



Lexmark™

Imaging Toolkit

Integration Guide for Java

August 2015

www.lexmark.com

Contents

- Overview..... 3**
- Installing the application.....5**
- Licensing the application..... 8**
 - Obtaining electronic licenses for the Imaging Toolkit.....8
 - Installing the application on your printer.....9
 - Licensing the printer.....10
- Configuring the application..... 12**
 - Changing the Imaging Toolkit properties.....12
 - Connecting to the database.....13
 - Using the Imaging Toolkit in a cluster environment.....13
 - Scan settings.....13
 - Incompatible scan settings.....17
 - Custom scan sizes.....18
 - File types created by the Imaging Toolkit.....19
- Using the application API..... 20**
 - Required functions.....20
 - Creating an application.....25
- Using the application from the command line..... 30**
 - Using the JAR file.....30
 - Creating a scan settings property file.....31
 - Possible command line-related error messages.....31
- Notices..... 33**
- Index..... 34**

Overview

Use the *Imaging Toolkit* to integrate scanning capabilities directly into an application.

When the Imaging Toolkit is called by a host program, a scan profile is sent to the designated multifunction product (MFP). A listening port on the host computer remains open until the end user goes to the MFP, selects the profile, and initiates a scan. When you start a scan, the scan data is sent back to the Imaging Toolkit, and then it passes the scan job back to the calling program.

This package consists of a Java archive library (JAR) that provides the imaging functionality necessary for integration with your application.

This document provides information on application programming interface (API), command-line usage, and a sample application to help you get started with JAR integration.

Note: For Linux environments, change the host files to assign the true IP address to the host name of the server where it is installed.

Integrating the Imaging Toolkit involves four components:

- Installing the Imaging Toolkit
- Licensing a printer (or printers) for use with the Imaging Toolkit
- Creating an application
- Executing the application

Files included with the Imaging Toolkit and their content

File or folder	Contains
License.rtf	The <i>End User License Agreement</i> .
ReadMe.txt	<ul style="list-style-type: none"> • The most recent technical updates. • The list of supported printers. • The list of known issues.
esf-license-app	<ul style="list-style-type: none"> • ImagingToolkit.flis—Used to license individual or multiple printers. • LicenseServer.exe—Used to license multiple printers on a network.
java-sdk	<ul style="list-style-type: none"> • docs folder—<i>Integration Guide for Java</i>. • sample folder, which contains: <ul style="list-style-type: none"> – src—Sample source code. – bin—Sample applications, config.properties file, and supporting JAR files including LexImgTk-x.x.jar, where x.x is the version number of the JAR file. For example, LexImgTk-3.2.jar. <p>Notes:</p> <ul style="list-style-type: none"> – If you plan to use <i>dynamic link library</i> (DLL), then see the <i>Integration Guide for DLL</i> located in the native-sdk\docs folder. – For database support, LexImgTk-x.x needs the jaybird-full-2.2.0.jar file in the Class Path. The jaybird-full-2.2.0.jar file is located in the java-sdk\sample\bin folder.
linux_dbsetup	The database setup file for Linux.

File or folder	Contains
native-sdk	Files needed for using DLLs in a Microsoft® Windows® operating system environment. If you plan to use DLLs, then see the <i>Integration Guide for DLL</i> , located in the native-sdk\docs folder.

Supported operating systems

- Windows 8 Enterprise
- Windows 7 Ultimate
- Windows Vista® Business
- Windows Vista Ultimate
- Windows Server® 2012 R2 Standard
- Windows Server 2008 R2 EE (64-bit) with clustering
- Red Hat Linux 5 (32-bit)
- Debian Linux 3.1
- SUSE Linux 11.2 (64-bit)
- SUSE Linux 9.3 (32-bit)
- SUSE Linux Enterprise Server 10.1 (64-bit)

Supported Java Virtual Machine

Sun Java Virtual Machine 1.4.2 or later

Supported cluster

Active/active environment

Installing the application

You can install the Imaging Toolkit with or without a database. To set up the Imaging Toolkit with a database, you must do the following:

- Install the Firebird database setup and setting the database file
- Extract the Imaging Toolkit components

For 32-bit Windows operating system

- 1 From the folder where the LIT.zip files are extracted, run the executable file.
- 2 From the Imaging Toolkit screen, do either of the following:

Install the application with a database

- a Click **Install Imaging Toolkit with DB**.
- b Follow the instructions on the computer screen.

The following components are included with the Imaging Toolkit:

- Imaging Toolkit database
- Firebird ODBC driver
- Firebird database

Notes:

- For JAR, you need only the java-sdk folder that is extracted to the destination directory.
- You can skip third-party installations. However, Firebird database is required for the Imaging Toolkit to work properly.
- For more information on installing Firebird on Windows, see the documentation for Firebird.

Install the application without a database

- a Click **Install Imaging Toolkit without DB**.
- b Follow the instructions on the computer screen.

Note: For JAR, you need only the java-sdk folder that is extracted to the destination directory.

For 64-bit Windows operating system

- 1 From the folder where the LIT.zip files are extracted, run the executable file.
- 2 From the Imaging Toolkit screen, do either of the following:

Install the application with a database

- a Click **Install Imaging Toolkit with DB**.
- b Follow the instructions on the computer screen.
- c Select **32-bit Installation**.

The following components are included with the Imaging Toolkit:

- Imaging Toolkit database
- Firebird ODBC driver
- Firebird database

Notes:

- For JAR, you need only the java-sdk folder that is extracted to the destination directory.
- You can skip third-party installations. However, Firebird database is required for the Imaging Toolkit to work properly.
- For more information on installing Firebird on Windows, see the documentation for Firebird.

Install the application without a database

- a Click **Install Imaging Toolkit without DB**.
- b Follow the instructions on the computer screen.
- c Select **32-bit Installation**.
Only the Imaging Toolkit is installed.

Note: For JAR, you need only the java-sdk folder that is extracted to the destination directory.

Installing the Linux database

- 1 Locate the following files in the linux_dbsetup folder:
 - FirebirdCS-2.1.3.18185-0.amd64.rpm (64-bit Linux)
 - FirebirdCS-2.1.3.18185-0.i686(2).rpm (32-bit Linux)
 - FirebirdSS-2.1.3.18185-0.i686.rpm (32-bit Linux)
 - LITJar.FDB
- 2 Copy the appropriate package (.rpm) file to a Linux machine.
- 3 Install the .rpm file. The installer creates the needed files in **/opt/firebird**.
- 4 Change the Firebird SYSDBA password to **masterkey**.
 - a Using a text editor, from the /opt/firebird folder, open the **SYSDBA.password** file.
 - b Note the current Firebird-generated password for user SYSDBA (ISC_PASSWD).
 - c Run the **changeDBAPassword.sh** script from the **/opt/firebird/bin** folder to prompt for the current password for the SYSDBA user.

```
# cd /opt/firebird/bin
# ./changeDBAPassword.sh

Please enter current password for SYSDBA user:
```
 - d Type the current Firebird-generated password, and then press **Enter**.
The script prompts for a new password for the SYSDBA user.
 - e Type **masterkey**, and then press **Enter**.

Note: For more information, see the documentation for Firebird.

- 5 Copy **LITJar.FDB** to a location that you want. Both this location and LITJar.FDB need full access permissions.

Note: For more information on installing Firebird in a Linux environment, see the documentation for Firebird.

Licensing the application

Before using the Imaging Toolkit, obtain a license for each printer that you plan to use with the Imaging Toolkit. When licensing printers, you need to do the following:

- Decide if you want to install a license on individual printers or provide licenses through your network from a license server.
- Determine if your printers require an electronic or paper license.
- Collect host ID information.
- Contact your Lexmark representative to order the appropriate type and number of licenses.
- Install the licenses either on individual printers or on your license server.

Obtaining electronic licenses for the Imaging Toolkit

Installing individual or network licenses

If you are using the Toolkit with a few printers, then install the individual licenses locally on each printer. If you are using the Imaging Toolkit with several printers, then install the license server and obtain a network license for the appropriate number of devices.

Determining whether your printer needs an electronic or paper license

The instructions in this guide describe how to assign an electronic license to a printer running the Embedded Solutions Framework (eSF). If you are using a non-eSF printer, then contact your Lexmark representative to obtain a paper license. To determine which type of printer you have:

- 1 Open a Web browser, and then type the printer IP address.

Note: View the IP address in the TCP/IP section of the Network/Ports menu.

- 2 Click **Settings** or **Configuration**.

Your printer supports eSF if you see any of the following links:

- Apps Management
- Solutions (eSF)
- Embedded Solutions

Collecting host ID information

For individual (local) licensing

- 1 Open a Web browser, and then type the printer IP address.

Note: View the IP address in the TCP/IP section of the Network/Ports menu.

- 2 Click **Reports** > **Device Information**.

- 3 Note the value associated with “Serial Number.”

For network licensing

The host ID for network licensing is the MAC address of the computer where the license server is located. To find the MAC address, contact your administrator, or do the following:

- 1 After the license server is installed, from the computer, navigate to the list of installed programs.
- 2 Click **Lexmark > License Server > License Administrator Tools > System Settings**.
- 3 Note the value associated with the Ethernet address (MAC address).

Note: The host ID can also be located from the command line by entering the license server installation path, followed by the `>lmutil hostid` command.

Installing a license server

A license server is intended for use with Windows operating systems.

- 1 From the Imaging Toolkit package, launch the **LicenseServer.exe** file.
This executable file is located in <install location>\esf-license-app.
- 2 Click **Next**.
- 3 Select the installation method you want to use, and then click **Next**.
- 4 If you want to install the server in an alternate location, then click **Browse**.
- 5 Click **Finish** to complete the installation.
- 6 Click **Done**.

Installing the application on your printer

- 1 Open a Web browser, and then type the printer IP address.
Note: View the IP address in the TCP/IP section of the Network/Ports menu.
- 2 Click **Settings** or **Configuration**.
- 3 Depending on your printer model, do one of the following:
 - Click **Apps > Apps Management**.
 - Click **Device Solutions > Solutions (eSF)**.
 - Click **Embedded Solutions**.
- 4 Click **Install a New App** or **Install**.
- 5 Browse to the Imaging Toolkit flash file at <install location>\esf-license-app.
- 6 Install the application.

Licensing the printer

Using a local license

- 1 Open a Web browser, and then type the printer IP address.
Note: View the IP address in the TCP/IP section of the Network/Ports menu.
- 2 Click **Settings** or **Configuration**.
- 3 Depending on your printer model, do one of the following:
 - Click **Apps > Apps Management**.
 - Click **Device Solutions > Solutions (eSF)**.
 - Click **Embedded Solutions**.
- 4 Click the license status of the application from the list.
- 5 Select **Local**, and then update the license.

Using a network license

Notes:

- This feature is available only in some printer models.
 - Before installing a network license, copy the license files to the license server.
- 1 Copy the license file to the following directory on the network license server:
C:\Program Files\Lexmark\LicenseServer\Licenses
 - 2 Open Control Panel.
 - 3 Click **Administrative Tools > Services > License Server**.
 - 4 Restart the **License Server** service.

With the Embedded Web Server

- 1 Open a Web browser, and then type the printer IP address.
Note: View the IP address in the TCP/IP section of the Network/Ports menu.
- 2 Click **Settings** or **Configuration**.
- 3 Depending on your printer model, do one of the following:
 - Click **Device Solutions > Solutions (eSF)**.
 - Click **Embedded Solutions**.
- 4 From the Network License tab, type the IP address or host name of the server and port where the license is stored.
- 5 Set the heartbeat period to specify how often the application searches for the network license server to check for updates. This value is also used to maintain access to an electronic license.
- 6 Specify the maximum number of times the application can try obtaining a license.
- 7 Click **Apply**.

- 8 From the list of solutions, click the license status of the application.
- 9 Select **Network**, and then update the license.

Configuring the application

Changing the Imaging Toolkit properties

The config.properties file stores the configuration settings for the Imaging Toolkit, and must be located in the same directory as LexImgTk-x.x.jar. If the Imaging Toolkit does not find the config.properties file, then it uses the default settings. The application properties are set using the following keys:

Note: Key values are case-sensitive.

Key	Description	Default value
FirstAvailableShortcut	If set to true , then the Imaging Toolkit creates a profile with the next available shortcut on the device. For more information, see the shortcut properties for “Scan settings” on page 13 .	true
MinShortcut	The minimum shortcut number. For more information, see the shortcut properties for “Scan settings” on page 13 .	1
MaxShortcut	The maximum shortcut number. For more information, see the shortcut properties for “Scan settings” on page 13 .	16535
DefaultPort	The listener port used by the Imaging Toolkit.	9750
ActivateUnreachableProfile	If set to true , then the Imaging Toolkit removes and resends the uploaded profile.	false
NumConsecutiveRetryPorts	If SinglePort is set to false , then this key value is the number of times to attempt to resend data when the default port is busy.	10
SinglePort	If set to false , then each profile uses a different port. If set to true , then all uploaded profiles use the port number specified in DefaultPort. Only one instance of the Imaging Toolkit runs in this specific port.	false
Cluster	To use the Imaging Toolkit in a cluster environment, set to true .	false
ClusterAddress	The IP address of the cluster.	No default value
ApplyDatabase	If you are using the Imaging Toolkit with a database, then set to true .	false
DataHost	The IP address or host name of the machine where the database is located.	localhost
DataPort	Use this key to change the port for the Firebird database service. This field is optional.	3050
DataSourcePath	The absolute path of the database location.	C:/Program Files/Lexmark/database/LITJar.FDB

Connecting to the database

The Imaging Toolkit must be able to connect to the LITJar.FDB database file.

Store LITJar.FDB in another location, and then change the following fields in config.properties:

- **ApplyDatabase**—Set this to “true” to use the Firebird database.
- **DataHost**—Specify the IP address or host name of the machine where LITJar.FDB is located.
- **DataPort**—If the Firebird database server is running on a port other than the default port, then specify the port number. This field is not required if using the default port.
- **DataSource**—Specify the absolute path of LITJar.FDB.

Notes:

- It is recommended that the operating system type and bit version be the same for the database server and the host computer from which the Imaging Toolkit connects to the database.
- Linux users must change the location settings in config.properties even if storing LITJar.FDB in the same directory as LexImgTk-x.x.jar.

Using the Imaging Toolkit in a cluster environment

To use the Imaging Toolkit in a Network Load Balancing cluster environment, change the following settings in the config.properties file:

- **SinglePort**—Set to “true.”
- **Cluster**—Set to “true.”
- **ClusterAddress**—Specify the IP address of the cluster.
- **ApplyDatabase**—Set to "true."
- **DataHost**—Specify the IP address of the machine where the database is located.
- **DataSourcePath**—Specify the absolute path location of the database.

Notes:

- The configuration file must be identical for all nodes in the cluster.
- In a cluster environment, the Imaging Toolkit cannot be executed from the command line.
- To run the Imaging Toolkit in a cluster environment, the Imaging Toolkit must be connected to the LITJar.FDB database file. For more information, see [“Connecting to the database” on page 13](#).

Scan settings

Notes:

- Some settings are available only in select devices. Contact your Lexmark representative for details.
- Some scan settings do not have default values.
- Default settings are marked with an asterisk (*).
- Scan settings are case-sensitive.
- Resolution, compression, depth, and type must be set such that they are compatible. Configuring these four settings in an incompatible way can lead to printer error. For information about incompatible scan settings, see [“Incompatible scan settings” on page 17](#).

Scan setting	Legal values
backgroundRemoval	-4 to 4
brightness	0 to 8 (4 is the default) Darkness on the scanned image can be controlled by adjusting this setting. Darkness is brightness+1.
colorBalanceBlue	-5 to 5
colorBalanceGreen	-5 to 5
colorBalanceRed	-5 to 5
compression	<ul style="list-style-type: none"> • G31D • G32D • G4 • JPEG* • PACKBITS • ZLIB • LZW • NONE
content	<ul style="list-style-type: none"> • MIXED • PHOTO • PHOTO CONTONE • TEXT • GRAPHIC <p>Note: Content is not a required field. The scan quality can be also be customized by changing the contrast.</p>
contrast	<ul style="list-style-type: none"> • 0 to 5 • MIXED • PHOTO • PHOTO CONTONE • TEXT* • GRAPHIC
depth	<ul style="list-style-type: none"> • 1 • 8* • 24

Scan setting	Legal values
documentSource	<ul style="list-style-type: none"> • BLACK AND WHITE LASER* • COLOR LASER • INKJET • PHOTOGRAPH • MAGAZINE • NEWSPAPER • PRESS • GENERAL OFFICE <p>Note: This field is optional. It is required only if Content is set to GRAPHIC.</p>
duplex	<ul style="list-style-type: none"> • SINGLE*/SIMPLEX • BOTH • DUPLEXSHORTEDGE • DUPLEXLONGEDGE
edgeToEdge	<ul style="list-style-type: none"> • TRUE • FALSE*
fileType	<ul style="list-style-type: none"> • JPEG* • TIFF • PDF • PS • RAW
invertImage	<ul style="list-style-type: none"> • TRUE • FALSE*
jobBuild	<ul style="list-style-type: none"> • TRUE • FALSE*
jpegQuality	5 to 90(80 is the default)
linearXfer	<ul style="list-style-type: none"> • TRUE • FALSE*
mirrorImage	<ul style="list-style-type: none"> • TRUE • FALSE*
multiPageTiff	<ul style="list-style-type: none"> • TRUE • FALSE*
orientation	<ul style="list-style-type: none"> • PORTRAIT* • LANDSCAPE

Scan setting	Legal values
paperSize	<ul style="list-style-type: none"> • 11x17 • 3x5 PHOTO • 4x6 PHOTO • A3 • A4 • A5 • A6 • AUTO SIZE • B3 • B4 • B5 • BOOK • BOOK ORIGINAL (same as BOOK) • BUSINESS CARD • EXECUTIVE • FOLIO • ID-1 • LEGAL • LETTER* • MIXED SIZES • OFICIO • STATEMENT • TABLOID (same as 11x17) • UNIVERSAL
pdfVersion	<ul style="list-style-type: none"> • 1.2 • 1.3 • 1.4 • 1.5 • 1.6 • A-1a
resolution	<ul style="list-style-type: none"> • 75 • 150* • 200 • 300 • 400 • 600 • 1200
scanPreview	<ul style="list-style-type: none"> • TRUE • FALSE*

Scan setting	Legal values
scanRecovery	<ul style="list-style-type: none"> • TRUE • FALSE*
shadowDetails	-4 to 4
shortcut	<p>The default value is 0.</p> <p>For FirstAvailableShortcut=FALSE:</p> <ul style="list-style-type: none"> • If the shortcut value is 0, then the Imaging Toolkit creates a profile without any shortcut. • If the shortcut specified is already in use, then a profile is not created. • If the shortcut specified is available, then a profile is created with the given shortcut value. <p>For FirstAvailableShortcut=TRUE:</p> <ul style="list-style-type: none"> • If the shortcut value is 0, then a profile is created with any available shortcut value from a specified range. • If the shortcut specified is already in use, then a profile is created with any available shortcut value from a specified range. • If the shortcut specified is available, then a profile is created with the given shortcut value.
source	<ul style="list-style-type: none"> • ADF • FLATBED • ANY • ADF1PAGE

Incompatible scan settings

Some scan settings do not work well in combination, and results will vary. While others may exist, the following combinations are known to be incompatible:

Scan setting	Does not work with
compression= NONE	<ul style="list-style-type: none"> • TYPE = JPEG • TYPE = TIFF • TYPE = PS • TYPE = RAW
compression = JPEG	TYPE = TIFF
compression = ZLIB	<ul style="list-style-type: none"> • TYPE = JPEG • TYPE = TIFF • TYPE = PS • TYPE = RAW
compression = PACKBITS	<ul style="list-style-type: none"> • TYPE = JPEG • TYPE = PDF • TYPE = PS • TYPE = RAW

Scan setting	Does not work with
compression = LZW	<ul style="list-style-type: none"> • TYPE = JPEG • TYPE = PDF • TYPE = PS • TYPE = RAW
depth = 1	TYPE = JPEG
resolution = 1200	Not supported; results will vary. Some printers will default to a scan resolution of 150 dpi if this value is used.

Custom scan sizes

Size	Actual size of scanned image
ID CARD	This will scan the image to Custom Scan Size 1 paper size.
CUSTOM SCAN SIZE 2	This will scan the image to Custom Scan Size 2 paper size.
CUSTOM SCAN SIZE 3	This will scan the image to Custom Scan Size 3 paper size.
CUSTOM SCAN SIZE 4	This will scan the image to Custom Scan Size 4 paper size.
CUSTOM SCAN SIZE 5	This will scan the image to Custom Scan Size 5 paper size.
CUSTOM SCAN SIZE 6	This will scan the image to Custom Scan Size 6 paper size.
Note: Size names are not case sensitive.	

You can also rename Custom Scan Sizes, or specify new scan size settings. The Toolkit will treat the renamed Custom Scan Size as a valid paper size name when scanning. To change Custom Scan Size names or settings:

- 1 Type the IP address or host name of the printer in the address field of your Web browser to access the printer Embedded Web Server.
- 2 From the Embedded Web Server home screen browse to **Settings > Paper Menu > Custom Scan Sizes**.
- 3 Click **ID Card** or **Custom Scan Size (1-6)**.
- 4 Use the Settings screen to specify the settings for your Custom Scan Size.
 - Scan Size Name—Leave blank to use the default label, or type a new name for your custom scan (64-character maximum).
 - Width—Type a number for the width (in inches) of your scan (1.00-8.50).
 - Height—Type a number for the length (in inches) of your scan (1.00-14.00).
 - Orientation—Use the menu to choose between Portrait and Landscape.
 - 2 scans per side—Click the check box to enable the printer to place two scanned images on the same side of the scanned output.
- 5 Click **Submit** to save changes and return to the Settings screen, or **Reset Form** to clear your changes.

Note: If you use a standard name such as Letter or Legal as a Custom Scan Size Name, the Toolkit will use the dimensions specified in the Custom Scan Settings when performing a scan.

File types created by the Imaging Toolkit

Image file

The image file that is returned when the scan completes has the same base name as the file name passed back to the JAR file.

If the file type does not support multiple-page files (for example, JPEG), then one file is returned for each page of the scan. The image contains the same base name, followed by a number indicating the page number of the scan (for example, image1.jpg).

Error file

The file has the same base name as the file name passed to the JAR, appended with a *.err extension. The file contains a description of the error that occurred.

- For API-related errors, the error file is located in the directory you specified as the output directory when you created the context.
- For command line-related errors, the error file is located in the same directory as the command line where you called the JAR file.

Note: You cannot change the output location of the error file.

Using the application API

Required functions

The Imaging Toolkit allows applications to perform the imaging process without any specific knowledge of how to communicate with a printer. With the Imaging Toolkit, you need only to create an instance of LITScan, and then call those APIs to complete the imaging process. The Imaging Toolkit API consists of public functions of the class LITScan, not all of which are needed for every application. This Imaging Toolkit supports profile names and file names that can contain ISO-8859-1/2/5 characters.

In order for the imaging process to take place, you must create a profile and receive the scanned image.

Creating a profile

A profile describes scan job parameters such as the contrast, depth, duplex, orientation, resolution, size, and type. The profile also determines where the scanned image is sent. This is specified by an IP address and a TCP port number.

As part of creating a profile, a socket must be created to listen for a connection on the TCP port that is specified by the profile.

Note: Make sure the socket is created *before* you send the profile to the printer. This way, the socket can be bound to the exact port number that was sent to the printer.

Receiving a scanned file

When you select the profile on the printer control panel, the printer connects to the socket and sends the image(s), along with metadata. Metadata is not part of the actual image data, but is rather information used as a communication tool between the printer and the computer.

For example, if you scan more than one page in a format that does not allow multiple pages, the printer sends each page as a separate file with metadata saying the file is complete, and another file is starting. When the printer is done sending the images, it closes the connection and the process is complete.

Basic API functions of LITScan

Basic API	What it does
<pre>public LITScan() throws Exception</pre>	<p>This creates an instance of LITScan after initializing the Toolkit with the default configuration. The default configuration file, config.properties, should reside in the same location where the .jar file resides. If the Toolkit fails to load the file, then the Toolkit initializes with the default values.</p> <p>During initialization, the Toolkit:</p> <ul style="list-style-type: none"> • configures the library with the database connection and then connects to the library. • if used in a cluster environment, sets the Toolkit for cluster. If SinglePort=TRUE, then this API creates a receiving port using the port number specified in the configuration file. Since only a single socket can be opened with a specific port number, another instance of LITScan with the same port number cannot be created. <p>Exceptions:</p> <ul style="list-style-type: none"> • SQLException, if the database connection fails • Socket-in-use exception, if a server socket cannot be created
<pre>public LITScan(String configFile) throws Exception</pre>	<p>This creates an instance of LITScan with a user-defined configuration. The user application can pass the user-defined path where the configuration file resides. If the Toolkit fails to load the file, then the Toolkit initializes with the default values.</p>
<pre>public ScanInfo[] getInitScanContext()</pre>	<p>This returns the context ID, MFP IP address, output location, and profile name of the scan contexts that were not released by the Toolkit. This API is generally used after creating an instance of the LITScan as it returns the scan contexts created at the previous instance. For more information, see “Persistency and removing an unreachable profile” on page 24.</p>

Basic API	What it does
<pre>public long createContext (final String ipAdd, final String profile, final String filename) throws Exception</pre>	<p>This creates a data structure called a scan context, which contains information about the MFP, scan settings, and label name, among others, and returns a unique long integer value that identifies the scan context.</p> <p>To use this API, the user has to send the following parameters:</p> <ul style="list-style-type: none"> ipAdd—This is the IP address of the MFP used to upload the scan profile. profile—This is the name of the scan profile on the MFP. filename—This is the full output file name for stored images. <p>Exceptions:</p> <ul style="list-style-type: none"> • <code>IllegalArgumentException</code>, if the IP address of the MFP, profile name, or directory is not valid • “Illegal MFP” message, if the MFP is not found on the network • “Image acquisition failed. Unlicensed MFP” message, if the MFP is not licensed
<pre>public long createContext (final String ipAdd, final String profile, final int port) throws Exception</pre>	<p>This creates a data structure called a scan context which contains information about the MFP, scan settings, and label name, among others, and returns a unique long integer value that identifies the scan context.</p> <p>To use this API, the user has to send the following parameters:</p> <ul style="list-style-type: none"> ipAdd—This is the IP address of the MFP used to upload the scan profile. profile—This is the name of the scan profile on the MFP. port—This is the number of the port that receives the images. <p>Note: If zero is passed as a TCP port number, then the image is scanned on the first available port.</p> <p>Exceptions:</p> <ul style="list-style-type: none"> • <code>IllegalArgumentException</code>, if the IP address of the MFP, profile name, or directory is not valid. • “Illegal MFP” message, if the MFP is not found on the network. • “Image acquisition failed. Unlicensed MFP” message, if the MFP is not licensed.
<pre>public void setObserver(final Observer observer)</pre>	<p>This sets the user application as an observer that gets notifications of scans from the Toolkit.</p>
<pre>public void setSettings(final long contextID, Map sm) throws Exception</pre>	<p>This lets users change the multiple scan settings of a particular scan context specified by the context ID at any one call by passing the map of the key/value pair. "Key" is the setting name, and "value" is the supported scan value for that field. For more information about scan settings that the Toolkit supports and their supported values, see “Scan settings” on page 13.</p> <p>Exception:</p> <ul style="list-style-type: none"> <code>IllegalArgumentException</code> exception is thrown, if the new <code>Map</code> fails to pass the scan settings verification.

Basic API	What it does
<pre>public Map getSettings(final long contextID)</pre>	<p>This returns the map that holds the scan settings and its value in key/value pair of the scan context identified by the index. Index is the unique long integer value returned by the createContext API.</p>
<pre>public void cancelScan(final long contextID)</pre>	<p>This cancels the scan job of the scan context identified by the context ID.</p>
<pre>public void existScan()</pre>	<p>This terminates the entire operation handled by the Toolkit.</p>
<pre>public int removeProfile(final long contextID) throws Exception</pre>	<p>This removes the uploaded profile from the MFP that is specified in the scan context.</p> <p>Parameter: contextID—This is the ID of the scan context which is identified by a unique long integer value.</p> <p>Exception: ObjectStoreException, if the API fails to remove the profile from the MFP</p>
<pre>public int removeProfile (String profileName, String mfpIPAddress) throws Exception</pre>	<p>This removes the uploaded profile from the MFP that is specified in the profile name and MFP IP address in the parameters.</p> <p>Parameter: profileName—This is the name of the scan profile on the MFP. mfpIPAddress—This is the IP address of the MFP.</p> <p>Exception: ObjectStoreException, if the API fails to remove the profile from the MFP</p>
<pre>public void scanBlock(final long contextID) throws Exception</pre>	<p>This creates and sends a scan profile to the printer, and saves the returned image after the profile is executed. Until the profile is executed or cancelled, this API does not return to the user application. This API cannot be used if SinglePort is set to TRUE.</p> <p>Parameter: contextID—This is the ID of the scan context which is identified by a unique long integer value.</p> <p>Exception: “Scan Context is in use” message, if the specified scan context is currently in operation</p>
<pre>public void scanNoBlock(final long contextID) throws Exception</pre>	<p>This creates and sends a scan profile to the printer, and saves the returned image after the profile is executed. This is a non-blocking call.</p> <p>Parameter: contextID—This is the ID of the scan context which is identified by a unique long integer value.</p> <p>Exception: “Scan Context is in use” message, if the specified scan context is currently in operation</p>

Basic API	What it does
<pre>public void releaseContext(final long contextID)</pre>	<p>This deletes a specified scan context. If you are using the Toolkit with a database, then this deletes data from the database.</p>
<pre>public void setOutputFile(final long contextID, final String directory, final String fileName)</pre>	<p>This provides the output path of the directory where the scanned image file is stored.</p>
<pre>public void setPortRange (int lowerPort, int upperPort)</pre>	<p>This is used as an API to set lower and upper port range values (between 1024 and 65535).</p> <p>If port range settings are left blank, then the server will attempt to open a socket on the first available port. If both the lower and upper port numbers are set to -1, then the port range function will be disabled, and any previously set port ranges will be deleted.</p> <p>An <code>IllegalArgumentException</code> will occur with the message "Illegal port range" if:</p> <ul style="list-style-type: none"> • The lower port number is less than 1024. • The upper port number is greater than 65535. • The lower port number is greater than the upper port number.

Receiving all images on the same port

You can set the Imaging Toolkit to receive all scanned images on a specific port using the `SinglePort` setting. To specify a particular port, change the following settings in `config.properties`:

- **SinglePort**—Set to **true**.
- **DefaultPort**—Specify the number of the port where all images should be received.

Notes:

- If `SinglePort` is set to "false," then each profile will send scanned images to a different port. To launch multiple instances, you must have a separate configuration file specifying a separate database for each instance.
- If `SinglePort` is set to "true," then only one instance of LITScan is created per application.

Persistency and removing an unreachable profile

Users can store the created scan context to the Firebird database. If `ApplyDatabase` is set to "true" in the configuration file and database connection is successful, then the scan context data will be stored in the database. It will remain in the database until you call the **releaseContext** API.

This stored information can be retrieved by the user using the **ScanInfo[] getInitScanContext()** API. This API returns all the scan context that has been created in the previous instance of the Imaging Toolkit as a list of `ScanInfo` objects. Each `ScanInfo` object contains the scan context ID, profile name, MFP IP address, and output directory of the previously created scan context. Use this API after creating an instance of LITScan during the initialization of the application.

The attributes of `ScanInfo` are `contextID` (long), `mfpIpAddress` (string), `outputLocation` (string), and `ProfileName` (string). Users can access them by using the following functions:

ScanInfo class function	What it does
<code>public long getContextID()</code>	Returns the scan context ID as a long integer
<code>public String getMfpIPAddress()</code>	Returns MFP IP address of the scan context
<code>public String getOutputLocation()</code>	Returns the output location of the scan context
<code>public String getProfileName()</code>	Returns the name of the profile

Profiles that are uploaded to the MFP and not removed if the Imaging Toolkit unexpectedly shuts down are no longer executable. These profiles are called steal or unreachable profiles. If `ActivateUnreachableProfile` is set to "true," then the API removes the profiles and resends them to be executed when launching the application again.

Creating an application

- 1 Create an instance.
- 2 Create a context.
- 3 If applicable, set the port range.
- 4 Specify scan settings.
- 5 If applicable, set the Observer.
- 6 Call one of the scan functions.
- 7 Get previously created scan context.

Note: All examples in the following sections are shown using Java.

Creating an instance

In order to use `LexImgTk-x.x.jar`, you must:

- 1 Create an instance variable:

```
LITScan litScan;
```

- 2 Create an instance where **`config_pro_file_path`** is the absolute path of the `config.properties` file:

```
litScan=new LITScan(config_pro_file_path);
```

You can also create the instance using the default configuration:

```
litScan=new LITScan();
```

Note: A default configuration cannot be used in a Linux environment, because the location of the `config.properties` file must be passed.

Creating a context

```
litScan.createContext().
```

This function returns a long integer value that is used as a handle to refer to a particular scan.

In order to create the handle, you must provide a valid printer address, a profile name, and a TCP port number for the Toolkit to listen on for printer connections. The function will use the default scan values for each created scan context.

Setting	Default value
brightness	4
colorDepth	8
compression	JPEG
contrast	TEXT
duplex	SINGLE
edgeToEdge	FALSE
jobBuild	FALSE
jpegQuality	80
linearXfer	FALSE
orientation	PORTRAIT
paperSize	LETTER
resolution	150
scanPreview	FALSE
shortcut	0
type	JPEG

Sample code for creating a context

```
String profile = profileText.getText();
String ipAddress = mfpIPText.getText();
.....
index = litScan.createContext(ipAddress, profile, 9750);
```

Setting the port range

The Set Port Range function is called to designate which ports the Toolkit will use for receiving images. Lower and upper port range values are passed as a parameter, and must be set between 1024 and 65535. If both the lower and upper port numbers are set to -1, the port range function will be disabled, and any previously set port range values will be deleted.

```
if(isEmptyLowerPort && isEmptyUpperPort){
    try{
        litScan.setPortRange(-1,-1);
        .....
    }
else if(!isEmptyLowerPort && !isEmptyUpperPort &&
        !isNotIntegerLowerPort && !isNotIntegerUpperPort){
    try{
        litScan.setPortRange(lowPort, upperPort);
        .....
    }
}
```

Specifying scan settings

The Imaging Toolkit allows scan parameters to be set individually for each context. Call `litScan.setSettings()` to adjust settings, or `litScan.getSettings()` to view the current set of parameters for a given context.

Note: Scan parameters *are* case-sensitive. For more information about legal values for scan settings, see [“Scan settings” on page 13](#).

Sample code for specifying scan settings using `litScan.getSettings()` and `litScan.setSettings()`

```
Map temp=new HashMap();

Map sm=litScan.getSettings(idScanBack);

temp.putAll(sm);
.....
temp.put("resolution","200");
.....

litScan.setSettings(idScanBack,temp);
```

Calling the scan functions

```
litScan.scanBlock()
litScan.ScanNoBlock()
```

The two scan function options enable you to scan with or without a blocking function. The non-blocking version allows the application to respond to your input. When you call the function, the profile is sent to the printer and the Toolkit waits for the printer to connect back to it.

Sample code to call the API for non-blocking scanning

```
String str=(String)model.getElementAt(selectedIndex);
int idScanBack=Integer.parseInt(str);

litScan.scanNoBlock(idScanBack);
```

Getting a previously created scan context

You can use the following sample code to display a previously created context in the context list when the application is launched:

```
ScanInfo[] scList=lit.getInitScanContext();
.....
while(index< scList[index].getContextID();
listModel.addElement(String.valueOf(contextID));
String profile=scList[index].getProfileName();
String ipAddress=scList[index].getMfpIPAdress();
String fileName=scList[index].getOutputLocation();
.....
}
```

Using the sample application

Using the Java Swing sample

The sample application is a Java Swing application that uses LexImgTk-x.x.jar to send scan profiles to a printer and receive the scanned images back from the printer. The LITScan APIs exposed by LexImgTk-x.x.jar consist of public functions of the class LITScan that facilitate the imaging process. The sample application can be run from the command line, as long as LexImgTk-x.x.jar has been included in the CLASSPATH.

Executing the application

- 1 Create a context by specifying the following:
 - The IP address or host name of the destination printer
 - A profile name

Once this information has been typed into the appropriate fields, click **Create Context**. An integer should appear in the Contexts list. Create Context calls the LITScan.createContext function to create the context and the LITScan.setObserver function to enable your application to receive status information from the JAR file.

- 2 Set the port range.

Lower and upper port range values can be set by typing a value (between 1024 and 65535) in the corresponding field and clicking **Set Port Range**.

If port range settings are left blank, then the server will attempt to open a socket on the first available port. In this case, the port range function will be disabled, and any previously set port ranges will be deleted.

After a context has been created, default values will be shown for all scan settings.

- 3 Adjust scan settings, if necessary, by typing new values in the appropriate fields, and then click **Set Scan Values** to call the LITScan.setSettings function. Invalid settings will not be saved. For more information about legal values for scan settings, see [“Scan settings” on page 13](#).

Note: To test whether your new settings have been accepted, click **Get Scan Values** to call LITScan.getSettings and display the saved values.

- 4 Type the full path of the icon file.

If the printer has a control panel with a welcome screen, then an icon will be sent to the printer and placed on the welcome screen.

Note: Icon images can be GIF, JPEG, or PNG, with dimensions of 120 x 75 pixels that is 11KB or smaller.

- 5 Click **Start Scan** to send a profile to the printer by calling the LITScan.scanNoBlock() function. This is a non-blocking call. Updates on status come through the observer, which outputs text into the output window.
- 6 Initiate a scan at the printer by choosing the appropriate profile name, shortcut number, or icon on the printer control panel.

Additional options and functionality

Click	To
Cancel Scan	Cancel the scan job.
Release Context	Release any resources used by the scan context.
Remove Profile	Delete the profile currently selected in the profile name field.

Click	To
Exit	Terminate the application.

Possible API-related error messages

The following is a list of possible error messages that you can receive while using the Imaging Toolkit. Information contained in brackets is a variable, and is filled in by the code when the error occurs.

Note: Only Toolkit-specific error messages are provided. Messages resulting from unexpected behavior within Java are not listed.

- Unable to remove profile: *<profileName>*
- Failed to load default file *<DefaultFileName>*
- Scan Settings failed verification.
- Illegal port range.
- Invalid output path or directory does not exist.
- Image acquisition failed. No available socket ports.
- Image acquisition failed. Profile failed to upload.
- Image acquisition failed. Connection failure.
- Image acquisition failed. Socket exception.
- Image acquisition failed. Unlicensed MFP.
- Image acquisition failed. Profile missing from MFP.
- Failed to copy temporary file.
- Failed to rotate image.
- User canceled.
- Corrupt defaults file.
- *<propertyName>* *<value>* is not supported. Use one of the following: *<list of legal values>*
- Wrong jpeg compression.
- Shortcut *<shortcut value>* cannot be a negative number.

Possible OptralImage error messages

The following error messages are specific to OptralImage™ printers only:

- Application error
- MFP Error - Bad Field
- MFP Error - Max Storage Reached
- MFP Error - Function Disabled
- MFP Error - Duplicate Shortcut

Using the application from the command line

Using the JAR file

When LexImgTk-x.x.jar is run from the command line, a scan profile is sent to the designated MFP. A listening port on the host computer remains open until the end user visits the MFP, selects the profile, and initiates a scan. Once the scan is initiated, scan data is sent to the specified output directory. There are eight parameters that can be used with the JAR to facilitate the imaging process, but only two are required.

Variable	Description
address (required)	This is the IP address of the printer that scans the document.
file (required)	This is the file name of the image or images to be returned by the printer. Note: The file format is specified in the Scan Settings property file. If a file extension is supplied here, then it is ignored.
-l <i>label</i>	This is the text to display on the printer control panel for the profile. If a label is not supplied, then the file name is used.
-p <i>settings_file</i>	This is the name of the Scan Settings property file. If a file name is not specified, then the JAR file looks for a file named "LexmarkImagingToolkit.properties." If that file is not found, then the default values are used.
-s <i>shortcut_number</i>	This is the shortcut number for the profile created on the printer.
-r <i>label</i>	This removes the icon and profile for the specified label.
-v	This returns the version number for the LexImgTk-x.x.jar file.
-k	This specifies the port range in which the server socket can be created.

Note: When run from the command line, optional arguments such as [shortcut_number] will override corresponding values present in the scan properties file.

JAR file syntax

Purpose	Correct syntax	Example
To call the JAR file	java -jar LexImgTk-x.x.jar [address] [file] -l [label] -p [settings_file] -s [shortcut_number] -k [lower_port-upper_port]	java -jar LexImgTk-3.2.jar 198.162.0.10 "C:\temp\MyNewImage" -l "My Job" -p "C:\temp\MySettings.txt" -s "7" -k "2100-2200"
To remove a profile	java -jar LexImgTk-x.x.jar [address] -r [label]	java -jar LexImgTk-3.2.jar 198.162.0.10 -r "My Job"
To list the version number of the JAR file	java -jar LexImgTk-x.x.jar -v	java -jar LexImgTk-3.2.jar -v

Note: Make sure to replace **x.x** in **LexImgTk-x.x.jar** with the latest version number of the JAR file.

Creating a scan settings property file

If you want to specify multiple scan settings for the profile, then you can do so with a scan settings property file. Scan parameters *are* case-sensitive. For more information about legal values for scan settings, see [“Scan settings” on page 13](#). To create a scan settings property file:

- 1 Create a text file with the settings you want.
- 2 Place the text file in a folder on your computer.
- 3 When you execute the JAR file, point to the scan settings file you created.

Note: If you do not specify the location of the scan settings property file when you execute the JAR file, then the LexmarkImagingToolkit.properties file is located. If that file is not found, then the default settings are used.

You can also update the scan settings by using the LexmarkImagingToolkit.properties file contained within the JAR file.

- 1 Extract LexmarkImagingToolkit.properties from the LexImgTk-x.x.jar file.
- 2 Save the file to a local directory.
- 3 Update the file with the current settings you need.

Possible command line-related error messages

The following is a list of possible error messages that can occur while using the Imaging Toolkit from the command line. Information contained in brackets is a variable, and is filled in by the code when the error occurs.

Note: Only Toolkit-specific error messages are provided. Messages resulting from unexpected behavior within Java are not listed.

- Missing required command line argument: ip
- Using Defaults - Missing required command line argument: path/file
- Unable to remove profile: *<profileName>*
- Illegal command line option *<arg>*
- Unknown command line option *<arg>*
- Failed to load default file *<DefaultFileName>*
- Failed to load properties file *<propFileName>*
- Scan Settings failed verification.
- Illegal port range.
- Invalid output path or directory does not exist.
- Image acquisition failed. No available socket ports.
- Image acquisition failed. Profile failed to upload.
- Image acquisition failed. Connection failure.
- Image acquisition failed. Socket exception.
- Image acquisition failed. Unlicensed MFP.
- Image acquisition failed. Profile missing from MFP.
- Failed to copy temporary file.
- Could NOT find file *<DefaultFileName>*, make sure to include its path. ERROR: *<file not found error details>*

- Failed to rotate image
- User canceled.
- Corrupt defaults file
- Using Default Settings – *<file not found error details>*
- *<propertyName> <value>* is not supported. Use one of the following: *<list of legal values>*
- Wrong jpeg compression
- Shortcut *<shortcut value>* cannot be a negative number.

Possible JAR file error messages

Note: The following error messages are specific to OptraImage printers only:

- Application error
- MFP Error – Bad Field
- MFP Error – Max Storage Reached
- MFP Error – Function Disabled
- MFP Error – Duplicate Shortcut
- MFP Error – Invalid Value for Scan Type
- MFP Error – Bad IP Address
- MFP Error: *<error returned by MFP>*

Notices

Edition notice

August 2015

The following paragraph does not apply to any country where such provisions are inconsistent with local law: LEXMARK INTERNATIONAL, INC., PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in later editions. Improvements or changes in the products or the programs described may be made at any time.

References in this publication to products, programs, or services do not imply that the manufacturer intends to make these available in all countries in which it operates. Any reference to a product, program, or service is not intended to state or imply that only that product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any existing intellectual property right may be used instead. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by the manufacturer, are the user's responsibility.

For Lexmark technical support, visit <http://support.lexmark.com>.

For information on supplies and downloads, visit www.lexmark.com.

© 2015 Lexmark International, Inc.

All rights reserved.

GOVERNMENT END USERS

The Software Program and any related documentation are "Commercial Items," as that term is defined in 48 C.F.R. 2.101, "Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. 12.212 or 48 C.F.R. 227.7202, as applicable. Consistent with 48 C.F.R. 12.212 or 48 C.F.R. 227.7202-1 through 227.7207-4, as applicable, the Commercial Computer Software and Commercial Software Documentation are licensed to the U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein.

Trademarks

Lexmark and the Lexmark logo are trademarks of Lexmark International, Inc., registered in the United States and/or other countries.

Optralmage is a trademark of Lexmark International, Inc.

Windows, Windows Server, Windows Vista, and Visual Basic are either registered trademarks or trademarks of the Microsoft group of companies in the United States and other countries.

All other trademarks are the property of their respective owners.

Index

A

API
calling the scan functions 27
creating a context 25
creating an instance 25
error messages 29
setting port range 25
specifying scan settings 27
API commands
for LITScan 21
application
installing 9
application properties 12
application settings 12

C

cluster
supported 4
cluster environment
using the Imaging Toolkit in 13
command line
error messages 31
JAR file syntax 30
scan settings property file 31
using the JAR file 30
configuration settings 12
context
creating for API 25
custom scan sizes 18

D

database
connecting to 13
determining printer license 8

E

error files 19
error messages
API-related 29
command line 31

F

file types
error files 19
image files 19

G

getting a previously created scan context 27

H

host ID
for individual licenses 8
for network licenses 8

I

image files 19
images
receiving on the same port 24
implementing the application
creating a profile 20
creating a socket 20
receiving a scanned file 20
incompatible scan settings 17
individual licenses 8
installing the application 5, 9
instance
creating for API 25

J

JAR file
using from the command line 30
JAR file syntax 30
Java Swing sample code 28
Java Virtual Machine
supported 4

L

license
electronic 8
individual 8
network 8
paper 8
license server
installing 8
using to find host ID 8
licensing the application 8
licensing your printers
network 10
LITJar.FDB 13
LITScan
basic API commands 21

local license
using 10

N

network licenses 8

O

operating systems
supported 4
overview 3

P

persistence 24

S

sample application 28
sample code 28
scan functions
calling for API 27
scan settings 13
incompatible 17
property file for command line 31
specifying for API 27
ScanInfo API 24
SinglePort 24
supported cluster 4
supported Java Virtual Machine 4
supported operating systems 4

U

unreachable profiles 24