|  |
| --- |
| logo-wygwam |
| Software Architecture |
| OpenXML4J |
|  |
| **Julien Chable** |
| **4/19/2007**  **Draft – v0.1** |

|  |
| --- |
| This is the specification of the Open XML API for Java (openxml4j), an open source API for office documents using the ECMA 376 Office Open XML File Format, based on System.IO.Packaging .NET API. |

# History

|  |  |  |
| --- | --- | --- |
| Date | Version | Comment |
| 4/1/2007 | 0.1 | Initial version |

# Contents

Contents

[1. History 2](#_Toc169497961)

[2. Contents 3](#_Toc169497962)

[3. References 4](#_Toc169497963)

[4. Definition and acronyms 4](#_Toc169497964)

[5. Overview 4](#_Toc169497965)

[6. Software architecture 5](#_Toc169497966)

[1. Overall architecture 5](#_Toc169497967)

[2. Components overview 6](#_Toc169497968)

[1. External component 6](#_Toc169497969)

[2. Common component 6](#_Toc169497970)

[a. Open Packaging Convention 6](#_Toc169497971)

[b. Shared ML library 6](#_Toc169497972)

[3. Implementation specific components 6](#_Toc169497973)

[a. Strongly Typed Part 6](#_Toc169497974)

[b. Object model 6](#_Toc169497975)

[7. Detail OPC architecture 7](#_Toc169497976)

[8. Testing 7](#_Toc169497977)

# References

This document refers to the following documents:

* ECMA-376 Office Open XML File Format : <http://www.ecma-international.org/publications/standards/Ecma-376.htm>
* Open Packaging Convention specifications : <http://www.ecma-international.org/publications/files/ECMA-ST/Office%20Open%20XML%20Part%202%20(DOCX).zip>

# Definition and acronyms

OOXML : Office Open XML

API : Application Programming Interface

# Overview

This document describes the architecture for the OpenXML4J API, an open source implementation of the ECMA-376 standard (Office Open XML File Format) for Java 5.

# Software architecture

## Overall architecture

This project is broken into several subprojects, each representing an aspect of the specifications. The architecture is based on a stack of components (i.e. part of the specifications), each component using the services from the component beneath :

This leads to the following subprojects cutting :

* Open Packaging Convention implementation (Part 2 – Open Packaging Convention)
* Shared markup language :
  + DrawingML (Part 4: Markup Language Reference - §5)
  + MathML (Part 4: Markup Language Reference - §7.1)
  + Extended Properties (Part 4: Markup Language Reference - §7.2)
  + Custom Properties (Part 4: Markup Language Reference - §7.3)
  + Custom XML Data Properties (Part 4: Markup Language Reference - §7.5)
  + [Compatibilty & extensibility]
    - VML
    - …
* WordprocessingML :
  + Strongly Typed Parts
  + Object Model
* SpreadsheetML :
  + Strongly Types Part
  + Object Model
* PresentationML :
  + Strongly Types Part
  + Object Model

## Components overview

The following is a brief description of the components.

### External component

The external components are third parties components which are reused in this project:

* **DOM4J** is a Free library to easily manipulate XML documents. Its approach makes developers very efficient to do some heavy XML tasks quickly. This library will be use when the use of XMLBeans doesn’t make sense to generate Java Types in order to manipulate XML data.
* **XMLBeans (+Saxon)** is a technology for accessing XML by binding it to Java types. XMLBeans provides several ways to access the XML. First, through XML Schema that has been compiled to generate Java types that represent schema types; then, with a cursor model and finally with the XML DOM.

### Common component

The common components are a collection of components shared across the upper components:

### Open Packaging Convention

The Open Packaging Convention component will be use as a fundamental by the upper components (WordprocessingML, SpreadsheetML and PresentationML component).

In the first releases, only the mandatory feature will be implemented (some ‘Should’ features will be implemented as well).

Reference: Open Packaging Convention specifications: <http://www.ecma-international.org/publications/files/ECMA-ST/Office%20Open%20XML%20Part%202%20(DOCX).zip>

### Shared ML library

This component will use by upper components (WordprocessingML, SpreadsheetML and PresentationML) to create or manipulate the shared XML content of Open XML document like DrawingML (for graphic rendering) or MathML (to describe mathematical content).

These components will be developing in priority:

* DrawinML
* Extended Properties

### Implementation specific components

### Strongly Typed Part

This first step will provide specific markup language part like a Wordprocessing document with specific properties (background color, headers and footers, …) or PresentationML with specific methods (move or remove a particular slide, …).

A STP doesn’t truly manipulate the content of a document like a paragraph or cells content. It only

### Object model

The object model provides a set of classes to create content in a document.

# Detail OPC architecture

TODO

# Testing

Unit testing is a good practice to ensure that what you think you’re doing is what you actually want to do. To guarantee a good quality of the implementation and to validate the compliance to the specifications, we have to build several test suites.

We use the JUnit library for functional and unit testing.