



The XML PDF Access API for Java™ Technology (XPAAJ)

Duane Nickull
Senior Technology Evangelist
Adobe Systems

TS-93260

Agenda

Using Java™ technology to manipulate PDF

The session will provide a general overview of Adobe's XPAAJ.jar for Java technology developers, its key methods and show some code samples to manipulate PDF documents

Agenda—60 Minutes

XPAAJ.jar Overview

API Review

Code: Basic Opening of PDF

Code: Some Basic Methods

Code: Advanced Methods

Code: Extracting XMP Metadata From PDF

Wrap Up—Q&A

Agenda

XPAAJ.jar Overview

API Review

Code: Basic Opening of PDF

Code: Some Basic Methods

Code: Advanced Methods

Code: Extracting XMP Metadata From PDF

Wrap Up—Q&A

XPAAJ.jar Overview

Java libraries for manipulation of PDF

- Java API 4 for developers to build applications for PDF documents and forms
- Free download (3.87 mb zip) from Adobe developer center:
<http://www.adobe.com/devnet/lifecycle/downloads/xpaaj.html>
- You can incorporate XPAAJ into standalone Java applications or deploy it into either a Web container or an application server container

XPAAJ.jar Enables:

- Text extraction
- XML form data import/export
- XMP metadata export
- Annotation import/export
- Conversions between PDF and XDP
- File attachment import/export/delete

XPAAJ.jar Includes:

Written by Java technology Developers
FOR Java technology Developers!

- XPAAJ .jar file
- Javadoc™ tools
- Getting Started Guide
- Developer Guide
- API Reference
- Sample Applications Guide
- Command line and Web application sample code
- Update Oct 2006 includes Cold Fusion samples!

System Requirements:

Operating System	System Updates	Minimum Memory	Processor
Windows® XP	SP1	512 MB of RAM	Intel 1.0 GHz, 512 cache
Windows 2000	SP3	512 MB of RAM	Intel 1.0 GHz, 512 cache
Windows 2000 server			Intel 1.0 GHz, 512 cache
Sun™ Solaris™ 8 SPARC 32bit		512 MB of RAM	UltrasparcIII, 300 MHz or faster processor
Sun Solaris 9 SPARC 32bit		512 MB of RAM	UltrasparcIII, 300 MHz or faster processor
RedHat® 7.2 for Intel®		512 MB of RAM	Intel 1.0 GHz, 512 cache
RedHat 8.0 for Intel		512 MB of RAM	Intel 1.0 GHz, 512 cache



Agenda

XPAAJ.jar Overview

API Review

Code: Basic Opening of PDF

Code: Some Basic Methods

Code: Advanced Methods

Code: Extracting XMP Metadata from PDF

Wrap Up—Q&A

PDFFactory Class

- PDFFactory defines a static PDFFactory object implements the PDFDocument interface
 - associate data stream → PDF/XDP file with a PDFDocument
- openDocument()
`public static PDFDocument
 openDocument(java.io.InputStream input)
 throws java.io.IOException, PDFException`
- newFileAttachment()
`public FileAttachment
 newFileAttachment(java.io.InputStream src,
 java.lang.String filename
 java.lang.String mimetype)`

PDFDocument Interface

- Defines a public interface for manipulating PDF documents
- 18 core methods
 - deleteAnnotations
 - deleteFileAttachment
 - exportAnnotations
 - exportFileAttachment
 - exportFormData
 - exportXMP
 - getFileAttachmentNames
 - getFormType
 - getImages
 - getNumberOfPages
 - getVersion
 - GetWords
 - importAnnotations
 - importFileAttachment
 - ImportFormData
 - importXMP
 - save
 - saveasXDP

FileAttachment Interface

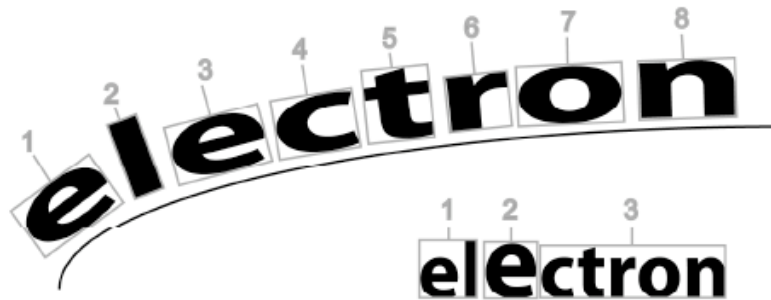
- Public interface for referencing file streams that can be archived or transmitted with PDF
- Three core methods:
 - `getData()`
 - `getFilename()`
 - `getMimetype()`

PDFImage Interface

- Public interface → exporting/importing image XMP metadata
- Each image in PDF document is represented by a PDFImage object
- Can obtain a list of PDFImage objects via `java.util.List` object (created by calling the `getImages()` method on PDFDocument object)
- Two core methods:
 - `exportXMP()`
 - `importXMP()`

PDFWord Interface

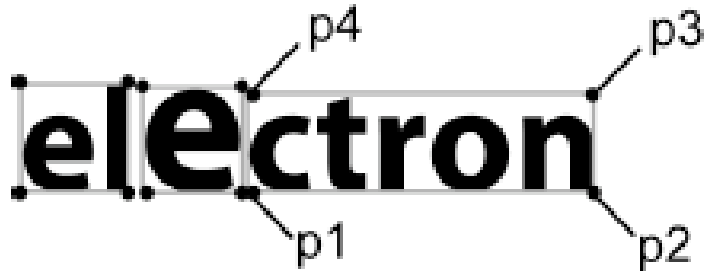
- Public interface for accessing one word from a list of the static-text words in a PDF doc
- Three core methods:
 - getPageNumber() -gets the page number on which the word occurs.
 - getQuadList() -gets list of quad objects associated with word



- getString() -gets the unicode characters that represent word

Quad Interface

- Obtain location/orientation of characters in a word



Quad Interface Example:

```
ListIterator words = pdfDocument.getWords();
while (words.hasNext()) {
    PDFWord word = (PDFWord)words.next();

    //Iterate over every Quad object in the list.
    ListIterator quads = word.getQuadList();
    while (quads.hasNext()) {
        Quad quad = (Quad)quads.next();
        //Get the point locations.
        Point p1, p2, p3, p4;
        p1 = quad.p1();
        p2 = quad.p2();
        p3 = quad.p3();
        p4 = quad.p4();
    }
}
```


Point Interface

- Public interface → get X and Y values associated with one corner of a Quad
- X = horizontal position, Y = vertical position
- Both positions are relative to the upper left corner of the PDF page (X-Y = 0,0)
- After you create a Quad object, you can call the p1, p2, p3, or p4 method to get the X-Y coordinate associated with one corner

Point Interface Example:

```
double x1, y1, x2, y2, x3, y3, x4, y4;  
// p1, p2, p3, p4 are obtained quad objects  
// Get the values for point p1.  
x1 = p1.x()  
y1 = p1.y()  
  
//Get the values for point p2.  
x2 = p2.x();  
y2 = p2.y();  
  
// etc.  
x3 = p3.x();  
y3 = p3.y();
```

Other API Information

- Related classes:
 - FormType
 - FormDataFormat
- XPAAJ has standard Java platform Exceptions

Exception	Problem
PDFException	The base exception for all PDF exceptions raised by XML/PDF Access API for Java.
PDFParseException	Parsing a given PDF document. Base exception: PDFException
PDFSecurityException	The given PDF document is encrypted. Base exception: PDFException
XFDFormatException	Importing or exporting XFD form data. Base exception: PDFException
XFDFormatException	Parsing a given XFD stream. Base exception: XFDFormatException

More Exceptions

Exception	Problem
<code>XFAException</code>	Importing or exporting XFA form data. Base exception: <code>PDFException</code>
<code>XFAParseException</code>	Parsing a given XFA stream. Base exception: <code>XFAException</code>
<code>XDPEXception</code>	Opening or saving a XDP stream. Base exception: <code>PDFException</code>
<code>XDPParseException</code>	Parsing a given XDP stream. Base exception: <code>XDPEXception</code>
<code>XMPEXception</code>	Importing or exporting a XMP stream. Base exception: <code>PDFException</code>
<code>XMPParseException</code>	Parsing a given XMP stream. Base exception: <code>XMPEXception</code>

Agenda

XPAAJ.jar Overview

API Review

Code: Basic Opening of PDF

Code: Some Basic Methods

Code: Advanced Methods

Code: Extracting XMP Metadata from PDF

Wrap Up—Q&A

Code-DuanePDFClass2

```
import java.io.*;
import java.util.*;
import java.awt.image.DataBuffer;
import com.adobe.pdf.*;

public class DuanePDFClass2 {

    public static void main(String[] args)
        throws FileNotFoundException, IOException

    // do something here...

}
```

Opening the File...

```
public static void PDFExtract(String inPdfName)
    throws FileNotFoundException, IOException
{
    System.out.println("\nOpening PDF...");
    PDFDocument doc = null;
    boolean b = false;
    FileInputStream inPdfFile = new
        FileInputStream(inPdfName);
    try {
        doc = PDFFactory.openDocument(inPdfFile);
    } catch (IOException e) {
        System.out.println("Error opening:" + inPdfName);
        System.out.println(e);
    }
    if(doc == null)
        System.out.println("Cannot open PDF");
    else
        System.out.println( inPdfName + " opened.");
}
```

Saving the PDF File...

```
System.out.println ("\nSaving document ... ");
int j = inPdfName.lastIndexOf(".");

//write out the file saved with a new name
String outPdfName = inPdfName.substring(0, j) +
    "_saved_by_Duane_at_JavaOne2007" + ".pdf";

InputStream inputStream; inputStream = doc.save();
    b = false; // boolean
    try {
        b = saveFile(inputStream, outPdfName);
    } catch (Exception e) {
        System.out.println("Error saving PDF file.");
        System.out.println(e);
    }
}
```




DEMO

Opening Basic PDF Document—DuaneClass2.java



Agenda

XPAAJ.jar Overview

API Review

Code: Basic Opening of PDF

Code: Some Basic Methods

Code: Advanced Methods

Code: Extracting XMP Metadata From PDF

Wrap Up—Q&A

PDF File Versions

- PDF files have versions (current 1.7)
- getVersion(), part of PDFDocument interface, returns a String representing version
- Throws PDFException
- Syntax:

```
System.out.println ("Document version = " +  
    doc.getVersion() );
```

PDF Length

- `getNumberOfPages()`, part of `PDFDocument` interface, returns integer
- Throws `PDFException`
- Syntax:

```
System.out.println ("PDF is " + doc.getNumberOfPages () + "  
pages long" );
```



DEMO

`getNumberOfPages()`

`getVersion()`

`DuaneClass2.java`



Agenda

XPAAJ.jar Overview

API Review

Code: Basic Opening of PDF

Code: Some Basic Methods

Code: Advanced Methods

Code: Extracting XMP Metadata From PDF

Wrap Up—Q&A

PDFExtractData

Operations:

- Opens document
- Exports all image metadata to “imagemetadax.Xml”
- Exports annotations data to file “anntations.Xml”
- Exports form data to file “formdata.Xml” *
- Exports all file attachments
- Saves the PDF to pdf and xdp files
- Extracts all words in the document to a text file with same name



DEMO

PDFExtractData.java



Agenda

XPAAJ.jar Overview

API Review

Code: Basic Opening of PDF

Code: Some Basic Methods

Code: Advanced Methods

Code: Extracting XMP Metadata From PDF

Wrap Up—Q&A

Working With XMP...

```
InputStream myXMPStream = doc.exportXMP();

//Get the byte size of the InputStream object.
int numBytes = myXMPStream.available();

System.out.println("\nNumber of XMP Bytes found is " +
numBytes + "\n");

// Read into a Buffered Reader Stream.
BufferedReader d = new BufferedReader(new
    InputStreamReader(myXMPStream));

    String xmpLine;
    while((xmpLine = d.readLine()) != null){
        System.out.println(xmpLine);
    }
```



DEMO

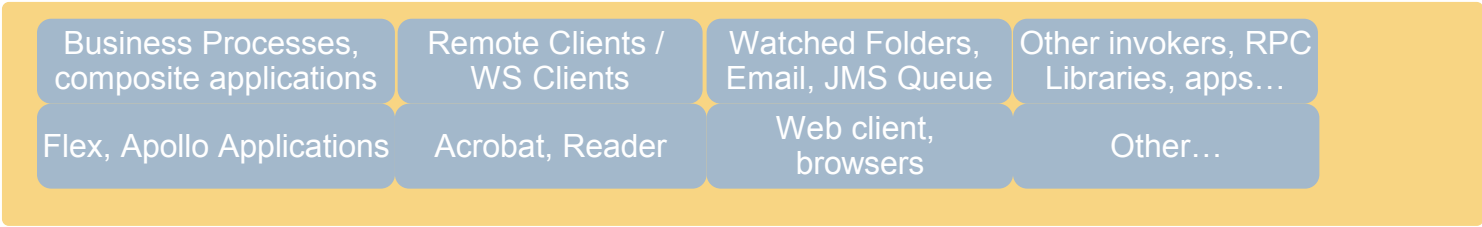
XMPEExtractSample.java



Summary

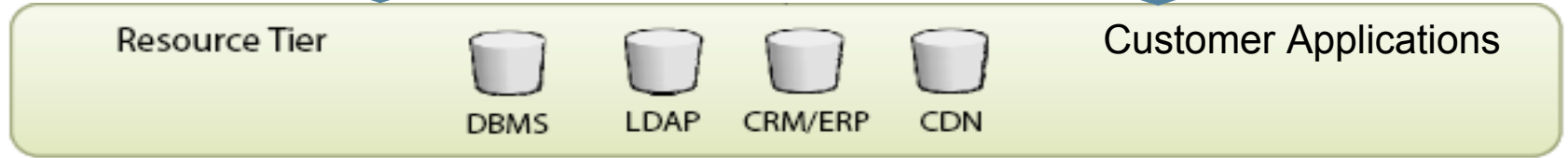
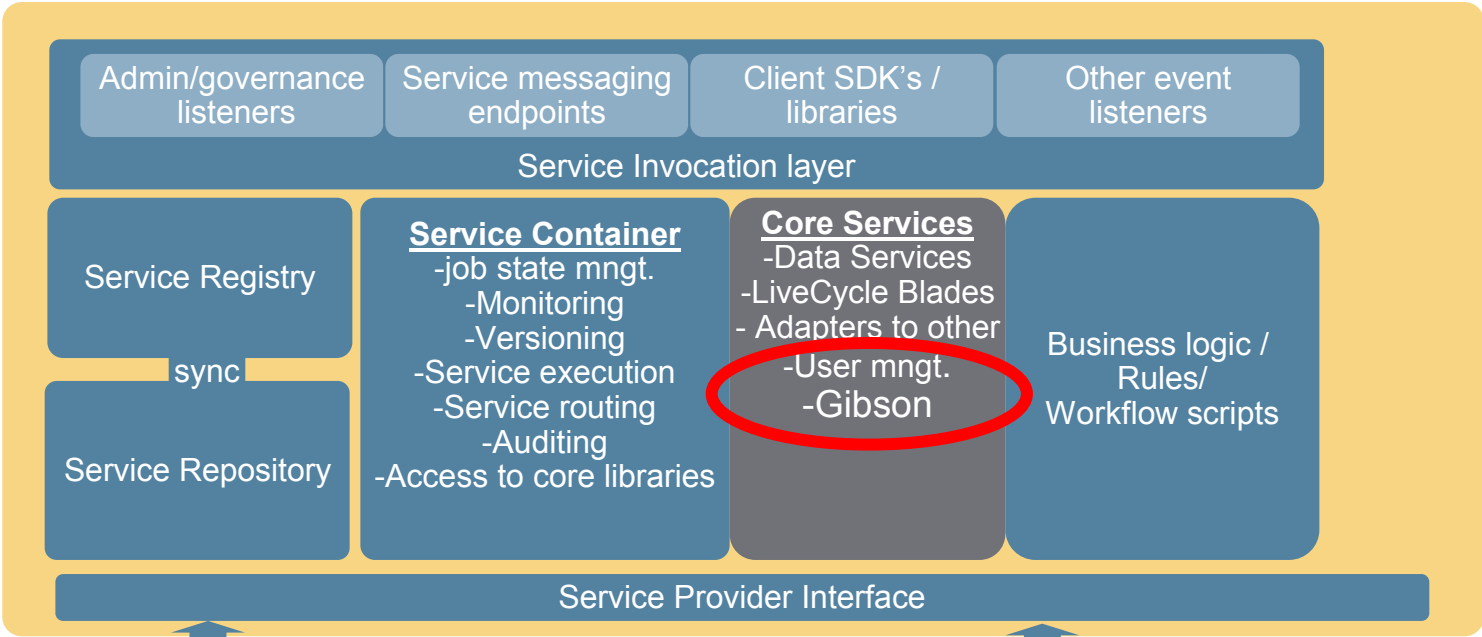
- XPAAJ is powerful, easy to use
- Java technology Developers also might want to look into LiveCycle

Clients



Design and Governance Tools

- LC Designer
- LC Workflow IDE's
- Flex Builder
- ...



JMS = Java Message Service

For More Information

- Adobe Enterprise Developer Program
 - <http://www.adobe.com/enterprise/developer/>
- Adobe Labs
 - <http://labs.adobe.com>



Q&A

Duane Nickull

<http://technoracle.blogspot.com>



The XML PDF Access API for Java™ Technology (XPAAJ)

Duane Nickull
Senior Technology Evangelist
Adobe Systems

TS-93260