



QS\_Toolbox

## Table

<b>Introduction .....</b>	<b>3</b>
What is QS_Toolbox .....	3
Demonstration version vs Commercial version .....	3
Component installation .....	4
Information structure in QS_Toolbox .....	5
<b>General functions.....</b>	<b>6</b>
<b>DB Overview .....</b>	<b>7</b>
<b>Method management .....</b>	<b>7</b>
<b>Method Overview .....</b>	<b>8</b>
Buttons bar .....	8
Search area .....	8
The magnifying glass .....	9
The pen.....	9
The printer.....	10
The page .....	10
Eyes, cursor and folder.....	11
<b>Method Analyzer .....</b>	<b>13</b>
<b>4D Command .....</b>	<b>14</b>
<b>Methods Pallet.....</b>	<b>15</b>
<b>Form Explorer .....</b>	<b>16</b>
<b>Structure Analysis .....</b>	<b>18</b>
Structure .....	18
Display table & field definition data .....	18
Specific options .....	19
Table Relations.....	19
Search & Control .....	20
Check & Update .....	22
4D Link Editor.....	24
<b>Converter EN / FR.....</b>	<b>25</b>
<b>Version Control System .....</b>	<b>27</b>
<b>Reports .....</b>	<b>29</b>
<b>Setup .....</b>	<b>30</b>
<b>New DB Analysis .....</b>	<b>30</b>
<b>Restart 4D.....</b>	<b>30</b>

## Introduction

QS\_Toolbox was developed with 4D v17 and compiled in 4D v17.3.

Because it follows the standard version branch, it can be used without any problem on the R version branch.

## What is QS\_Toolbox

It is a component for 4D developers to use when analyzing their code, transferring code or migrating.

QS\_Toolbox allows in-depth analysis of the structure in order to help you detect possible errors and programming difficulties.

QS\_Toolbox does not modify your data without your permission. Except for the method parameters and the replacement of "disapproved" commands, no changes can be made in your code.

## Demonstration version vs Commercial version

QS\_Toolbox is free, and the source code is available up to version 2.2 of the component.

Since version 3.0, QS\_Toolbox has become a commercial component whose cost is excessively low compared to the services it is able to provide you.

The demo version is fully functional for 60 days.

## Component installation

The component (or its shortcut / alias) must be placed in the 'Components' folder of your application to scan.

To start it, you can set the following code in 'on startup database' method:

```
ARRAY TEXT($Txt_Components;0)
COMPONENT LIST($Txt_Components)
If (Not(Is compiled mode))

    // Help to get back the hand if an issue on startuo
    If ((Application type=4D Local mode) & (Is compiled mode=False) & (Shift down))
        TRACE
    End if

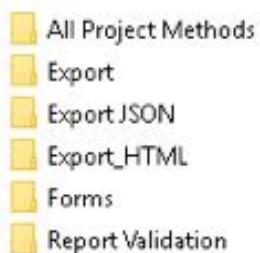
    // ASSERT on
    SET ASSERT ENABLED(True)

    // QS_Toolbox
    If (Find in array($Txt_Components;"QS_Toolbox@")>0)
        EXECUTE METHOD("QS_Toolbox")
    End if

Else
    // ASSERT off
    SET ASSERT ENABLED(False)
End if
```

At the first launch, a folder is created on the local disk and located:

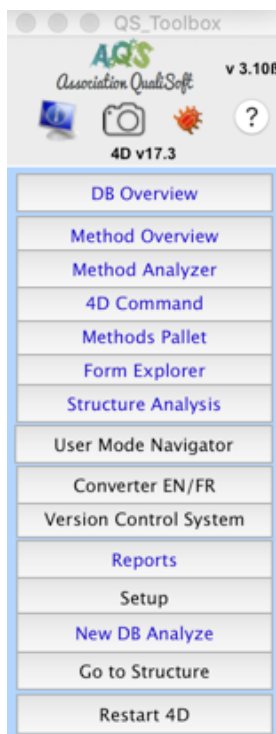
- PC : 'drive':\my documents\QS\_Toolbox\<base name>
- Mac : /documents/QS\_Toolbox/<base name>



In the folder <base name>, several sub-folders are created to organize the output information.

A data file is also created in this folder and is used to manage parameters and save methods as part of version management. This data file is called 'QST\_External\_DB'.

## Information structure in QS\_Toolbox



After launch, the component is presented in the form of a pallet accessible in structure mode.

The entire interface is in English.

All different options are accessible through menus. Those in blue are only accessible once the startup database analysis is completed. The duration of this analysis depends on the complexity of the structure (number of tables, forms, methods, ...).

The themes of QS\_Toolbox are:

- Methods Analysis
- Analysis of Forms
- Analysis of the database structure
- Code conversion (EN <-> FR)
- 4D method version management
- Automated reporting

Note: When the component is launched, it starts the analysis of your database. This analysis can temporarily slow down your application and allow a smooth navigation between the different elements of the component.

In case of a major change, you can restart the analysis of your database at any time. To do this, we recommend that you close all windows of the component.

## General functions



The upper part of the tool palette provides you with the following information:

- Name of the analyzed application (here QS\_Toolbox itself)
- Version number of the component (here 3.00)
- 4D version number used (here v17.3)

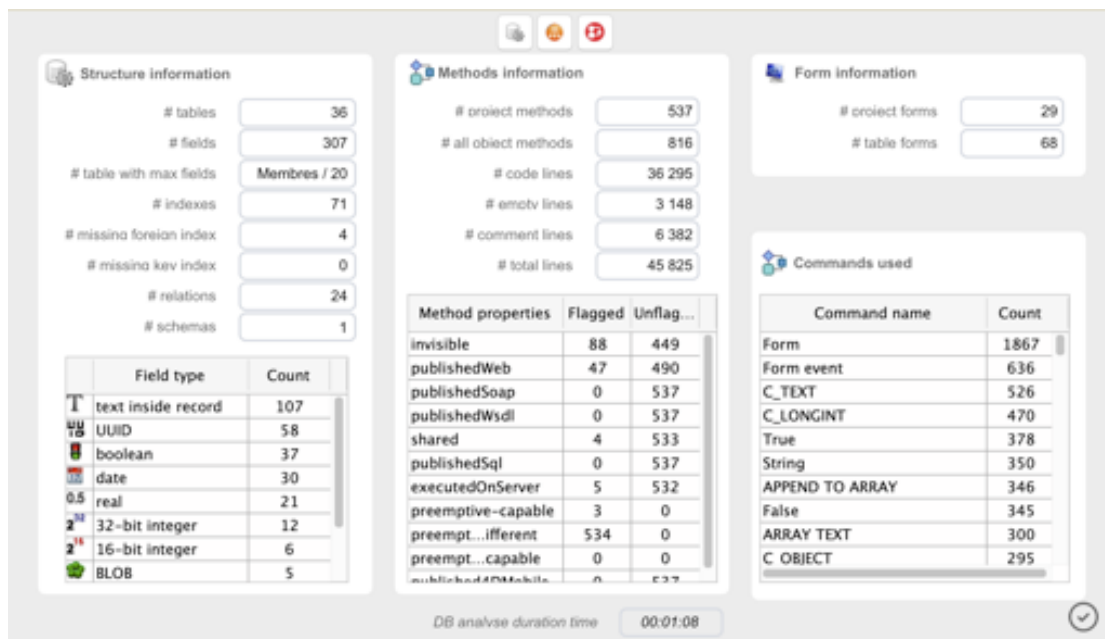
The available tools are, from left to right:

- Storage of system information
- Screen copy automatically saved in the Log folder of the database
- Access to the site of the "bugs list" of the major version of 4D in progress
- The 'About' of the component.

The license number is also entered from this location

Generally speaking, a double click on the name of a method opens it in the 4D editor. In case of changes, they will only be included in the QS\_Toolbox analysis after reanalysis of your database. This has been done voluntarily so as not to have to create a process that runs in the background and slows down your application.

## DB Overview



This module displays a synthetic view of the application:

- Structure information
- Information on methods
- Information on the commands used (updating is done asynchronously).



Clicking on each icon displays a dialog specific to the structure, links and index, or to restart the analysis of the commands used.

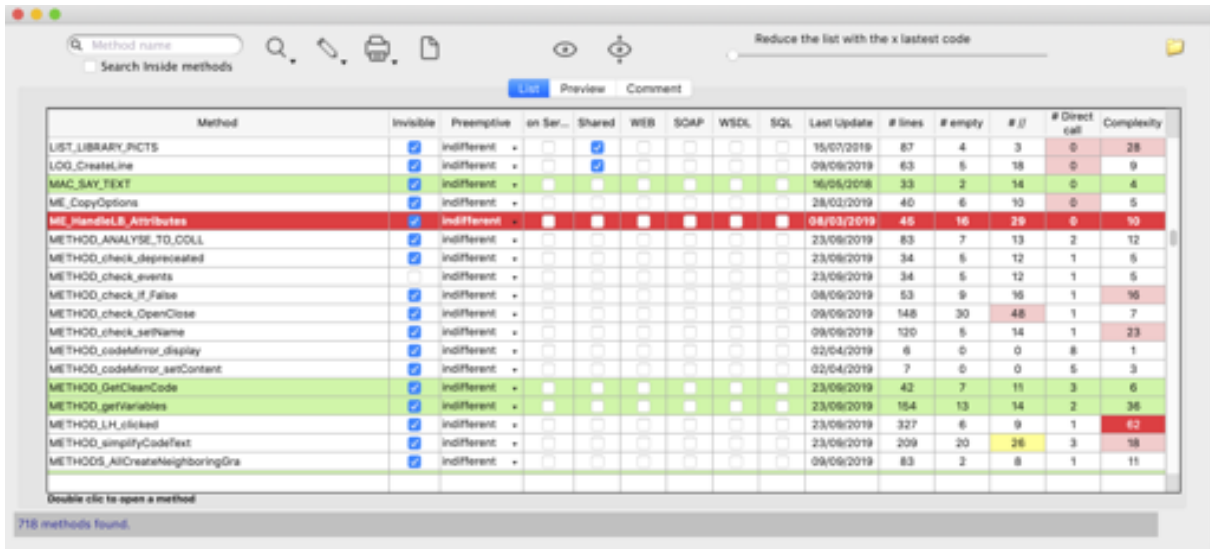
## Method management

The management of methods is divided into 4 distinct parts that we will study below:

- Method Overview: view a set of statistics on methods and modify their properties
- Method Analyser: in-depth analysis of your methods
- 4D command: lists methods from 4D commands and also provides other information
- Pallet methods: quick access to all your code, including form widgets



## Method Overview



Method	Invisible	Preemptive	on Ser...	Shared	WEB	SOAP	WSDL	SQL	Last Update	# lines	# empty	# j	# Direct call	Complexity
LIST_LIBRARY_PICTS	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	15/07/2019	87	4	3	0	28
LOO_CreateLine	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	09/09/2019	63	5	18	0	9
MAC_SAY_TEXT	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	16/05/2018	33	2	14	0	4
ME_CopyOptions	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	28/02/2019	40	6	10	0	5
ME_HandleLB_Attributes	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	08/03/2019	45	16	29	0	10
METHOD_ANALYSE_TD_COLL	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019	83	7	13	2	12
METHOD_check_deprecated	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019	34	5	12	1	5
METHOD_check_events	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019	34	5	12	1	5
METHOD_check_if_False	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	08/09/2019	53	9	16	1	16
METHOD_check_OpenClose	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	09/09/2019	148	30	48	1	7
METHOD_check_setName	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	09/09/2019	120	5	14	1	23
METHOD_codeMinor_display	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	02/04/2019	6	0	0	8	1
METHOD_codeMinor_setContent	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	02/04/2019	7	0	0	5	3
METHOD_GetCleanCode	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019	42	7	11	3	6
METHOD_getVariables	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019	154	13	14	2	36
METHOD_LH_clicked	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019	327	6	9	1	62
METHOD_simplifyCodeText	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019	209	20	26	3	18
METHODS_AllCreateNeighboringDns	<input checked="" type="checkbox"/>	Indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	09/09/2019	83	2	8	1	11

This is the basic module of the component for method management.

This module allows you to, among other things:

- View and modify the parameters of your methods
- To visualize the statistics carried out on the methods
- To visualize the method
- To edit the comments of the method
- Export the catalogue of your methods in HTML format

### Buttons bar

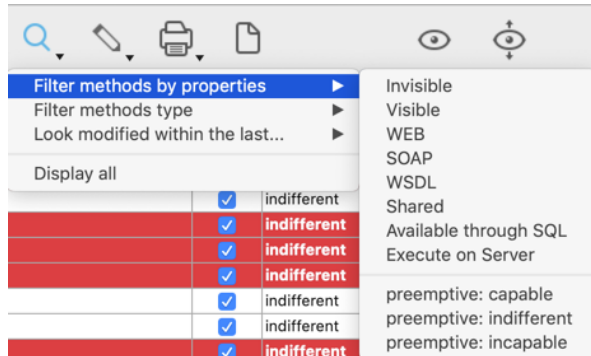
#### Search area

Allows you to reduce the list of displayed methods by searching on the name and content of the method, if the option is enabled.



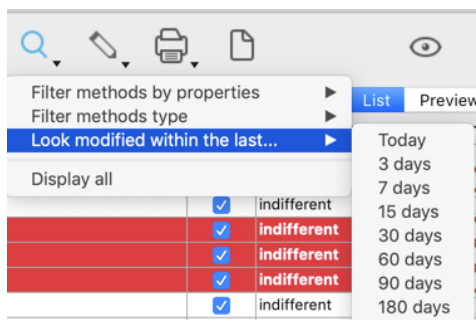
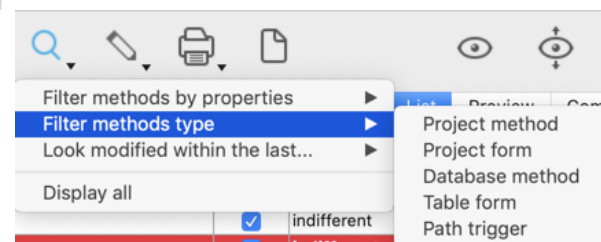
## The magnifying glass

The button bar associated with this module allows you to quickly make selections:



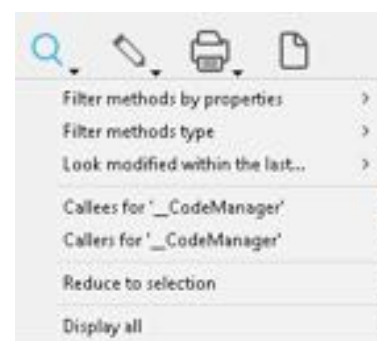
Selection based on method properties

## Selection based on the nature of the method



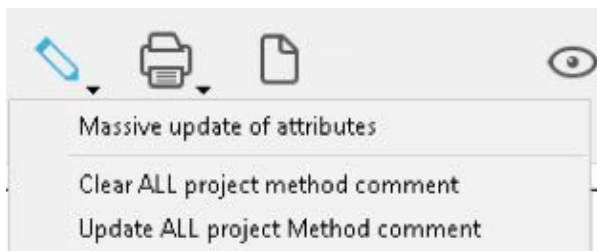
Selection based on the age of the methods

Note that if a method is selected, access to "Called methods" and "Calling methods" is possible. In this case, the content of the list is updated.



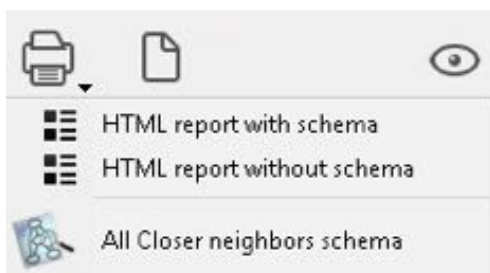
## The pen

The pen allows you to make a generalized modification of the properties of the selected methods or to act directly on the comments of the methods.



- Deleting comments from all methods
- Integration of comments located at the beginning of the code as method comments. By default, only the first 10 lines are taken into account (adjustable option in the Setup menu).

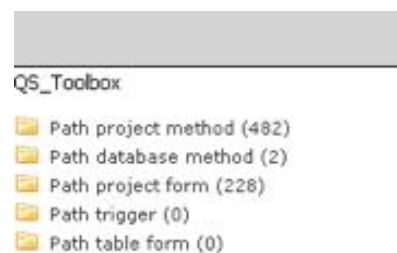
## The printer



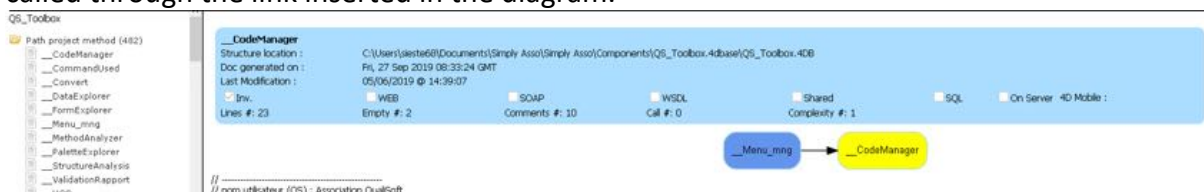
Allows the HTML export of the catalog of methods with or without schema generated using Graphviz that must be installed on your computer.

At the end of the export, you will be able to navigate through the methods between them.

The export creates a menu and shows you the number of methods available depending on their location.



Clicking on the name of the method displays the information about it and the callers and called through the link inserted in the diagram.



## The page

Allows you to copy the list in memory in order to paste it into a spreadsheet.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Method list from QS_Toolbox: 27/09/2019 @ 10:59:28													
2														
3	Method name	Attr. invisibl	Attr. On Serv	Attr. Shared	Attr. Web	Attr. SOAP	Attr. WSDL	Attr. SQL	Last Update	Nb lines	Nb empty	Nb Commen	Nb Nested	Complexity
4														
5	_Menu_mng	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	218	9	12	16	54
6	_Start	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	47	5	12	1	3
7	FE_analyseElementClicked	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	26/09/2019	224	5	8	2	35
8	QS_PARAM_AFFECTATION	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	22	2	10	2	1
9	QS_PARAM_SAVE	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	49	6	10	1	5
10	QST_DB_Setup	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	91	8	11	3	8
11	[projectForm]/_FormExplorer/myLH								26/09/2019	53	2	1	0	13
12	[projectForm]/_Setup/RestoreSetup								25/09/2019	4	0	0	0	1

## Eyes, cursor and folder

The first eye is used to reduce the displayed list to the manual selection made in the method list. This allows you to compare properties of methods that must share the same methods more quickly.

The second eye is used to retrieve the list of methods.

The cursor allows you to reduce the list of methods based on the date they were modified. This makes it possible to identify very quickly the last methods affected.

The folder provides quick access to the information stored on the local hard disk:

- Windows: 'drive':\my documents\QS\_Toolbox\<base name>
- Mac: /documents/QS\_Toolbox/<base name>

## Statistics' interpretation

Colour codes can be used to draw your attention to different elements:

# lines	# empty	# //	# Direct call	Complexity
87	4	3	0	28
63	5	18	0	9
33	2	14	0	4
40	6	10	0	5
45	16	29	0	10
83	7	13	2	12
34	5	12	1	5
34	5	12	1	5
53	9	16	1	16
148	30	48	1	7
120	5	14	1	23
6	0	0	8	1
7	0	0	5	3
42	7	11	3	6
154	13	14	2	36
327	6	9	1	62
209	20	26	3	18
83	2	8	1	11

The statistics correspond to:


- Total number of lines per method
- Number of empty lines
- Number of comment lines
- Number of method calls
- Relative complexity of the method

A cell highlighted by a color quickly allows you to visualize if a verification action is to be considered on your method. 3 color levels have been defined: yellow, pink and red.

Invisible	Preemptive	on Ser...	Shared	WEB	SOAP	WSDL	SQL	Last Update
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	15/07/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	09/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	16/05/2018
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	28/02/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	08/03/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019
<input type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	08/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	09/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	09/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	02/04/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	02/04/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23/09/2019
<input checked="" type="checkbox"/>	indifferent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	09/09/2019

The method properties are accessible and can be modified directly from the graphical interface.

A simple click allows you to modify the desired property.

In case of modification, the icon  appears in the top left corner of the module to save the modifications.

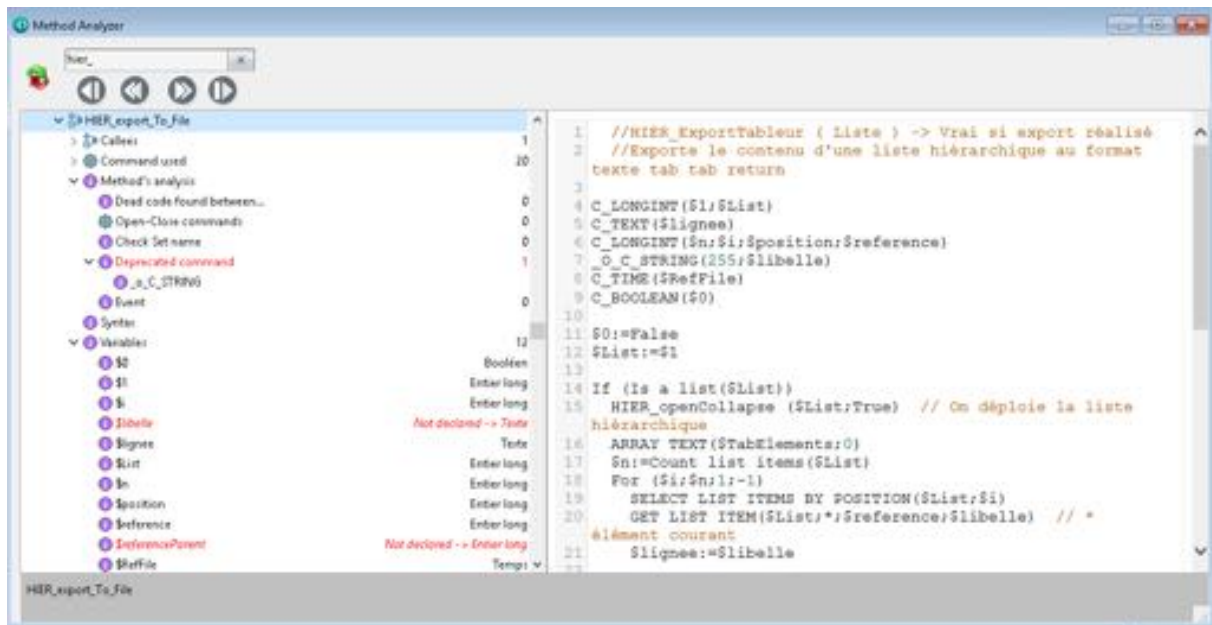
A *green line* indicates that the 'Preemptive' property of the method can be set to 'capable'. This indication is only available if the typing file was generated during compilation.



Similarly, a line not highlighted in green and whose "Preemptive" property is "capable" must be changed to "indifferent" unless you correct the code and restart the compilation for verification.

A *red line* indicates that the method is "useless" because the sum of the empty and commented lines is equal to the total number of lines of code of the method.

## Method Analyzer



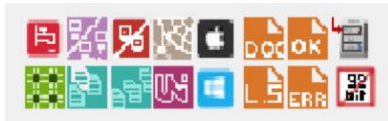
Second module on methods. The latter lists your methods using a hierarchical list, much like the method explorer, with the following differences:

- You have access to calling and called methods directly from the interface
- You have access to the list of commands used in the code
- You have access to the list of constants used in the code
- An analysis of your method is carried out in order to highlight points of attention such as the presence of dead code, the verification of the names of assemblies, the use of "end-of-life" commands...
- You have access to the syntax of your method\*
- You have access to the list of variables of your method and their declaration. In case of non-reporting, QS\_Toolbox offers you the type to report. \*

\* If the generation of the symbol file was performed during compilation, the information presented by QS\_Toolbox is more relevant.

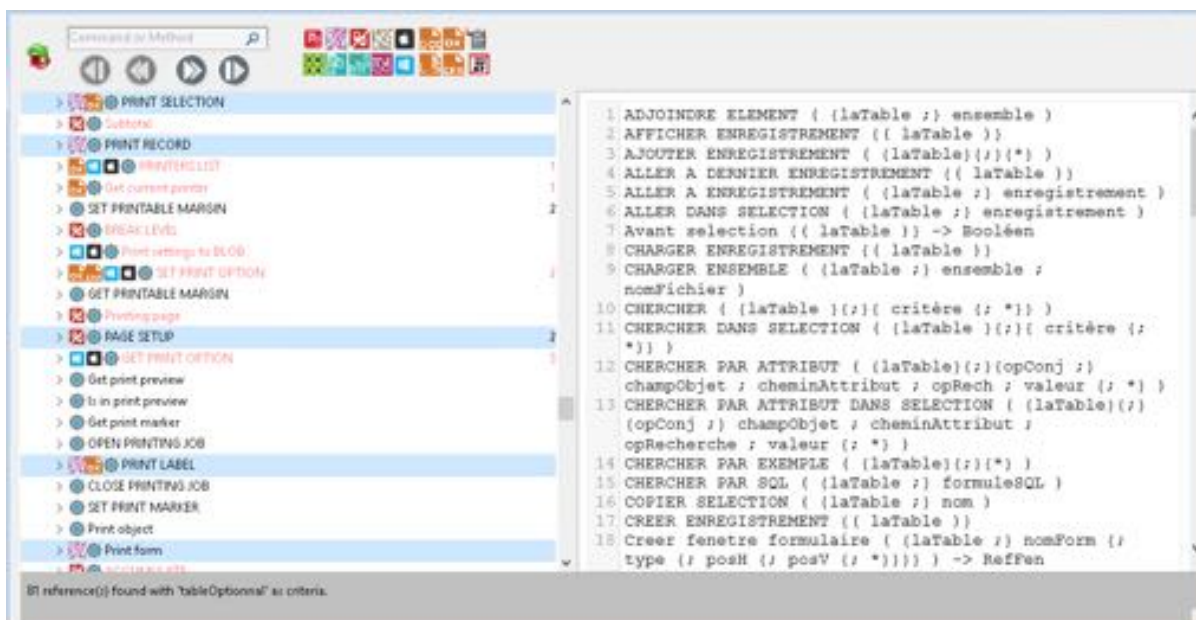
## 4D Command

This module allows you to list the methods used through the 4D command catalog, but also to list commands with a behavior such as a different client-server operation, those that modify the OK variable, ...



The buttons are "identical" to those found in 4D's documentation (<https://doc.4d.com/> or <https://livedoc.4d.com/>)

Clicking on one of the buttons displays the relevant commands and selects them for quick navigation between them.



Example of selecting commands whose table specification is optional when using "Default table".

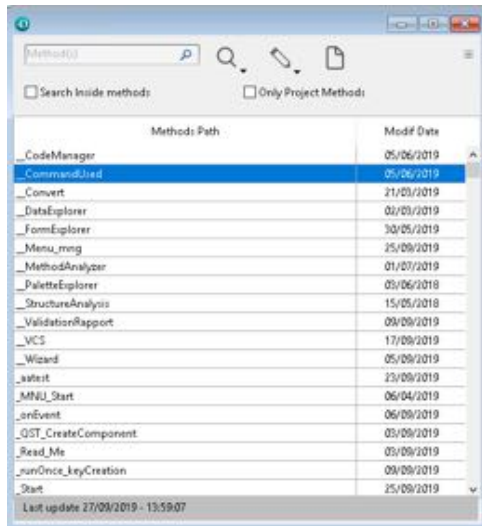


Selecting a command allows you to quickly know the number of methods using it and to extract:

- His token
- Its compatibility to be Thread-safe
- Its syntax
- The list of methods using this command, the code of which is displayed in case of selection




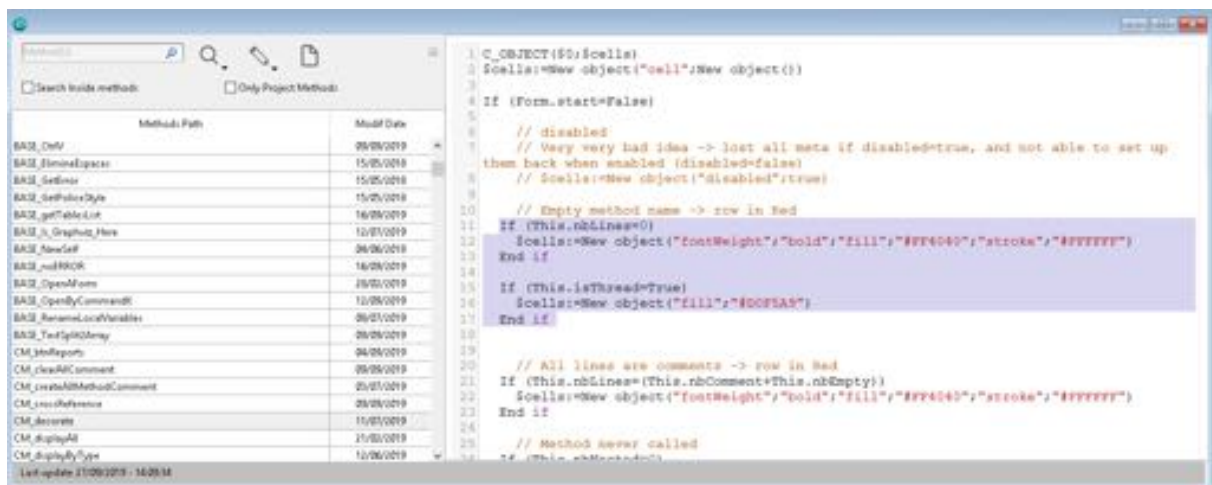
## Methods Pallet



This module allows you to have the list of methods on hand to view or copy all or part of the code.

In addition to the selection tools identical to those in the Method overview module, you can search the code for project methods only, if the option is selected.

Clicking on it  displays the right panel where you will find the code of the selected method and from which you can extract a portion of the code before copying it.



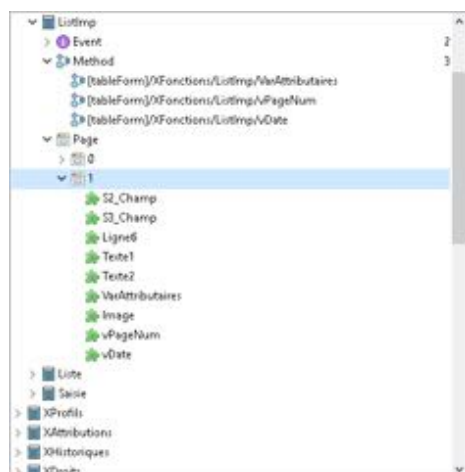
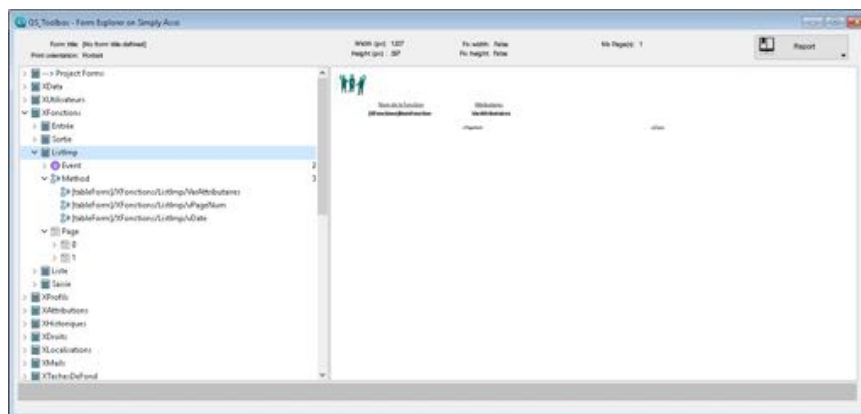
## Form Explorer

This module allows you to view a set of characteristics of your forms in a different way than 4D's Form Explorer.

The top banner displays the main attributes of the form, such as the name of the form, heights and widths, orientation and number of pages.

Form title: {No form title defined}	Width (px): 802	Fix width: False	Nb Page(s): 5
Print orientation: Portrait	Height (px): 679	Fix height: False	

Selecting a form from the list allows you to retrieve, in real time, the events checked for that form, the list of widgets containing a method and the number of pages used in that form.

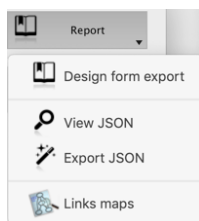
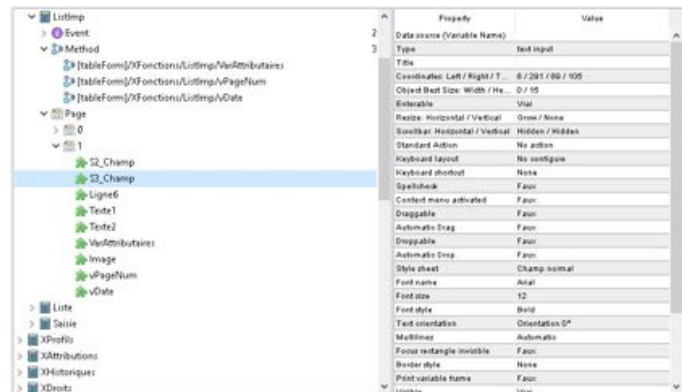


Clicking on the form name displays the screenshot of page 1 of the form.

Clicking on a page number not only displays the screen copy of the content of that page, but also lists the list of widgets that make up the page.



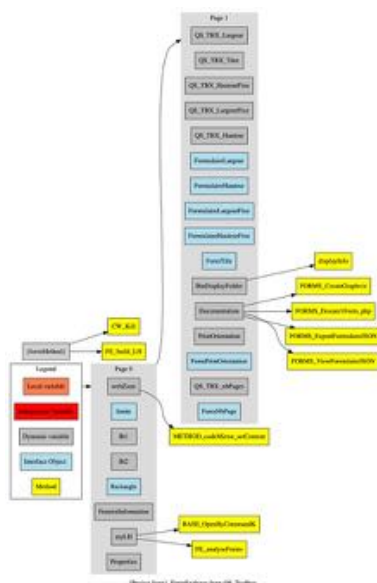
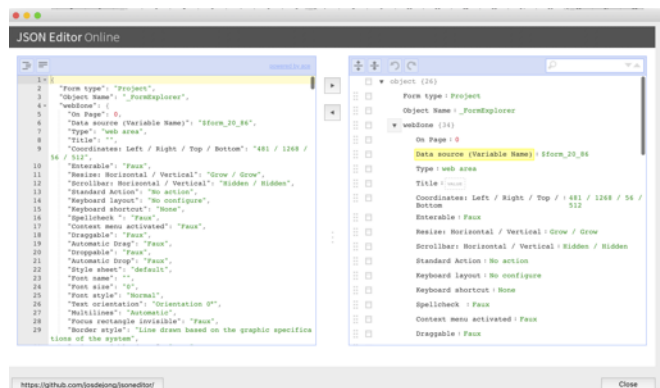
Selecting a widget allows you to display all the properties of the same widget.  
The list consists of all the properties available using the 4D commands.



The 'Reports' button allows you to retrieve all the information in the form in the form:

- From an export to DOCX format
- An export in JSON format
- An export of a graphical representation of widgets, page by page
- The map of the form links

The visualization of the parameters of a form and its widgets in JSON format is available directly from QS\_Toolbox.



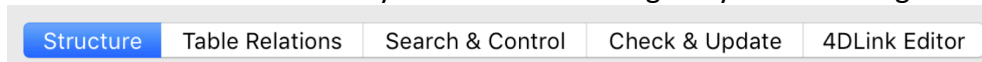
The link map is an interesting tool.  
It highlights the links between the widgets on the different pages of the form as well as the links with the project methods called from the widgets in a graphical form.

## Structure Analysis

This module corresponds to the toolbox of QS\_Toolbox. Indeed, it allows you to:

- View all field and table attributes
- View the definition of links
- To check, to look for what is wrong with your code.
- To execute SQL or 4D code under your sole responsibility
- Edit 4Dlink links

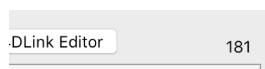
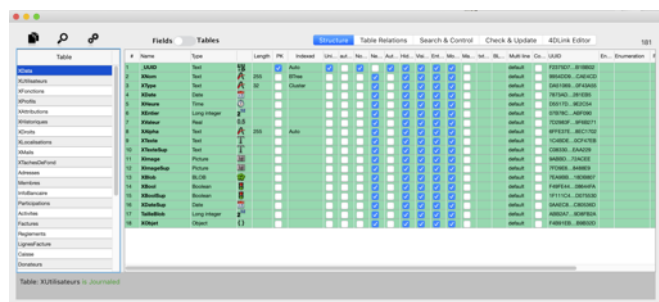
The transition from one functionality to another is managed by the following selection bar:



## Structure

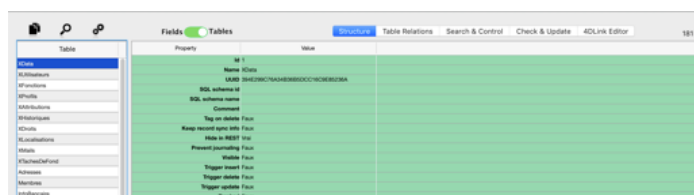
Display table & field definition data

When selecting a table, the list of fields that make it up is displayed. The background color is the one defined in the structure editor.



At the top right, QS\_Toolbox indicates the number of records contained in this table.

When the selector is set to Tables, then the information about the Table is displayed.



## Specific options

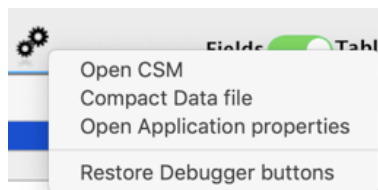
The specific options are available through the following icons:

- Copying information
- Displaying or exporting the structure schema
- Database management tools



Allows you to copy the definition of the fields of the selection table, all tables and all fields as well as to store the code for adding a record using the 4D code.

Display of the database structure and export of the structure image (PNG format).



The last icon allows you to launch the CSM dialogs, the application properties, compact the data file by command line or restore the debugger buttons when they have disappeared.

## Table Relations

This feature allows you to view all the links defined in the database structure. It is not possible to define a link from this dialog. You must open the structure editor and add / modify the link from there.

Select a relation: ProfilsToFunction - FunctionToProfils

Name for N to 1: ProfilsToFunction

Name for 1 to N: FunctionToProfils

UUID: E2BF582DB94244ECA6613339EC7C0A93

Integrity: None

Choice field: 0

Automatic load N to 1 ☐

Automatic load 1 to N ☐

Auto Wildcard support ☐

Prompt if related one does not exist ☐

Auto assign related value in subform ☐

Kind	Table name	Field name
source	XProfils	UUIDFunction
destination	XFunctions	PK_UUID

## Search & Control

This is the feature of QS\_Toolbox that will allow you to perform various and varied searches on your methods

The specific options are available through the following icons:

- Copy of the result of the last search in the clipboard.
- The magnifying glass allows you to search, display tips, extract data ....
  - As in the previous menu, the wheels allow you to launch the CSM dialogs, the application properties, compact the data file by command line or export the image library.



The available menus are:

### Check

- Check names of tables, fields and methods
- Command requiring to be "closed" after a call
- Declaration of the names of the assemblies

### Search For

- Impaired orders
- Images referenced in the library
- Big tables
- A specific event
- A specific widget
- Improvements for use with 4D Server
- The use of data formatting
- the presence of dead code such as 'if (1=2)' ; 'if (false)'...

### TIPS

- View all tips and tricks
- Show tips after analyzing project methods

### Analyse

- Comparison of variable declarations

### List

- Lists all objects in the database
- database parameters (SET DATABASE PARAMETER)
- lists the information of the WEB server
- lists the WEB server options

### Get

- lists the indexes
- lists the primary keys
- lists the style sheets
- lists plug-ins
- lists the components

- copies the path of the application to memory
- copies the path of the data file to memory
- retrieves the external IP
- lists license information

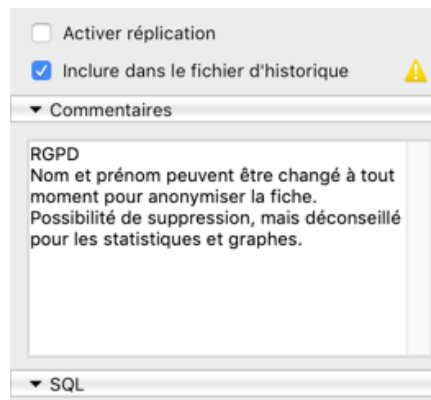
### Copy system configuration

Copies all system information to memory:


- system configuration
- volume information
- network information
- printer list
- CODECS list
- component list
- plug-in list
- table fragmentation
- cache information
- ...

### Get GRPD information

Retrieves information placed in the 'Comments' area and identified by GPDR



☐ Activer réplication

☒ Inclure dans le fichier d'historique 

▼ Commentaires

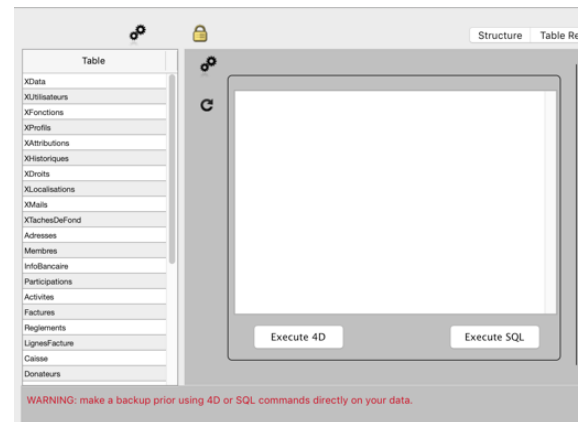
RGPD  
Nom et prénom peuvent être changé à tout moment pour anonymiser la fiche.  
Possibilité de suppression, mais déconseillé pour les statistiques et graphes.

▼ SQL

## Check & Update

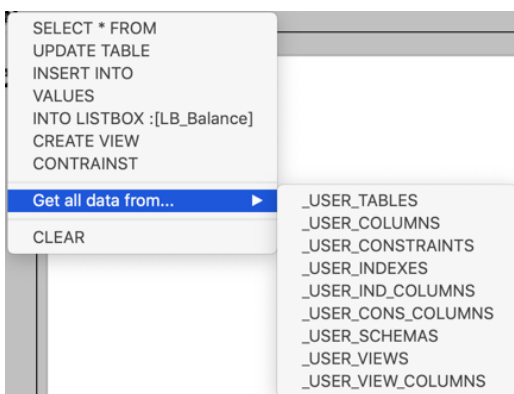
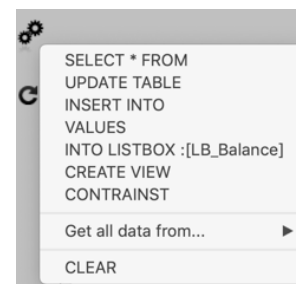
The features in this section may be dangerous for your data if the code executed makes a change to it.  
An input field is available and allows you to execute either 4D code or SQL code directly from QS\_Toolbox.

No verification of the entry is performed by the component.



A double click on the name of a table allows you to add the name of this table in the editor


To help you build a SQL query, the main usable keywords are available from the wheel icon.  
The 'CLEAR' line allows you to delete any entry from the field.



The menu "This all data from..." allows you to obtain a list of the information of the system tables that are read-only.

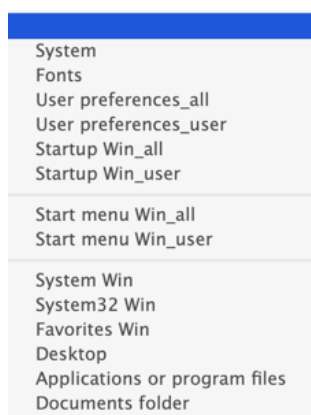
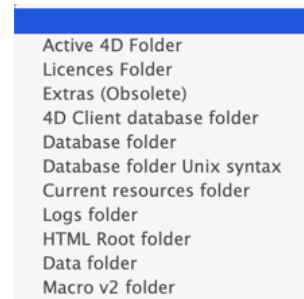
The result of a SQL query of type 'SELECT' is displayed in a listbox.

IN DE X ID	IN DE X NAME	IN DE X TYPE	IN DE X ORD	TABLE NAME	UNIQUE NAME	TABLE ID	IN DE X ID
1	SPRINGER	TABLE	1	SPRINGER	SPRINGER	1	1
2	SPRINGER	TABLE	2	SPRINGER	SPRINGER	2	2
3	SPRINGER	TABLE	3	SPRINGER	SPRINGER	3	3
4	SPRINGER	TABLE	4	SPRINGER	SPRINGER	4	4
5	SPRINGER	TABLE	5	SPRINGER	SPRINGER	5	5
6	SPRINGER	TABLE	6	SPRINGER	SPRINGER	6	6
7	SPRINGER	TABLE	7	SPRINGER	SPRINGER	7	7
8	SPRINGER	TABLE	8	SPRINGER	SPRINGER	8	8
9	SPRINGER	TABLE	9	SPRINGER	SPRINGER	9	9
10	SPRINGER	TABLE	10	SPRINGER	SPRINGER	10	10
11	SPRINGER	TABLE	11	SPRINGER	SPRINGER	11	11
12	SPRINGER	TABLE	12	SPRINGER	SPRINGER	12	12
13	SPRINGER	TABLE	13	SPRINGER	SPRINGER	13	13
14	SPRINGER	TABLE	14	SPRINGER	SPRINGER	14	14
15	SPRINGER	TABLE	15	SPRINGER	SPRINGER	15	15
16	SPRINGER	TABLE	16	SPRINGER	SPRINGER	16	16
17	SPRINGER	TABLE	17	SPRINGER	SPRINGER	17	17
18	SPRINGER	TABLE	18	SPRINGER	SPRINGER	18	18
19	SPRINGER	TABLE	19	SPRINGER	SPRINGER	19	19
20	SPRINGER	TABLE	20	SPRINGER	SPRINGER	20	20

The icon  allows you to delete any entry from the field.

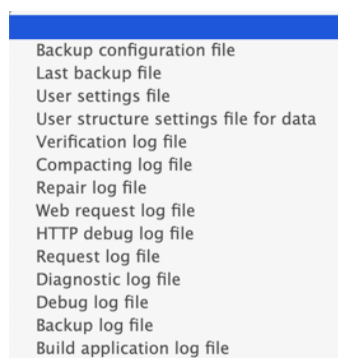
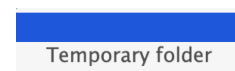
The right part of the Check & Update feature allows you to access the different folders of the system:

Access to all available folders thanks to the "4D Folder" command



Access to all available folders through the 'System folder' command

Access to the temporary folder thanks to the command "Temporary folder".

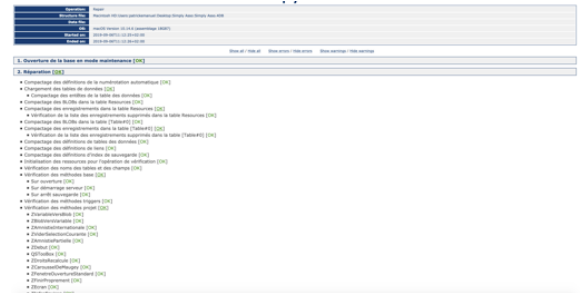


Access to all available folders thanks to the "4D File" command

Open a log file...

- Verification log file
- Compacting log file
- Repair log file
- Build application log file

Open log with consideration of "LogStyleSheet.xsl".

[illegible]

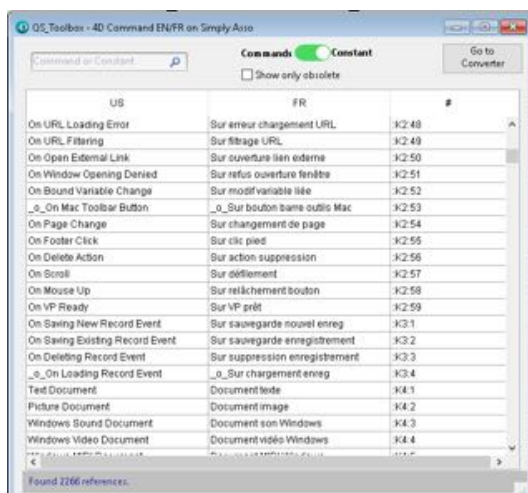
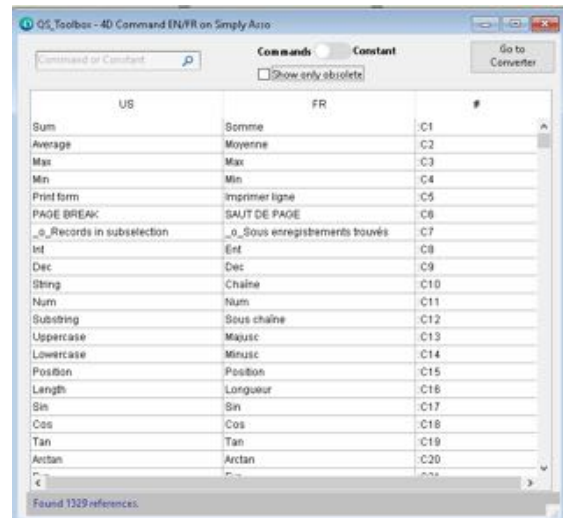
Page 24 sur 30



## Converter EN / FR

This module allows you to have a list of the commands and constants of the current version of 4D.

You can search by part of the name of an order or display only 'obsolete' orders with a single click.



To work with constants, simply switch the selector to the right side.

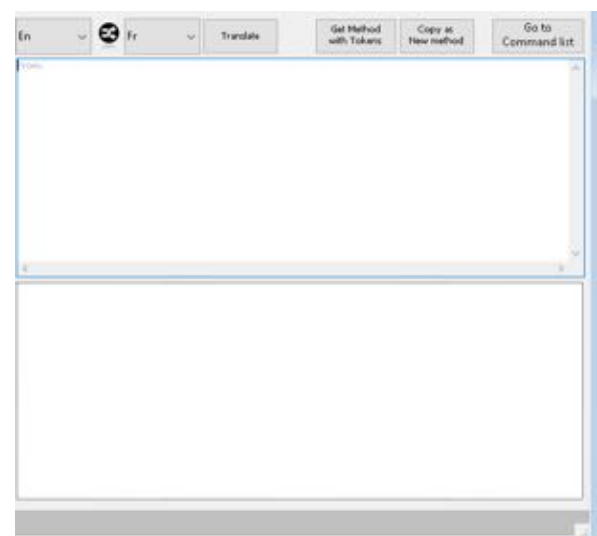
The searches are carried out in the same way as for orders.

To switch to the code translator, simply click on the 'Go to converter' button.

The screen that appears will allow you to enter your code (or paste it) in the From section.

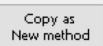
By default, the translation is: EN -> FR.


If you want to do FR -> EN, just click on the icon



Once the code to be translated is present in the upper part, simply click on the "Translate" button to obtain the translated code. Tokens are present in the translated code.



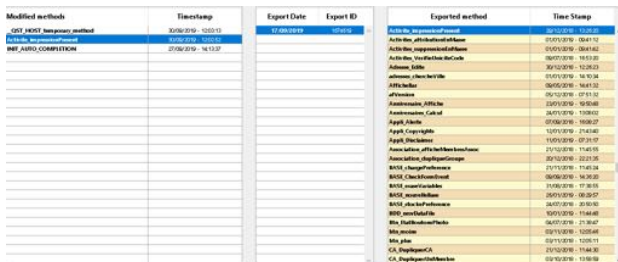
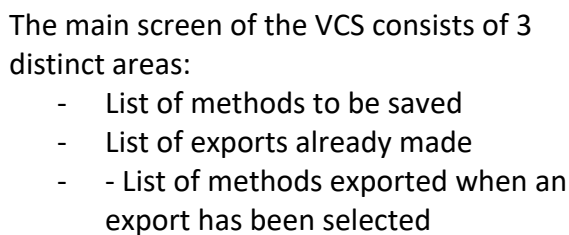
The button  allows you to save the translated code in a new project method. Despite all the controls performed for the translation, it is necessary that you proofread the code to ensure that it does not contain any errors related to the translation.

The button  allows you to read the code of a project method and to display the tokens associated with each command / constant.

This feature allows you to save and compare different versions of your methods, all types combined.

All methods are saved in the external database data file 'QST\_External\_DB'.

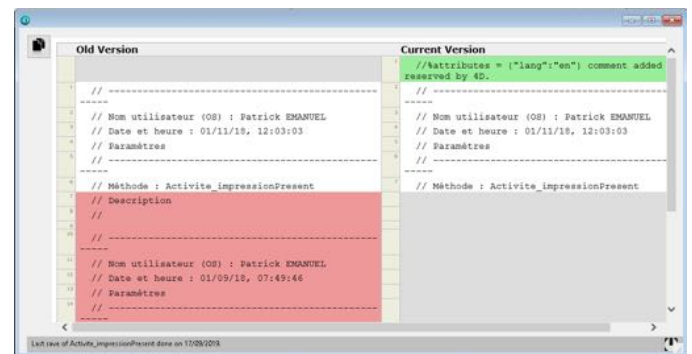
- To start the backup of the methods modified since the last backup
- To compare 2 versions of the same method
- Work on all backups of the current database (project)
- - To work on all backups of all databases (projects) that use the VCS of QS Toolbox




A new method backup will clear the first list and add a new line to the second.

Modified methods	Timestamp	Export Date	Export ID
		30/09/2019	1675518
		17/09/2019	1674619

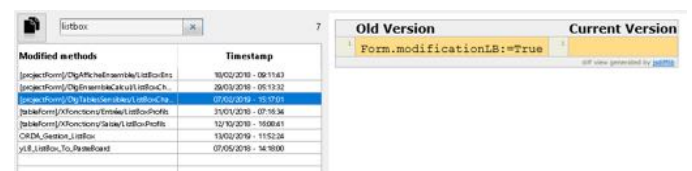
The comparison between the 2 methods uses a Java applet that displays the deletions in red and the additions in green



The icon  allows you to copy the content of OldVersion into memory in order to retrieve it in a text editor or to copy directly into the method editor.

Access to all backups displays all saved methods.

A search box allows you to narrow down the list in order to compare the code of the selected method with that of the current version. The number at the search level corresponds to the number of methods listed. Selecting a method automatically starts the comparison. In the following example, the code has simply been deleted.



Projects	Export Date	Export ID
Q5_Toolbox4DB	22/09/2019	220139
Simply Area4DB	18/09/2019	219900
	18/09/2019	0
	18/09/2019	0
	18/09/2019	0
	18/09/2019	0
	17/09/2019	219617
	28/08/2019	207427
	04/03/2019	0
	26/02/2019	0

Exported method	Time Stamp
about	21/09/2019 - 21:30:01
[projectHome]/VCS/[methodName]	19/09/2019 - 15:50:34
[projectHome]/VCS/AllVersion_0	20/09/2019 - 10:00:30
[projectHome]/VCS/MethodOfVersion	19/09/2019 - 17:37:19
[projectHome]/VCS/MethodPage1	19/09/2019 - 21:04:51
[projectHome]/VCS/Exported_0	20/09/2019 - 10:00:52
[projectHome]/VCS/Exported_01	19/09/2019 - 21:42:28
[projectHome]/VCS/Exported_01	20/09/2019 - 10:05:12
[projectHome]/VCS/Modified_0	19/09/2019 - 13:15:28
[projectHome]/VCS/Project_0	20/09/2019 - 10:02:30
[projectHome]/VCS/SearchPicker	19/09/2019 - 15:57:40
[projectHome]/VCS/SearchPicker1	20/09/2019 - 10:06:30
CRM_getCodeToDisplay	20/09/2019 - 10:49:42
CRM_getCodeToDisplay	20/09/2019 - 10:47:02

Access to all projects displays the list of projects saved in the QS\_Toolbox VCS and allows you to navigate through all the backups performed. This has a twofold benefit:

- Check that the code is the same on projects using "identical" methods
- Copy code from another project without having to open it

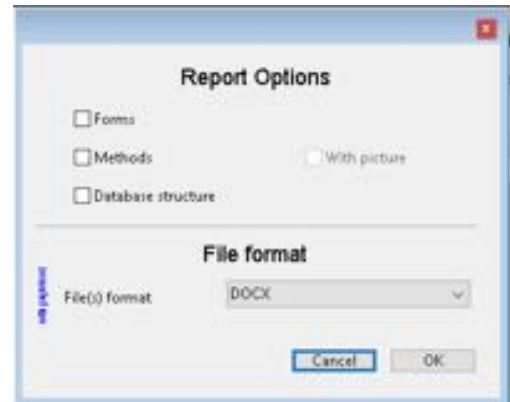
## Reports

The reports created allow the export to text format of all the information concerning:

- The forms
- The methods
- The structure of the database

The export formats are:

- DOCX
- HTML
- RTF
- ODT



Nom	Modifié	Type	Taille
Project Form - _Menu.docx	11/07/2019 13:29	Document Micros...	61 Ka
Path project method - 1.docx	06/07/2019 09:50	Document Micros...	180 Ka
Project Form - _Windows.docx	24/06/2019 15:22	Document Micros...	106 Ka
Project Form - _VCS.docx	24/06/2019 15:22	Document Micros...	256 Ka
Project Form - _StructureAnalysis.docx	24/06/2019 15:22	Document Micros...	604 Ka
Project Form - _Setup.docx	24/06/2019 15:21	Document Micros...	106 Ka
Project Form - _ObsoleteCommands.docx	24/06/2019 15:21	Document Micros...	122 Ka
Project Form - _Methods_Pallet.docx	24/06/2019 15:21	Document Micros...	111 Ka
Project Form - _MethodManager.docx	24/06/2019 15:21	Document Micros...	337 Ka
Project Form - _MethodAnalyzer.docx	24/06/2019 15:20	Document Micros...	85 Ka
Project Form - _LinkCreateStructureFile.d	24/06/2019 15:20	Document Micros...	53 Ka
Project Form - _LinkCreateDataFile.docx	24/06/2019 15:20	Document Micros...	51 Ka
Project Form - _joineditor_dialog.docx	24/06/2019 15:20	Document Micros...	62 Ka
Project Form - _FormExplorer.docx	24/06/2019 15:20	Document Micros...	150 Ka
Project Form - _FormSortie.docx	24/06/2019 15:20	Document Micros...	75 Ka
Project Form - _DataExplorer.docx	24/06/2019 15:20	Document Micros...	58 Ka
Project Form - _dataByOrds_detail.docx	24/06/2019 15:20	Document Micros...	61 Ka
Project Form - _dataByOrds.docx	24/06/2019 15:19	Document Micros...	69 Ka
Project Form - _Copyrights.docx	24/06/2019 15:19	Document Micros...	241 Ka
Project Form - _Converter.docx	24/06/2019 15:19	Document Micros...	127 Ka
Project Form - _Commands.docx	24/06/2019 15:19	Document Micros...	128 Ka

The generated exports can provide a basis for documenting your application's sources. The content includes all the available information on each of the exported elements.

The export of forms systematically includes:

- A screen copy of each page of the form
- A table defining the properties for each of the widgets used

The method export includes the methods of all objects in the database.

The export of the structure includes:

- The image of the database structure.
- The definition of each of the tables
- The definition of each of the fields

## Setup

The different options of the application allow:

- To set the limits used for the definition of the colours used in the Method Overview section
- Set the number of times a file must be tested when it is written to disk. On VERY fast systems, this number must be increased
- To define (in second) the time between 2 modified method checks (only for the VCS).



## New DB Analysis

Restarts an analysis of the database.

During the analysis, the application may be slightly slowed down. This is related to the use of resources.

## Restart 4D

Restarts the application to consider the latest changes applied to the structure such as defining links for use by ORDA.