

# TELITAB2HTML\$

TELITAB2HTML\$ returns a HTML table for a given [TeLiTab](#) set

## Syntax

1. TELITAB2HTML\$(TeLiTab\$, "NullString", Option%)

### Arguments

- [TeLiTab\\$](#) is a string parameter containing [TeLiTab](#) data;
- [NullString](#), obsolete parameter;
- Option%, option to get a HTML with or without table tag
  - Option%=0, (or not given) will return the HTML table including the <table> </table> tags;
  - Option%=1, will return the HTML table **without** the <table> </table> tags. In this way you can add several [TeLiTab](#) HTML tables to each other (especially when these are list values).

## Remarks

1. The goal of this function is to create HTML tables to be used as part of a script to generated an HTML document (using for instance the [TEMP LATES\(\)](#) function);
2. The use of this function assumes some basic HTML skills, meaning you are familiar to the HTML code required for creation of an HTML document. This in order to understand the result obtained from this function. Please visit [\[W3C\]](#) for more background information;
3. A [TeLiTab](#) will contain a list and a table. This is transformed into HTML as follows:
  - a. When Option%=0 is used, the a table of the list and a separate table of the table of the [TeLiTab](#) is created
  - b. When Option%=1 is used, the table tags are removed and both the HTML as created for for Option%=0 is place behind each other
  - c. The HTML table created for the list will always have the following structure (without the HTML tags...):  
Reference text | Parameter name | Parameter value | Parameter dimension
  - d. The HTML table created for the table will always have the following struncture (without the HTML tags...):

*Parameter Name | Parameter Name | Parameter Name | etc...*

*Dimension | Dimension | Dimension | etc...*

*Parameter value | Parameter value | Parameter value | etc.*

4. By removing the <table></table> tag, it is made possible to add HTML tables to each other. Use Option%=1 within the TELITAB2HTML function for each [TeLiTab](#) you want to add and place "<table>" before and "</table>" after the relation (see example);
5. By standard, [Quaestor](#) will include "class" attributes to the table tag. This enables you to use separate style sheets (\*.CSS files) to define the style of the table. The following attributes are placed:
  - For the list part of the [TeLiTab](#):
    - (with Option%=0) The <table> of the list includes *class="list"*
    - The <tr> tag includes *class="list"*
    - The <td> for reference includes *class="listref"*
    - The <td> for the parameter name includes *class="listpar"*
    - The <td> for the parameter value includes *class="listval"*
    - The <td> for the parameter dimension includes *class="listdim"*
    - For the table part of the [TeLiTab](#):
      - (with Option%=0) The <table> of the list includes *class="table"*
      - The <tr> tag includes *class="list"*
      - The <td> for reference includes *class="tableref"*
      - The <td> for the parameter name includes *class="tablepar"*
      - The <td> for the parameter value includes *class="tableval"*
      - The <td> for the parameter dimension includes *class="tabledim"*
1. [Quaestor](#) will also place a <a ...> </a> tag around the parameter name. This will link to separate HTML files created automatically based on all reference text and binary data in the knowledge base (see [Documentation of knowledge](#) )

## Examples

### Example 1: using Option%=0

Further on the example of TeLiTab#(). You have a [TeLiTab](#) C0#:

```

0
2 "A" "B"
"1" 1 2
"2" 2 4
"3" 3 6
"4" 4 8
"5" 5 10
"6" 6 12
"7" 7 14
"8" 8 16
"9" 9 18
"10" 10 20

```

Relation:

HTML\_C0# = TELITAB2HTML\$(C0#, "NullString", 0)

will return for HTML\_C0# (in text):

```

<table class="table">
<tr class="table"><td class="tableref">&nbsp;</td><td class="tableref"><a href="file:///E:/my documents/my knowledge/kbs/_Telitab2html/Html
/documents/A.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('A', event);" onMouseOut="return !hideCurrentPopup();">A
</a></td><td class="tableref"><a href="file:///E:/my documents/my knowledge/kbs/_Telitab2html/Html/documents/B.htm" tabindex="1000" class="
popupLink" onMouseOver="return !showPopup('B', event);" onMouseOut="return !hideCurrentPopup();">B </a></td></tr>
<tr class="table"><td class="tableref">&nbsp;</td><td class="tabledim">[-]</td><td class="tabledim">[-]</td></tr>
<tr class="table"><td class="tableref">"1"</td><td class="tableval">1 </td><td class="tableval">2</td></tr>
<tr class="table"><td class="tableref">"2"</td><td class="tableval">2 </td><td class="tableval">4</td></tr>
<tr class="table"><td class="tableref">"3"</td><td class="tableval">3 </td><td class="tableval">6</td></tr>
<tr class="table"><td class="tableref">"4"</td><td class="tableval">4 </td><td class="tableval">8</td></tr>
<tr class="table"><td class="tableref">"5"</td><td class="tableval">5 </td><td class="tableval">10</td></tr>
<tr class="table"><td class="tableref">"6"</td><td class="tableval">6 </td><td class="tableval">12</td></tr>
<tr class="table"><td class="tableref">"7"</td><td class="tableval">7 </td><td class="tableval">14</td></tr>
<tr class="table"><td class="tableref">"8"</td><td class="tableval">8 </td><td class="tableval">16</td></tr>
<tr class="table"><td class="tableref">"9"</td><td class="tableval">9 </td><td class="tableval">18</td></tr>
<tr class="table"><td class="tableref">"10"</td><td class="tableval">10 </td><td class="tableval">20</td></tr>
</table>

```

As you see, there are no list values. This will result in the following HTML table:

	A	B
	[-]	[-]
"1"	1	2
"2"	2	4
"3"	3	6
"4"	4	8
"5"	5	10
"6"	6	12
"7"	7	14
"8"	8	16
"9"	9	18
"10"	10	20

If only A is varied and B = 2 (in order to have a TeLiTab with a list and table value), the same relation will result in:

```
<table class="list">
<tr class="list"><td class="listref">A&#32;value</td><td class="listpar"><a href="file:///E:/my documents/my knowledge/kbs/_Telitab2html/Html
/documents/B.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('B', event);" onMouseOut="return !hideCurrentPopup();">B
</a></td><td class="listval">2</td><td class="listdim">[-]</td></tr>
</table>
<br><table class="table">
<tr class="table"><td class="tableref">&nbsp;</td><td class="tableref"><a href="file:///E:/my documents/my knowledge/kbs/_Telitab2html/Html
/documents/A.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('A', event);" onMouseOut="return !hideCurrentPopup();">A
</a></td></tr>
<tr class="table"><td class="tableref">&nbsp;</td><td class="tabledim">[-]</td></tr>
<tr class="table"><td class="tableref">"1"</td><td class="tableval">1</td></tr>
<tr class="table"><td class="tableref">"2"</td><td class="tableval">2</td></tr>
<tr class="table"><td class="tableref">"3"</td><td class="tableval">3</td></tr>
<tr class="table"><td class="tableref">"4"</td><td class="tableval">4 </td></tr>
<tr class="table"><td class="tableref">"5"</td><td class="tableval">5</td></tr>
<tr class="table"><td class="tableref">"6"</td><td class="tableval">6</td></tr>
<tr class="table"><td class="tableref">"7"</td><td class="tableval">7</td></tr>
<tr class="table"><td class="tableref">"8"</td><td class="tableval">8</td></tr>
<tr class="table"><td class="tableref">"9"</td><td class="tableval">9</td></tr>
<tr class="table"><td class="tableref">"10"</td><td class="tableval">10</td></tr>
</table>
```

This will result in the following HTML tables:

A value	B	2	[-]
---------	---	---	-----

	A
	[-]
"1"	1
"2"	2
"3"	3
"4"	4
"5"	5
"6"	6
"7"	7
"8"	8
"9"	9
"10"	10

**Example 2: using Option%=1**

Assume you have two separate TeLiTab sets with list values:

Set 1, C1#:

"A" 1  
"B" 2

Set 2, C2#:

"C" 3  
"D" 4

If you would use the following relation to create an HTML table of the two lists, two separate HTML tables will be created:

```
HTML1#=TELITAB2HTML$(C1#,"NullString",0)+TELITAB2HTML$(C2#,"NullString",0)
```

Resulting into:

```
<table class="list">
<tr class="list"><td class="listref">A&#32;value</td><td class="listpar"><a href="file:///E:/my documents/my knowledge/kbs/_Telitab2html/Html
/documents/A.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('A', event);" onMouseOut="return !hideCurrentPopup();">A
</a></td><td class="listval">1</td><td class="listdim">[-]</td></tr>
<tr class="list"><td class="listref">A&#32;value</td><td class="listpar"><a href="file:///E:/my documents/my knowledge/kbs/_Telitab2html/Html
/documents/B.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('B', event);" onMouseOut="return !hideCurrentPopup();">B
</a></td><td class="listval">2</td><td class="listdim">[-]</td></tr>
</table><table class="list">
<tr class="list"><td class="listref">A&#32;far&#32;longer&#32;description&#32;for&#32;parameter&#32;C</td><td class="listpar"><a href="file:///E:/my
documents/my knowledge/kbs/_Telitab2html/Html/documents/C.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('C',
event);" onMouseOut="return !hideCurrentPopup();">C </a></td><td class="listval">3</td><td class="listdim">[-]</td></tr>
<tr class="list"><td class="listref">A&#32;description&#32;for&#32;D</td><td class="listpar"><a href="file:///E:/my documents/my knowledge/kbs
/_Telitab2html/Html/documents/D.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('D', event);" onMouseOut="return !
hideCurrentPopup();">D </a></td><td class="listval">4</td><td class="listdim">[-]</td></tr>
</table>
```

Resulting in the following two tables:

A value	A	1	[-]
A value	B	2	[-]

  

A far longer description for parameter C	C	3	[-]
A description for D	D	4	[-]

You see that the description and values of the two sets are not align properly. This is because for HTML you have two tables with individual layout. To avoid this, you would like to have the two sets in one table right away.

For sure you can doe this by first carrying out a union on the two sets using the [UNION#\(\)](#) function. However, you can also present the two tables into one HTML table by using Opion%=1 in combination with an HTML table tag in front and closing tag at the end:

```
HTML1#="<table>" + TELITAB2HTML$(C1#,"NullString",1)+TELITAB2HTML$(C2#,"NullString",1) + "</table>"
```

Resulting into:

```
<table><tr class="list"><td class="listref">A&#32;value</td><td class="listpar"><a href="file:///E:/my documents/my knowledge/kbs/_Telitab2html/Html
/documents/A.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('A', event);" onMouseOut="return !hideCurrentPopup();">A
</a></td><td class="listval">1</td><td class="listdim">[-]</td></tr>
<tr class="list"><td class="listref">A&#32;value</td><td class="listpar"><a href="file:///E:/my documents/my knowledge/kbs/_Telitab2html/Html
/documents/B.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('B', event);" onMouseOut="return !hideCurrentPopup();">B
</a></td><td class="listval">2</td><td class="listdim">[-]</td></tr>
<tr class="list"><td class="listref">A&#32;far&#32;longer&#32;description&#32;for&#32;parameter&#32;C</td><td class="listpar"><a href="file:///E:/my
documents/my knowledge/kbs/_Telitab2html/Html/documents/C.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('C',
event);" onMouseOut="return !hideCurrentPopup();">C </a></td><td class="listval">3</td><td class="listdim">[-]</td></tr>
<tr class="list"><td class="listref">A&#32;description&#32;for&#32;D</td><td class="listpar"><a href="file:///E:/my documents/my knowledge/kbs
/_Telitab2html/Html/documents/D.htm" tabindex="1000" class="popupLink" onMouseOver="return !showPopup('D', event);" onMouseOut="return !
hideCurrentPopup();">D </a></td><td class="listval">4</td><td class="listdim">[-]</td></tr>
</table>
```

Resulting in the following table:

A value	A	1	[-]
A value	B	2	[-]
A far longer description for parameter C	C	3	[-]
A description for D	D	4	[-]

# Example knowledgebases

## Download

An example knowledgebase of this function can be downloaded here.

## Description

The knowledge base provides the two methods above.

[TopGoal HTML\\_C0#](#) generates the HTML table as described in Method 1, [HTML\\_C1\\_C2#](#) generates the HTML table as described in Method 2.

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