Apache POI - Component Overview

by Andrew C. Oliver, Rainer Klute, David Fisher

1. Apache POI Project Components

1.1. POIFS for OLE 2 Documents

POIFS is the oldest and most stable part of the project. It is our port of the OLE 2 Compound Document Format to pure Java. It supports both read and write functionality. All of our components ultimately rely on it by definition. Please see the POIFS project page for more information.

1.2. HSSF and XSSF for Excel Documents

HSSF is our port of the Microsoft Excel 97(-2007) file format (BIFF8) to pure Java. XSSF is our port of the Microsoft Excel XML (2007+) file format (OOXML) to pure Java. SS is a package that provides common support for both formats with a common API. They both support read and write capability. Please see the HSSF+XSSF project page for more information.

1.3. HWPF and XWPF for Word Documents

HWPF is our port of the Microsoft Word 97 (-2003) file format to pure Java. It supports read, and limited write capabilities. It also provides simple text extraction support for the older Word 6 and Word 95 formats. Please see the HWPF project page for more information. This component remains in early stages of development. It can already read and write simple files.

We are also working on the XWPF for the WordprocessingML (2007+) format from the OOXML specification. This provides read and write support for simpler files, along with text extraction capabilities.

1.4. HSLF and XSLF for PowerPoint Documents

HSLF is our port of the Microsoft PowerPoint 97(-2003) file format to pure Java. It supports read and write capabilities. Please see the HSLF project page for more information.

We are also working on the XSLF for the PresentationML (2007+) format from the OOXML specification.

1.5. HPSF for OLE 2 Document Properties

HPSF is our port of the OLE 2 property set format to pure Java. Property sets are mostly use to store a document's properties (title, author, date of last modification etc.), but they can be used for application-specific purposes as well.

HPSF supports both reading and writing of properties.

Please see the HPSF project page for more information.

1.6. HDGF for Visio Documents

HDGF is our port of the Microsoft Viso 97(-2003) file format to pure Java. It currently only supports reading at a very low level, and simple text extraction. Please see the HDGF project page for more information.

1.7. HPBF for Publisher Documents

HPBF is our port of the Microsoft Publisher 98(-2007) file format to pure Java. It currently only supports reading at a low level for around half of the file parts, and simple text extraction. Please see the HPBF project page for more information.

1.8. HMEF for TNEF (winmail.dat) Outlook Attachments

HMEF is our port of the Microsoft TNEF (Transport Neutral Encoding Format) file format to pure Java. TNEF is sometimes used by Outlook for encoding the message, and will typically come through as winmail.dat. HMEF currently only supports reading at a low level, but we hope to add text and attachment extraction shortly. Please see the HMEF project page for more information.

1.9. HSMF for Outlook Messages

HSMF is our port of the Microsoft Outlook message file format to pure Java. It currently only some of the textual content of MSG files, and some attachments. Further support and documentation is coming in slowly. For now, users are advised to consult the unit tests for example use. Please see the HPBF project page for more information.

Microsoft has recently added the Outlook file format to its OSP. More information is now

available making implementation of this API an easier task.

2. What is it?

The Apache POI project is the master project for developing pure Java ports of file formats based on Microsoft's OLE 2 Compound Document Format. OLE 2 Compound Document Format is used by Microsoft Office Documents, as well as by programs using MFC property sets to serialize their document objects.

Apache POI is also the master project for developing pure Java ports of file formats based on Office Open XML (ooxml.) OOXML is part of an ECMA / ISO standardisation effort. This documentation is quite large, but you can normally find the bit you need without too much effort! ECMA-376 standard is here, and is also under the Microsoft OSP.

3. Component Map

The Apache POI distribution consists of support for many document file formats. This support is provided in several Jar files. Not all of the Jars are needed for every format. The following tables show the relationships between POI components, Maven repository tags, and the project's Jar files.

Component	Application type	Maven artifactId
POIFS	OLE2 Filesystem	poi
HPSF	OLE2 Property Sets	poi
HSSF	Excel XLS	poi
HSLF	PowerPoint PPT	poi-scratchpad
HWPF	Word DOC	poi-scratchpad
HDGF	Visio VSD	poi-scratchpad
HPBF	Publisher PUB	poi-scratchpad
HSMF	Outlook MSG	poi-scratchpad
XSSF	Excel XLSX	poi-ooxml
XSLF	PowerPoint PPTX	poi-ooxml
XWPF	Word DOCX	poi-ooxml
OpenXML4J	OOXML	poi-ooxml-schemas, ooxml-schemas

This table maps artifacts into the jar file name. "version-yyyymmdd" is the POI version stamp. For the latest release it is 3.6-20091214.

Maven artifactId	Prerequisites	JAR
poi	commons-logging, log4j	poi-version-yyyymmdd.jar
poi-scratchpad	poi	poi-scratchpad-version-yyyymmdd.jar
poi-ooxml	poi, poi-ooxml-schemas, dom4j	poi-ooxml-version-yyyymmdd.jar
poi-ooxml-schemas (*)	xmlbeans, stax-api-1.0.1	poi-ooxml-schemas-version-yyyymmdd.jar
poi-examples (*)	poi, poi-scratchpad, poi-ooxml	poi-examples-version-yyyymmdd.jar
ooxml-schemas	xmlbeans, stax-api-1.0.1	ooxml-schemas-1.1.jar

(*) starting with 3.6-beta1-20091124.

poi-ooxml requires poi-ooxml-schemas. This is a substantially smaller version of the ooxml-schemas jar (ooxml-schemas-1.1.jar for POI 3.7 or later, ooxml-scheams-1.0.jar for POI 3.5 and 3.6). The larger ooxml-schemas jar is normally only required for development

The OOXML jars require a stax implementation. Previously we suggested "geronimo-stax-api_1.0_spec", but now "stax-api-1.0.1" is used for better compatibilty with other Apache projects. Any compliant implementation should work fine though.

4. Examples

Small sample programs using the POI API are available in the *src/examples* directory of the source distribution. Before studying the source code you might want to have a look at the "Examples" section of the POI API documentation.

Also note that we now include all of the examples in the distribution.

5. Contributed Software

Besides the "official" components outlined above there is some further software distributed with POI. This is called "contributed" software. It is not explicitly recommended or even maintained by the POI team, but it might still be useful to you.

5.1. POI Browser

The POI Browser is a very simple Swing GUI tool that displays the internal structure of a Microsoft Office file and especially the property set streams. Further information and

instructions how to execute it can be found in the POI Browser package description.