

TELITAB2WORDS\$

TELITAB2WORD\$ returns a MS Word macro to create a table in Word based on a given [TeLiTab](#) set

Syntax

TELITAB2WORD\$(Telitab\$, ParameterStyle(=-1,0,1,2,3,4), TableStyle\$, Transposed(=0,1), [Mode=0,1])

Arguments

- Telitab\$ is a string parameter containing [TeLiTab](#) data;
- ParameterStyle, option indicating the type of discription you want to use for the parameters in the table to be created;
- TableStyle\$, is a document containing a description of possible styles for the table. The syntax for the style definitions in TableStyle\$ do have a direct resemblance to the VBA code of Word. Feel free to experiment with the syntax. In the examples tested combination are presented. The following can be specified in this style document:
 - Formatted (True, False), do you want the formatted values (presented in the number of decimals presented in the [Workbase](#)) or the actual values;
 - ParameterBold (True, False), do you want bold parameter names;
 - Font, font type (such as "arial" or "courier") and size (such as 8 or 11);
 - OutsideBorder, the outside border of the table, defined into:
 - wdLineStyleSingle, Word syntax for single line;
 - wdLineWidth100pt, Word syntax for line thickness;
 - wdColorAutomatic, Word syntax for automatic color;
 - InsideGrid, the inside border for the table, defined into:
 - wdLineStyleSingle, Word syntax for single line;
 - wdLineWidth050pt, Word syntax for line thickness;
 - wdColorAutomatic, Word syntax for automatic color;
 - CaseNumbers (True, False), whether you want to see case numbers;
 - TableAlignment, specify the alignment of the table:
 - wdAlignRowCenter, Word syntax to align in the center;
 - wdAlignRowLeft, Word syntax to align on left side
- Transposed, to indicate whether the table should be presented transposed or not;
- Mode, option to determine the result you will receive:
 - Mode=0, (or not given) is the default. In this mode [Quaestor](#) will create a separate .bas file and use this file on the indicated spot in the Word template as macro to create the table. In other words, the parameter used with TELITAB2WORD will contain the link to this file;
 - Mode=1, is a advanced or debut mode. It will return the script itself (so the parameter using the TELITAB2WORD function will contain the text of the macro itself). When Mode=1 is selected, for instance for additional manipulation by the knowledge engineer, it should be saved using a PUT\$() function.

Remarks

1. The goal of this function is to create a VBA macro to created marked up tables in a Word document.
2. This function will only create the VBA code. To see the table you should place the parameter in a Word template and use the [WINWORD\\$\(\)](#) function to create the document based on this template (remember always to check your modified RTF template...)

Please note that we have limited ourselves in the number of possible styles as to be defined in TableStyle\$ parameter. However, the definition of the styles in TableStyle\$ does have direct resemblance with the VBA code of Word. You can use the macro record function in Word and investigate the generated VBA code to get inspiration for the TableStyle\$ content. In the examples tested combination are presented.

Examples

The TELITAB2WORDS\$() function behaves very much like the [TELITAB2HTML\\$\(\)](#) function in this sense that, depending on the [TeLiTab](#), it will generate the appropriate list and table as separate tables.

Assuming you have the following [TeLiTab](#) connected to the parameter MainDimensions#:

```
3"
BOT" 4
"CB" .65
"LOB" 6
4 "B" "LPP" "T" "VOL"
"1" 8.33 50.0 2.08 564
"2" 9.17 55.0 2.29 751
"3" 10.00 60.0 2.50 975
"4" 10.83 65.0 2.71 1240
"5" 11.67 70.0 2.92 1548
```

To generate the Word document containing this dataset as proper Word table, we use the following relation (see also [WINWORD\\$\(\)](#)):

MyReport\$ = WINWORD\$ Appendix_element.rtf("Report_" + STR\$(TIME(0)) + "_Set_" + STR\$(ORCA(1)) + ".doc", 0, VBA_Macro\$, ItemName)

As you see, in this relation VBA_Marco\$ and ItemName are requested (and they are put in the Word template, see the example knowledge base below).

The parameter ItemName is just a parameter to give a name in the generated document. VBA_Marco\$ contains the actual TELITAB2WORD\$ function to generate the VBA code required as Word macro in the document to generate the table. VBA_Marco\$ could have the following relation (in this case we have made the relation fully parametric but you can also provide the above described values right away):

VBA_Macro\$ = TELITAB2WORD\$(MainDimensions#, ParameterStyle, TableStyle\$, Transposed, Mode)

With

MainDimensions#, the above dataset;

ParameterStyle, user input for the ParameterStyle, for instance a select box using an <EQ> construction in the reference of the parameter and the @E QEXPLAIN attribute in the data slot of the parameter;

TableStyle\$, contains either a text document with descriptions for the Table style (such as different example presented below) or determined by an expression to generated this document for TableStyle\$ using separate parameters

FormattedBln\$, ParameterBoldBln\$, FontStyle\$, OutsideBorder\$, InsideGrid\$, CaseNumbersBln\$, andTableAlignment\$.The relation could be (see also the example knowledge base):

TableStyle\$ = TEMPLATE\$(TEXTITEM\$(1),0, FormattedBln\$,ParameterBoldBln\$,FontStyle\$,OutsideBorder\$,InsideGrid\$,CaseNumbersBln\$,TableAlignment\$)

with in the dataset:

TEXTITEM1=
|Formatted:=~FormattedBln\$
ParameterBold:=~ParameterBoldBln\$
Font:=~FontStyle\$
OutsideBorder:=~OutsideBorder\$
InsideGrid:=~InsideGrid\$
CaseNumbers:=~CaseNumbersBln\$
TableAlignment:=~TableAlignment\$|

Examples of TableStyle\$ definitions:

Centred, unformatted:

Formatted:=False
ParameterBold:=True
Font:=Arial 8.5
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=True
TableAlignment:=wdAlignRowCenter

Centred, formatted, with black parameters:

Formatted:=True
ParameterBold:=True,
ParameterBackColor:=wdColorBlack,
Font:=Arial 8.5
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=True
TableAlignment:=wdAlignRowCenter

Left aligned, formatted, with normal parameters:

Formatted:=True
ParameterBold:=False,
Font:=Arial 8.5
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=True
TableAlignment:=wdAlignRowLeft

Left aligned, formatted, with normal parameters, transposed table:

Formatted:=True
ParameterBold:=False,
Font:=Arial 8.5
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=True
TableAlignment:=wdAlignRowLeft

Centred, formatted, with black parameters, table without case numbers:

Formatted:=True
ParameterBold:=True,
ParameterBackColor:=wdColorBlack
Font:=Arial 8.5
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=False
TableAlignment:=wdAlignRowCenter

Centred, formatted, with black parameters, table without case numbers, transposed:

Formatted:=True
ParameterBold:=True,
ParameterBackColor:=wdColorBlack
Font:=Arial 8.5
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=False
TableAlignment:=wdAlignRowCenter

Left aligned, formatted, with normal parameters, larger courier font:

Formatted:=True
ParameterBold:=false,
Font:=Courier 14
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=True
TableAlignment:=wdAlignRowLeft

Left aligned, formatted, with normal parameters, no case numbers, larger courier font:

Formatted:=True
ParameterBold:=false,
Font:=Courier 14
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=False
TableAlignment:=wdAlignRowLeft

Left aligned, formatted, with normal parameters with no outside grid lines:

```
Formatted:=True
ParameterBold:=False,
Font:=Arial 8.5
OutsideBorder:=wdLineStyleNone
InsideGrid:=wdLineStyleSingle wdLineWidth050pt wdColorAutomatic
CaseNumbers:=True
TableAlignment:=wdAlignRowLeft
```

Left aligned, formatted, with normal parameters with no inside grid lines:

```
Formatted:=True
ParameterBold:=False,
Font:=Arial 8.5
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleNone
CaseNumbers:=True
TableAlignment:=wdAlignRowLeft
```

Left aligned, formatted, with normal parameters with no inside grid lines, no case numbers:

```
Formatted:=True
ParameterBold:=False,
Font:=Arial 8.5
OutsideBorder:=wdLineStyleSingle wdLineWidth100pt wdColorAutomatic
InsideGrid:=wdLineStyleNone
CaseNumbers:=False
TableAlignment:=wdAlignRowLeft
```

Transposed, user input to present the table transposed

Mode, user input for the mode of the function

The result of

MyReport\$

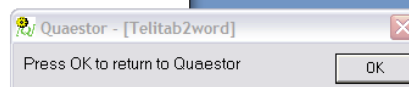
will be a Word document containing the table. For instance:

Table example

CB	0.650	[-]
LOB	6.000	[-]
BOT	4.000	[-]

LPP	[m]	50.0	55.0	60.0	65.0	70.0
B	[m]	8.33	9.17	10.00	10.83	11.67
T	[m]	2.08	2.29	2.50	2.71	2.92
VOL	[m^3]	564	751	975	1240	1548

The above table is generated using the macro of the *VBA_Macro\$* parameter.



Example knowledgebases

Download

An example knowledgebase of this function can be downloaded here.

Description

By using MyReport\$ as [TopGoal](#), the knowledge base generated the Word table based on:

1. An template that is included as binary (click here to download the separate template);
2. A TableStyle\$ definition that is highly parametric enabling you to make several style changes and see the result. In all cases the TableStyle\$ parameter will be generated on the basis of the individual choices.

Quick links: [Functions overview](#) | [Attribute overview](#) | [Constants overview](#) | [Dimensions overview](#)