

# Jakarta POI - Java API To Access Microsoft Format Files

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

## 1. 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 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.

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'll soon have file format interpreters. When practical, we donate components directly to those projects for POI-enabling them.

### 1.1. 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 or write an Excel file using Java (XLS). You can also read and modify spreadsheets using this API, although right now writing is more mature.

## **2. Components To Date**

### **2.1. Overview**

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

### **2.2. 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.

### **2.3. HSSF for Excel Documents**

HSSF is our port of the Microsoft Excel 97(-2002) file format (BIFF8) to pure Java. It supports read and write capability. Please see [the HSSF project page](#) for more information.

### **2.4. HWPF for Word Documents**

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

### **2.5. HPSF for Document Properties**

HPSF is our port of the OLE 2 property set format to pure Java. Property sets are mostly used 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 reading and writing of properties. However, with the current POI release only reading is possible. In order to write properties, you'll have to fetch the latest POI version [from the Subversion repository](#).

Please see [the HPSF project page](#) for more information.

## **3. Contributing**

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

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. 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: 376062 \$ \$Date: 2006-02-08 20:50:58 +0000 (Wed, 08 Feb 2006) \$