

EVALUATE#

EVALUATE# returns a [TeLiTab](#) containing a subset (derived) of two Telitab sets normalised on the axis distribution of first Telitab

Syntax

EVALUATE#(Mode%, Telitab1\$, Telitab2\$, [InpVar1\\$](#)[:Label1\$], ..., [InpVarN\\$](#)[:LabelN\$])

Arguments

- Mode% controls the interpolation method by which to normalise data from Telitab2\$ according to the distribution of [InpVar1\\$](#) in Telitab1\$

1. Mode% = 1 -> discrete interpolation
2. Mode% = 2 -> linear interpolation
3. Mode% = 3 -> double quadratic interpolation
4. Mode% = 4 -> spline interpolation
5. Mode% = 5 -> least square approximation

- Telitab1\$ is a string expression containing/returning the first Telitab set
- Telitab2\$ is a string expression containing/returning the second Telitab set
- [InpVar1\\$](#) is the name of the axis parameter for which to normalise data in Telitab2\$, or an expression on the basis of parameters present in the sets, e.g. "Rtot*Vs".
- If an expression is provided, a parameter Label1\$ should be provided, eg. "Rtot*Vs"."PE".
So, [InpVar1\\$](#)/Label1\$ will be the first parameter in the returned Telitab result. Got to [General Quaestor syntax](#) why you have to place parameters between quotes or not.
- [InpVar2\\$](#)[:Label2\$] is the name of the first (normalised) parameter or expression:Label. If [InpVar2\\$](#) is present in both Telitab1\$ and Telitab2\$, the values from Telitab1\$ are preferred. If you want to use the same parameter (e.g. "Power") from Telitab1\$ and Telitab2\$ for computing e.g. the difference and name that e.g. "Dpow", you can realise that by placing an underscore _ before the parameter by which you address the parameter in Telitab2\$, so like this: "Power-_Power"."Dpow"
- [InpVar1\\$](#) to [InpVarN\\$](#)[:LabelN\$] will be included in the resulting Telitab set.

Remarks

1. The EVALUATE# function can be used to carry out operations on one or two [TeLiTab](#) sets.
2. If you use parameters of which the values are not given in either one of the input [TeLiTab](#) sets, the system will search their value in the current case of the solution. If not available, '-999999' will be returned in the returned [TeLiTab](#) set.
3. Telitab2\$ may be empty ("[NullString](#)"). In that case, the data in Telitab1\$ is used in the evaluation and creation of the result [TeLiTab](#) set.
4. It is not allowed within a single EVALUATE#() function to add new parameters and use them immediately in an expression within the same call. If you want to do so, it can be done by nesting EVALUATE#() functions.
5. You are allowed to use standard functions of Quaestor as well as defined parameters available in the solution.
6. You are not allowed to provide ranges to other than the first [InpVar\\$](#):Label\$ combination in the EVALUATE#() expression.
7. If an argument in the EVALUATE# function is between (), its value is used in the evaluation but not included in its resulting [TeLiTab](#) set.

Examples

Example 1

The expression

```
EVALUATE#(3, TEXTITEM$(1), TEXTITEM$(2), "V", "P-_P"."DeltaP")
```

containing the following data in its data slot:

```
TEXTITEM1=
```

```
|2 "V" "P"  
"1" 10 10000  
"2" 12 12000  
"3" 14 14000  
"4" 16 16000|
```

```
TEXTITEM2=
```

returns:

```
|2 "V" "P"
"1" 10 9300
"2" 12 11400
"3" 14 13500
"4" 16 15600|
```

```
0
2 "V" "DeltaP"
"1" 10 700
"2" 12 600
"3" 14 500
"4" 16 400
```

Example 2

Let Telitab1\$ contain the dataset:

```
0
2 "RTot" "V"
"1" 31.967 12.00
"2" 41.914 13.00
"3" 49.684 14.00
"4" 51.768 15.00
"5" 53.848 16.00
"6" 55.900 17.00
"7" 57.897 18.00
"8" 60.430 19.00
"9" 64.614 20.00
"10" 70.409 21.00
"11" 76.435 22.00
```

and Telitab2\$ contain the dataset:

```
0
2 "RTot" "V"
"1" 32.434 12.00
"2" 50.228 14.00
"3" 53.734 16.00
"4" 57.111 18.00
"5" 63.964 20.00
"6" 75.760 22.00
```

RTot in kN, V in knots.

We can compare these two speed/resistance curves on the basis of speed in m/s as follows:

```
EVALUATE#(3, Telitab1$, Telitab2$, "V*0.51444":"Vs", "RTot/_RTot":"ResRatio", "RTot*V*0.51444":"PE1", "_RTot*V*0.51444":"PE2")
```

returns (values rounded):

```
0
6 "Vs" "ResRatio" "PE1" "PE2"
"1" 6.173 0.9856 197.341 200.224
"2" 6.687 0.9720 280.309 288.354
"3" 7.202 0.9891 357.832 361.750
"4" 7.716 0.9789 399.472 408.069
"5" 8.231 1.0021 443.225 442.286
"6" 8.745 1.0124 488.872 482.866
"7" 9.259 1.0137 536.121 528.843
"8" 9.774 1.0069 590.664 586.572
"9" 10.288 1.0101 664.800 658.112
"10" 10.803 1.0168 760.645 748.060
"11" 11.317 1.0089 865.066 857.427
```

Example 3

Create a dataset on an equidistant axis "V" on the basis of an axis definition in Teltab1\$ and a non-equidistant set V-RTot in Telitab2\$.

Let Telitab1\$ contain:

Let Telitab2\$ contain:

```
0
1 "V"
"1" 12.00
"2" 14.00
"3" 16.00
"4" 18.00
"5" 20.00
"6" 22.00
```

```
0
2 "V" "RTot"
"1" 12.10 33.663
"2" 14.30 51.529
"3" 15.90 53.579
"4" 18.10 57.369
"5" 20.40 65.928
"6" 21.70 73.675
```

The expression:

```
EVALUATE#(3, Telitab1$, Telitab2$, "V", "RTot")
```

returns (values rounded):

```
0
2 "V" "RTot"
"1" 12.00 32.436
"2" 14.00 50.119
"3" 16.00 53.723
"4" 18.00 57.107
"5" 20.00 63.990
"6" 22.00 75.762
```

Example 4

If Telitab1\$="NullString" (and if Telitab2\$="NullString" as in this example), you can pass table information in the first [InpVar\\$](#):Label\$ combination, e.g.:

```
EVALUATE#(3,"NullString", "NullString", "2(1)6": "X", "X^2": "Y")
```

returns:

```
0
2 "X" "Y"
"1" 2 4
"2" 3 9
"3" 4 16
"4" 5 25
"5" 6 36
```

This construction allows you to provide the equidistant range of a table to be normalised as in example 5.

Example 5

If Telitab1\$="NullString" and if Telitab2\$ contains a table to be normalised:

```
0
2 "V" "RTot"
"1" 12.10 33.663
"2" 14.30 51.529
"3" 15.90 53.579
"4" 18.10 57.369
"5" 20.40 65.928
"6" 21.70 73.675
```

you can pass table information in the first [InpVar\\$:](#)Label\$ combination, e.g.:

```
EVALUATE#(3, "NullString", Telitab2$, "12(1)22":"V", "RTot")
```

returns (values rounded):

```
0
2 "V" "RTot"
"1" 12 32.436
"2" 13 43.077
"3" 14 50.119
"4" 15 53.031
"5" 16 53.723
"6" 17 55.135
"7" 18 57.107
"8" 19 60.071
"9" 20 63.990
"10" 21 69.242
"11" 22 75.762
```

Example 6

If an argument in the EVALUATE# function is between (), its value is used in the evaluation but not included in its resulting [TeLiTab](#) set.

In the example below, EVALUATE# is used to generate a table containing R, v1, v2, v3, v4, DIAM en MCAM based on GEOTABLE\$ and an additional operations on content in GEOTABLE\$ using the [DQUAD](#) function and the parameters DIAM, MCAM and CMAXTDIST#. However, in the result we do not want to have the values of CMAXTDIST#.

We can achieve this without, for instance, having to filter out this data using the [UNFOLD#](#) function by writing the expression as follows:

```
EVALUATE#(2, GEOMTABLE$, "Nullstring", "R", "v1":"v1", "v2":"v2", "v3":"v3", "DQUAD(CMAXTDIST#,2,'RADIUS','Yradius',R/(0.5*DIAM),1)":"v4",
DIAM, MCAM, (CMAXTDIST#))
```