

# Apache POI - Java API To Access Microsoft Format Files

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

## 1. POI 3.1-FINAL Released (2008-06-29)

The POI team is pleased to announce the release of 3.1 FINAL, the latest release of Apache POI. There have been many important bug fixes since the 3.0.2 release and a lot of new features.

A full list of changes is available in [the changelog](#), and [download](#) the source and binaries from your [local mirror](#).

The release is also available from the central Maven repository under Group ID "org.apache.poi" and Version "3.1-FINAL".

## 2. Office Open XML Support - POI 3.5

We are currently working to support the new Office Open XML file formats, such as XLSX and PPTX, which were introduced in Office 2007.

Support for these is currently only available in the 3.5 series, which is in a svn branch (not trunk). POI 3.5.1 beta 1 was released on 2008-07-18, and is our first release including the OOXML support. It is available to download from your favourite [local mirror](#).

Please note that the documentation on the website is for the 3.1 series of POI. Documentation on the new OOXML support is included in the 3.5.1 beta 1 distribution.

## 3. Purpose

The POI project consists of APIs for manipulating various file formats based upon Microsoft's OLE 2 Compound Document format using pure Java. In short, you can read and write MS Excel files using Java. Soon, you'll be able to read and write Word, PowerPoint and Visio files using Java. POI is your Java Excel solution as well as your Java Word solution. 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.

At this time, none of our releases support the new Office Open XML file formats, such as .xlsx or .docx. Work to support these is in progress, and people interested should follow the [dev list](#). We expect this support to make it into a full release by the summer.

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.

### **3.1. 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 or HSSF?

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, write or modify an Excel file using Java (XLS).

## **4. Components To Date**

### **4.1. Overview**

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

### **4.2. 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.

### **4.3. HSSF for Excel Documents**

## *Apache POI - Java API To Access Microsoft Format Files*

HSSF is our port of the Microsoft Excel 97(-2003) file format (BIFF8) to pure Java. It supports read and write capability. (Support for Excel 2007 .xlsx files is in progress). Please see [the HSSF project page](#) for more information.

### **4.4. 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!

### **4.5. 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](#).

### **4.6. 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.

### **4.7. 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](#).

### **4.8. 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](#).

## **5. Contributing**

So you'd like to contribute to the project? Great! We need enthusiastic, hard-working,

## *Apache POI - Java API To Access Microsoft Format Files*

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. We could especially use some help documenting the HSSF file format! 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. \$Revision: 690517 \$ \$Date: 2008-08-30 15:47:33 +0100 (Sat, 30 Aug 2008) \$