Apache POI - HWPF - Java API to Handle Microsoft Word Files

Overview

by Nicola Ken Barozzi, Andrew C. Oliver, Ryan Ackley, Rainer Klute

1. Overview

HWPF is the name of our port of the Microsoft Word 97(-2007) file format to pure Java. It also provides limited read only support for the older Word 6 and Word 95 file formats.

The partner to HWPF for the new Word 2007 .docx format is *XWPF*. Whilst HWPF and XWPF provide similar features, there is not a common interface across the two of them at this time.

HWPF is still in early development. It is in the scratchpad section of the SVN. You will need to ensure you either have a recent SVN checkout, or a recent SVN nightly build (including the scratchpad jar!)

Source code in the *org.apache.poi.hdf* tree is the old legacy code. Source in the *org.apache.poi.hwpf.model* tree is the old legacy code refactored into an new object model. Those packages contains Java representation of internal Word format structure. This code is "internal", it shall not be used by your code. Because of backward-compatibility some API still has references to those packages. They are subject to be deprecated and removed. Code from *org.apache.poi.hwpf.usermodel* package is actual public and user-friendly (as much as possible) API to access document parts. Source code in the *org.apache.poi.hwpf.extractor* tree is a wrapper of this to facilitate easy extraction of interesting things (eg the Text), and *org.apache.poi.hwpf.converter* package contains Word-to-HTML and Word-to-FO converters (latest can be used to generate PDF from Word files when using with Apache FOP). Also there is a small file-structure-dumping utility in *org.apache.poi.hwpf.dev* package, primally for developing purposes.

The main entry point to HWPF is HWPFDocument. Currently it has a lot of references both to internal interfaces (*org.apache.poi.hwpf.model* package) and public API (*org.apache.poi.hwpf.usermodel*) package. It is possible that it will be split into two different interfaces (like WordFile and WordDocument) in later versions.

Copyright © 2002-2011 The Apache Software Foundation All rights reserved.

Word document can be considered as very long single text buffer. HWPF API provides "pointers" to document parts, like sections, paragraphs and character runs. Usually user will iterates over main document part sections, paragraphs from sections and character runs from paragraph. Each such interface is a pointer to document text subrange along with additional properties (and they all extends same Range parent class). There is additional Range implementations like Table, TableRow, TableCell, etc. Some structures like Bookmark or Field can also provide subranges pointers.

Changing file content usually requires a lot of synchronized changes in those structures like updating property boundaries, position handlers, etc. Because of that HWPF API shall be considered as not thread safe. In addition, there is a "one pointer" rule for changing content. It means you should not use two different Range instances at one time. More precisely, if you are changing file content using some range pointer, all other range pointers except parents' ones become invalid. For example if you obtain overall range (1), paragraph range (2) from overall range and character run range (3) from paragraph range and change text of paragraph, character run range is now invalid and should not be used, but overall range pointer still valid. Each time you obtaining range (pointer) new instance is created. It means if you obtained two range pointers and changed document text using first range pointer, second one became invalid.

2. XWPF Patches Required!

At the moment, XWPF covers many common use cases for reading and writing .docx files. Whilst this is a great thing, it does mean that XWPF does everything that the current POI committers need it to do, and so none of the committers are actively adding new features.

If you come across a feature in XWPF that you need, and isn't currently there, please do send in a patch to add the extra functionality! More details on contributing patches are available on the "Contribution to POI" page.

3. HWPF Pointman Needed!

At the moment we unfortunately do not have someone taking care for HWPF and fostering its development. What we need is someone to stand up, take this thing under his hood as his baby and push it forward. Ryan Ackley, who put a lot of effort into HWPF, is no longer on board, so HWPF is an orphan child waiting to be adopted.

If **you** are interested in becoming the new HWPF pointman, you should look into the Microsoft Word internals. A good starting point seems to be Ryan Ackley's overview. Full details on the word format is available from Microsoft, but the documentation can be a little hard to get into at first... Try reading the overview first, and looking at the existing code, then finally look up the documentation for specific missing features.

Copyright © 2002-2011 The Apache Software Foundation All rights reserved.

As a first step you should familiarize yourself with the source code, examples, test cases, and the HWPF patches available at Bugzilla (if any). Then you should compile an overview of

- the current HWPF status,
- the patches in Bugzilla to be checked in (and those that should better be ditched),
- the available test cases and the test cases still to be written,
- the available documentation and the docs to be written,
- anything else that seems reasonable

When you start coding, you will not yet have write access to the SVN repository. Please submit your patches to Bugzilla and nag the dev list until someone commits them. Besides the actual checking in of HWPF patches, current POI committers will also do some minor reviews now and then of your source code patches, test cases and documentation to help ensure software quality. But most of the time you will be on your own. However, anyone offering useful contributions over a period of time will be offered committership!

Please do not forget to write JUnit test cases and documentation! We won't accept code that doesn't come with test cases. And please consider that other contributors should be able to understand your source code easily. If you need any help getting started with JUnit test cases for HWPF, please ask on the developers' mailing list! If you show that you are prepared to stick at it you will most likely be given SVN commit access. See "Contribution to POI" page for more details and help getting started.

Of course we will help you as best as we can. However, presently there is no committer who is really familiar with the Word format, so you'll be mostly on your own. We are looking forward for you and your contributions! Honor and glory of becoming a POI committer are waiting!

Copyright © 2002-2011 The Apache Software Foundation All rights reserved.