Добрый день, Третья и улучшенная версия zcl\_docx

Для начала нам понадобятся:

1. 7zip <http://7zip.org/>
2. Notepad++ <https://notepad-plus-plus.org/download/v7.5.4.html>
3. Notepad ++ xml plugin <https://sourceforge.net/projects/npp-plugins/files/XML%20Tools/Xml%20Tools%202.4.9%20Unicode/>

В скачанном архиве плагина будет вполне понятное readme

Мой инструмент позволяет делать очень сложные документы, настолько сложные, что вам такие в общем случае не нужны. Сейчас я расскажу необходимый минимум для построения сложного документа, применив его рекурсивно вы можете построить настолько сложный документ, насколько позволяет ваша фантазия или требования тз.

Хотелось бы обойтись без вот этих рекурсий, но требования тз безжалостны.

Мы начнем с простого варианта и будем его усложнять.

Минимальная программа выглядит так

\*&---------------------------------------------------------------------\*  
\*& Report Z\_TEST\_DOCX1  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
report z\_test\_docx1.  
  
  
\* тут класс  
include zcl\_docx\_class.  
  
START-OF-SELECTION.  
  
data  
      : lo\_docx type ref to lcl\_docx  
      , lr\_data type ref to lcl\_recursive\_data  
      .  
  
  
create object lr\_data.  
  
lr\_data->append\_key\_value(  iv\_key = 'NAME' iv\_value = sy-uname ).  
lr\_data->append\_key\_value(  iv\_key = 'DATE' iv\_value = |{ sy-datum date = environment }| ).  
lr\_data->append\_key\_value(  iv\_key = 'TIME' iv\_value = |{ sy-uzeit time = environment }| ).  
  
  
create object lo\_docx .  
  
lo\_docx->load\_smw0( 'Z\_TEST\_DOCX1' ).  
  
  
lo\_docx->map\_data( exporting ir\_data = lr\_data ).  
  
  
call method lo\_docx->save  
  exporting  
    on\_desktop   = 'X'  
    iv\_folder    = 'report'  
    iv\_file\_name = 'Z\_TEST\_DOCX1.docx'  
    no\_execute   = ''.

Грузим шаблон из smw0, сохраняем на рабочем столе, в папке report, под именем Z\_TEST\_DOCX1.docx и открываем офисом.

Если не на рабочем столе то во временной папке, если папка пуста, то прямо там, если       no\_execute   = 'X' , то только сохранить, не открывать.

В шаблоне могут быть переменные.

Небольшое соглашение по именам:

Переменные в шаблоне берутся в фигурные скобки и пишутся капсом, например {DATE}.

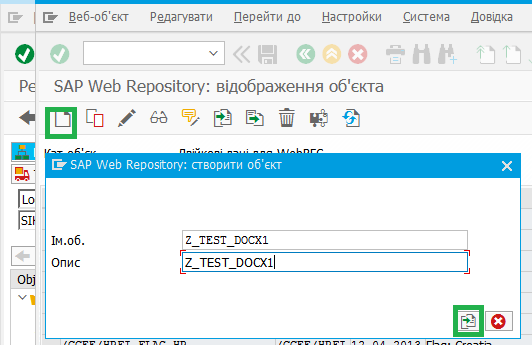
В программе они пишутся в любом регистре и без квадратных скобок.

В шаблоне {DATE}, в программе

lr\_data->append\_key\_value(  iv\_key = 'date' iv\_value = |{ sy-datum DATE = ENVIRONMENT }| ).

Загрузка шаблона в smw0 .

Транзакция smw0, двоичные данные



Следующий по сложности элемент – таблицы.

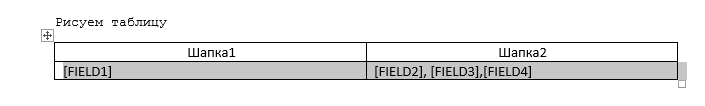
Рисуем таблицу

|  |  |
| --- | --- |
| Шапка1 | Шапка2 |
| {FIELD1} | {FIELD2}, {FIELD3},{FIELD4} |

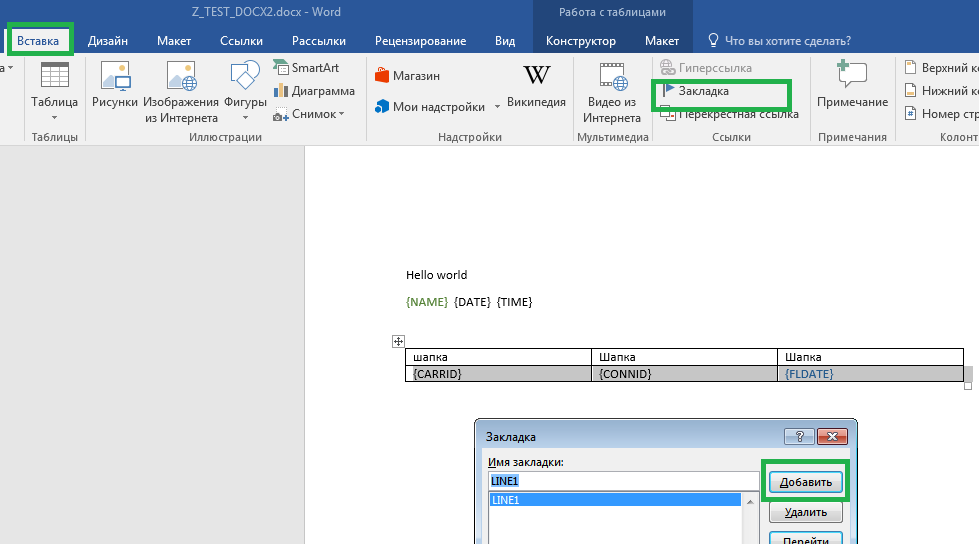
Шапка и строка таблицы,

Имена переменных, как имена полей в структуре, капсом и в фигурных скобках, можно по одному в ячейку, можно по несколько в ячейку.

Выделяем строку таблицы



Вставка->Закладка-> имя закладки капсом -> приисвоить



В программе это будет выглядеть так

lr\_data->append\_key\_table( iv\_key = 'LINE1' iv\_table = lt\_total ).

Таблица должна содержать поля, которые есть в шаблоне, если их больше, ничего страшного, если их меньше, то вы увидите часть шаблона в итоговом документе.

Это пример Z\_TEST\_DOCX2.

С таблицами приходится делать дополнительные манипуляции с шаблоном.

Для распаковки запаковки шаблона есть файлик 123.bat находит последний документ в папке и распаковывает его, если он был распакован то запаковывает обратно. Во избежание кладите шаблон и файлик в отдельную папку.

@echo off

FOR %%i IN (\*.docx) DO set foo=%%i

Set 7zip="C:\Program Files\7-Zip\7z.exe"

FOR /f "tokens=1 delims=." %%a IN ("%foo%") do set folder=%%a

IF EXIST %folder%\NUL goto compress ELSE goto extract

:extract

"C:\Program Files\7-Zip\7z.exe" x %foo% -o\* -aoa

goto end

:compress

echo "Compress"

"C:\Program Files\7-Zip\7z.exe" a %foo% .\%folder%\\*

RMDIR /S /Q %folder%

goto end

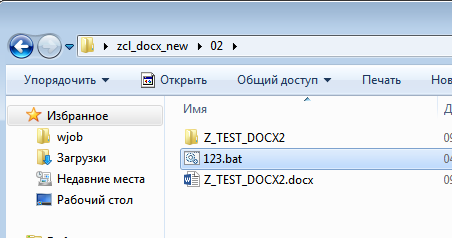
:end

echo "end"

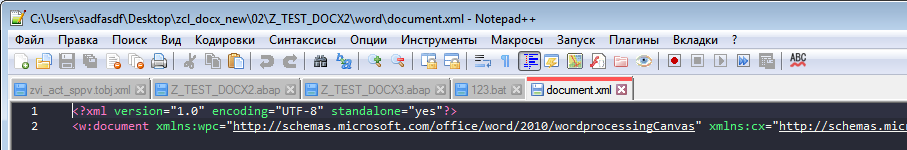
для его работы необходим установленный 7zip по пути "C:\Program Files\7-Zip\7z.exe"

либо подправить пути в файле.

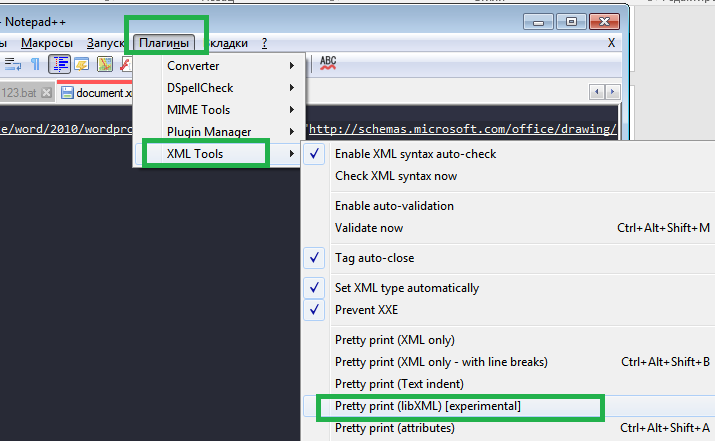
Распаковываем

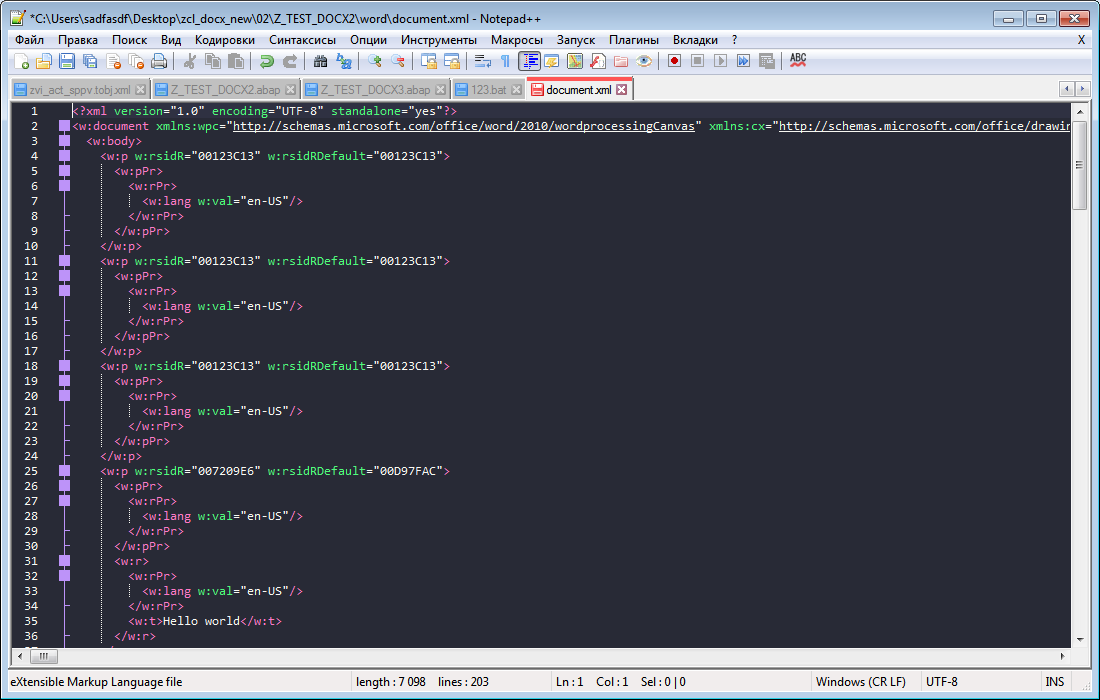


Идем внутрь, папка ворд, открываем document.xml нотепадом++

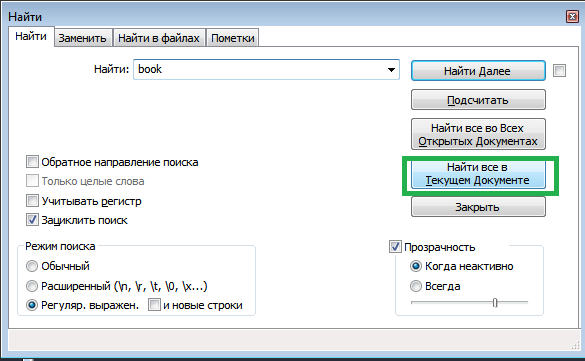


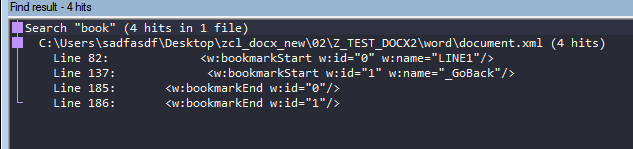
Приводим в читаемый вид



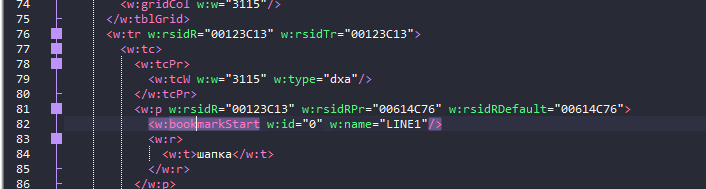


Ищем закладки

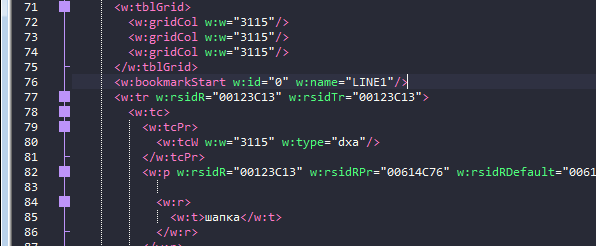




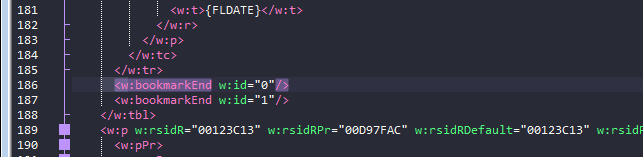
Открывающая в 82 строке, закрывающая в 185, у них будут одинаковые id



82 нужно перенести после 75, чтоб наши закладки полностью охватывали всю строку

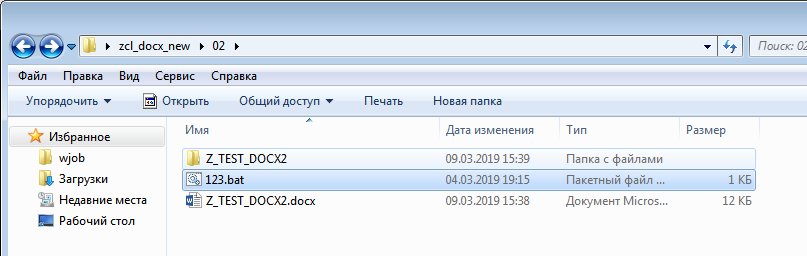


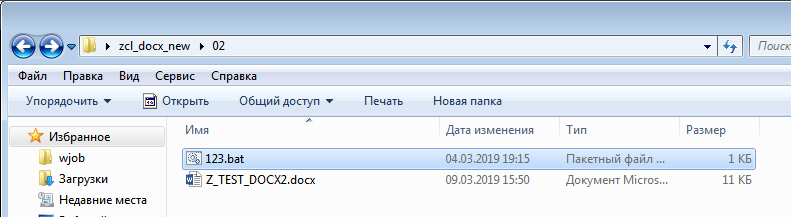
Закрывающая на месте,



Она стоит после закрывающего тега.

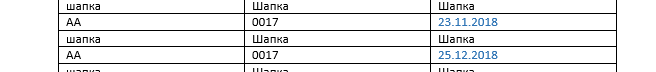
Сохраняем документ, поднимаеся выше, еще раз запускаем 123.bat и он запаковывает обратно документ





Грузим его в smw0.

Если откроете вордом и пересохраните, то процедуру придется повторить.



Как оказалось захватили шапку.

Открывающую ссылку нужно перенести на 128 строку. Запаковываем, грузим проверяем.



Теперь нормально.

Шаблон может иметь бесконечное количество значений key\_value, и бесконечное количество таблиц, которые key\_table.

Все было бы просто, но нам нужны таблицы таблиц, с уровнем вложенности около 5-7 или сколько требует тз.

Мы переходим к задаче со звездочкой, которая называется рекурсия.

Пример Z\_TEST\_DOCX3

И так, наш документ может содержать key\_value, key\_table.

Если некоторою конструкцию нужно повторить несколько раз, то мы ее обзываем документом.

Любая конструкция сложнее key\_value или key\_table – документ.

Документ содержит в себе неограниченное количество key\_value, неограниченное количество key\_table, а также неограниченное количество документов, содержащих в себе key\_value, key\_table и документов и так далее, ну вы поняли, пока здравый смысл не покинет вас.

Key\_value, сколько нужно

Key\_table, сколько нужно

Итак документ содержит в себе неограниченное количество key\_value, key\_table, и других документов, содержащих в себе…

Вот пример с ложного документа

Привет! Я ведомость№

Тут у меня всякий красивый текст, в который должны попасть поляI00142 23.02.2018 22:49:51

Табличка:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CARRID | CLASS | FORCURAM | | FORCURKEY | LOCCURAM | LOCCURKEY | ORDER\_DATE |
| LH | C | 899.00 | | DEM | 899.00 |  | 26.02.1995 |
| LH | F | 890.00 | | DEM | 890.00 |  | 27.02.1995 |
| LH | Y | 850.00 | | DEM | 850.00 |  | 27.02.1995 |
| LH | Y | 1450.00 | | DEM | 1450.00 |  | 01.11.1995 |
| Промежуточный | | 12678.00 | | | 7687.00 | | |
| SQ | F | 849.00 | | DEM | 849.00 |  | 21.02.1995 |
| SQ | C | 835.00 | | DEM | 835.00 |  | 22.02.1995 |
| Промежуточный | | 1684.00 | | | 1684.00 | | |
| Общий итог по таблице | | | 14362.00 | | 9371.00 | | |

Опять какой-то бла-бла-бла текст

Блок подписей формируется динамически, может вообще не содержатьданныха может состоять из 1-10 строк вида:

|  |  |  |
| --- | --- | --- |
|  |  | SebastianFuidl |
|  |  | SebastianFuidl |
|  |  | John Doe |
|  |  | Developer |
|  |  |  |

Должность Подпись Фамилия, инициалы

Потом еще какой-то текст

Реквизиты сторон (статический блок без переноса на следующею страницу):

Продавец: Покупатель:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Это итоговый документ котрый получился в нашей системе

В нем мы видем

key\_value, поляI00142 23.02.2018 22:49:51

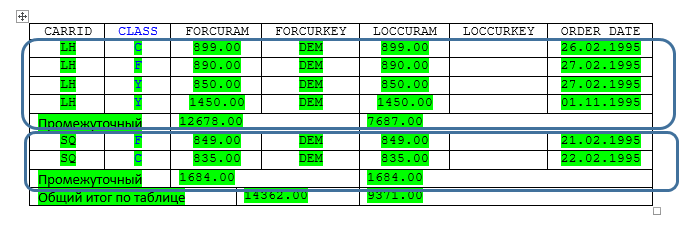
2 key\_table

|  |  |  |
| --- | --- | --- |
|  |  | John Doe |



Одну строку таблицы можно было оформить как набор key\_value, но key\_table из одной строки проще.

И два документа



Вкаждом из котром есть

Key\_table из нескольких строк, и key\_table из одной строки

Схлопываем повтряющиеся куски шаблона, и наш шаблон превращается вот в это

Привет! Я ведомость №

Тут у меня всякий красивый текст, в который должны попасть поля {NAME} {DATE} {TIME}

Табличка:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CARRID | CLASS | FORCURAM | | FORCURKEY | LOCCURAM | LOCCURKEY | ORDER\_DATE |
| {CARRID} | {CLASS} | {FORCURAM}} | | {FORCURKEY} | {LOCCURAM} | {LOCCURKEY} | {ORDER\_DATE} |
| Промежуточный | | {FORCURAM} | | | {LOCCURAM} | | |
| Общий итог по таблице | | | {FORCURAM} | | {LOCCURAM} | | |

Опять какой-то бла-бла-бла текст

Блок подписей формируется динамически, может вообще не содержать данных а может состоять из 1-10 строк вида:

|  |  |  |
| --- | --- | --- |
|  |  | {NAME\_FIRST} {NAME\_LAST} |

Должность Подпись Фамилия, инициалы

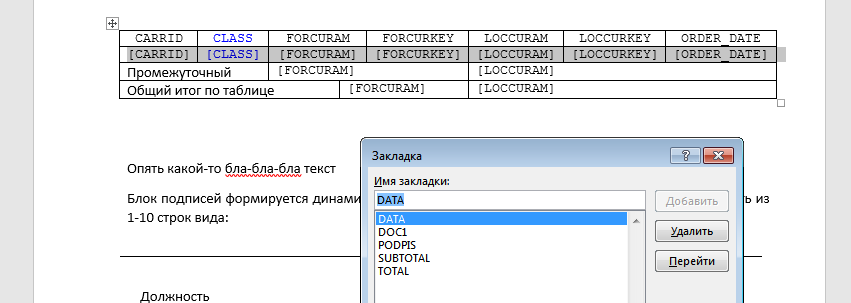
Потом еще какой-то текст

Реквизиты сторон (статический блок без переноса на следующею страницу):

Продавец: Покупатель:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

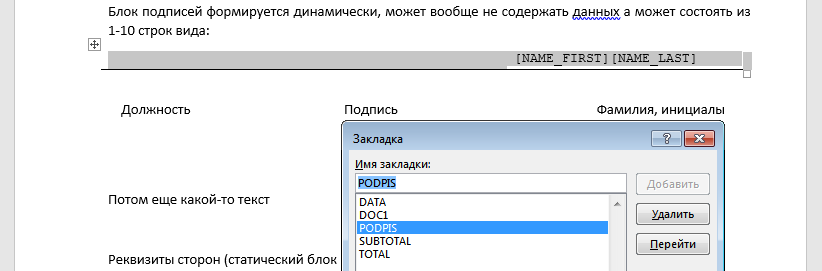
Выделяем первую строку и называем ее DATA



Вторая строка SUBTOTAL

Третья строка TOTAL

Последняя строка, подпись, она же sign, в суржеке PODPIS



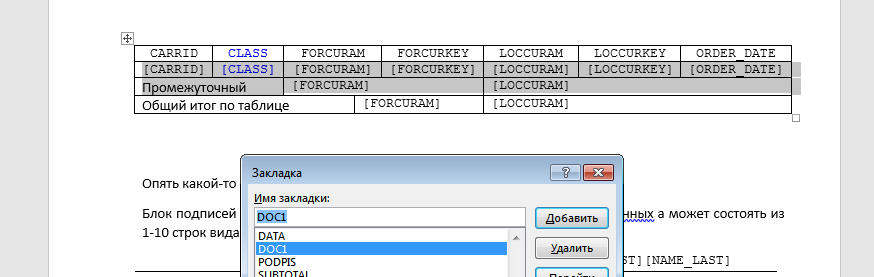
Ее можно сделать не таблицей, а фрагментом текста, но по факту таблицей здесь можно оформить все, а програмно

lr\_data->append\_key\_table( iv\_key = 'podpis' iv\_table = lt\_adrp ).

Прощще, чем

lr\_data->append\_key\_value(  iv\_key = 'name' iv\_value = sy-uname ).  
  lr\_data->append\_key\_value(  iv\_key = 'date' iv\_value = |{ sy-datum DATE = ENVIRONMENT }| ).  
  lr\_data->append\_key\_value(  iv\_key = 'time' iv\_value = |{ sy-uzeit TIME = ENVIRONMENT }| ).

Выделяем 2 сторки которые DATA и Subtotal и называем их DOC1



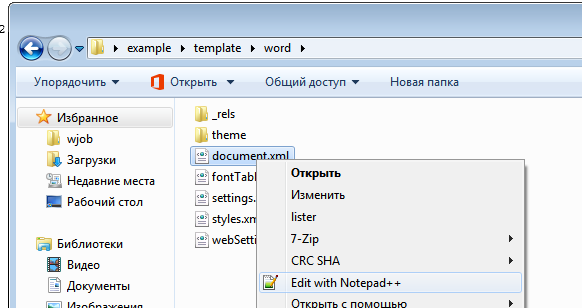
Это будет наш вложенный документ, в котором будет таблица и сторка

Сохраняем наш шаблон т переходим к следующему шагу.

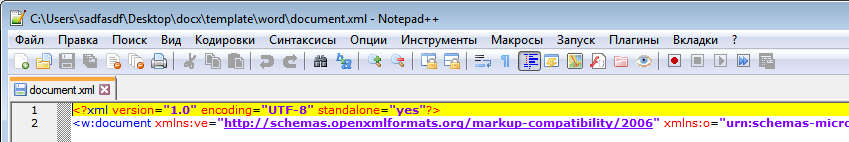
Распаковываем наш шаблон

В самом начале мы установили notepad++ и xml плагин к нему.

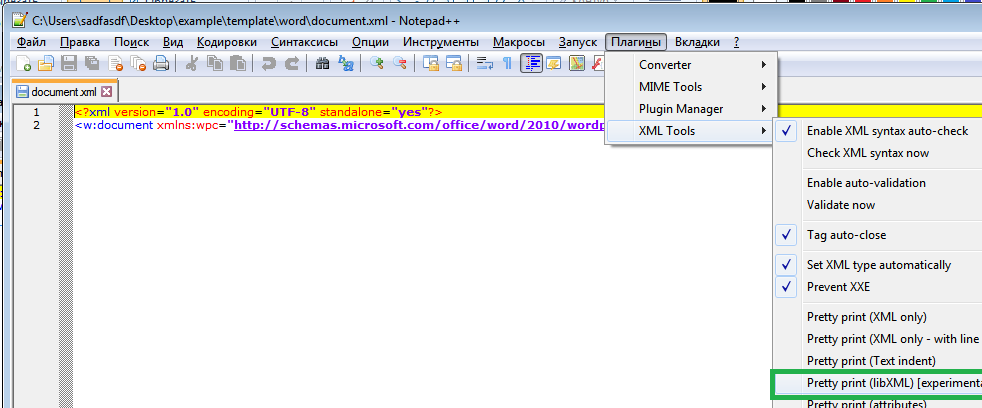
Открываем с помощью notepad++ document.xml который в папке word

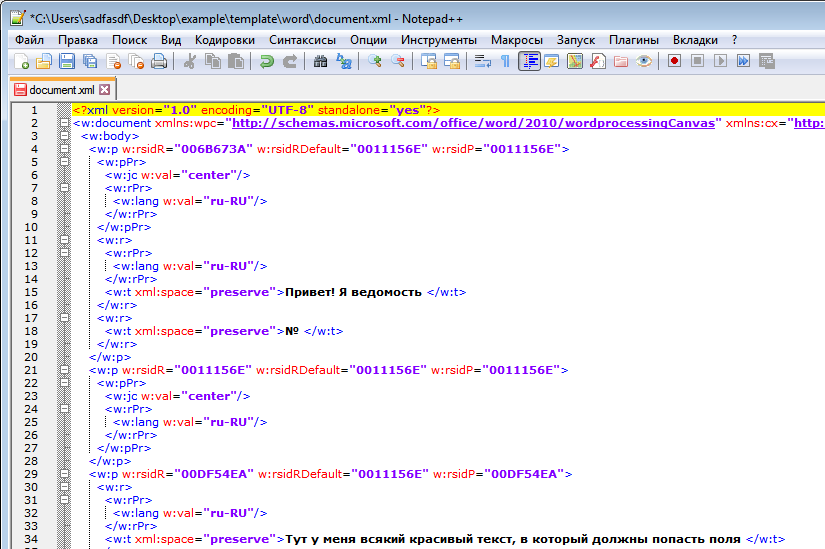


И видим примерно следующее



Переходим в плагины -> XML Tools -> pretty print(libXML) {experimental} и видим что-то более менее читабельное





Еще один важный момент это закладки которые мы сделали у ворда свое незензурное представление о том как раcставлять теги меток.

Искать их нужно по имени bookmarkStart



У открывающего тега есть id и name

У закрывающего только id

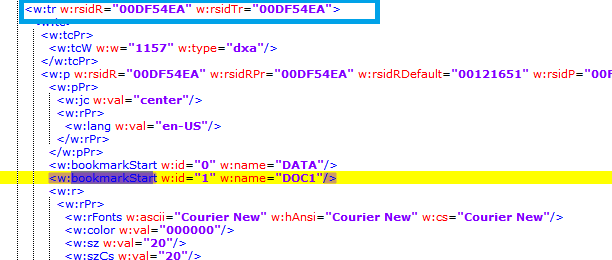
bookmarkEnd



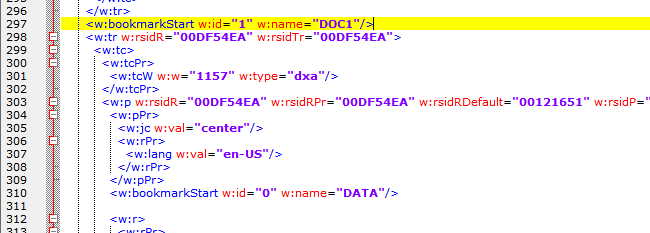
По id мы определяем какой закрывающий относится к какому закрывающему.

Наша задача переставить эти теги так, чтоб между ними было одинаковое количество открывающих и закрывающих тегов, и в нем содержался необходимый валидный нам фрагмент

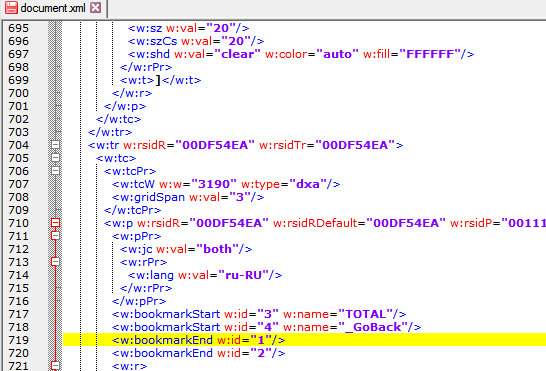
Как мы помним наш документ содержит 2 строчки таблицы, в документе мы видем следуеещее



Тег <w:tr> говорит о начале строки таблицы, выдеоен синим, а наша закладка лежит внутри строки, нам нужно перенести ее до открытия тега <w:tr>



C закрывающим тегом все тоже самое



Он лежит в следующей строке таблицы, которая относится к итогам, его нужно перенести после закрывающего тега </w:tr>



Вот такие сложности только с вложенными документами, если у вас нет вложенных документов то вам проще.

Сохраняем результат и собираем документ обратно

грузим это в smw0.

Теперь переходим к програмной части

Кроме всего прочего необходимого для получения данных нам нужны 2 переменных

        : lo\_docx TYPE REF TO lcl\_docx  
        , lr\_data TYPE REF TO lcl\_recursive\_data

Первая это про наш документ, получение шаблона из smw0, применения данных к шаблону и сохранение,

Вторая это наши данные.

  CREATE OBJECT lr\_data.  
  
  lr\_data->append\_key\_value(  iv\_key = 'name' iv\_value = sy-uname ).  
  lr\_data->append\_key\_value(  iv\_key = 'date' iv\_value = |{ sy-datum DATE = ENVIRONMENT }| ).  
  lr\_data->append\_key\_value(  iv\_key = 'time' iv\_value = |{ sy-uzeit TIME = ENVIRONMENT }| ).

Создаем объект и добавляем в него объекты key\_value.

Вот наши документы

    lr\_tmp\_data = lr\_data->create\_document( 'doc1' ).

В текущий документ добавляем новый документ который относится к метке DOC1  
  
lr\_tmp\_data->append\_key\_table( iv\_key = 'data' iv\_table = lt\_tmp ).  
lr\_tmp\_data->append\_key\_table( iv\_key = 'subtotal' iv\_table = lt\_sub\_total\_tmp ).

В толко что созданый документ добавляем 2 таблицы . При желании в этом документе можно создать другие документы, в них добавить key\_value, key\_table и других документов.

В конце к основному документу добавляем итоги и подписи

  lr\_data->append\_key\_table( iv\_key = 'total' iv\_table = lt\_total ).  
  lr\_data->append\_key\_table( iv\_key = 'podpis' iv\_table = lt\_adrp ).

И финальный штрих

  CREATE OBJECT lo\_docx .  
  
  lo\_docx->load\_smw0( 'Z\_TEST\_DOCX3' ).  
  
  
  lo\_docx->map\_data( EXPORTING ir\_data = lr\_data ).  
  
  
  CALL METHOD lo\_docx->save  
    EXPORTING  
      on\_desktop   = 'X'  
      iv\_folder    = 'report'  
      iv\_file\_name = ' Z\_TEST\_DOCX3.docx'  
      no\_execute   = ''.

Грузим шаблон, применяем трансформацию, сохраняем результат, все счастливы, пьют шампанское, летят на мальдивы.

Естественно там много мест где можно выстрелить себе в ногу, как настреляете будем исправлять.