Tool to create Microsoft Word docx from abap.

Installation

Install package via ABAPGIT <https://docs.abapgit.org/guide-install.html>

1

2

3

4

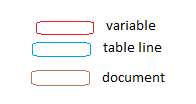
5

Where there are mistakes, just click Pull\_zip twice.

For example, the following document should be created:

6

Initially, define variables and repeated parts



Variable – simple text.

Table line – contains table row that may consist of several or zero variables and text.

Document – contain several or zero texts, variables, table row.

We have something like this

7

Let reduce our document

8

Structure of data for our document shoudt be like

├── name

├── date

├── time

├── document1

│   ├── data\_row

│   │ ├── CARRID

│   │ ├── CLASS

│   │ ├── FORCURAM

│   │ ├── LOCCURAM

│   │ ├── LOCCURKEY

│   │ └── ORDER\_DATE

│   │ .

│   │ .

│   ├── data\_row

│   │ ├── CARRID

│   │ ├── CLASS

│   │ ├── FORCURAM

│   │ ├── LOCCURAM

│   │ ├── LOCCURKEY

│   │ └── ORDER\_DATE

│   └── subtotal

│   ├── FORCURKEY

│   └── LOCCURAM

│   .

│   .

├── document1

│   ├── data\_row

│   │ ├── CARRID

│   │ ├── CLASS

│   │ ├── FORCURAM

│   │ ├── LOCCURAM

│   │ ├── LOCCURKEY

│   │ └── ORDER\_DATE

│   │ .

│   │ .

│   ├── data\_row

│   │ ├── CARRID

│   │ ├── CLASS

│   │ ├── FORCURAM

│   │ ├── LOCCURAM

│   │ ├── LOCCURKEY

│   │ └── ORDER\_DATE

│   └── subtotal

│   ├── FORCURKEY

│   └── LOCCURAM

├── total

│   ├── FORCURKEY

│   └── LOCCURAM

├── sign

│   ├── NAME\_FIRST

│   └── NAME\_LAST

│ .

│ .

└── sign

    ├── NAME\_FIRST

    └── NAME\_LAST

Or simple

├── name (value)

├── date (value)

├── time (value)

├── document1 (document repeated)

│   ├── data\_row (table repeated)

│   │ ├── CARRID

│   │ ├── CLASS

│   │ ├── FORCURAM

│   │ ├── LOCCURAM

│   │ ├── LOCCURKEY

│   │ └── ORDER\_DATE

│   └── subtotal (table of 1 row)

│   ├── FORCURKEY

│   └── LOCCURAM

├── total (table of 1 row)

│   ├── FORCURKEY

│   └── LOCCURAM

└── sign (table repeated)

    ├── NAME\_FIRST

    └── NAME\_LAST

Let sign variable placeholder.

At first toggle developer toolbar. <https://www.google.com/search?q=microsoft+office+16+toggle+developer+toolbar>

Select

9

Make tag

10

Click properties

11

Enter tag name

12

Tag name is case insensitive; all the way, it will be converted to uppercase

Value can have any tag name, value inside table row must have name of field of row structure

For example, tag name for sign row

13

Name all variable placeholder

In design time it must look like this

14

At second we mark placeholder for table row(data\_row, subtotal, total, sign)

Place mouse cursor to the left of row

15

Click

16

17

Properties, data\_row

18

Subtotal

19

Total

20

Sign

21

Then template in design mode must look like

22

Now, we move on to the task with an asterisk. In 99% cases you do not need this. I just show opportunity how to make more complex document.

If you want, you can make infinite depth of your document.

We need join 2 rows in one placeholder. Unfortunately, Microsoft Office cannot make placeholder for 2 rows. It can make placeholder for one row or for a whole table.

There we have 2 way:

1. Cheap and wrong
2. More complex and True

First way:

Copy your table 3 times

23

Cut unnecessary part from each table

24

Make placeholder for whole second table

25

Remove spaces between tables

In my case, it look like

26

I don’t like this, maybe, I cannot work with tables.

Second way I prefer:

Select two rows, make placeholder. Office create placeholder for first row. It is ok.

27

28

Now, save the template, close office.

Rename template.docx to template.zip

29

30

Unpack to subfolder

31

Navigate inside, then in subfolder ‘word’

32

We need notepad++ <https://notepad-plus-plus.org/downloads/>

Open document.xml with notepad++

33

Navigate plugin->plugins admin..

34

Search “Xml tools”

35

Install

Now you can pretty print xml document

36

37

Find placeholder we created. In our case it document2

38

Ctrl+f

39

We can see our placeholder

<w:tag w:val="document2"/>

40

It starts in line 1293 with tag <w:sdt>

Collapse tree to see where it end

41

It end near line 1621

42

Now collapse next placeholder to see where it ends

43

It ends near line 1757

44

Expand all.

Go to line 1621

Cut 2 lines 1619, 1620

45

</w:sdtContent>

</w:sdt>

46

Navigate to line 1757

47

Insert 2 lines before line 1757

48

Yellow – inserted lines.

Save, close.

Navigate 1 level up

49

50

Select all file at this level, add to zip archive

51

52

Rename template\_docx.zip to template\_docx.docx

Now we can see placeholder hold 2 rows

53

Go to transaction smw0

Select Binary data, enter

54

Object name Z\_TEST\_DOCX2

55

Create

56

57

58

Now let’s create test program.

You can read this program with comments or see program Z\_TEST\_DOCX2 that already exists in the package.

Template Z\_TEST\_DOCX2 also exists in the package

*\*&---------------------------------------------------------------------\**  
*\*& Report Z\_TEST\_DOCX\_2*  
*\*&---------------------------------------------------------------------\**  
*\*--------------------------------------------------------------------\**  
*\*  Autor: Anton.Sikidin@gmail.com*  
*\*--------------------------------------------------------------------\**  
  
REPORT z\_test\_docx\_2.  
  
START-OF-SELECTION.  
  
  
  TYPES  
*"  structure to hold our data*  
  : BEGIN OF t\_data  
  ,   carrid  TYPE s\_carr\_id  
  ,   class  TYPE s\_class  
  ,   forcuram  TYPE s\_f\_cur\_pr  
  ,   forcurkey  TYPE s\_curr  
  ,   loccuram  TYPE s\_l\_cur\_pr  
  ,   loccurkey  TYPE s\_currcode  
  ,   order\_date  TYPE s\_bdate  
  , END OF t\_data  
  .  
  
  
  
  DATA  
        : lt\_carrid TYPE TABLE OF s\_carr\_id  
        , lt\_total TYPE TABLE OF t\_data  
        , lt\_sub\_total TYPE TABLE OF t\_data  
        , lt\_sub\_total\_tmp TYPE TABLE OF t\_data  
        , lt\_data TYPE TABLE OF t\_data  
        , lt\_tmp TYPE TABLE OF t\_data  
  
        , lt\_adrp TYPE TABLE OF adrp  
  
        .  
  
  *" select data for "sign" placeholder*  
  SELECT \* INTO TABLE lt\_adrp FROM adrp UP TO 5 ROWS.  
  
*\*  REFRESH lt\_adrp.*  
*\**  
*\**  
*\*  APPEND INITIAL LINE TO lt\_adrp ASSIGNING FIELD-SYMBOL(<fs\_adrp>).*  
*\*  <fs\_adrp>-name\_first = 'Renee'.*  
*\*  <fs\_adrp>-name\_last = 'Villegas'.*  
*\*  APPEND INITIAL LINE TO lt\_adrp ASSIGNING <fs\_adrp>.*  
*\**  
*\*  <fs\_adrp>-name\_first = 'Meerab'.*  
*\*  <fs\_adrp>-name\_last = 'Finnegan'.*  
*\*  APPEND INITIAL LINE TO lt\_adrp ASSIGNING <fs\_adrp>.*  
*\**  
*\*  <fs\_adrp>-name\_first = 'Jozef'.*  
*\*  <fs\_adrp>-name\_last = 'Beil'.*  
*\*  APPEND INITIAL LINE TO lt\_adrp ASSIGNING <fs\_adrp>.*  
*\**  
*\*  <fs\_adrp>-name\_first = 'Leonard'.*  
*\*  <fs\_adrp>-name\_last = 'Yates'.*  
*\*  APPEND INITIAL LINE TO lt\_adrp ASSIGNING <fs\_adrp>.*  
*\**  
*\*  <fs\_adrp>-name\_first = 'Kyron'.*  
*\*  <fs\_adrp>-name\_last = 'Stevens'.*  
  
  
  
  
  
  *" select data to display in main table,*  
  *" may be it may create in easy and in proper way,*  
  *" but now it just data for proof of concept*  
  
  SELECT \*  
    INTO CORRESPONDING FIELDS OF TABLE lt\_tmp  
    FROM sbook  
    .  
  
  lt\_data = lt\_tmp. *"make backup, because our data corrupted, while calculate total and subtotal*  
  
  LOOP AT lt\_tmp ASSIGNING FIELD-SYMBOL(<fs\_tmp>).  
    COLLECT <fs\_tmp>-carrid INTO lt\_carrid.  *" every document1 or document2 hold data by single carrid*  
  
    CLEAR  
    : <fs\_tmp>-class  
    , <fs\_tmp>-forcurkey  
    , <fs\_tmp>-loccurkey  
    , <fs\_tmp>-order\_date  
    .  
  
    COLLECT <fs\_tmp> INTO lt\_sub\_total . *"calculate  subtotal*  
  
    CLEAR  
    : <fs\_tmp>-carrid  
    .  
  
    COLLECT <fs\_tmp> INTO lt\_total. *" calculate total.*  
  
  ENDLOOP.  
  
  
  lt\_tmp = lt\_data.  
  
  REFRESH lt\_data.  
  
  LOOP AT lt\_sub\_total ASSIGNING FIELD-SYMBOL(<fs\_sub\_total>).  
  
    DATA  
          : lv\_i TYPE i  
          .  
    CLEAR lv\_i.  
  
    LOOP AT lt\_tmp ASSIGNING <fs\_tmp> WHERE carrid = <fs\_sub\_total>-carrid.  
      ADD 1 TO lv\_i.  
  
      IF lv\_i > 4 . *" four row for proof of concept is enought*  
        EXIT.  
      ENDIF.  
  
      APPEND <fs\_tmp> TO lt\_data.  
  
    ENDLOOP.  
  
  ENDLOOP.  
  
  
  *" craete main wariable to hold our data*  
  DATA(lr\_data) = zcl\_docx2=>create\_data( ).  
  
  
  *" fill place holder for simple variable, iv\_key case insensitive*  
  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 }| ).  
  
  *" fill place holder for total ( table of 1 row)*  
  lr\_data->append\_key\_table( iv\_key = 'total' iv\_table = lt\_total ).  
  
  *"fill  placeholder for sign, iv\_key always case insensitive*  
  lr\_data->append\_key\_table( iv\_key = 'sign' iv\_table = lt\_adrp ).  
  
  
  *" fill our main table*  
  
  LOOP AT lt\_sub\_total ASSIGNING <fs\_sub\_total>.  
  
    REFRESH  
    : lt\_tmp  
    , lt\_sub\_total\_tmp  
    .  
  
    APPEND <fs\_sub\_total> TO lt\_sub\_total\_tmp. *" subtotal table of 1 row*  
    LOOP AT lt\_data ASSIGNING FIELD-SYMBOL(<fs\_data>) WHERE carrid = <fs\_sub\_total>-carrid.  
      APPEND <fs\_data> TO lt\_tmp.  *" table for placeholder "data\_row"*  
    ENDLOOP.  
  
  
    *" create  variable to hold data as subchild of our main variable*  
  
    DATA(lr\_document1) = lr\_data->create\_document( 'document1' ).  
  
    *" fill placeholder "data\_row" in document "document1"*  
    lr\_document1->append\_key\_table( iv\_key = 'data\_row' iv\_table = lt\_tmp ).  
  
    *" fill placeholder "subtotal" in document "document1"*  
    lr\_document1->append\_key\_table( iv\_key = 'subtotal' iv\_table = lt\_sub\_total\_tmp ).  
  
  
    *" our temlate contain 2 variant of main table: easy and wrong, more complex and true*  
  
    *"create variable to hold data for document2*  
    DATA(lr\_document2) = lr\_data->create\_document( 'document2' ).  
  
    *" fill placeholder "data\_row" in document "document2"*  
    lr\_document2->append\_key\_table( iv\_key = 'data\_row' iv\_table = lt\_tmp ).  
  
    *" fill placeholder "subtotal" in document "document2"*  
    lr\_document2->append\_key\_table( iv\_key = 'subtotal' iv\_table = lt\_sub\_total\_tmp ).  
  
  ENDLOOP.  
  
  
  
*\*final moment get document*  
  
  DATA  
        : lv\_document TYPE xstring  *" variable to hold generated document, can be omitted*  
        .  
  
  
  
  lv\_document = zcl\_docx2=>get\_document(  
      iv\_w3objid    = 'Z\_TEST\_DOCX2' *" name of our template, obligatory*  
*\*      iv\_on\_desktop = 'X'           " by default save document on desktop*  
*\*      iv\_folder     = 'report'      " in folder by default 'report'*  
*\*      iv\_path       = ''            " IF iv\_path IS INITIAL  save on desctop or sap\_tmp folder*  
*\*      iv\_file\_name  = 'report.docx' " file name by default*  
*\*      iv\_no\_execute = ''            " if filled -- just get document no run office*  
*\*      iv\_protect    = ''            " if filled protect document from editing, but not protect from sequence*  
                                     *" ctrl+a, ctrl+c, ctrl+n, ctrl+v, edit*  
      ir\_data       = lr\_data        *" root of our data, obligatory*  
*\*      iv\_no\_save    = ''            " just get binary data not save on disk*  
      ).

Run program and get something like this

59

60

Both variants seem acceptable. For the record, second gives you the most options.

Use it to create whatever you want.