

Create .docx with Tcl

Rolf Ade

A .docx is just a zip archive

- See a simple example on the right
- Except for media data, all files are XML documents
- The format is “Office Open XML”, specified by ECMA-376 and ISO/IEC 29500
- Prior art: ooxml for .xlsx by Alexander Schöpe

```
.
├─ [Content_Types].xml
├─ docProps
│   └─ app.xml
│   └─ core.xml
├─ _rels
│   └─ .rels
└─ word
    ├── document.xml
    ├── fontTable.xml
    ├── _rels
    │   └─ document.xml.rels
    ├── settings.xml
    └─ styles.xml
```

WordprocessingML

- Still the default document format of MS Word up to 365
- Libreoffice also reads and writes it very well
- The specification defines “consumers” and “producers”
- MS Word and libreoffice are both, consumers and producers
- The library presented here is just a producer
- WordprocessingXML is more like a translation of a binary format into XML than a well-designed markup language

Create a .docx

- Create a docx object command:
`set docx [docx new]`
- Add content top down
- Write the thing with:
`$docx write MyDocument.docx`
- The basic methods to create content are:
paragraph
append
image
table
textbox

Create a .docx II

- Auxiliary methods are:

comment	configure	field
footer	header	import
jumpto	mark	numbering
pagebreak	pagesetup	readpart
sectionend	sectionstart	settings
simplecomment	simpletable	style
tablecell	tablerow	url
writepart		

Fine control

- Every method expects (after a few mandatory arguments in some cases) optional option value pairs
- Often, the value of an option has to be in key value format
- Unknown options are reported together with the list of the known options
- In case of key value options, unknown keys are reported together with the list of known keys
- In case of enumeration as value in a key value pair, unknown values are reported together with the list of the allowed values
- The stack trace points directly to the user code line

Text and Formatting

- Text is added in paragraphs
- Headings and listing points are just special formatted paragraphs
- If a part of a paragraph has to be formatted differently use the method `append` to append to the current paragraph with other formatting values
- Then append the rest of the paragraph

```
$mydocx paragraph "Lorem ipsum "  
$mydocx append "important" -bold yes  
$mydocx append $restOfTheParagraph
```

Use Styles

- Same advice as for users of the gui interfaces
- There are styles for paragraphs, ranges of characters, (numbered or bullet point) listings and tables
- With listings, you have no choice but to format them with styles. For paragraphs, text and tables it is most certainly advisable
- Styles may be created from scratch
- Some simple facilities to import styles from other .docx files are available

So, what is missing?

- Footnotes/endnotes
- Bullet points from custom images
- Just kidding
- Documentation (repeating larger parts of the 1500 pages of the standard?)
- Field testing
- Developer documentation
- Higher-level scripted markup on the base of this WordprocessingML primitives.

So, what is missing, continued

- Images always need options for both width and height
- Higher-level styles import
- More fine grain document settings control.
- A libreoffice extension to put placeholders into a docx documents given, that are replaced with content while generated (template style of generating .docx)

Lessons

- The tDOM *fromScript feature provides a mighty and convenient pattern.
- It is not easy to write a scripted library that's "behaving nicely", with helpful error messages and stack traces
- While catch and return -code error \$msg helps in case of basic needs (at the cost of more ugly code) more problems arise due to script arguments (evaluated arguments).
- I want my Tcl even more lisp-ish

Questions?