

GET\$

GET\$ runs a satellite program and returns the contents of a file

Syntax

GET\$(FileName\$, Process\$, [InpVar])

Arguments

- FileName\$ is a string expression containing the name of a file of which the contents has to be returned. If FileName\$ does not contain a path, the file is assumed to be present on the current [Quaestor](#) Report directory.
- Process\$ is a name of satellite program to be invoked, e.g. producing FileName\$ as output file. If Process\$="NullString" or "", no program is started and simply the contents of file FileName\$ is returned. If Process\$ does not contain a path, the program is assumed to be present on the current [Quaestor](#) Applications directory.
- InpVar is an optional list of parameters or functions. The list may contain one or more PUT\$() function calls which writing data to input file(s) required by Process\$.

Remarks

1. The GET\$ function can be used in two ways; either to retrieve data from an existing file, or to run another program that generates a file from which GET\$ retrieves data into a [TeLiTab](#).
2. GET\$() is an alternative for GET\$ as it returns ALL output in file FileName\$ in the form of a single string. The result can be assigned to a string parameter which in its term can be used as data source Telitab\$ for other functions such as interpolation or integration. GET\$() can be used with programs using and producing Telitab data but also by those working with file IO in an arbitrary format. GET\$ checks the necessity of rerunning a program by comparing the current input with previous input. GET\$() always reruns without comparing previous with new input. GET\$() in combination with [TEMPLATES\\$\(\)](#) and PUT\$() is comparable with [GET\\$](#) using an input template, viz. a structural recipe for the (single) input file. GET\$() can create multiple input files each using their own template ([TEMPLATES\\$\(\)](#)).
3. Please note that the Process\$ can be the execution of a batch file that is carrying out several command line actions. Furthermore, this batch file itself can be generated knowledge based by means of a PUT\$() combined with a [TEMPLATES\\$\(\)](#) action. Using the report path where the batch file is saved in the Process\$ (use [SYSTEMVAR\\$\(\)](#) to request the path from [Quaestor](#)) the approach enables many programs and processes to be executed in a very flexible way. Look on the internet and documentation of the processes you want to execute about detailed possibilities.
4. Furthermore, please realise that you can use the [@COPYFIRST](#) attribute to make sure some required external files are copied to the working directory prior to running the program specified in the GET\$. Furthermore, when you add the process itself to the [@COPYFIRST](#) [Quaestor](#) will remove any path information given in front of the process in the GET\$ function.

Examples

Example 1: Retrieving data from a file

Using the relation:

```
A$ = GET$("DATAFILE.DAT", "")
```

If the file DATAFILE.DAT contains:

```
0
2 "XC" "YC"
"1" 1 1
"2" 2 4
"3" 3 9
"4" 4 16
"5" 5 25
"6" 6 36
"7" 7 49
"8" 8 64
"9" 9 81
"10" 10 100
```

The string A\$ will contain the [TeLiTab](#):

```
0
2 "XC" "YC"
"1" 1 1
"2" 2 4
"3" 3 9
"4" 4 16
"5" 5 25
"6" 6 36
"7" 7 49
"8" 8 64
"9" 9 81
"10" 10 100
```

Example 2: Running a satellite program (combination with PUT\$)

Let the relation OUTPUT\$ be defined by

```
OUTPUT$ = GET$("OUTPUT.EPO", "SATTELITE_PROGRAM.EXE", PUT$("INPUT.EPI", INPUT$))
```

This relation results in the following:

An input string (INPUT\$) is placed in an input file (INPUT.EPI). This file will be used by the process (SATTELITE_PROGRAM.EXE) to generate an output file (OUTPUT.EPO). This output file is then brought into [Quaestor](#) as an output string (OUTPUT\$). Within [Quaestor](#), this string or [TeLiTab](#) can be used for all kinds of purposes.

Example 3: Executing an embedded file

The relation

```
A$ = GET$("NullString", "NullString")
```

Embed a Binary by means of selecting the relation and right click select: "Include Binary in frame...". When A\$ is requested, the embedded content is assumed to be an executable file. If the embedded object has the ".EXE" extension, this executable is started without expecting an output file.

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