

Apache POI - Java API To Access Microsoft Format Files

by Andrew C. Oliver, Glen Stampoultzis, Avik Sengupta, Rainer Klute

1. 28 September 2009 - POI 3.5-FINAL available

The Apache POI team is pleased to announce the release of 3.5 FINAL. This release brings many improvements including support for the new OOXML formats introduced in Office 2007, such as XLSX and DOCX.

A full list of changes is available in the [change log](#). People interested should also follow the [dev list](#) to track progress.

See the [downloads](#) page for more details.

2. Purpose

The POI project consists of APIs for manipulating various file formats based upon Microsoft's OLE 2 Compound Document format, and Office OpenXML format, using pure Java. In short, you can read and write MS Excel files using Java. In addition, you can read and write MS Word and MS PowerPoint files using Java. POI is your Java Excel solution (for Excel 97-2007). However, we have a complete API for porting other OLE 2 Compound Document formats and welcome others to participate.

OLE 2 Compound Document Format based files include most Microsoft Office files such as XLS and DOC as well as MFC serialization API based file formats.

Office OpenXML Format based files include the new (2007+) xml based file formats, including Microsoft office files such as XLSX, DOCX and PPTX.

As a general policy we try to collaborate as much as possible with other projects to provide this functionality. Examples include: [Cocoon](#) for which there are serializers for HSSF; [Open Office.org](#) with whom we collaborate in documenting the XLS format; and [Lucene](#) for which we provide format interpreters. When practical, we donate components directly to those projects for POI-enabling them.

2.1. Why/when would I use POI?

We'll tackle this on a component level. POI refers to the whole project.

So why should you use POIFS, HSSF or XSSF?

You'd use POIFS if you had a document written in OLE 2 Compound Document Format, probably written using MFC, that you needed to read in Java. Alternatively, you'd use POIFS to write OLE 2 Compound Document Format if you needed to inter-operate with software running on the Windows platform. We are not just bragging when we say that POIFS is the most complete and correct implementation of this file format to date!

You'd use HSSF if you needed to read or write an Excel file using Java (XLS). You'd use XSSF if you need to read or write an OOXML Excel file using Java (XLSX). The combined SS interface allows you to easily read and write all kinds of Excel files (XLS and XLSX) using Java.

3. Components To Date

3.1. Overview

The following are components of the entire POI project and a brief summary of their purpose.

3.2. 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.

3.3. 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. They both supports read and write capability. Please see [the HSSF+XSSF project page](#) for more information.

3.4. HWPF for Word Documents

HWPF is our port of the Microsoft Word 97 file format to pure Java. It supports read, and

limited write capabilities. Please see [the HWPF project page for more information](#). This component is in the early stages of development. It can already read and write simple files.

Presently we are looking for a contributor to foster the HWPF development. Jump in!

3.5. HSLF 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](#).

3.6. HPSF for 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.

3.7. HDGF for Visio Documents

HDGF is our port of the Microsoft Visio 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](#).

3.8. 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](#).

3.9. Component map

The POI distribution consists of several JAR files. Not all of them are needed in every case. The following table shows the relationships between POI components and the JAR files.

Component	JAR	Maven artifactId
POIFS	poi-version-yyyyymmdd.jar	poi
HPSF	poi-version-yyyyymmdd.jar	poi
HSSF	poi-version-yyyyymmdd.jar	poi

XSSF	poi-ooxml-version-yyyyymmdd.jar	poi-ooxml
HLSF	poi-scratchpad-version-yyyyymmdd.jar	poi-scratchpad
HWPE	poi-scratchpad-version-yyyyymmdd.jar	poi-scratchpad
HDGF	poi-scratchpad-version-yyyyymmdd.jar	poi-scratchpad
HPBF	poi-scratchpad-version-yyyyymmdd.jar	poi-scratchpad
HSMF	poi-scratchpad-version-yyyyymmdd.jar	poi-scratchpad

4. Contributing

So you'd like to contribute to the project? Great! We need enthusiastic, hard-working, talented folks to help us on the project in several areas. The first is bug reports and feature requests! The second is documentation - we'll be at your every beck and call if you've got a critique or you'd like to contribute or otherwise improve the documentation. Last, but not least, we could use some binary crunching Java coders to chew through the complexity that characterizes Microsoft's file formats and help us port new ones to a superior Java platform!

So if you're motivated, ready, and have the time, join the mail lists and we'll be happy to help you get started on the project!

Copyright (c) @year@ The Apache Software Foundation. All rights reserved.