

UniDocWrapper

A Minimalist Wrapper of Unioffice

PREPARED BY RICHARD J. CORDES
RJ.Cordes@COGSEC.org

PREPARED June 2019

Contents

Executive Summary 1

Introduction 1

 Acronyms and Terms 1

 Requirements 2

 Constraints and Concerns 2

Package Overview 3

 Architecture and Components 4

 Exported 4

Current Status and Future Development 4



Executive Summary

A simple wrapper around Unioffice Docx features for reading and editing of pre-defined templates found in docx files.

Introduction

At the moment, we're in need of a means of reading and editing structured .docx files for organisation of case/document notes. The Unioffice package has met this need, however, for our purposes, much of the package is unnecessary and its implementation has been unwieldy. In order to remedy this, UniDoc Wrapper (UDW) has been proposed.

Acronyms and Terms

For the purposes of this document, the following acronyms and terms have been clarified:

Term	Definition
UDW	UniDoc Wrapper
Docx	A Microsoft Word "Open XML" Format file. Docx files are smaller than their predecessor "DOC", as they are built using XML and zipped. They can also be opened and read by Google Docs.
DTD	Document Type Definition. A definition of a document type as per SGML markup language standards. XML and HTML are examples of mark-up languages.

Table 1: Terms and Acronyms



Requirements

This is a highly minimalist package and its only requirements are as follows:

- Read Docx in
- Search Docx by heading
- Insert and Append Sections of text
- Convert document to map
- Save Docx

Constraints and Concerns

Any system which is inflexible is by its nature fragile unless its inputs are highly standardised. In this case, there is little option but to be inflexible, as docx files are complicated. All documents will have to fit to a certain style and structure, as will all expected inputs. Continuing along this thread, is the acknowledgement that images and other complementary material often found within docx files will have to be ignored for the time being.

The resulting files will unfortunately have to be treated carefully in order to maintain their stability and readability by the proposed wrapper. However, the wrapper's development is time sensitive, and future development may be focused on reducing fragility. This is not so much a constraint on the system as it is on the end-user but dealing with docx files which have been edited is of concern.

Finally, Unioffice, by Baliance, is governed by a GNU 3.0 license. Our use has not been considered "commercial". General commercial use is defined 18 USCS §31, and, while it states for use "directly or indirectly" with any business, it indicates that business is meant to be considered some undertaking intended for profit. The use of this package has no connection to any undertaking for profit and is intended to be released under the license used by Baliance for use by others under the same conditions and restrictions. I am not an expert on the use and interpretation of licenses, but from all accounts, this appears to be fair use. In addition, other definitions seem to state that so long as it is released for free, (without intent to use it as advertisement) it's non-commercial usage.



Package Overview

The Package, being minimalist, is not expected to require any notable amount of code and will be written in Golang.

Incoming Docx

Incoming Docx is expected to conform to a pre-defined structure which begins with a paragraph-normal heading that serves as a Document Type Definition (DTD). If this DTD heading is not found, the document will not be read. Other similar headings may be made in the future, if other templates come into use.

```
$UDW_1A
```

Code Snippet 1: DTD heading

The rest of the document will require the following structure, or it will be ignored by the reader to prevent over-writing work.

```
TITLE
SUBTITLE
HEADING1
----- (Content Delimiter)
CONTENT a
CONTENT a
-----
CONTENT b
-----
HEADING1
CONTENT1
```

Code Snippet 2: Document Definition



Architecture and Components

The structure is straightforward and is detailed in the following sections.

Exported

There should be very few exported items. After proposal moves forward, they will be defined here.

Current Status and Future Development

Currently in review.

