image is available in memory in a compressed file format, you can use your choice of database engines to store and retrieve the data.

Finally, you may want to consider using your computer's or network's file system to store the images, use a database to store the file paths as strings, and write your own code to do the synchronization. In some cases, this could be a better solution than storing the images directly in the database.

Introduction: Data Transfer

When receiving transferred data, as in an Internet application or a FAX modem transmission, you often want to display the image as it is received.

To meet that need, LEADTOOLS provides simple functions that let you load image data into a bitmap as it is received. You can also use the LEADTOOLS paint-while-load feature to display the bitmap as it is received.

For Internet support, LEADTOOLS provides a plug-in DLL that can be used on WIN32 systems to view compressed images with Netscape Navigator or Microsoft Internet Explorer.

Introduction: Animation

GIF files are commonly used for small animated images in applications. These are multipage files, and animation is achieved by processing the list of images. (This is not the same as AVI or other full-motion video formats.)

With DLL functions or the ActiveX control, LEADTOOLS lets you load a multipage file into a list of bitmaps; then create and play an animation sequence using the list. With the paint-while-load feature, you can also play the animation while the list is being loaded.

LEADTOOLS takes advantage of the special animation features in GIF files, such as using one palette for the whole list of bitmaps. But you can create, play, and save animations in any multipage format, including GIF, TIFF, and PCX. You can also create a list of bitmaps from an AVI file, but this is a read-only feature.

Summary of All Supported Image File Formats

Color and Grayscale

JPEG and LEAD Compressed (JPG and CMP)

GIF and TIFF with LZW Compression

TIFF Without LZW Compression

BMP Formats

Icons and Cursors (ICO and CUR)

PCX Formats (PCX and DCX)

Kodak Formats (PCD and FPX)

DICOM Format (DIC)

Exif Formats (TIFF and JPG)

Windows Metafile Format (WMF)

PhotoShop 3.0 Format (PSD)

Portable Network Graphics Format (PNG)

Truevision TARGA Format (TGA)

Encapsulated PostScript (EPS)

SUN Raster Format (RAS)

WordPerfect Format (WPG)

Macintosh Pict Format (PCT)

Bitonal (1-Bit)

TIFF CCITT and Other FAX Formats

LEAD 1-Bit Format (CMP)

Miscellaneous 1-Bit Formats (MAC, IMG, and MSP)

JPEG and LEAD Compressed (JPG and CMP)

LEADTOOLS reads and writes the following LEAD and JPEG compressed formats:

- **LEAD.** This is the LEAD CMP compressed format for grayscale and color images. This format results in smaller files and better image quality than industry-standard formats.
- **Progressive LEAD.** This is a LEAD format that is useful for transmitting images, because the first part of the file contains the full dimensions of the image. Therefore, in a paint-while-load routine, you can display the whole image, then progressively clarify it as the rest of the file loads.
- **JFIF.** This is the JPEG File Interchange Format. LEADTOOLS supports YUV 4:4:4, 4:2:2, and 4:1:1 color spacing, and YUV 4:0:0 for grayscale.
- **Progressive JPEG.** This is a JFIF format that is useful for transmitting images, because the first part of the file contains the full dimensions of the image. Therefore, in a paint-while-load routine, you can display the whole image, then progressively clarify it as the rest of the file loads. LEADTOOLS supports YUV 4:4:4, 4:2:2, and 4:1:1 color spacing, and YUV 4:0:0 for grayscale.
- **JTIF.** This is the JPEG Tagged Interchange Format. LEADTOOLS supports YUV 4:4:4, 4:2:2, and 4:1:1 color spacing, and YUV 4:0:0 for grayscale.

In all of these formats, you can read and write 8 or 24 bits per pixel, where 8 is for grayscale and 24 is for color.

GIF and TIFF with LZW Compression

LEADTOOLS reads and writes the following LZW-compressed formats. Note that these formats require a separate license, as explained in LZW License Instructions.

- **GIF.** This is the Graphics Interchange Format created by CompuServe for storing and exchanging color raster images.

  You can read and write the following bits per pixel: 1, 2, 3, 4, 5, 6, 7, 8.
- **Animated GIF.** This is a multipage GIF format that can be played as an animation by displaying the images sequentially. LEADTOOLS lets you create and play the animations.

  You can read and write the following bits per pixel: 1, 2, 3, 4, 5, 6, 7, 8.
- **TIFF LZW.** These files use the Tagged Image File Format with LZW compression. LEADTOOLS can read and write TIFF files that use RGB, CMYK, or YCbCr color space.

  You can read the following bits per pixel, without regard for the color space: 1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32.

  You can write the following bits per pixel, with RGB color space: 1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32.

  You can write the following bits per pixel, with CMYK color space: 24, 32.

  You can write the following bits per pixel, with YCbCr color space: 24.

TIFF Without LZW Compression

LEADTOOLS reads and writes several TIFF formats that **do not** require an LZW license.

TIFF is a tag-based file format designed to promote universal interchanges of digital image data. Because TIFF files do not have a single way to store image data, there are many versions of TIFF. LEADTOOLS supports the most common TIFF formats. The usual file extension is TIFF for single-image files, and MPT for multipage files (which can contain many images).

LEADTOOLS can read and write TIFF files that use RGB, CMYK, or YCbCr color space. LEADTOOLS can also read and write files with PackBits compression.

You can read the following bits per pixel, without regard for the color space: 1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32.

You can write the following bits per pixel, with RGB color space: 1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32.

You can write the following bits per pixel, with CMYK color space: 24, 32.

You can write the following bits per pixel, with YCbCr color space: 24.

BMP Formats

LEADTOOLS reads and writes the following BMP formats:

- **Windows Bitmap (BMP).** This is a file format created by Microsoft. Some BMP images are compressed with an RLE-type compression.

  You can read and write the following bits per pixel, without RLE compression: 1, 4, 8, 16, 24, 32.

  You can read and write the following bits per pixel, with RLE compression: 4, 8.

- **OS/2 Bitmap (OS/2 BMP).** These are files created on an OS/2 operating system. LEADTOOLS supports both 1.x and 2.x formats.

  You can read and write the following bits per pixel, without RLE compression: 1, 4, 8, 24.

  You can read (but not write) the following bits per pixel, with RLE compression: 4, 8.

Icons and Cursors (ICO and CUR)

LEADTOOLS reads and writes Windows Icon and Cursor files. LEADTOOLS is not designed as an icon editor, but this file support lets you create a bitmap from an icon or cursor, or create an icon or cursor from a bitmap. The library (DLL) to support these formats is the same as for BMP files.

LEADTOOLS supports these file formats as follows:

- **Windows Icon (ICO).** This is a multipage file format for storing one or more Windows icons. The background color when you read these images is always black. The size when you write these images can be no larger than 255-by-255 pixels. (Neither the width nor height can be greater than 255.)

  You can read multipage files with the following bits per pixel: 1, 4, 8.

  You can write single-page files with the following bits per pixel: 1, 4, 8.

- **Windows Cursor (CUR).** This is a file format for storing Windows cursors (mouse pointers). LEADTOOLS does not load or save transparency information in these files. The size when you write these images should be exactly 32-by-32 pixels.

  You can read and write 1-bit, single-page files.

PCX Formats (PCX and DCX)

LEADTOOLS reads and writes the following PCX formats:

- **PCX.** This is a file format created by ZSoft. This format compresses its image data with the RLE type compression.
- **DCX.** This is a multipage PCX format that enables a file to contain more than one image. It is handled the same as a regular PCX file, except for the multipage feature.

For PCX and DCX files, you can read and write the following bits per pixel: 1, 4, 8, 24.

Kodak Formats (PCD and FPX)

LEADTOOLS supports the following Kodak formats:

- **PhotoCD (PCD).** This is a common high-resolution format for images on CD-ROM. An image file can contain one or more physical resolutions, with fixed dimensions. You can choose which resolution to load, and thus control the size of the resulting bitmap.

  You can read the following bits per pixel: 8, 24.

  You cannot write PhotoCD files. They are read-only.
- **FlashPix (FPX).** (WIN32 only) This Kodak format is newer than PhotoCD. This format can also contain more than one physical resolution, but without fixed dimensions. You can choose which resolution to load, and thus control the size of the resulting bitmap.

  You can read and write the following bits per pixel: 8, 24.

DICOM Format (DIC)

Note: This format is supported only in Express editions.

DICOM (DIC) files are commonly used in medical images. LEADTOOLS reads a broad range of DICOM formats, converting them to LEAD-supported bits-per-pixel as necessary. LEADTOOLS writes DICOM files in the LEAD-supported bits-per-pixel.

You can read DICOM files as follows:

- Grayscale files in the 1- to 16-bit range are converted to 8-bit grayscale.
- Palettized color files in the 1- to 16-bit range are converted to 8-bit palettized or 24-bit unpalettized. By default, they are converted to 24-bit only if the palette contains more than 256 colors.
- Unpalettized color files (RGB, ARGB, or CMYK) with 1 to 16 bits per plane are converted to 24-bit RGB.

You can write DICOM files as 8-bit grayscale, 8-bit palettized color, or 24-bit RGB color (8 bits per plane).

Exif Formats (TIFF and JPG)

Exif is a standard for image files created with digital cameras and other input devices. The standard is set by the Japan Electronic Industry Development Association, and formally it is called the *Digital Still Camera Image File Format Standard.*

Exif files contain either uncompressed TIFF images or compressed JPEG images. In addition, these files can contain many comment fields of various types.

LEADTOOLS reads and writes the following Exif formats:

- **TIFF.** This is a tag-based file format designed to promote universal interchanges of digital image data. If the color-space model is YCbCr, the Exif standard recommends that the width of the image be a multiple of 2. LEADTOOLS **does not** enforce this standard; enforcement is up to you.

  You can read and write 24 bits per pixel.
- **JPG.** This is the JPEG format with YUV 4:2:2 sampling. The Exif standard requires the width of the image to be a multiple of 16 and the height to be a multiple of 8. LEADTOOLS **does not** enforce this standard; enforcement is up to you.

  You can read and write 24 bits per pixel. (Grayscale is not supported.)

An Exif JPG file can also contain a stamp (thumbnail) image, which is a small copy of the main image. The stamp must be 24 bits per pixel, and it cannot be larger than 64K bytes. Generally, it should fit within an 80-by-80-pixel square, preserving the aspect ratio.

Windows Metafile Format (WMF)

The **Windows Metafile (WMF)** format is a vectored format that may or may not also contain a raster image. When reading a WMF file, LEADTOOLS converts all of the image data to a raster image. When writing a WMF file, LEADTOOLS saves all of the image data as a raster image within the WMF file. Therefore, if you load and save a WMF file, you convert any vectored image data that the original contained.

For WMF files, you can read and write the following bits per pixel: 8, 24.

Note: When working with LEADTOOLS annotations (rather than image files), you can save the annotations as WMF files containing vectored graphics that can be used in other applications.

PhotoShop 3.0 Format (PSD)

Photoshop 3.0 (PSD) is the format produced by the Adobe Photoshop graphics editor.

For PSD files, you can read and write the following bits per pixel: 1, 8, 24.

Portable Network Graphics Format (PNG)

PNG (Portable Network Graphics) is a replacement for the GIF format. It is a full-featured (non-LZW) compressed format intended for widespread use without legal restraints.

You can read the following bits per pixel: 1, 4, 8, 16, 24, 32.

You can write the following bits per pixel: 1, 4, 8, 24.

LEADTOOLS supports reading (but not writing) interlaced PNG files.

Truevision TARGA Format (TGA)

Truevision TGA (TARGA) is a file format created by Truevision Inc. LEADTOOLS supports all uncompressed and RLE compressed TGA file formats.

For TGA files, you can read and write the following bits per pixel: 8, 16, 24, 32.

Encapsulated PostScript (EPS)

PostScript Raster (Encapsulated PostScript). These files are used primarily on PostScript printers. These printers usually offer more variety of fonts and higher resolution than standard laser printers. EPS files will work on any PostScript compatible printer and any end-user application that supports placement of EPS files in its work space.

The image that you read from an EPS file can be either a PostScript raster image or an embedded TIFF image. The image that you write to an EPS file is always a PostScript raster image.

For a PostScript raster image, you can read the following bits per pixel: 1, 8.

For an embedded TIFF image, you can read the following bits per pixel: 1, 4, 8, 16, 24, 32.

You can write only an 8-bit PostScript raster image, which is grayscale.

SUN Raster Format (RAS)

SUN Raster (RAS) is a format native to Sun UNIX platforms.

For RAS files, you can read and write the following bits per pixel: 1, 4, 8, 24, 32.

WordPerfect Format (WPG)

The **WordPerfect (WPG)** format can contain vectored or raster images. LEADTOOLS handles only the raster images.

For WPG files, you can read and write the following bits per pixel: 1, 4, 8.

Macintosh Pict Format (PCT)

Macintosh Pict (PCT) files are produced using Macintosh QuickDraw, and are used in desktop publishing and imaging applications.

For PCT files, you can read and write the following bits per pixel: 1, 4, 8, 24.

TIFF CCITT and Other FAX Formats

LEADTOOLS supports reading and writing of the following 1-bit formats that are commonly used in FAX transmissions:

- **TIFF CCITT.** These are compressed TIFF files that are commonly used for FAX transmission and document imaging.
- **TIFF CCITT Group 3.** These are TIFF CCITT files in a format that is more advanced and more compressed than TIFF CCITT. LEADTOOLS supports both 1-dimension and 2-dimension variations of this format.
- **TIFF CCITT Group 4.** These are TIFF CCITT files in a format that is more advanced and more compressed than TIFF CCITT Group 3.
- **CALS Raster.** These are CCITT Group 4 CALS raster files. CALS is a United States government standard.
- **IOCA (ICA).** This is the Image Object Content Architecture developed by IBM. LEADTOOLS supports these files in an MO:DCA wrapper with embedded 1-bit CCITT Group 3 or Group 4 images. LEADTOOLS also supports IOCA files without an MO:DCA wrapper.
- **WinFax Group 3.** This is a FAX format created by Delrina for Group 3 support.
- **WinFax Group 4.** This is a FAX format created by Delrina for Group 4 support.
- **FAX Group 3.** This is a raw FAX format (without a header) for Group 3 support. LEADTOOLS supports both 1-dimension and 2-dimension variations of this format.
- **FAX Group 4.** This is a raw FAX format (without a header) for Group 4 support.
- **AWD.** (WIN32 only) This is a Microsoft FAX format.

LEAD 1-Bit Format (CMP)

LEAD 1BIT is a proprietary 1-bit LEAD compressed format that has better compression and performance than other 1-bit formats.

You can read and write files in this format.

Miscellaneous 1-Bit Formats (MAC, IMG, and MSP)

LEADTOOLS supports reading and writing of the following 1-bit formats:

- **MacPaint (MAC).** These Macintosh Paint files are commonly used for monochrome clip art.
- **GEM Image (IMG).** These files are native to the Graphical Environment Manager developed by Digital Research.
- **Microsoft Paint (MSP).** These files from early versions of Windows are used for black-and-white drawings and clip art.

Copyright Notice

Address

LEAD Technologies, Inc.
900 Baxter Street
Charlotte, NC 28204

(704) 332-5532
FAX: (704) 372-8161
BBS: (704) 334-9045
Technical support: (704) 372-9681 or Support@Leadtools.com
CompuServe: 71333,2237 or GO LEADTECH
Home page: http://www.leadtools.com

Summary of LEADTOOLS Products

LEAD Technologies, Inc.	(704) 332-5532
900 Baxter Street	FAX: (704) 372-8161
Charlotte, NC 28204	BBS: (704) 334-9045
Sales@Leadtools.com	Tech Support: (704) 372-9681
www.leadtools.com	CompuServe: 71333,2237 or GO LEADTECH

LEADTOOLS is a family of comprehensive imaging toolkits designed to help programmers integrate black & white, grayscale and color images into their applications as quickly as possible. LEADTOOLS gives developers access to imaging technology that is typically only seen in professional end user applications such as PhotoShop, and allows them to add the technology into their programs without the expertise, resources or time investment that it might take them to write the code on their own.

LEADTOOLS has almost 300 functions that handle all aspects of imaging, from the initial import of the image with TWAIN scanners or digital cameras, to the viewing and processing of an image, to the final output via printers. Whether the programmer is overlaying annotations on one million checks, or zooming in on a high resolution satellite photograph, from A to Z LEADTOOLS is the solution.

LEADTOOLS supports the importing and exporting of more than 40 of the most popular image file formats including TIFF, TGA, BMP, PNG, PCX and GIF. LEAD's file format support does not just stop with the most popular flavor of a particular format. LEAD has spent the time to ensure that LEADTOOLS supports as many of the known flavors and bits per pixel (1-32) as possible. Developers will feel confident that TIFF support in LEADTOOLS means TIFF is supported! GIF support includes the ability to handle 1-8 bits per pixel, as well as interlaced, transparent, animated, and embedded text variations. (Note: because of the LZW patent, there is a license required from Unisys to gain access to the GIF code). The image compression support in LEADTOOLS is the most comprehensive of any toolkit on the market, with both industry standard methods such as JPEG (multiple flavors including progressive) and CCITT G3/G4, as well as, proprietary alternatives such as LEAD CMP which produces image sizes that are smaller, and image quality that is better, than the standard compressed formats.

LEADTOOLS contains more than 50 different image processing filters (such as intensity, sharpen, emboss, mosaic, and histogram) to create special effects or to provide image clean-up and photo manipulation. Programmers who need more control over image processing can take advantage of powerful "region of interest" capabilities.

LEADTOOLS provides special effects for developing multimedia projects and/or presentation applications. Use 113 Paint Effects, 64 Dissolve Effects, and 24 Transition Effects with delays, grain sizes, pattern brushes (up to 64 passes), a colored wand and a transparent color. The combinations are unlimited.

LEADTOOLS provides flexible paint functions for dithering, zooming and scrolling. In fact, LEAD's color reduction and dithering code is considered among the best in the business. The programmer has the flexibility to choose between fixed or optimized palettes, and eight (8) different dithering techniques including Floyd-Stein and Burkes.

LEADTOOLS has specific features designed for the imaging database developer, such as VB data binding, 32-bit ODBC, a customized OLE 2.0 in-place server, Load/Save memory, and Load/Save file offset. Programmers will find that LEADTOOLS makes integrating images into a database as easy as possible. Web developers will appreciate a special DLL plug-in which allows Netscape Navigator to read LEADTOOLS images, and a net aware activex which allows Microsoft Internet Explorer to read LEADTOOLS images.

All of the imaging technology described above is included in the entire LEADTOOLS product line, except that the annotation features, and several 1-bit imaging filters such as scale to gray and deskew, are only available in the Pro Express. Additionally the JPEG, CCITT G3/G4, and LEAD CMP compressed file formats in the Pro Express have been optimized for superior speed and performance. (These compressed formats are also available in all other LEADTOOLS products). The Pro Express requires a per copy royalty for each distributed copy of the application software developed with LEADTOOLS.

Payment options include single unit fixed price, quantity discounts for units purchased in volume blocks, and fixed price for annual unlimited quantity. All other products permit royalty FREE distribution of the end user application. Please see the following product listings for complete details.

LEADTOOLS Pro Express $1,995 (requires per copy royalties)

LEAD's Pro Express is the most comprehensive imaging toolkit on the market. No other toolkit provides as many imaging features for color, gray-scale, and black & white images in one development package, and for one price. The Pro Express version is a collection of LEAD's best code, and is designed for the programmer who is looking for the fastest performance, and/or comprehensive color and document imaging capabilities. In addition to all the features contained in all other LEADTOOLS products, the Pro Express contains powerful annotation capabilities and optimized document image processing filters, and high performance compression, decompression and viewing of JPEG, LEAD CMP, and CITT G3/G4. Newly added is the Dicom file format for medical imaging. The annotation toolkit enables the programmer to add text, highlights, sticky notes, audio, ellipses, redaction (blackout), buttons, hotspots, freehand scribble, pointers, rectangles, polygons and more to black & white and color images. Programmers working with document imaging will also find special processing filters such as despeckle, deskew, favor black and white 1bit images. The toolkit includes WIN32 DLLs, WIN16 DLLs, a 32 bit ActiveX (OCX), a 16 bit ActiveX (OXC), and a VBX, and ships with several example applications, with source code, for C/C++, MFC, Visual Basic (3.0, 4.0, 5.0), Visual Fox, Access (2.0 & 7.0), Delphi (1.0 & 2.0), VB script and Java script.

LEADTOOLS WIN32 Pro $995 (royalty FREE)

The WIN32 Pro is designed for the programmer who needs a 32 bit tool for development with C/C++, and who wants the ultimate control over imaging. This toolkit includes both a C/C++ API for the DLLs and an ActiveX. The ActiveX is documented to work with the C/C++ as well as Visual Basic, Access 7.0, and Delphi. The toolkit contains 32 bit DLLs and a 32 bit ActiveX (OCX), and ships with several example applications, with source code, for C/C++, MFC, Visual Basic, Visual FoxPro, Access 7.0, Delphi 2.0, VB script and Java script.

LEADTOOLS WIN16 Pro $795 (royalty FREE)

The WIN 16 Pro is designed for the programmer who needs a 16 bit tool for development with C/C++, and who wants the ultimate control over imaging. This toolkit includes both a C/C++ API for the DLLs and an ActiveX. The ActiveX is documented to work C/C++, as well as with Visual Basic and Access. The toolkit contains 16 bit DLLs, a 16 bit ActiveX (OCX), and a VBX, and ships with several example applications, with source code, for C/C++, MFC, Visual Basic, Access 2.0, and Delphi 1.0.

LEADTOOLS WIN16/WIN32 Pro $1495 (royalty FREE)

The WIN16/32 is a bundle of the WIN32 Pro and the WIN 16 Pro. If the application being developed needs to run on both 16 bit and 32 bit platforms, the purchase of the bundle, instead of each product separately, will save almost $300.

LEADTOOLS ActiveX 32 (OCX) Pro $395 (royalty FREE)

The ActiveX 32 (OCX) Pro is designed for the programmer who wants to create applications with a 32 bit custom control. The ActiveX 32 contains a WIN32 OLE custom control and ships with several example applications, with source code, for Visual Basic, Visual FoxPro, Access 7.0 Delphi 2.0, VB script and Java script. Although Visual C++ and MFC can use ActiveX controls, this version of LEADTOOLS does not provide the C/C++ documentation. Programmers looking to use the ActiveX within a 32 bit "C" environment should purchase the LEADTOOLS WIN32 product.

LEADTOOLS ActiveX 16 (OCX) Pro $395 (royalty FREE)

The ActiveX 16 (OCX) Pro is designed for the programmer who wants to create applications with a 16 bit custom control. The ActiveX contains a WIN 16 OLE custom control and ships with several example applications, with source code, for Visual Basic, and Access 2.0. Although Visual C++ and MFC can use

ActiveX controls, this version of LEADTOOLS does not provide the C/C++ documentation. Programmers looking to use the ActiveX within a 16 bit "C" environment should purchase the LEADTOOLS WIN16 product.

LEADTOOLS ActiveX(OCX) 16/32 Pro $495 (royalty FREE)

The ActiveX 16/32 is a bundle of the ActiveX 32 and the ActiveX 16. If the application being developed needs to run on both 16 and 32 bit platforms, purchasing the bundle instead of each control separately, will save almost $300.

LEADTOOLS VBX Pro $295 (royalty FREE)

The VBX is designed for programmers creating applications in a 16 bit environment. The VBX ships with example applications, and source code, for Visual Basic v3. and Delphi 1.0.

If you need a complete list of the LEAD supported 40+ file formats or 50+ image processing filters, or if you want a printout of the functions, properties or methods, please visit LEAD Technologies' web site at **http://wwwleadtools.com,** or call for a copy to be faxed.

LEAD is the developer and publisher of the award winning LEADTOOLS imaging toolkits, and has been developing and supplying imaging toolkits to programmers for over seven years. LEAD produces code that is mature, stable, easy to use, and optimized for high performance. After 7 years of working with thousands of programmers, LEAD understands that issues such as documentation, function samples, example source code, and timely technical support, are very important to a developer. These non object code components can make the difference when meeting a deadline, or when trying to win Best of Show. LEADTOOLS is market tested and proven, and has been shipped to millions of customers through the integration into products by Microsoft, Corel, Delrina, Xerox, Hewlett Packard, and many other satisfied customers. Whether a programmer is creating a web page, a document imaging system, a paint/drawing program, or an imaging database that will run on a desktop, or over an Intranet or the Internet, LEADTOOLS is the best choice in imaging toolkits.

All products are sold in the U.S. and Canada with a 30-day money back guarantee.

LEADTOOLS Pro Express

LEADTOOLS Pro Express is the most comprehensive imaging toolkit on the market. No other toolkit provides as many imaging features for color, gray-scale and black & white images in one development package, and for one price. The Pro Express version is LEAD's best code and is designed for the programmer who is looking for the fastest performance, and/or the most extensive color and document imaging capabilities available.

The Pro Express is LEAD's top of the line in the LEADTOOLS family of imaging toolkits. Designed to be the most comprehensive imaging technology available in a development toolkit, LEADTOOLS is a collection of nearly 300 functions, properties and methods, falling into 12 general categories: scanning, color conversion, display, multi-media special effects, annotation, image processing, animation, compression, image format import/export filters, Internet imaging, database imaging, and printing.

The Pro Express contains all the features in every other LEADTOOLS product. In addition, the Pro Express contains powerful annotation capabilities and optimized document image processing filters, and high performance compression, decompression and viewing of JPEG, LEAD CMP, and CCITT G3/G4. Pro Express now supports the medical imaging standard Dicom. LEADTOOLS Pro Express offers the fastest software-only JPEG decoder available. The annotation capabilities enable the programmer to add text, highlights, sticky notes, audio, ellipses, buttons, lines, arrows, rectangles, polygons, redaction (blackout), hotspots, freehand scribble, pointers, bitmap and text stamps, to black & white and color images. Programmers working with document imaging will also find special processing filters such as despeckle, deskew, favor black, and scale to gray to enhance their black and white 1 bit images. Additionally, you can now achieve faster rotation in 90-degree increments by applying a rotated view perspective to the bitmap.

Please refer to the table below for a quick comparison between the Pro Express and LEAD's other Pro versions: (For more information, please contact the Sales department)

	Pro versions	**Pro Express**
Scanning	✓	✓
Color Conversion	✓	✓
Display - general	✓	✓
scale to gray		✓
favor black		✓
Multimedia Special Effects	✓	✓
Annotations		✓
Image Processing - general	✓	✓
despeckle		✓
deskew		■
region of interest	■	■
Animation	■	■
Compression - general	■	■
LEAD's fastest JPEG		■
LEAD's fastest CMP		■
LEAD's fastest G3/G4		■
Image Formats	■	■

Imaging for the Internet	■	■
Imaging for Databases	■	■
Printing	■	■
Royalty	no	yes

The Pro Express toolkit includes WIN32 DLLs, WIN16 DLLs, a 32 bit ActiveX(OCX), a 16 bit ActiveX (OCX), and a VBX, and ships with several example applications, with source code, for C/C++, MFC, Visual Basic (3.0, 4.0, 5.0), Visual Fox, Access (2.0 & 7.0), Delphi (1.0 & 2.0), VB and Java script. The Pro Express does require royalties to be paid on the distribution of the application software that is developed with LEADTOOLS. Please contact LEAD's Sales department to determine the license fee package that will best suit your needs.

LEAD Technologies, Inc.
(704) 332-5532
www.leadtools.com