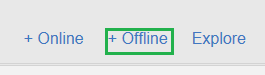
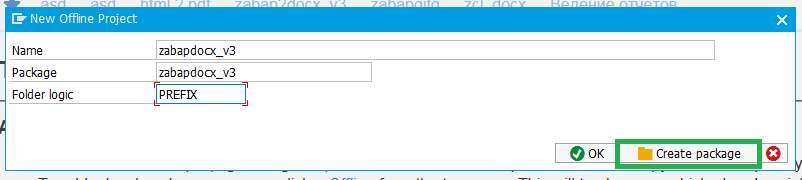
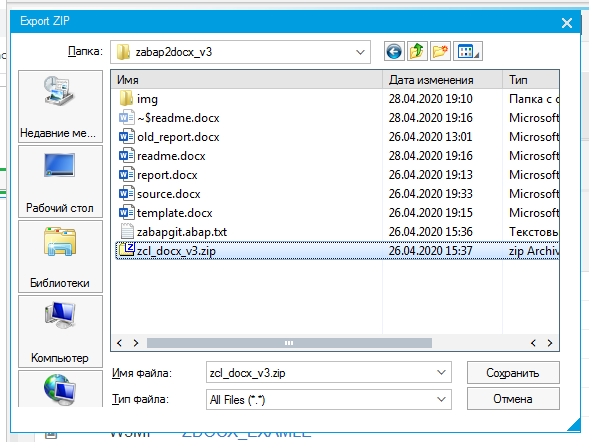
#Best way to create Microsoft Word docx from ABAP.

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



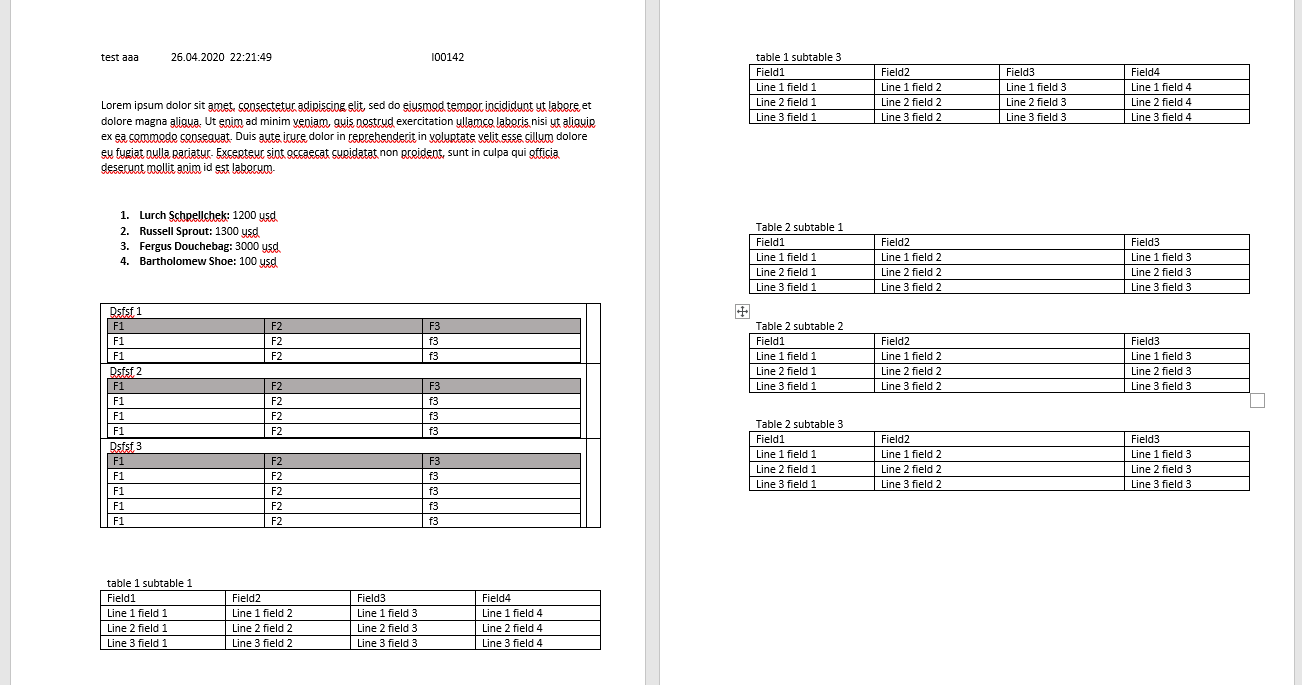


alt_text



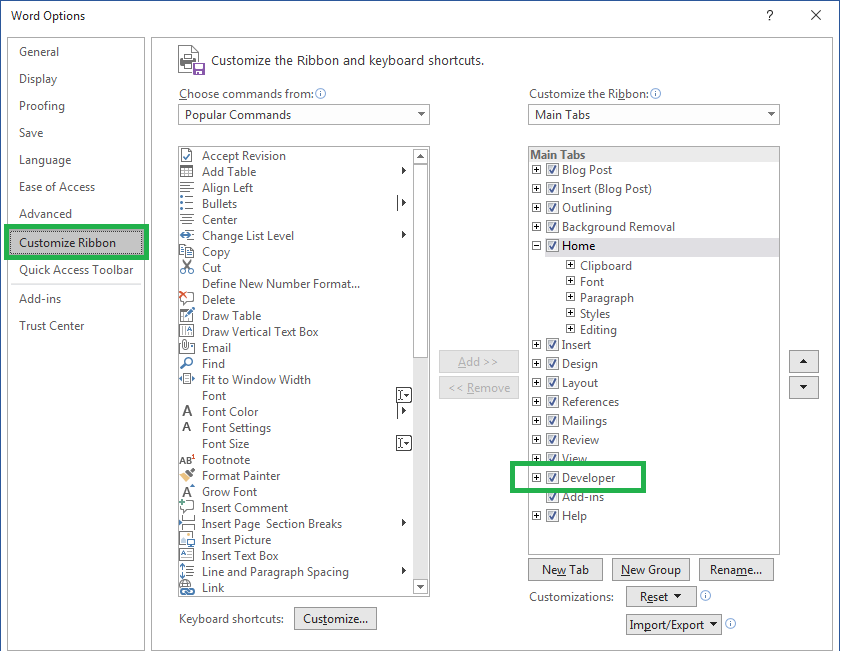
alt_text

For example, the following document should be created:

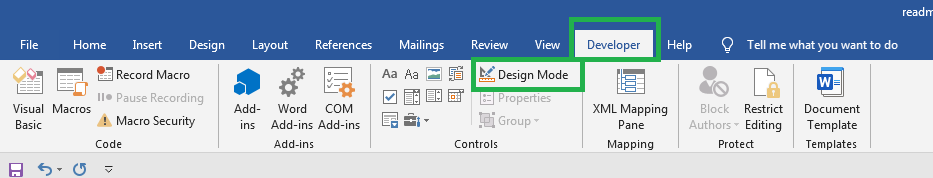


At first toggle developer toolbar.

File -> Options -> Customize ribbon

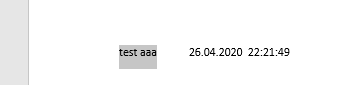


Go to developer tab, turn on “design mode”.

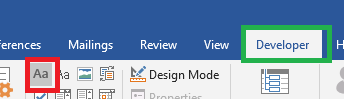


![alt\_text](images/z002\_08.png "image\_tooltip")

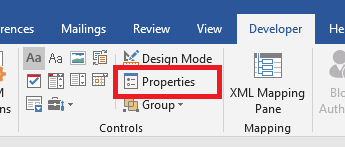
Select text that will be replaced.



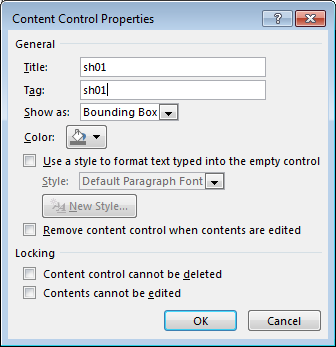
Make tag.



Click properties.

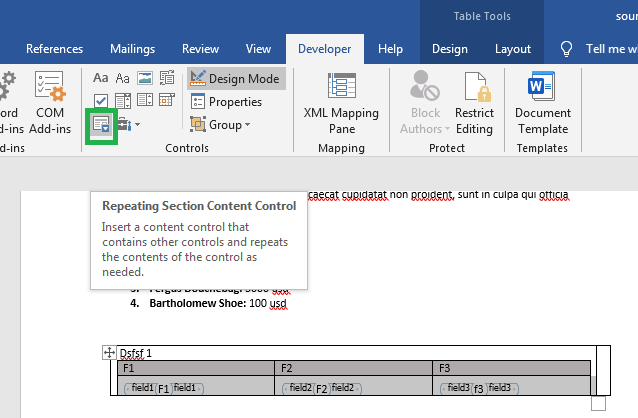


Enter tag name.

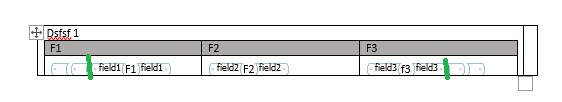


Repeat for all variable.

For repeated rows or text fragment - select row or text fragment, make repeated control.



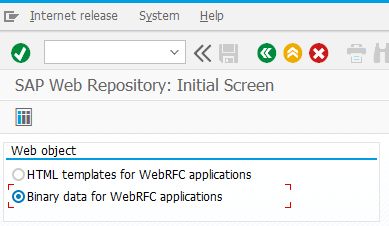
To enter properties of repeated control place cursor in the begin or end control.



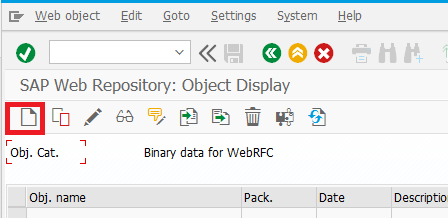
Tag all variables and repeated part.

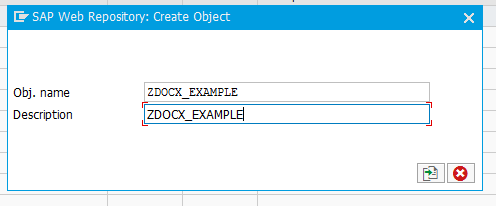
Save document.

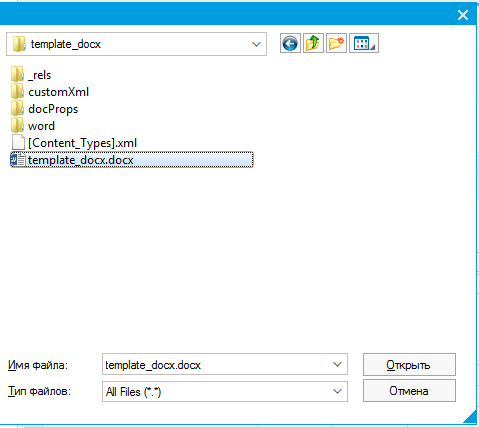
Go to transaction smw0 Select Binary data, enter.

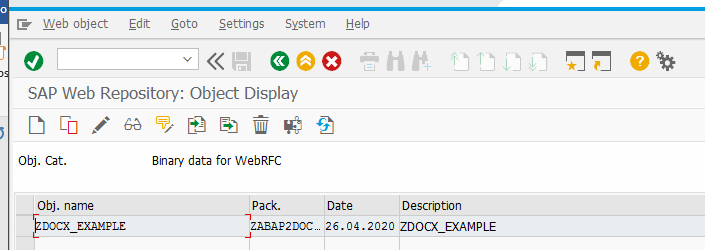


Object name zdocx\_example.





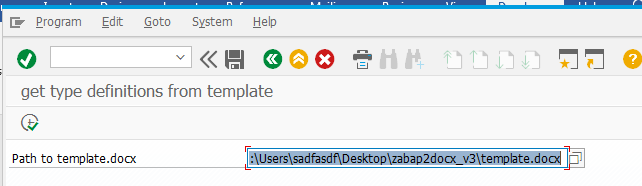




Go to se38.

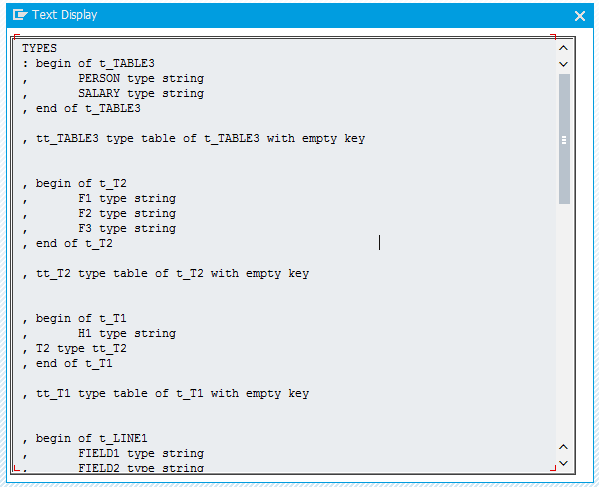
Program ZDOCX\_GET\_TYPES.

Navigate to your template.



Run.

Program generate data types, based on your document structure.



Copy to your program.

Define variable.

```

Data

: gs\_templ\_data TYPE t\_data

.

```

fill structure with your data.

Then get document

```

zcl\_docx3=>get\_document(  
    iv\_w3objid    = 'ZDOCX\_EXAMLE'   *" name of our template*  
*\*      iv\_template   = ''            " you can feed template as xstring instead of load from smw0*  
\*      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  
    iv\_data       = gs\_templ\_data  " root of our data, obligatory  
\*      iv\_no\_save    = ''            " just get binary data not save on disk  
    ).

```

Whole program “ZDOCX\_EXAMPLE”

```

\*&---------------------------------------------------------------------\*  
\*& Report ZDOCX\_EXAMPLE  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT zdocx\_example.  
  
\*describe types  
  
TYPES  
: begin of t\_TABLE3  
,       PERSON type string  
,       SALARY type string  
, end of t\_TABLE3  
  
, tt\_TABLE3 type table of t\_TABLE3 with empty key  
  
  
, begin of t\_T2  
,       F1 type string  
,       F2 type string  
,       F3 type string  
, end of t\_T2  
  
, tt\_T2 type table of t\_T2 with empty key  
  
  
, begin of t\_T1  
,       H1 type string  
, T2 type tt\_T2  
, end of t\_T1  
  
, tt\_T1 type table of t\_T1 with empty key  
  
  
, begin of t\_LINE1  
,       FIELD1 type string  
,       FIELD2 type string  
,       FIELD3 type string  
,       FIELD4 type string  
, end of t\_LINE1  
  
, tt\_LINE1 type table of t\_LINE1 with empty key  
  
  
, begin of t\_TAB1  
,       TITLE1 type string  
, LINE1 type tt\_LINE1  
, end of t\_TAB1  
  
, tt\_TAB1 type table of t\_TAB1 with empty key  
  
  
, begin of t\_LINE2  
,       FIELD1 type string  
,       FIELD2 type string  
,       FIELD3 type string  
, end of t\_LINE2  
  
, tt\_LINE2 type table of t\_LINE2 with empty key  
  
  
, begin of t\_TAB2  
,       TITLE2 type string  
, LINE2 type tt\_LINE2  
, end of t\_TAB2  
  
, tt\_TAB2 type table of t\_TAB2 with empty key  
  
  
, begin of t\_data  
,       SH01 type string  
,       DATE type string  
,       TIME type string  
,       USER type string  
, TABLE3 type tt\_TABLE3  
, T1 type tt\_T1  
, TAB1 type tt\_TAB1  
, TAB2 type tt\_TAB2  
, end of t\_data  
  
, tt\_data type table of t\_data with empty key  
  
  
.  
  
  
  
\*some variable  
DATA  
      : gs\_templ\_data TYPE t\_data  
      , lv\_index TYPE i  
      , lv\_index2 TYPE i  
      , lv\_index3 TYPE i  
      .  
  
  
\*fill data  
  
gs\_templ\_data-DATE = |{ sy-datum date = environment }|.  
gs\_templ\_data-TIME = |{ sy-uzeit   TIME = ENVIRONMENT }|.  
gs\_templ\_data-USER  = sy-uname.  
  
  
\*1.  Lurch Schpellchek: 1200 usd  
\*2.  Russell Sprout: 1300 usd  
\*3.  Fergus Douchebag: 3000 usd  
\*4.  Bartholomew Shoe: 100 usd  
  
APPEND INITIAL LINE TO gs\_templ\_data-table3 ASSIGNING FIELD-SYMBOL(<ls\_3>).  
<ls\_3>-person = 'Lurch Schpellchek'.  
<ls\_3>-salary = '1200'.  
  
APPEND INITIAL LINE TO gs\_templ\_data-table3 ASSIGNING <ls\_3>.  
<ls\_3>-person = 'Russell Sprout'.  
<ls\_3>-salary = '1300'.  
  
APPEND INITIAL LINE TO gs\_templ\_data-table3 ASSIGNING <ls\_3>.  
<ls\_3>-person = 'Fergus Douchebag'.  
<ls\_3>-salary = '3000'.  
  
APPEND INITIAL LINE TO gs\_templ\_data-table3 ASSIGNING <ls\_3>.  
<ls\_3>-person = 'Bartholomew Shoe'.  
<ls\_3>-salary = '100'.  
  
  
  
gs\_templ\_data-sh01 = 'test aaa'.  
  
DO 3 TIMES.  
  lv\_index = sy-index.  
  
  APPEND INITIAL LINE TO gs\_templ\_data-tab1 ASSIGNING FIELD-SYMBOL(<ls\_tab1>).  
  <ls\_tab1>-title1 = |table 1 subtable { lv\_index }|.  
  
  
  DO 3  TIMES.  
    lv\_index2 = sy-index.  
    APPEND INITIAL LINE TO <ls\_tab1>-line1 ASSIGNING FIELD-SYMBOL(<ls\_line1>).  
  
    DO 4 TIMES.  
      lv\_index3 = sy-index.  
  
      ASSIGN COMPONENT lv\_index3 OF STRUCTURE <ls\_line1> TO FIELD-SYMBOL(<fs\_any>).  
  
      <fs\_any> = |Line { lv\_index2  } field { lv\_index3 }|.  
  
    ENDDO.  
  
  
  ENDDO.  
  
  
ENDDO.  
  
  
DO 3 TIMES.  
  lv\_index = sy-index.  
  
  APPEND INITIAL LINE TO gs\_templ\_data-tab2 ASSIGNING FIELD-SYMBOL(<ls\_tab2>).  
  <ls\_tab2>-title2 = |Table 2 subtable { lv\_index }|.  
  
  
  DO 3  TIMES.  
    lv\_index2 = sy-index.  
    APPEND INITIAL LINE TO <ls\_tab2>-line2 ASSIGNING FIELD-SYMBOL(<ls\_line2>).  
  
    DO 3 TIMES.  
      lv\_index3 = sy-index.  
  
      ASSIGN COMPONENT lv\_index3 OF STRUCTURE <ls\_line2> TO <fs\_any>.  
  
      <fs\_any> = |Line { lv\_index2  } field { lv\_index3 }|.  
  
    ENDDO.  
  
  
  ENDDO.  
  
  
ENDDO.  
  
gs\_templ\_data = VALUE #( BASE gs\_templ\_data  
  t1 = VALUE #(  
    ( h1 = |1| t2 = VALUE #(  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
      )  
    )  
    ( h1 = |2| t2 = VALUE #(  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
      )  
    )  
    ( h1 = |3| t2 = VALUE #(  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
        ( f1 = 'F1' f2 = 'F2' f3 = 'f3' )  
      )  
    )  
  )  
).  
  
  
\*get document  
  
DATA  
       : lv\_document TYPE xstring  " variable to hold generated document, can be omitted  
       .  
  
\*first case:  send document as attachment  
  
lv\_document = zcl\_docx3=>get\_document(  
    iv\_w3objid    = 'ZDOCX\_EXAMLE'  
    iv\_data       = gs\_templ\_data  
    iv\_no\_save    = 'X'   ).  
  
PERFORM send\_document\_as\_attachment USING lv\_document.  
  
  
  
\*second case: save on desctop and open document  
  
zcl\_docx3=>get\_document(  
    iv\_w3objid    = 'ZDOCX\_EXAMLE'   *" name of our template*  
*\*      iv\_template   = ''            " you can feed template as xstring instead of load from smw0*  
\*      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  
    iv\_data       = gs\_templ\_data  " root of our data, obligatory  
\*      iv\_no\_save    = ''            " just get binary data not save on disk  
    ).  
  
  
FORM send\_document\_as\_attachment USING iv\_doc TYPE xstring.  
  
\*  implement sending here  
  MESSAGE 'Doc sended' TYPE 'S'.  
ENDFORM.

```