Datalogics

Getting Started

Adobe PDF Java Toolkit Version 2.0

Document Updated 10/23/2013

For additional information, contact:

Datalogics, Incorporated
101 North Wacker Drive, Suite 1800
Chicago, IL 60606-7301
Phone: (312) 853-8200
Fax: (312) 853-8282
www.datalogics.com

Table of Contents

What is the Adobe PDF Java Toolkit?

Adobe PDF Java Toolkit is the core library used by Adobe LiveCycle, a product that allows users to build applications to automate business workflows. Popular with government agencies and service organizations, LiveCycle is an electronic form and document platform that is used to capture, process, and store information. For example, LiveCycle can be used to build tools to automate manual processes like setting up customer accounts, enrolling clients in benefit programs, and managing correspondence.

PDFJT is a Java Application Programming Interface (API) that provides much of the Adobe LiveCycle services for working with PDF files, forms, and the Adobe Reader. Developers can use PDFJT to build their own applications to offer a wide variety of services:

- Build your own electronic forms, make them available online, extract and populate data from those forms, and store this data for later analysis.
- Apply extensions to PDF files so that your users can interact with, edit, and save PDF forms with a viewing tool such as Adobe Reader. PDFJT can provide features for working with PDF files that are not normally available in Adobe Reader.
- Create and edit PDF documents, extract text and images, edit text, merge files, delete and insert pages, add watermarks, and apply digital signatures.
- Create PDF Portfolios, or collections of multiple types of files managed as part of a single PDF file.

Note that many parts of PDFJT serve Adobe LiveCycle but work independently of the product. So it is not necessary to have a licensed copy of LiveCycle to use PDFJT.

PDFJT is a Java library, pdfjt.jar, as well as a set of sample Java program files, classes, PDF input files, JavaScript files, and other supporting materials.

Adobe PDF Java Toolkit is only available through Datalogics.

What you receive from Datalogics

Datalogics will provide you with a zip file, the Adobe PDF Java Toolkit Release. Extract the documents and folders in this zip file to the local drive of your choice. When you unzip the file it will create three directories:

1. **Docs**. This folder holds the Javadoc files, a programming reference. Open Index.html to look for information about the Java packages and classes related to the product.
2. **Libs**. This folder holds one file, pdfjt.jar, the PDFJT library. You must include this JAR file in your projects to use PDFJT features.
3. **Samples**. The sample folder contains sample programs and supporting materials that demonstrate different kinds of tasks that you might complete using PDFJT.

   For example, look at the Java program file HelloWorldSample.java, found in the samples directory, src/com/adobe/pdfjt/samples/document. This program demonstrates how to create a new PDF file, insert a new page, add the text “Hello World” to the page, and save the document.

Setting up Test Credentials for Adobe Acrobat or Adobe Reader

PDF Java Toolkit works independently of all Adobe Systems software products. You can use PDF Java Toolkit to create and run applications on any workstation or server. You do not need to install Adobe Reader or Adobe Acrobat.

To review the output results from PDF Java Toolkit programs, however, you will need a viewer that will allow you to open PDF files. For most of the PDFJT sample programs the free Adobe Reader will do. Some samples require Adobe Acrobat, but the sample itself will make that plain to you.

To use either Adobe Reader or Adobe Acrobat with PDF Java Toolkit, you will need to install the Datalogics test credentials. We describe the process for both products below.

As it only takes a few moments, you might want to simply install the certificate for both Adobe Acrobat and Adobe Reader on your machine. If you are running both Acrobat and Reader on your machine, you can install the certificate once for both products, and you never need to install it again unless you upgrade either product to a new version.

Installing the Certificate

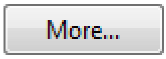
The PDF Java Toolkit supports creating PDF files that are compatible with Adobe Acrobat or Adobe Reader version 10 and later.

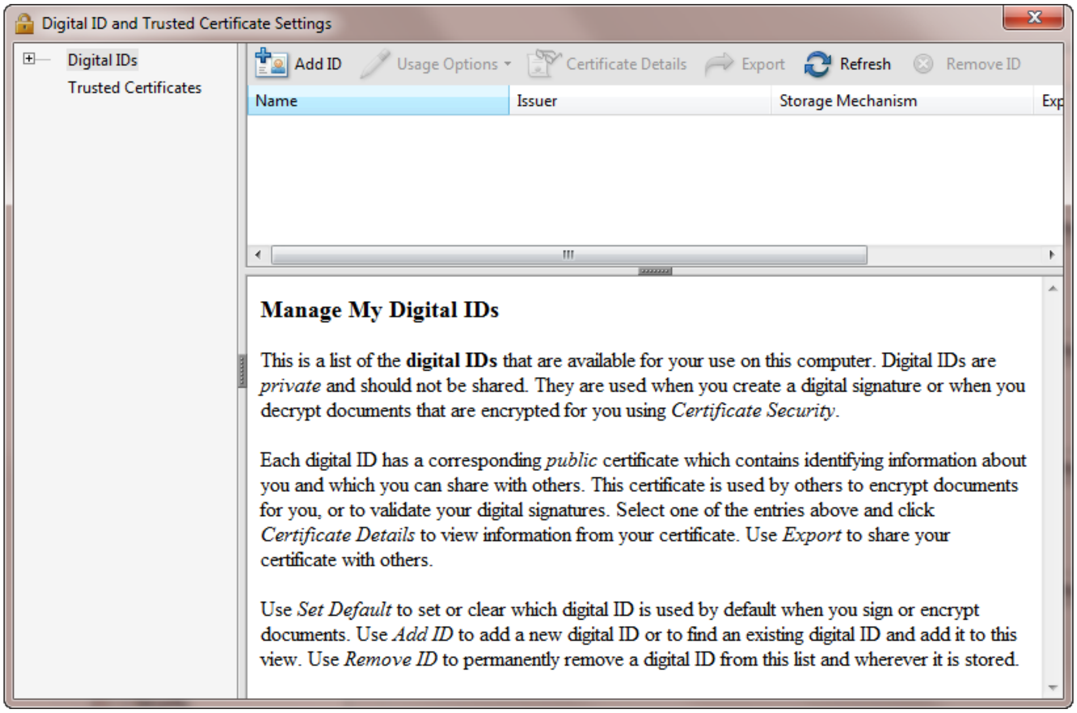
Open Adobe Acrobat or Adobe Reader. The process is the same for both systems.

Note that if you are reading this document as a PDF in Adobe Acrobat while following the steps below, you won't be able to scroll down in the PDF file. If you have both Reader and Acrobat installed on your machine, you could open this PDF file in Reader and run through the steps below using Acrobat.

From the Edit menu, click Preferences.

In the Preferences window, highlight Signatures on the left side of the screen.

Under the "Identities & Trusted Certificates" heading, click .



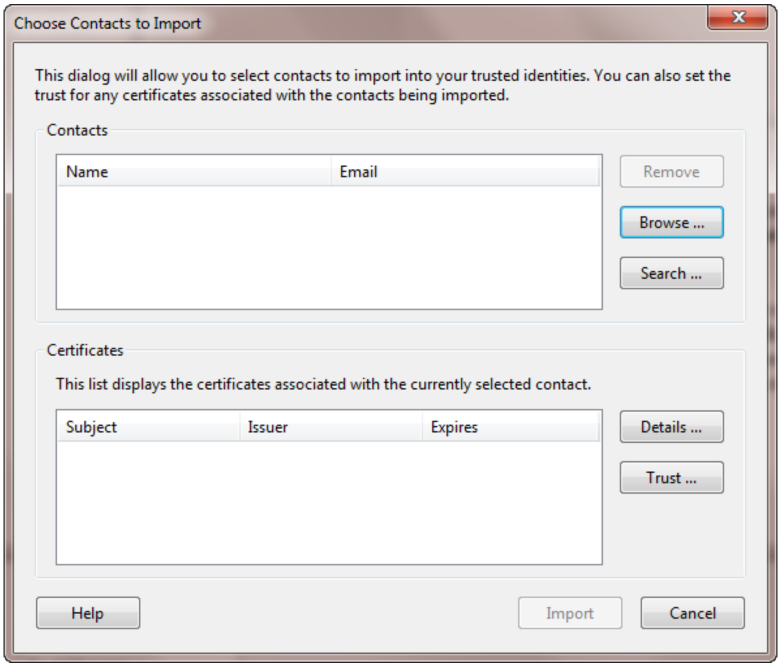
Manage My Digital IDs

This is a list of the **digital IDs** that are available for your use on this computer. Digital IDs are *private* and should not be shared. They are used when you create a digital signature or when you decrypt documents that are encrypted for you using *Certificate Security*.

Each digital ID has a corresponding *public* certificate which contains identifying information about you and which you can share with others. This certificate is used by others to encrypt documents for you, or to validate your digital signatures. Select one of the entries above and click *Certificate Details* to view information from your certificate. Use *Export* to share your certificate with others.

Use *Set Default* to set or clear which digital ID is used by default when you sign or encrypt documents. Use *Add ID* to add a new digital ID or to find an existing digital ID and add it to this view. Use *Remove ID* to permanently remove a digital ID from this list and wherever it is stored.

Click Trusted Certificates, and then click Import.



This dialog will allow you to select contacts to import into your trusted identities. You can also set the trust for any certificates associated with the contacts being imported.

This list displays the certificates associated with the currently selected contact.

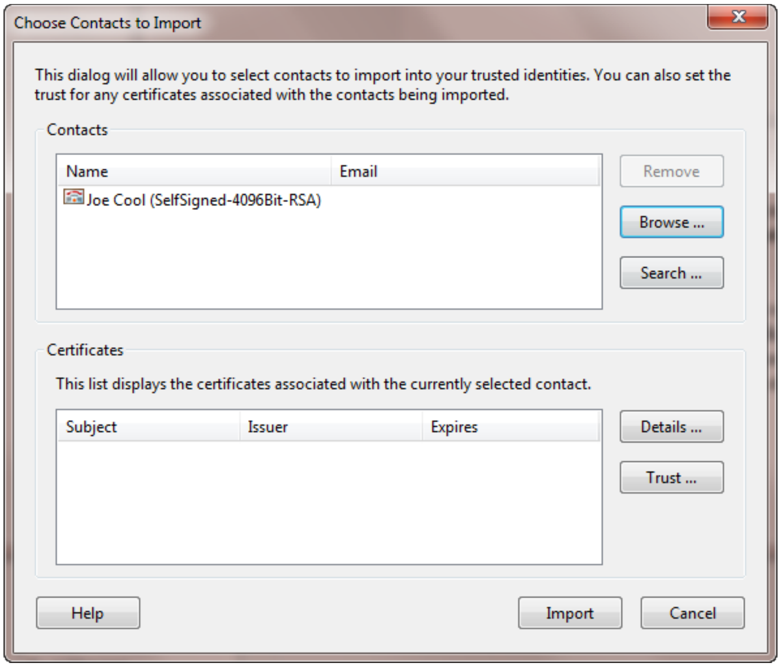
Click Browse ....

We provide a certificate file (.crt) with the PDF Java Toolkit software files.

Browse to the directory where you installed the PDF Java Toolkit. If you installed the software on your local machine, the path might look like this:

C:\PDF Java Toolkit\Samples\Input\credentials\4096.crt

Click Open.



Choose Contacts to Import

This dialog will allow you to select contacts to import into your trusted identities. You can also set the trust for any certificates associated with the contacts being imported.

Contacts

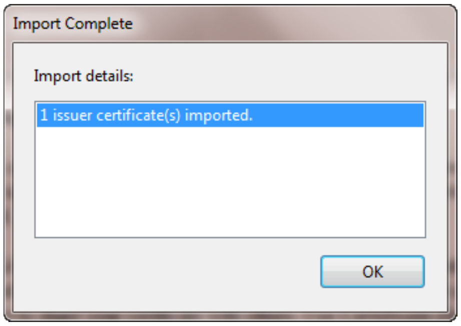
Name	Email
Joe Cool (SelfSigned-4096Bit-RSA)	

Certificates

This list displays the certificates associated with the currently selected contact.

Subject	Issuer	Expires

Click Import.



Import Complete

Import details:

1 issuer certificate(s) imported.

Click OK, and close the Certificate window.

Compiling and Running your Java Program Files

If you are working on a PDFJT project and plan to compile and run Java program files for that project at a command line, every time you enter these commands you will need to add pdfjt.jar to your class path. If PDFJT is the only external toolkit you use for this project, enter these commands:

```
javac -classpath path\to\pdfjt.jar;src\com\package\yourfile.java
java -classpath path\to\pdfjt.jar com.package.yourfile
```

We are assuming here that you are in the Samples subdirectory in the directory structure where you installed PDF Java Toolkit .

In these commands “path\to\” refers to the local or server directory where you stored the pdjft.jar library file, and “yourfile” path is the path and name of your Java source file.

The two commands need to be entered in sequence whenever you want to compile and run a Java program file.

The commands might look like this:

```
javac -classpath
..\libs\pdfjt.jar;src com\adobe\pdfjt\samples\document\HelloWorldSample.java
java -classpath
..\libs\pdfjt.jar;src com.adobe.pdfjt.samples.document.HelloWorldSample
```

The class path variable "..\libs\pdfjt.jar;src":

1. defines the location of the PDFJT.JAR library file, and then
2. directs the system to look in the "src" directory under the Samples directory to find the Java program file. In this example, the program file is called HelloWorldSample.Java.

Note the use of the semi-colon symbol, ";" in the above commands. This designates the beginning of the path for the sample Java program. Src is the name of the subdirectory under the Samples directory. Use ";" for Windows systems, and use ":" for Mac and Linux systems.

Configuring PDF Java Toolkit to Work with Eclipse

Installing Eclipse and Java

Datalogics recommends installing Eclipse Standard or Eclipse IDE for Java Enterprise Edition for Developers (www.eclipse.org), version 4.3 or later, and Java Developers Kit (JDK) version 5.0 or later (www.java.com). If you are running an earlier version of Eclipse on your machine it should work, however.

To go directly to the Java Development Kit 7 download page, use this link:

www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Setting up the PATH System Variable

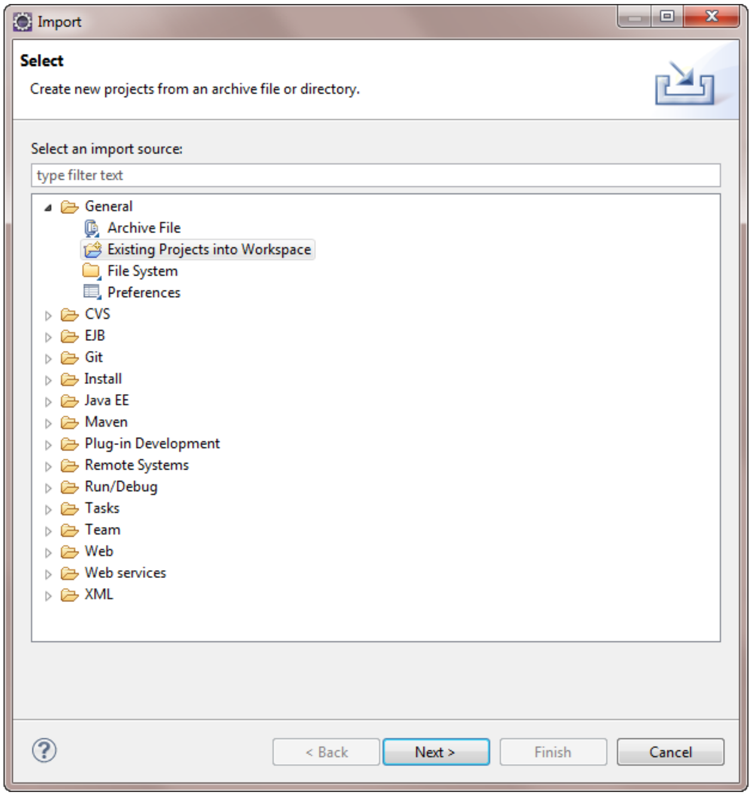
You might need to manually add Java to your PATH environment variable if you work in Windows or Linux. This is rare, but this Java web page provides an easy guide for the process:

www.java.com/en/download/help/path.xml

Importing the Samples Project into Eclipse

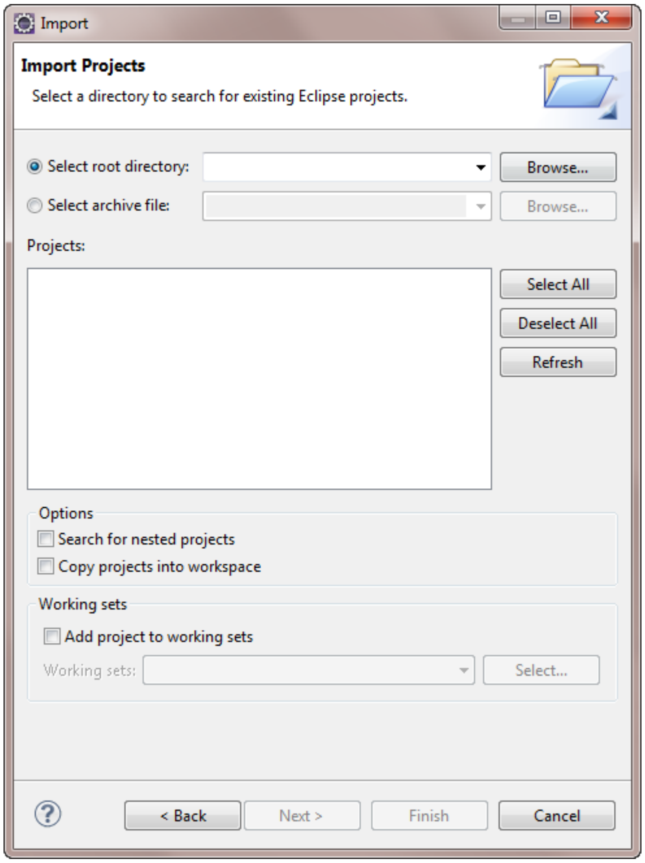
Open Eclipse and the Project Explorer window. If needed click Window, Show View, and Project Explorer. Note that you can work with the Navigator window instead, if you prefer Navigator.

Then, in the Project Explorer window, click your right mouse button and click Import... . Click the General folder to open it:



Create new projects from an archive file or directory.

Then, double click Existing Projects into Workspace:



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive file:

Projects:

Options

□ Search for nested projects

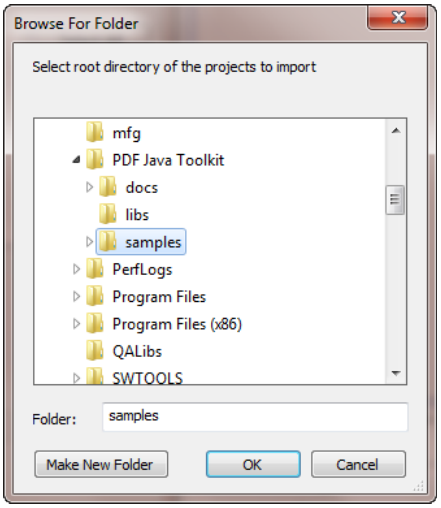
□ Copy projects into workspace

Working sets

□ Add project to working sets

Working sets:

Click Browse. From here you will navigate to the directory where the PDF Java Toolkit project files are stored.

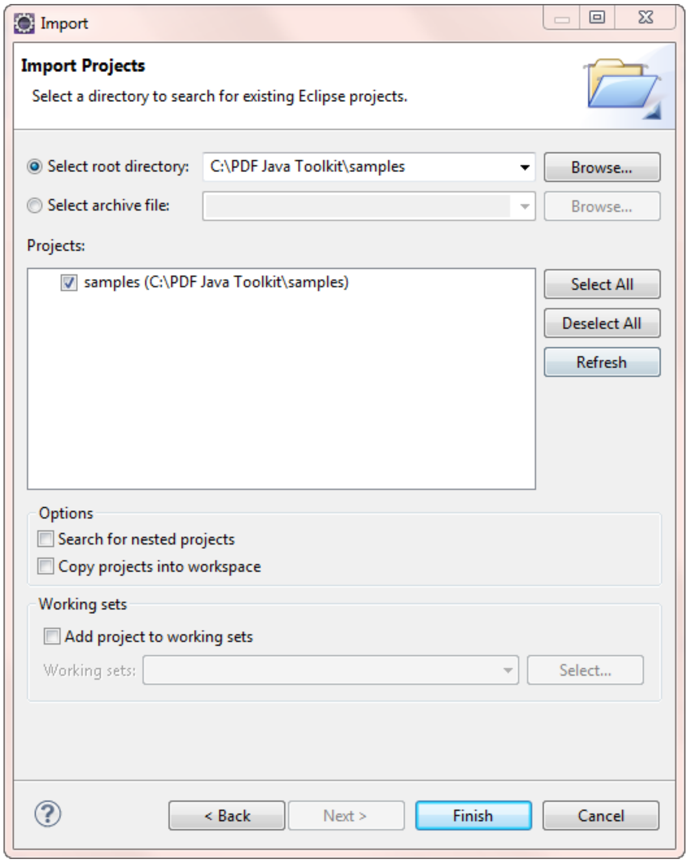


Browse For Folder

Select root directory of the projects to import

Folder: samples

For import projects, select the Samples directory and click OK:



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\PDF Java Toolkit\samples

○ Select archive file:

Projects:

☑ samples (C:\PDF Java Toolkit\samples)

Options

☐ Search for nested projects

☐ Copy projects into workspace

Working sets

☐ Add project to working sets

Working sets:

Click Finish.

Setting up a Java Toolkit Project in Eclipse

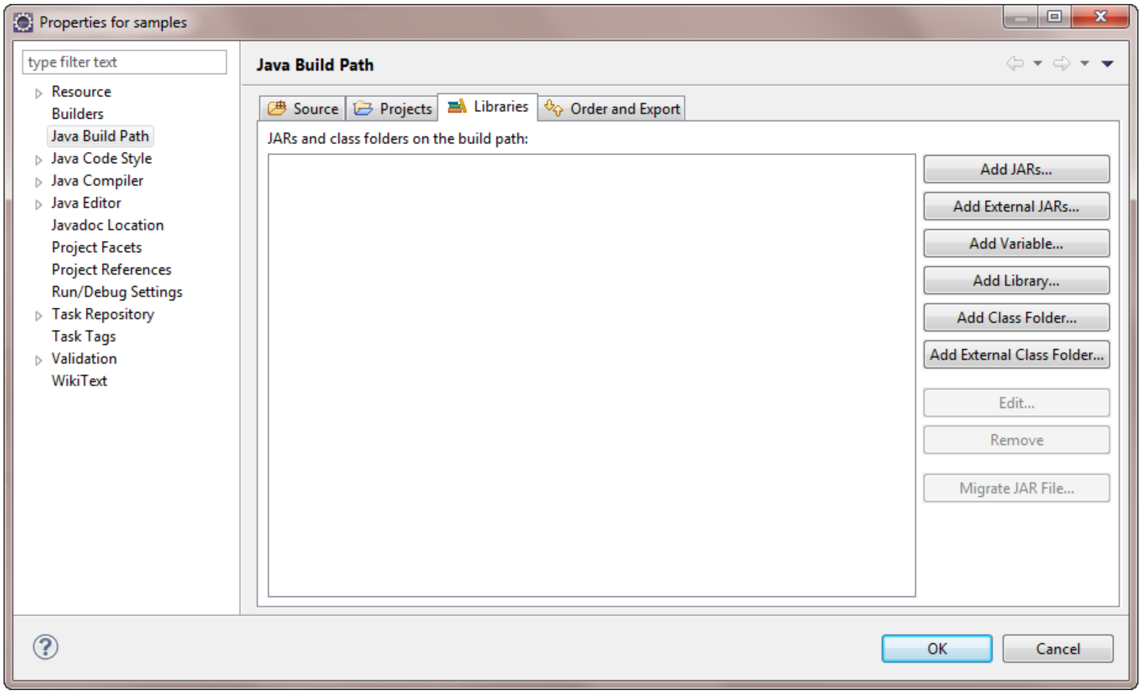
After you select Samples, you need to access the pdfjt.jar library file. This file holds all of the libraries that you will be using with the PDF Java Toolkit, and it is stored in the Libs directory. You can move this file to the Samples directory, so that it will be found within the project. Or you can leave it in Libs, and it will be outside the project.

Either way, you need to add the pdfjt.jar file to the Eclipse project build path.

Click the Project tab. Right click on the Samples directory.

Select Build Path and then Configure Build Path....

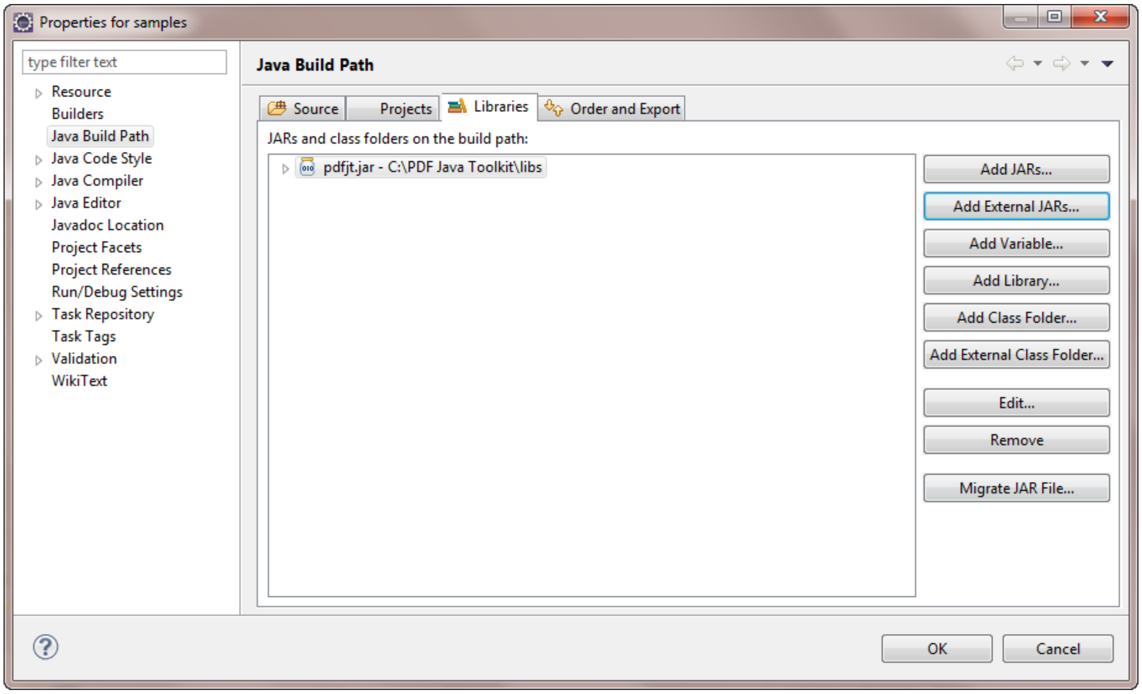
Click the Libraries tab. The available Jar files appear.



If the pdfjt.jar file is located within the project—you moved the file to the Samples directory—click Add JARs... .

If the pdfjt.jar file is external to your project—you left the file in its original directory, libs—click Add External JARs... .

Either way, navigate to the directory where pdfjt.jar is stored. Select pdfjt.jar and click Open.



pdfjt.jar - C:\PDF Java Toolkit\libs

Click OK.

Working with the PDF Java Toolkit

You can use the APIs within PDF Java Toolkit to complete a wide variety of tasks. The sample programs in PDF Java Toolkit allow you to create applications that have four basic functions:

1. Create digital signatures and assign them to PDF files
2. Build and manage data using PDF electronic forms
3. Create PDF archive files (PDF/A)
4. Manage PDF document processing, such as creating and editing PDF files, adding text and graphics, deleting pages, and extracting text and graphics

Digital Signatures

In recent years state and federal government agencies have encouraged the shift to electronic documents using digital signatures. These are preferred over forms and contracts that must be printed, manually completed, signed, and stored in boxes for long periods. Adobe Acrobat and Adobe Reader make it easy to generate electronic forms, complete them online, and then save them with the added information and with digital signatures. With PDF forms the digital signature is an encrypted token added to the file. The signature guarantees the legal authority the document by verifying the identity of the person who created it. The signature also locks the document and thus demonstrates that the document has not been altered since the owner signed it.

PDF Java Toolkit offers about 30 sample programs related to digital signatures in the digsig sample directory. You can use these sample programs to add a digital signature to a PDF file, add a time stamp, validate a signature, certify signed documents, search for signature records, remove a signature from a document, and other processes.

Managing PDF Forms

With Adobe Acrobat, you can use PDF files to create electronic forms for collecting and processing information. These forms are fast and easy to create, and thus serve as an alternative to Web-based forms that require programming skills. Users can also save a PDF form with added information, and the values entered in PDF forms can be extracted and stored in a database for later processing.

You can use PDF Java Toolkit to build electronic forms. You can also create an application that can collect completed copies of PDF form files from users or customers, dozens or hundreds at a time, extract the values entered in the form fields, and copy this data to a database. Several sample programs allow you to work with generating PDF forms, where you can:

- Add text fields and other kinds of form fields to a PDF file
- Define the font to be used within text fields in a form
- Identify and list out the types of form fields found within a PDF file, and the names of those fields

Working with PDF/A files

You can use PDF Java Toolkit to create a program that converts any PDF files you provide into the PDF/A format. PDF/A is an ISO-standard version the PDF format. Adobe introduced PDF in 1993, and the International Organization for Standardization (ISO) took management of PDF as an open standard in 2008. ISO released PDF/A as a lighter version of the original PDF format in 2005. It is designed to be used with PDF files that need to be archived and stored for long periods, and then accessed later. Thus PDF/A files must be self-contained so that they can be opened in future versions of standard viewing tools. For example, the fonts in a PDF/A file are embedded in the file itself, rather than accessed through a link to a font directory on a local server.

PDF Java Toolkit provides five sample programs that work with the PDF/A format in directories pdfa and pdfa2, that allow you to convert a standard PDF file to PDF/A and then verify that the conversion completed successfully.

You can also use EmbeddedFileValidationHandlerSample to verify any files embedded in a PDF file, such as an image, video, or file attachment, before the PDF is converted to PDF/A.

Managing PDF documents

You can use PDF Java Toolkit to create applications that can create and manipulate PDF files. With your custom code you can process large numbers of PDF files automatically, and complete tasks like:

- Extracting text or graphics
- Applying watermarks to pages
- Adding notes and comments to PDF pages
- Deleting and inserting pages
- Merging PDF files or splitting large files into smaller ones

PDF Java Toolkit provides several dozen sample programs to help you complete these tasks. For example, you could use the PDFWatermark.java sample program as a template to create a program that would apply a graphic to a PDF document. WaterMark.java allows you to select an import PDF file, define a graphic as a watermark, and describe how the watermark will be placed on the page, in terms of the size of the image and the location.

Use ExtractAndInsertPages to pull pages out of a PDF file you provide and save them to a separate file or set of files, or to insert new pages in that PDF.

The RemoveContent program allows you to search for and remove an object from every page of a PDF file. It works by identifying a sequence of instructions in the content stream for each page and then removing those instructions if they match the criteria described in the program.

The TextExtractionSample program allows you to identify all of the words in a PDF file and export them to a separate text file, where they can be edited.