.doc versus .docx

Author: Mihai Rotaru

Date: 15 Dec 2011

Ever since Microsoft Office 2007 was launched by Microsoft, a new headache began to plague office workers all around the world. The culprit was the new default format used by Microsoft's office suite, .docx. This format is intended to supersede the .doc format, which was perhaps perceived as obsolete by Microsoft.

The transition could have been smoother, if not for the fact that Microsoft Office versions older than 2007 cannot open the new format in their default configuration. It is possible for Office 2003 users to open .docx documents, after installing the Compatibility Kit, but they will be warned that the document might not be displayed as it would in later Office versions, and that some elements not supported by the older Word might not be present. This can lead to users making changes to a document, only to find out later that the document looks different, or has elements missing, when viewed with a different version of the Office suite.

I decided to perform a dissection of a .docx file, and get a glimpse at it's internal workings. I had a .docx, which was a BBC article I downloaded and converted into a .docx file for printing as part of an assignment. I changed it's extension to .zip, and indeed, the archive contains a number of folders, and a [Content_Types].xml file at it's root.

The [Content_Types].xml describes the contents of the other .xml files, scattered throughout the folder hierarchy. For the 'xml' extension, by default the `ContentType` is set to "application/xml", but certain files override this setting via `Override` elements. For example, "/word/document.xml" has the "application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml" content type.

The _rels folder for this document contains a single file, .rels - which doesn't have an xml extension, but is xml because it has the xml declaration as the first line, and is a well-formed xml document. It's root element is `Relationships`, which contains a number of `Relationship` elements, each having the `Id`, `Type` and `Target` attributes. I couldn't exactly figure out how are these used.

The docProps folder is a bit more interesting; it has two documents inside it, core.xml and app.xml. The core.xml is quite simple, and it is easy to see that it contains various meta-data about the document (`coreProperties`, as the root element is named).

I found out a number of interesting things - for example, the document had a `dc:creator` element (`dc` is the namespace alias for "http://purl.org/dc/elements/1.1/"), which has the value of my Middlesex University ID used for logging in to computers on the campus; I did not know Word automatically stores this information inside the file itself. I also noted that this file stores the dates the file was created and modified - therefore, .docx documents store this information independent of the file system. The file also contains information about who last modified the file.

The app.xml file is a bit of a mixed bag - I was expecting it to contain information about the version of Microsoft Word the .docx document was created with - which it did, inside the `AppVersion` element - but it also contains metadata about the document's contents; such as the number of paragraphs, pages, words and characters. In addition, it has a `CompanyName` element, which was bestowed the value 'Middlesex University'.

The documents inside the word folder contain the document data. The roles of most of the files inside it are not hard to guess; for example, the fontTable.xml file stores information about each of the fonts used in the document. The settings.xml file contains settings such as the zoom level set when the document was last edited, and the decimal symbol used. Images and other media will be stored inside the 'media' folder; in this case, it contained an image named image1.gif. This hierarchy resembles an XHTML webpage - the document written in a markup language, accompanied by resources (such as images) and stylesheets.

The document.xml file is at the core of the .docx folder hierarchy, containing the text and layout information. All the other .xml files are used to describe certain aspects of this file, and to support Word in deciding how to process or display it.

I then saved the file as a simple .doc; after which I renamed it to .zip and tried to open it, but the archiving software gave me an error. I then opened the document with a hex editor (010 Editor), and the reason became clear: the .doc file is a binary format. Some of the text was readable while browsing the hex data, but most of the file was difficult to make sense of.