Structure of a WordprocessingML document

01/12/2024

This topic discusses the basic structure of a **WordprocessingML** document and reviews important Open XML SDK classes that are used most often to create **WordprocessingML** documents.

The basic document structure of a **WordProcessingML** document consists of the `<document>` and `<body>` elements, followed by one or more block level elements such as `<p>`, which represents a paragraph. A paragraph contains one or more `<r>` elements. The `<r>` stands for run, which is a region of text with a common set of properties, such as formatting. A run contains one or more `<t>` elements. The `<t>` element contains a range of text.

Important WordprocessingML Parts

The Open XML SDK API provides strongly-typed classes in the DocumentFormat.OpenXML.WordprocessingML namespace that correspond to **WordprocessingML** elements.

The following table lists some important **WordprocessingML** elements, the **WordprocessingML** document package part that the element corresponds to (where applicable) and the managed class that represents the element in the Open XML SDK API.

Expand table

Package Part	WordprocessingML Element	Open XML SDK Class	Description
Main Document	document	Document	The root element for the main document part.
Comments	comments	Comments	The root element for the comments part.
Document Settings	settings	Settings	The root element for the document settings part.
Endnotes	endnotes	Endnotes	The root element for the endnotes part.
Footer	ftr	Footer	The root element for the footer part.
Footnotes	footnotes	Footnotes	The root element for the footnotes part.
Glossary Document	glossaryDocument	GlossaryDocument	The root element for the glossary document part.
Header	hdr	Header	The root element for the header part.
Style Definitions	styles	Styles	The root element for a Style Definitions part.

Minimum Document Scenario

A **WordprocessingML** document is organized around the concept of stories. A story is a region of content in a **WordprocessingML** document. **WordprocessingML** stories include:

- comment
- endnote
- footer
- footnote
- frame, glossary document
- header
- main story
- subdocument
- text box

Not all stories must be present in a valid **WordprocessingML** document. The simplest, valid **WordprocessingML** document only requires a single story—the main document story. In **WordprocessingML**, the main document story is represented by the main document part. At a minimum, to create a valid **WordprocessingML** document using code, add a main document part to the document.

The following information from the ISO/IEC 29500 introduces the WordprocessingML elements required in the main document part in order to complete the minimum document scenario.

The main document story of the simplest WordprocessingML document consists of the following XML elements:

- `document` — The root element for a WordprocessingML's main document part, which defines the main document story.
- `body` — The container for the collection of block-level structures that comprise the main story.
- `p` — A paragraph.
- `r` — A run.
- `t` — A range of text.

Open XML SDK Code Example

The following code uses the Open XML SDK to create a simple **WordprocessingML** document that contains the text passed in as the second parameter

C#

C#

```
using DocumentFormat.OpenXml.Packaging;
using DocumentFormat.OpenXml.Wordprocessing;

static void CreateWordDoc(string filepath, string msg)
{
    using (WordprocessingDocument doc = WordprocessingDocument.Create(filepath,
DocumentFormat.OpenXml.WordprocessingDocumentType.Document))
    {
        // Add a main document part.
        MainDocumentPart mainPart = doc.AddMainDocumentPart();

        // Create the document structure and add some text.
        mainPart.Document = new Document();
        Body body = mainPart.Document.AppendChild(new Body());
        Paragraph para = body.AppendChild(new Paragraph());
        Run run = para.AppendChild(new Run());

        // String msg contains the text from the msg parameter"
        run.AppendChild(new Text(msg));
    }
}
```

Generated WordprocessingML

After you run the Open XML SDK code in the previous section to generate a document, you can explore the contents of the .zip package to view the **WordprocessingML** XML code. To view the .zip package, rename the extension on the minimum document from .docx to .zip. The .zip package contains the parts that make up the document. In this case, since the code created a minimal **WordprocessingML** document, there is only a single part—the main document part.The following figure shows the structure under the word folder of the .zip package for a minimum document that contains a single line of text.**Art placeholder**The document.xml file corresponds to the **WordprocessingML** main document part and it is this part that contains the content of the main body of the document. The following XML code is generated in the document.xml file when you run the Open XML SDK code in the previous section.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:r>
        <w:t>The text passed as the second parameter goes here</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

Typical Document Scenario

A typical document will not be a blank, minimum document. A typical document might contain comments, headers, footers, footnotes, and endnotes, for example. Each of these additional parts is contained within the zip package of the wordprocessing document.

The following figure shows many of the parts that you would find in a typical document.

Figure 1. Typical document structure