POI-HSLF - A Quick Guide

Overview

by Nick Burch

1. Basic Text Extraction

For basic text extraction, make use of `org.apache.poi.hslf.extractor.PowerPointExtractor`. It accepts a file or an input stream. The `getText()` method can be used to get the text from the slides, and the `getNotes()` method can be used to get the text from the notes. Finally, `getText(true,true)` will get the text from both.

2. Specific Text Extraction

To get specific bits of text, first create a `org.apache.poi.hslf.usermodel.SlideShow` (from a `org.apache.poi.hslf.HSLFSlideShow`, which accepts a file or an input stream). Use `getSlides()` and `getNotes()` to get the slides and notes. These can be queried to get their page ID (though they should be returned in the right order).

You can then call `getTextRuns()` on these, to get their blocks of text. (One TextRun normally holds all the text in a given area of the page, eg in the title bar, or in a box). From the `TextRun`, you can extract the text, and check what type of text it is (eg Body, Title). You can allso call `getRichTextRuns()`, which will return the `RichTextRun`s that make up the `TextRun`. A `RichTextRun` is made up of a sequence of text, all having the same character and paragraph formatting.

3. Poor Quality Text Extraction

If speed is the most important thing for you, you don't care about getting duplicate blocks of text, you don't care about getting text from master sheets, and you don't care about getting old text, then `org.apache.poi.hslf.extractor.QuickButCruddyTextExtractor` might be of use.

QuickButCruddyTextExtractor doesn't use the normal record parsing code, instead it uses a

tree structure blind search method to get all text holding records. You will get all the text, including lots of text you normally wouldn't ever want. However, you will get it back very very fast!

There are two ways of getting the text back. `getTextAsString()` will return a single string with all the text in it. `getTextAsVector()` will return a vector of strings, one for each text record found in the file.

4. Changing Text

It is possible to change the text via `TextRun.setText(String)` or `RichTextRun.setText(String)`. It is not yet possible to add additional TextRuns or RichTextRuns.

When calling `TextRun.setText(String)`, all the text will end up with the same formatting. When calling `RichTextRun.setText(String)`, the text will retain the old formatting of that `RichTextRun`.

5. Adding Slides

You may add new slides by calling `SlideShow.createSlide()`, which will add a new slide to the end of the SlideShow. It is not currently possible to re-order slides, nor to add new text to slides (currently only adding Escher objects to new slides is supported).

6. Guide to key classes

- `org.apache.poi.hslf.HSLFSlideShow` Handles reading in and writing out files. Calls `org.apache.poi.hslf.record.record` to build a tree of all the records in the file, which it allows access to.
- `org.apache.poi.hslf.record.record` Base class of all records. Also provides the main record generation code, which will build up a tree of records for a file.
- `org.apache.poi.hslf.usermodel.SlideShow` Builds up model entries from the records, and presents a user facing view of the file
- `org.apache.poi.hslf.model.Slide` A user facing view of a Slide in a slidesow. Allows you to get at the Text of the slide, and at any drawing objects on it.
- `org.apache.poi.hslf.model.TextRun` Holds all the Text in a given area of the Slide, and will contain one or more `RichTextRun`s.
- `org.apache.poi.hslf.usermodel.RichTextRun` Holds a run of text, all having the same character and paragraph stylings. It is possible to modify text, and/or text stylings.
- `org.apache.poi.hslf.extractor.PowerPointExtractor` Uses the model code to allow extraction of text from files

- `org.apache.poi.hslf.extractor.QuickButCruddyTextExtractor`
  Uses the record code to extract all the text from files very fast, but including deleted text (and other bits of Crud).