





# **Datalogics**

DATALOGICS JAVA INTERFACE

User Guide

This guide is part of the *Adobe PDF Library v8.0* suite; 06/25/07.  
Copyright 1999-2007 Datalogics Incorporated. All Rights Reserved. Use of Datalogics software is subject to the applicable license agreement.

*Datalogics Java Interface* is a trademark of Datalogics Incorporated. Other products mentioned herein as Datalogics products are also trademarks or registered trademarks of Datalogics, Incorporated.  
Adobe, *Adobe PDF Library*, Portable Document Format (PDF), PostScript, *Acrobat*, *Distiller*, *Exchange* and *Reader* are trademarks of Adobe Systems Incorporated.

Microsoft, Windows and Windows NT are trademarks or registered trademarks of Microsoft Corporation.  
IBM, AIX, AS/400, OS/400, MVS, and OS/390 are registered trademarks of International Business Machines.

HP and HP-UX are registered trademarks of Hewlett Packard Corporation.

SAS/C is a registered trademark of SAS Institute Inc.

Java, J2EE, J2SE, J2ME, all Java-based marks, Sun and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group.

Linux is a registered trademark of Linus Torvalds.

All other trademarks and registered trademarks are the property of their respective owners.

For additional information, contact:

Datalogics, Incorporated  
101 North Wacker Drive, Suite 1800  
Chicago, Illinois 60606-7301

Phone: 312-853-8200

Fax: 312-853-8282

[www.datalogics.com](http://www.datalogics.com)  
[dlcomments@datalogics.com](mailto:dlcomments@datalogics.com)

# Table of Contents

---

<b>1 Getting Started</b>		<b>1.1</b>
An Introduction to the <i>Java Interface</i>	1.2	
What You Should Know	1.2	
How to Use this Book	1.3	
Related Documentation	1.4	

---

<b>2 Overview of the Java Interface</b>		<b>2.1</b>
Requirements	2.2	
Installation	2.2	
What's in the package	2.2	

---

<b>3 Samples Building and Running</b>		<b>3.1</b>
Building Sample Programs	3.2	
Running Sample Programs	3.2	

---

<b>4 Packages and Method Translation</b>		<b>4.1</b>
Packages	4.2	
Method Translation	4.2	

---

<b>5 Exceptions and Error Handling</b>		<b>5.1</b>
PDFLException Handling	5.2	



# Getting

## *Started*

This chapter introduces the *Datalogics Java Interface*.

## An Introduction to the *Java Interface*

The *Datalogics Java Interface* provides a Java-language wrapper to the *Adobe PDF Library* to facilitate the rapid creation of PDF documents via Java calls.

The *Adobe PDF Library* is C-based, not Java-sourced, but as it is usable by any application type which supports calls to a C-based library, the *Datalogics Java Interface* has been created.

For information on the various layers of *Adobe PDF Library* please see the Acrobat Core API Overview.

---

**NOTE:** Current releases of the *Java Interface* do not include *Datalogics Interface (DLI)* methods. These may be added in the future.

## What You Should Know

This document is intended for programmers who are familiar with the Java language, text composition and the creation of output drivers, or by application designers who are constructing an application based on the *Adobe PDF Library*.

You should have access to the *Adobe PDF Library Applications Programming Interface (API)* manual and the *Adobe PDF Specifications* manual for your system. You should find these documents provided within your release, accessible via the `referencelibrary.pdf` document using the copy of *Adobe Reader* provided (or any other PDF viewer utility).

For *Adobe PDF Library v7.x* releases, Adobe PDF Specification 1.6 is appropriate. For *Adobe PDF Library v8.x* releases, Adobe PDF Specification 1.7 is appropriate.

---

**NOTE:** Some structures permitted in Adobe PDF Specification 1.7 may not be permitted in Adobe PDF Specification 1.6, some structures defined in Adobe PDF Specification 1.6 are not available in Adobe PDF Specification 1.5, and so on.

The explanations, assumptions and samples provided in this guide refer to *Adobe PDF Library v8.0Plus* and *DLI v8.0* or higher.

## How to Use this Book

This book has been created to guide you through the process of creating PDF documents with the *Datalogics Java Interface*.

This chapter, Getting Started, outlines the chapters to follow, explains the document conventions used here, and lists other related documentation which you may find useful for your work. Follow-on chapters will outline the steps needed to create PDF and introduce the *Java Interface* in relation to the *Adobe PDF Library*, explain the methods used and how they fit together, and provide various samples.

The following list provides an outline of the chapters as well as a brief description of their contents. Click on each Chapter title below to jump to its first page.

Chapter 1: **"Getting Started"** (*This chapter*) This introduces the *Java Interface* and describes the contents of this book.

Chapter 2: **"Overview of the Java Interface"** gives a general explanation of the Java Interface to the *Adobe PDF Library*.

Chapter 3: **"Samples Building and Running"** describes the accompanying Java Interface sample applications.

Chapter 4: **"Packages and Method Translation"** discusses package and method handling within the Java Interface.

Chapter 5: **"Exceptions and Error Handling"** outlines the expected mechanism for catching and dealing with returned exceptions, which should be added to your application.

### Document Conventions

The terms *note*, *link* and *bookmark* are used in this book the same way they are in the user interface of *Adobe PDF Library v8.xPlus*®, *Adobe Acrobat*® and *Adobe Reader*®. These correspond to the *text annotation*, *link annotation* and *routine entry* structures (respectively) that appear in a PDF file. See the Portable Document Format Reference Manual for a description of the PDF file format.

The following documentation conventions appear throughout the manual to help you differentiate regular text from product and program names, and to distinguish command syntax.

- Product and program names are set in *italic* type.
- Multi-line examples are separated from the text and set in  
`Courier monospace`
- Directory names and filenames are contained within the text and set in `Courier monospace`.
- Commands are contained within the text and set in `Courier monospace`.
- New terms are *italicized*.
- Page numbers in this book do not correspond to page numbers in the PDF file. The numbering scheme (e.g. 4.1 or A.10) indicates the chapter number (4) or appendix letter (A) first, followed by the page number (1 or 10), separated by a period.

## Related Documentation

The following documents will be useful in developing applications using *Adobe PDF Library* and *DLI*.

### Datalogics Resources

**Adobe PDF Library and DLI Installation Guide** This document describes the installation requirements for using the *Adobe PDF Library* and *DLI* on the various platforms to which Datalogics has ported these products.

**Adobe PDF Library Developer Overview** This document is designed to aid developers with incorporating the API calls for the *Adobe PDF Library* into their composition application.

**DLI Implementation and Reference Guide** This document details the *Datalogics Interface*, a simplified interface to the COS Layer of the *Adobe PDF Library*.

**Java Interface User Guide** (*This book*) This document outlines the *Datalogics Java Interface*, a Java-language wrapper interface to the *Adobe PDF Library* and *DLI*.

## Adobe Resources

The following documents are distributed by Adobe as part of the original *Adobe PDF Library* release, and are redistributed by Datalogics without alteration. (While Adobe documentation is not edited by Datalogics, some *Adobe PDF Library* methods have been modified or enhanced by Datalogics for this "Plus" release. Those updated methods are documented in the "Modified Adobe Methods" chapter of the *DLI Implementation and Reference Guide*.) These and other documents may also be found on the Adobe website under the **Documentation** tab at <http://www.adobe.com/devnet/acrobat/>. (Descriptions below are, in part, provided by Adobe as part of their original accompanying PDFL\_SDK\_8\_Read\_Me.pdf file.)

**PDF Reference, Sixth Edition** This document describes PDF Standard 1.7 specifications.

**NOTE:** Adobe also provides an accompanying errata file for this manual (`pdf_reference_errata.pdf`), with last-minute updates and corrections. One copy is provided with this documentation (see your documentation file folder), and you can check for newer copies on the Adobe website under the **Documentation** tab at <http://www.adobe.com/devnet/acrobat/>.

**Addendum to PDF Reference, Sixth Edition** This document describes PDF Redaction.

**Adobe PDF Library Overview** This guide provides background and development information for the *Adobe PDF Library*. Read this document before beginning development for information such as supported platforms, known issues and development requirements.

**Acrobat and PDF Library API Reference** This is the reference manual for all of the *Acrobat* API methods made available to the *Acrobat* Viewer and *Adobe PDF Library*. It documents the parameters, return values and availability of each method, as well as specific implementation notes. This document is useful while developing with the *Adobe PDF Library* or planning development to determine method availability and capabilities.

**Developing Plug-ins and Applications** This is a developer guide for creating *Adobe PDF Library* applications as well as plug-ins for *Acrobat*. This includes the previous standalone publication "3D Annotations Tutorial" of the *Adobe PDF Library* v7.x release series.

**Snippet Runner Cookbook** This documents the useful `SnippetRunner` sample application development tool, provided by Adobe as a means of rapidly developing new functions using Library methods, and testing them within the context of a working Library application, using the Common Interface provided.



# Overview *of* *the Java Interface*

The *Java Interface* functions independently of the *Adobe PDF Library* as a wrapper making calls to it, as opposed to being a layer within it. That is, you can modify your Java code without disturbing the Library code, and update the Library release without needing to modify the Java calls.

## Requirements

For the *Datalogics Java Interface* itself, JRE v1.3 or higher is required. For the supported platform and operating-system requirements, please see either the accompanying *Adobe PDF Library* and *DLI* Installation Guide or the System Requirements page of the Datalogics Knowledgebase, available at the Datalogics website via <http://www.datalogics.com/support.asp>

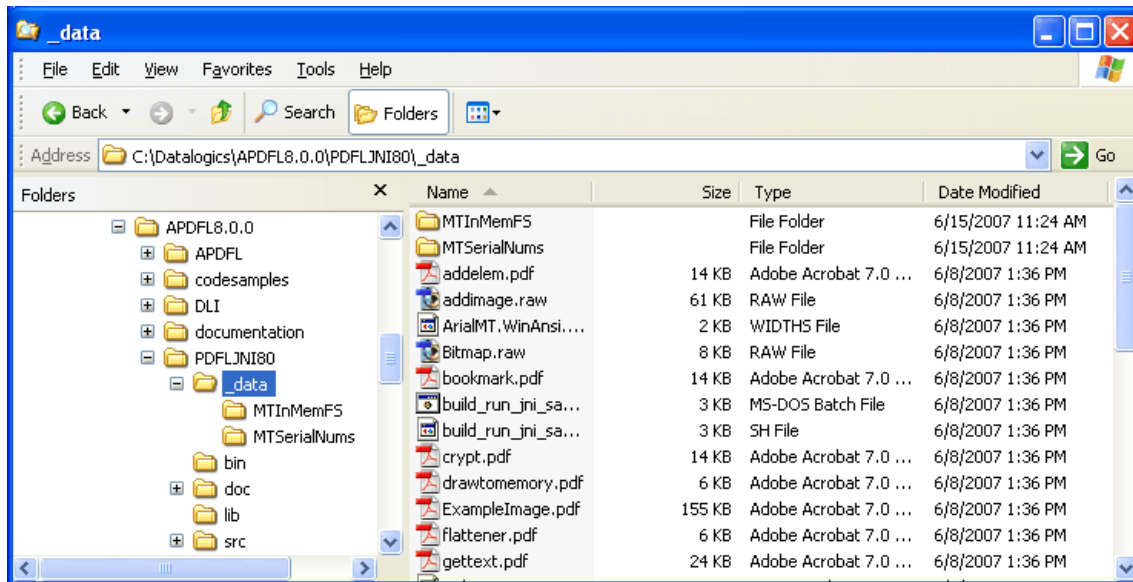
## Installation

As of the *Adobe PDF Library* v8.0.0PlusP1b release, installation of the *Java Interface* is included with the installation of the Library itself. A separate installation procedure for the *Java Interface* is no longer required.

## What's in the package

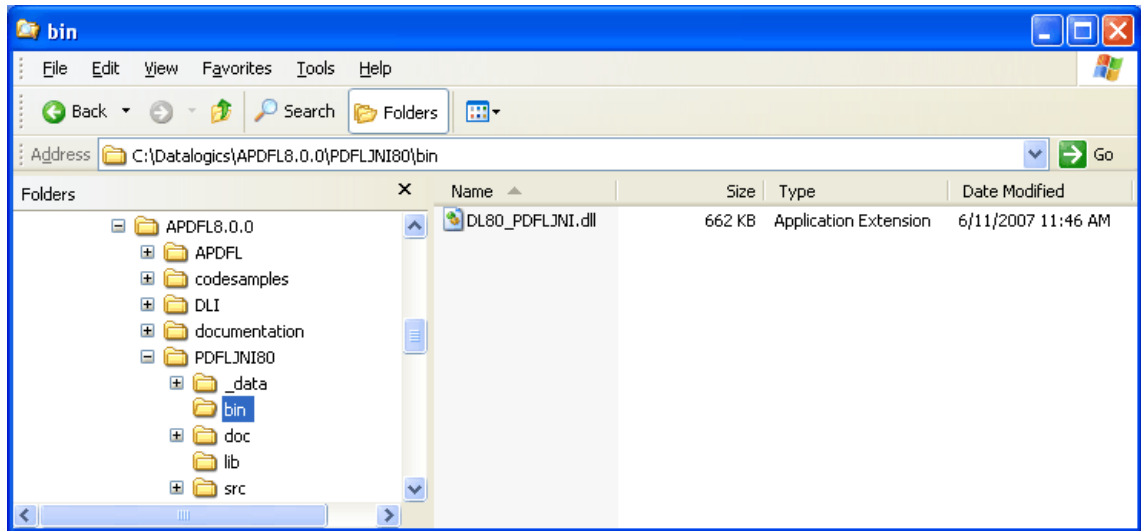
This is a representative list of package contents. Specific files and folder names may vary between releases or between platforms. Please contact your Datalogics representative if you have any questions regarding the specific contents of your *Java Interface* release.

~\\_data



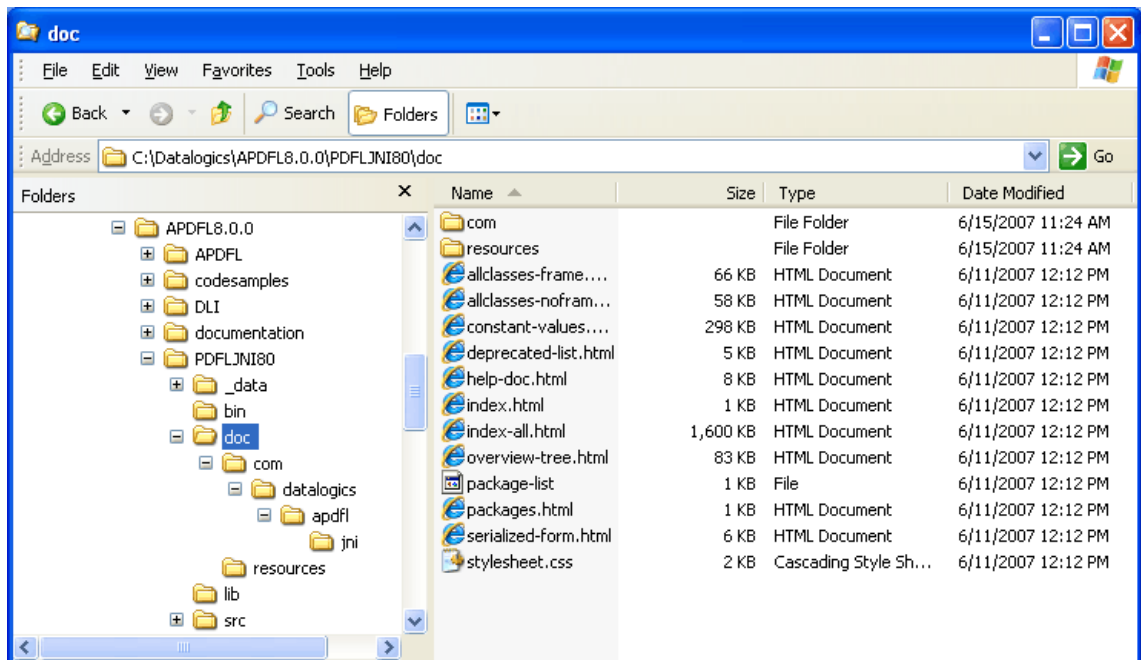
Input files for sample programs

~\bin



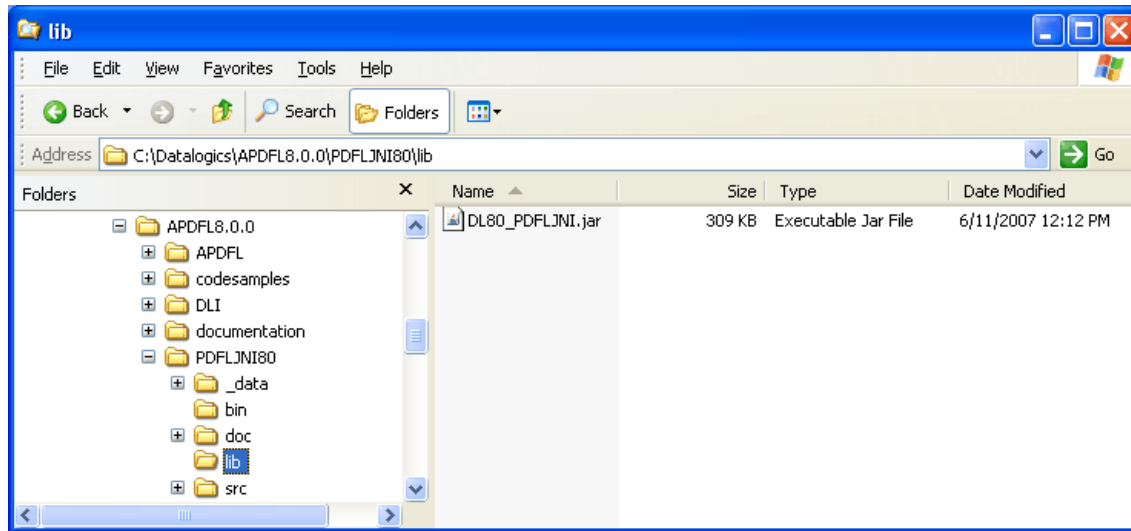
- DL80\_PDFLJNI.dll (JNI interface program)

~\doc



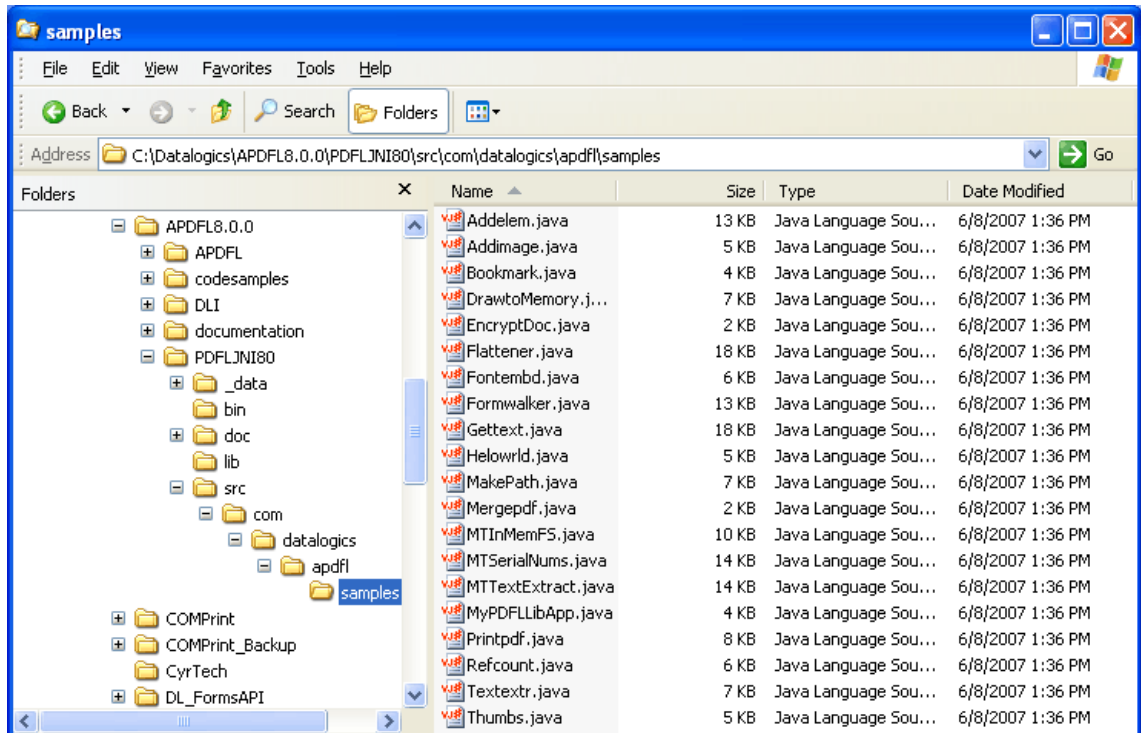
Java documentation

~\lib



- DL80\_PDFLJNI.jar (Java classes)

~\src\com\datalogics\apdf\samples



- Addelem.java
- Addimage.java
- Bookmark.java
- DrawToMemory.java
- EncryptDoc.java
- Flattener.java
- Fontembd.java
- Formwalker.java
- Gettext.java
- Helowrld.java
- Makepath.java
- Mergepdf.java
- MTInMemFS.java
- MTSerialNums.java
- MTextExtract.java
- MyPDFLLibApp.java
- Printpdf.java
- Refcount.java
- Textextr.java
- Thumbs.java



# Samples

## *Building and Running*

The *Java Interface* provides several samples written in Java. We recommend verifying at least one Java sample for proper installation and operation before proceeding to build your own application.

## Building Sample Programs

Change Directory to the `\samples` folder under the location where you installed your release, and issue the `javac` command as shown:

**NOTE:** These are syntax examples only. Actual file and pathnames may vary.

```
cd [...path here...]\src\com\datalogics\apdf1\samples
javac -classpath [...path here...]\lib\DL80_PDFLJNI.jar [Java file]
```

## Running Sample Programs

```
cd [...path here...]\@data
java -classpath
    [...path here...]\lib\DL80_PDFLJNI.jar; [...path here...]\src
    com.datalogics.samples.sample
```

For example, if the programs were installed in `c:\pdf1jni`, the command to run the Hello World example would be:

```
cd c:\pdf1jni\@data
java -classpath c:\pdf1jni\lib\DL80_PDFLJNI.jar;c:\pdf1jni\src
    com.datalogics.samples.Helowrld
```

# Packages *and*

## *Method Translation*

## Packages

The *Adobe PDF Library* classes are in the packages `com.datalogics.apdf1.jni.*`.

The samples are in the package `com.datalogics.apdf1.samples`.

## Method Translation

During development of the *Datalogics Java Interface*, the documented *Adobe PDF Library* methods and structures were translated into their Java equivalents.

The central class is `com.datalogics.apdf1.PDJNI`. This class contains the methods that are documented in the PDFL manual translated into their Java equivalents.

**NOTE:** Current releases of the *Java Interface* do not include *Datalogics Interface (DLI)* methods. These may be added in the future.

All of the methods in this class are public static. You do not need to instance this class.

### Class Translations

Each *Adobe PDF Library* call was translated into a more object-oriented style. The specific process used depended on the name of the original method parameter(s):

- 1 First, a translation was performed based on the first parameter. All methods that take a `PDDoc` were added to the `PDDoc` class; all methods that require a `PDPPage` as the first parameter were added to the `PDPPage` class, etc. The method signature was altered by removing the first parameter, removing the first part of the name if it matched the parameter type, and changing the first letter to lower case.

For example:

```
PDPPage PDDocCreatePage(PDDoc doc, int flags, ASFixedRect mediaBox);
```

was added to the `PDDoc` class as:

```
public PDPPage createPage(int flags, ASFixedRect rect) throws PDFLException;
```

- 2 If the first translation was unsuccessful, the process attempted to find a class that matched the leading characters of the method name. If a class was found, the class name was shortened and a static method

was added to the class:

For example:

```
PDDoc PDDocCreate();
```

was added to PDDoc class as:

```
public static PDDoc create();
```

### Returned Parameters

A *holder* class was substituted for any parameters that returned a value. There is a holder class that corresponds to each *Adobe PDF Library* class, primitive type, and Java-supplied class. The holder classes are named by appending “Holder” to the held Class Name (*e.g.* the holder for a `String` is `StringHolder`).

Each holder class contains a public field named `value`. On return this field will contain the actual object.

For example:

The *Adobe PDF Library* method:

```
int PDWordGetString(PDWord word, char * buffer, int len);
```

was translated to:

(*in class* `PDJNI`)

```
public int PDWordGetString(PDWord word, StringHolder stringHolder);
```

(*in class* `PDWord`)

```
public int getString(StringHolder stringHolder);
```

On return, the `stringHolder` value will contain the string.

Holders for primitive and Java-supplied types are in package `org.omg.CORBA` (part of the standard JDK).



# Exceptions

*and Error Handling*

## PDFLException Handling

All methods are declared to throw a `PDFLException`. You will be required to handle this exception. This takes the place of `DURING/HANDLER/END_HANDLER` macros used in the *Adobe PDF Library*.

# Index

## A

- Addelem.java 2.5
- Addimage.java 2.5
- Adobe Acrobat 1.4
- Adobe PDF Library
  - Applications Programming Interface (API) manual 1.2
  - Classes 4.2
  - Document Conventions 1.4
  - Method and Structure Equivalents 4.2
  - Methods
    - PDDocCreatePage 4.2
    - PDWordGetString 4.3
  - Samples 4.2
- Adobe Reader 1.2, 1.4

## B

- Bookmark.java 2.5

## C

- Conventions, Document 1.4

## D

- Datalogics website
  - System Requirements 2.2
- DL70\_PDFLJNI.dll 2.3
- DL70\_PDFLJNI.jar 2.4
- Document Conventions 1.4
- Documentation
  - Adobe
    - Acrobat and PDF Library API Reference 1.5
    - Addendum to PDF Reference, Sixth Edition 1.5
    - Adobe PDF Library Overview 1.5
    - Developing Plug-ins and Applications 1.5
    - PDF Reference, Sixth Edition 1.5
    - Portable Document Format Reference Manual
      - Errata file 1.5
    - Snippet Runner Cookbook 1.5
  - Datalogics
    - Adobe PDF Library and DLI Installation Guide 1.4
    - Adobe PDF Library Developer Overview 1.4
    - Adobe PDF Library Installation Guide 2.2
    - DLI Implementation and Reference Guide 1.4

- Java Interface User Guide 1.4
- DrawToMemory.java 2.5

## E

- EncryptDoc.java 2.5

## F

- Flattener.java 2.5
- Fontembd.java 2.5
- Formwalker.java 2.5

## G

- getString 4.3
- Gettext.java 2.5

## H

- Helowrld.java 2.5
- Holder classes 4.3
  - Package 4.3
  - StringHolder 4.3
  - value field 4.3

## J

- Java Interface
  - Class Translations 4.2
  - Code basis 1.2
  - Contents 2.2
    - ~\@data folder 2.2
    - ~\bin folder 2.3, 2.4
    - ~\doc folder 2.3
    - ~\src\com\datalogicsamples folder 2.5
  - Documentation (folder) 2.3
  - Exceptions and Error Handling 5.1
  - How to Use this Book 1.3
  - Input files for sample programs 2.2
  - Installation 2.2
  - Intended Audience 1.2
  - Introduction 1.2
  - Java/Library code independence 2.1
  - Method Translation 4.2
  - Packages 4.2
  - Requirements 2.2

Samples 3.1  
What You Should Know 1.2

## M

Makepath.java 2.5  
Mergepdf.java 2.5  
MTInMemFS.java 2.5  
MTSerialNums.java 2.5  
MTTextExtract.java 2.5  
MyPDFLibApp.java 2.5

## N

Notes

File and pathnames may vary 3.2  
No DLI Methods in current releases 1.2, 4.2  
PDF Reference Manual errata file available for  
download 1.5  
Structure Variations may exist between PDF 1.6  
and earlier 1.2

## P

PDDoc 4.2  
PDF  
File format 1.4  
PDFException 5.2  
PDJNI 4.3  
PDPage 4.2  
PDWord 4.3  
PDWordGetString 4.3  
Portable Document Format Reference Manual 1.4  
Printpdf.java 2.5

## R

RefCount.java 2.5  
Related Documentation 1.4

## S

Sample Programs  
Building 3.2  
Running 3.2  
Hello World 3.2

## T

Textextr.java 2.5  
Thumbs.java 2.5

## V

value (Holder class public field) 4.3

## W

Websites  
Datalogics  
System Requirements 2.2