

## Internal structure of ODF

Nowadays information technologies become more and more popular. And some of them became a part of our life. For example, all of us know what the program Microsoft Office means but no one knows how it works. In this article we consider such term as document electronic format also known as document markup.

There are so many document formats in the modern computer world. The most famous of them are PDF, ODF and DOC. We will consider ODF. It's free software and supported by many programmers in the world. The one of the best package of programs for working with documents of this format is LibreOffice. Before its creation it was Open Office. Before discussing of technology of template and placeholders there is some information about structure of this great format.

Each document ODT is a zip archive, so to get inside of the document, it is enough to rename the extension of the document to the zip and open this file with any zip-archiver[1]. After that we will see at least five xml-files and a few directories. These xml-files in addition to the direct content of the document there are so-called metadata. It is additional information that allows you to set certain parameters of the text, for example, font type and size, position on the page, print settings, or display monitor.

The most popular is now becoming a standard meta-information describing XML (eXtend Markup Language). The basic requirement of this standard is the principle of "user-friendly": xml-documents should be easily readable by a simple word processor, and xml-markup should be easily understood by a man.

The ODF is one of many implementations of the standard XML. The description logic language is based on standard XML [2]. So it will have a tree-like structure and consist of tags and attributes. This language described in this paper follows the system of rules and refers to declarative ones. It follows that instructions of this language defined data to be obtained rather than the process for

their obtainment. The structure of the language is represented as a tree of elements that are arranged in any order. XML script requires a main or root element. Therefore, opening the odf-document as a zip-file, to understand the structure of files and folders by their names will not be too big a problem. Especially for those people who at least once in thier life created html pages well at the simple level.

The main file with content is content.xml and style file is style.xml. The folders may be media files: images, audio and video files. In general, the ODF document is something like a website of age of static html-pages. And tags within xml-files are unlikely to be hard to understand:

```
<text:p text:style-name="norm"> Hello, ODF! </text:p>
```

For comparison here metadata might look like the in an old RTF format of "Microsoft":

```
{loch 0 s24lang1033i0? 0 u1055? U1088? U1080? U1074? U1077? U1090?,  
ODF!}
```

Metadata of the ODF look more readable than the meta-information of the RTF. I'm not talking about the DOC the metadata is stored in a binary form. Of course let's look at how the readability looks like in an alternative OOXML:

```
<w:t>Hello, ODF! </w:t>
```

It would seem that OOXML is simplier and more compact. This is especially notable for programmers who like to write all sorts of workflow automation. But this simplicity may be bad. For example construction "w: t" may mean many things. Markup OOXML is so to speak intuitively incomprehensible but the ODF is more readable for people[3].

Also it gives us the possibility to separate content and meta information from the file without even reading the documentation about all of standard tags. As is known "user-friendly" is when it is not necessary to read a lot of documentation.

The documentation, of course, programmers have to read. But the documentation ODF is also more convenient although only because of its small size documents ODF. Thus ODF is a champion of friendliness and convenience in comparison with other popular office formats.

In summary usage of ODF has following advantages:

Customers save their data in an open format such as ODF. They avoid a risk of being relied on a single supplier, they are free to choose other software if their current vendor will leave a market, raise prices, change their software or change the terms of the license agreement.

The OpenDocument is the only standard for editable office documents, approved by an independent committee on standards and implemented by multiple software vendors.

The ODF can be used by any software provider, including, inter alia, supplier and developer of proprietary software using free licenses[4]. It is the only standard that can be used by anyone without restrictions, without the use of software of a concrete vendor. Many applications already use this standard.

### **REFERENCE LIST**

1. ODF TC Creates Advanced Document Collaboration Subcommittee, 2010-12-05.
2. David Hunter, Jeff Rafter Beginning XML. M.: Williams, 2009. – 1344.
3. [https://en.wikipedia.org/wiki/Office\\_Open\\_XML](https://en.wikipedia.org/wiki/Office_Open_XML).
4. ODF TC Creates Advanced Document Collaboration Subcommittee, 2010-12-05, retrieved 2014-01-31