# MD2Word-Pandoc

## 简介

一个使用C#编写的、基于[Pandoc](https://github.com/jgm/pandoc)的markdown转换word工具。其本质是在Pandoc的基础上提供一个GUI界面，方便操作。是初学C#和Winform的作品，还在摸索阶段，有更好的思路、写法或找到了bug可以留issue，我都会看到并且回复的 :)

注意！本项目还在开发当中，并且要求系统安装Pandoc。已基本实现转换和样式保存导出功能，GUI界面还未完成，目前阶段的使用方法见 [现阶段使用方法](#现阶段使用方法)

## 为什么做这个

在接触了markdown之后，越发觉得Word调节样式非常繁琐了。能用markdown写作的环境坚决不会使用Word，但无奈平时的实验报告又不得不上交Word版本，转了一圈也没有找到一个好用、转换效果好的markdown转word工具，也曾尝试使用一些在线文档编写平台进行转换，比如：

* 石墨文档
* Teambition Thought
* 语雀文档
* 为知笔记
* ……

转换的效果并不理想。而Pandoc默认的Word样式在中文状态下不美观，于是萌生了编写此工具的想法。

## 基本目标

* ☒ 由.md文件转换为一个排版样式尚可的Word文档。转换出的Word的字符、段落样式符合中文的书写和显示习惯（如段首缩进、行间距、字体的选择等）
* ☐ 段落样式可供设置的内容：字号、字体、段落间距、颜色、（加粗、斜体、下划线？）
* ☐ 提供可视化界面，样式可由用户方便地设置
* ☒ 拖拽即可导入文件
* ☐ 软件自带有若干个样式预设
* ☐ 可以保存预设，可以导入、导出样式预设

## 更进一步

* ☐ 读取md文件的yaml信息来进行自动样式设置
* ☐ 提供论文排版模式，即在首页之前添加特定的封面、处理参考文献等
* ☐ 根据MD的大纲级别制作目录
* ☐ 制作一个精简版的pandoc依赖，只需要其中的md转换docx的功能，从而简化软件体积

## 现阶段使用方法

注意，要求系统已安装[Pandoc](https://github.com/jgm/pandoc)。测试所用的版本为2.7.2，理论上更高版本也可行，但未测试。

测试阶段，未来应该会完善 :)

1. Clone本项目，安装后文提到的依赖，使用Visual Studio打开并构建运行
2. 将resource文件夹中的default\_preset.xml文件复制到程序运行的preset文件夹下
3. 打开构建好的可执行程序，拖入md文件到窗口中
4. 下拉菜单选择预设
5. 执行转换

可对该xml文件进行修改，以满足自己的样式需求。转换出的docx文件将保存在与源md的同目录下。

## 实现思路

有更好的思路、方法可以留issue，我会看到的

1. 关键：根据用户的样式设置，生成一个可供pandoc转换时使用的临时template.docx——使用Office Open XML SDK操作
   1. 创建一个新的临时的docx
   2. 根据pandoc参照模板，生成若干段落样式
   3. 向临时的docx中添加这些段落样式
   4. 保存并关闭这个docx
2. 调用pandoc。使用以下命令：

* pandoc -o output.docx input.md --reference-doc=template.docx
* 解释：由源md文件input.md参照template.docx中的样式列表生成Word文档output.docx。

1. 使用JSON或XML保存样式预设，用于导入和导出分享、保存样式信息

### 核心步骤

string path = @"C:\Users\59838\Desktop\test.docx";  
  
// 1.创建一个新的临时的docx  
Console.WriteLine("CreateWordprocessingDocument at " + path);  
CreateWordprocessingDocument(path);  
  
// 2.打开刚刚创建的新文档  
Console.WriteLine("GetStyleDefinitionsPart...");  
WordprocessingDocument wordDocument =  
 WordprocessingDocument.Open(path, true);  
  
// 3.获取样式定义部分实例  
StyleDefinitionsPart part = GetStyleDefinitionsPart(wordDocument);  
  
// 4.根据要求生成样式Style实例  
Console.WriteLine("GenerateStyle...");  
Style style = GenerateStyle();  
  
// 5.将4中生成的实例，通过3获取的实例，添加到样式管理（文档的样式库）中  
Console.WriteLine("AddParagraphStyle...");  
AddParagraphStyle(part, style);  
  
// 6.将更改保存并关闭文件（重要）  
wordDocument.Close();

## 有用的资料

### Pandoc User’s Guide 节选

**Pandoc转换时参考样式的依据是styleID属性，而不是style的名字**

[完整的 Pandoc User’s Guide](Pandoc%20User's%20Guide.html)

--reference-doc=*FILE*

Use the specified file as a style reference in producing a docx or ODT file.

**Docx**

For best results, the reference docx should be a modified version of a docx file produced using pandoc. The contents of the reference docx are ignored, but its stylesheets and document properties (including margins, page size, header, and footer) are used in the new docx. If no reference docx is specified on the command line, pandoc will look for a file reference.docx in the user data directory (see --data-dir). If this is not found either, sensible defaults will be used.

To produce a custom reference.docx, first get a copy of the default reference.docx: pandoc -o custom-reference.docx --print-default-data-file reference.docx. Then open custom-reference.docx in Word, modify the styles as you wish, and save the file. For best results, do not make changes to this file other than modifying the styles used by pandoc:

#### 段落样式

|  |  |  |
| --- | --- | --- |
| 样式名 | 描述 | 必须 |
| Normal | 默认文本。在有序列表中会使用该样式 | 是 |
| Body Text | 正文文本。在正文段落中应用 | 是 |
| First Paragraph | 段落首段，一般和Body Text一样即可，在段落的第一行应用 | 是 |
| Compact | 在无序列表中应用 | 是 |
| Heading 1 | 一级标题 | 是 |
| Heading 2 | 二级标题 | 是 |
| Heading 3 | 三级标题 | 是 |
| Heading 4 | 四级标题 | 否 |
| Block Text | 引用块文字 | 是 |

更多：

* Heading 5
* Heading 6
* Heading 7
* Heading 8
* Heading 9
* Footnote Text“脚注文字，可选
* Definition Term：未知
* Definition：未知
* Caption：未知
* Table Caption：表的的注释，可选
* Image Caption：图片的注释，可选
* Figure：未知，可能是图表
* Captioned Figure：未知，可能是带标题的图表
* TOC Heading：目录标题
* Title：标题，可选
* Subtitle：副标题，可选
* Author：作者，可选
* Date：日期，可选
* Abstract：未知，可选
* Bibliography：参考文献，可选，具体使用未知

#### 字符样式

|  |  |  |
| --- | --- | --- |
| 样式名称 | 描述 | 必须 |
| Default Paragraph Font | 默认段落字体 | 是 |
| Body Text Char | 正文字体 | 是 |
| Verbatim Char | 等宽字体（在代码块中用到） | 是 |
| Footnote Reference | 脚注引用字体样式 | 否 |
| Hyperlink | 超链接字符样式 | 是 |

#### 表格样式

* Table：未知设置方法

### 构建所需依赖

* 必需：WindowsBase（.Net Framework自带）
* 必需：Open XML SDK 2.5 for Microsoft Office
  + 下载：https://www.microsoft.com/en-us/download/details.aspx?id=30425
  + 在VS中引入：https://blog.csdn.net/CoingSun/article/details/105754924
* 可选：OpenXMLSDKTool，用于查看Word文档中的各项内容

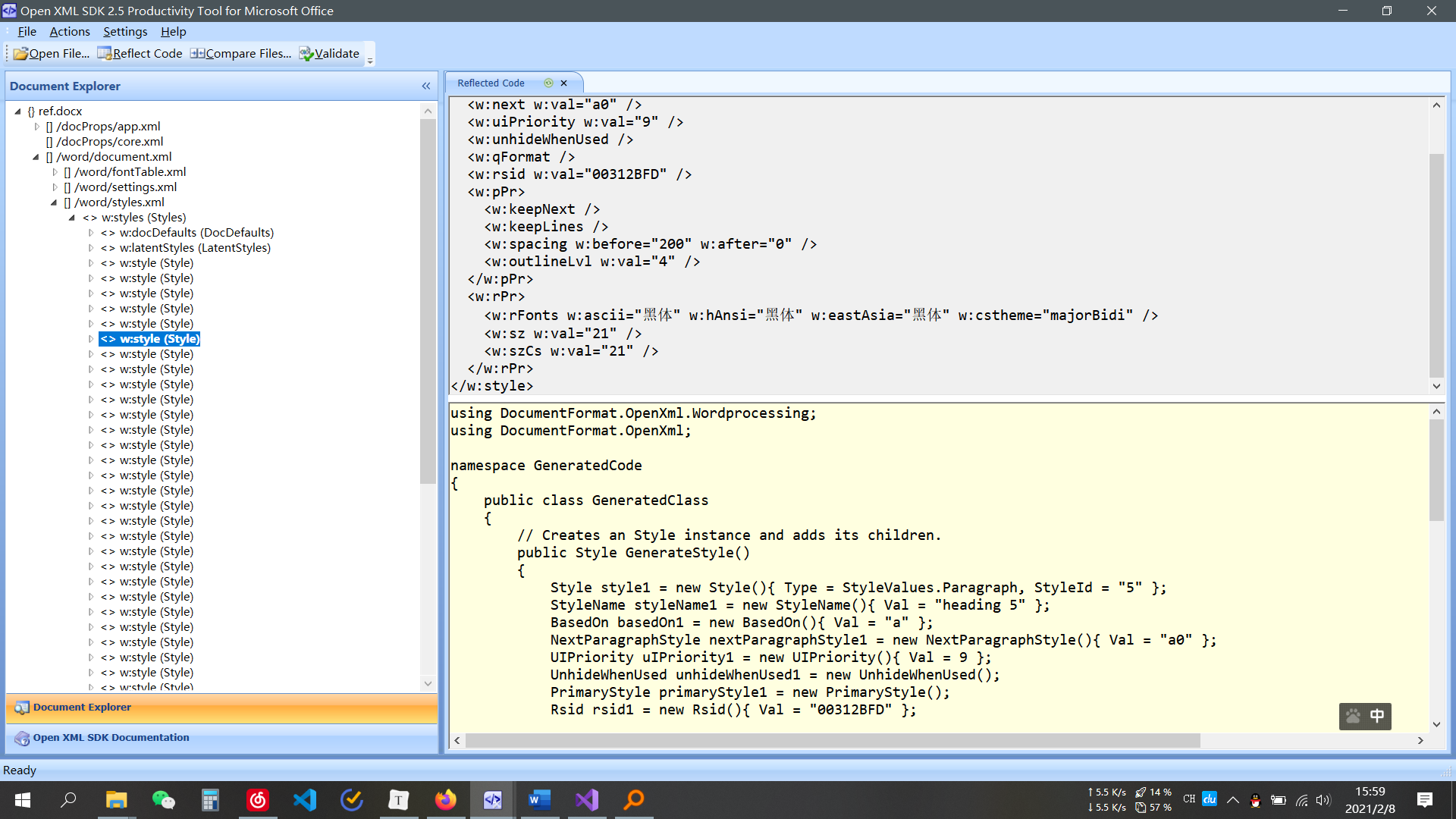


image-20210208155954767

### .NET 序列化、反序列化与XML

// 1. 首先要创建或者得到一个数据对象  
MyObject obj = new MyObject();  
  
// 2. 用序列化的方法生成XML  
string xml = XmlHelper.XmlSerialize(obj, Encoding.UTF8);  
  
// 3. 从XML读取数据并生成对象  
MyObject obj2 = XmlHelper.XmlDeserialize<obj>(xml, Encoding.UTF8);

可以直接序列化数组，但不能直接序列化ArrayList

### Microsoft 官方文档

* [主要] Office Open XML SDK 字处理 概述 https://docs.microsoft.com/zh-cn/office/open-xml/word-processing?view=openxml-2.8.1
* ~~为 Office 创建 VSTO 外接程序 / Word 解决方案 / Word 对象模型概述 / 处理文档~~ https://docs.microsoft.com/zh-cn/visualstudio/vsto/how-to-programmatically-open-existing-documents?view=vs-2019

### 字体大小转换

Word中设置字号大小n磅，那么在代码层面的值是2n 。比如在Word中字号设置为14磅，那么代码中的值取14×2=28

|  |  |
| --- | --- |
| 字号 | 磅 |
| 初号 | 42 |
| 小初 | 36 |
| 一号 | 26 |
| 小一 | 24 |
| 二号 | 22 |
| 小二 | 18 |
| 三号 | 16 |
| 小三 | 15 |
| 四号 | 14 |
| 小四 | 12 |
| 五号 | 10.5 |
| 小五 | 9 |