JOOMM

©Shane O'Neill
scjon@acm.org

10th November 2005

1 Overview

JOOMM is a class to provide mail-merge functionality with OpenOffice documents from Java programs. This class is a stand-alone class, and doesn't rely on any dependency classed/libraries.

The purpose of this document is to detail the usage of the class, as I've previously received an email requesting more information in its usage. Proper documentation was something I had wanted to do one way or another, as there are too many OS projects which sadly lack decent documentation. So, I will try to remedy that for this project at least.

2 Status

The project code has beta status, and any bugs I've come across have been fixed, that's not to say that there are none left, but for all intents and purposes the development of this project is final. However feedback and bug reports are always welcome. Any bugs brought to my attention will be fixed.

3 The Process in Brief

1. Enter the mapping place-holders into the mapping file.

2. A call to initMappings() should then be made, passing the name (and path if necessary) of the mapping file. This will then return a Vector of the place-holders that were contained within the mapping file.

3. parseDocument() can then be called, passing the name of the template file, actual data[1], and mapping entries.

[1] There is a one to one correlation between the actual data and the mapping entries passed, i.e. the first actual datum passed is used to replace the placeholder contained within the first node of the mapping Vector.

It's possible that the path and name of the merged file may need to be specified. These have default values, however, if your platform in not a Unix based system then the path will have to be specified as it defaults to '/tmp'.

4 Usage

The class requires a you to have a template (as with any mail-merge) document ready for insertion of data. The place-holders (patterns) used to mark particular areas of the document are quite simply anything enclosed in double angled brackets, i.e. <<persons_name>>. The information specified between the brackets is arbitrary, and is specified in a separate mapping file.

4.1 Mapping File Format

The mapping file is simply a pain text file, where the place-holders (patters) to be used within the template are defined. Unlike the template file, the place-holders within the mapping file should not contain angled brackets. Each placeholder should be on a separate line, e.g.

```
company_name
company_address
company_contact
```

Also, take care to note the order in which these place-holders are listed in the file, as data related to these must be passed to the parseDocument() method in the same order. The should be at least the same amount of entries in the mapping file as there is data being passed to parseDocument().

Comments can also be included within the mapping file, each line starting with a '#' character are considered to be comments and are ignored when parsing the file.