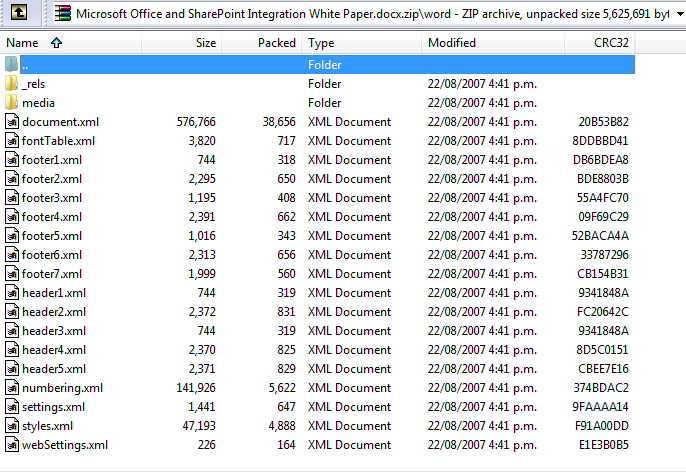
**Evaluation Only. Created with Aspose.Words. Copyright 2003-2020 Aspose Pty Ltd.**

A DOCX document generated by Aspose.Words follows the Open Packaging Convention and can be opened in a ZIP-capable application.



##### OOXML is Open, Why Use Aspose.Words?

Being XML-based, Office Open XML is heralded as an enabling technology. It is true that Office Open XML makes it possible to build document processing and generating applications using just the XML classes without relying on third-party libraries such as Aspose.Words. However, we strongly believe it is still very beneficial to use Aspose.Words when you have to deal with OOXML documents, rather than work through XML or other libraries.

The OOXML specification is several thousand pages long. Being open and standard does not mean being simple. To correctly process or generate OOXML documents one must invest in learning the format well.

In addition to making it simpler to correctly process and generate valid documents, Aspose.Words provides the following important features you would not have when working with OOXML files directly via XML or other third-party libraries:

* Quality conversions between many popular document formats, including conversion to PDF , XPS and printing.
* Ability to build documents from fragments from one or multiple documents, while automatically merging per-document structures such as styles and lists.
* High-level functions such as field update, accept revisions or mail merge can be invoked with just one line of code.
* Access to flat Range-like operations such as find and replace, get/set text of a bookmark, form field, document field or a node.

Consider the following example. It is a simple paragraph that contains text “Hello World” and the word “Hello” is bold. Now imagine you need to write a program that will search for all “Hello World” phrases in the document and replace them with “Goodbye Earth”.

What started out as a seemingly simple task of loading, modifying and saving an XML file does not look so simple anymore. In fact, it has become very complex. It takes a non-trivial algorithm to find and replace flat text across an XML tree. Have you ever wondered why standard XML classes such as **XmlDocument** do not offer find and replace functionality.