



QS\_Toolbox

## Table

<b>Introduction.....</b>	<b>3</b>
<b>What is QS_Toolbox .....</b>	<b>3</b>
<b>Demonstration version vs Commercial version .....</b>	<b>3</b>
<b>Component installation .....</b>	<b>3</b>
<b>Information structure in QS_Toolbox .....</b>	<b>4</b>
<b>General functions.....</b>	<b>5</b>
<b>Method management .....</b>	<b>6</b>
<b>Method Overview .....</b>	<b>6</b>
<b>Buttons bar .....</b>	<b>7</b>
Search area .....	7
The magnifying glass.....	7
The pen.....	8
The printer .....	8
The page .....	8
Eyes, cursor and folder .....	9
<b>Method Analyzer .....</b>	<b>11</b>
<b>4D Command.....</b>	<b>12</b>
<b>Methods Pallet .....</b>	<b>13</b>
<b>Form Explorer .....</b>	<b>14</b>
<b>Structure Analysis .....</b>	<b>16</b>
<b>Structure.....</b>	<b>16</b>
Display table & field definition data .....	16
Specific options.....	17
<b>Table Relations.....</b>	<b>17</b>
<b>Search &amp; Control .....</b>	<b>18</b>
<b>Check &amp; Update .....</b>	<b>20</b>
<b>4D Link Editor.....</b>	<b>22</b>
<b>Converter EN / FR .....</b>	<b>23</b>
<b>Version Control System .....</b>	<b>25</b>
<b>Reports.....</b>	<b>27</b>
<b>Setup.....</b>	<b>28</b>
<b>New DB Analysis.....</b>	<b>28</b>
<b>Restart 4D .....</b>	<b>28</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.







Each time the component is launched, it connects to our servers to check the validity of the license. Aware that you may be on the move and without a temporary connection, the latter accepts 10 offline launches before "blocking" until a new connection is detected.

### Component installation

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

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>

-  All Project Methods
-  Export
-  Export JSON
-  Export\_HTML
-  Forms
-  Report Validation

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 the event 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:

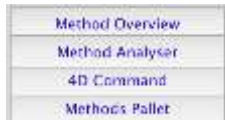
- Storage of system information
- Screen copy automatically saved in the Log folder of the database
- Access to the site of the "bugs list" of the major version of 4D in progress
- The 'About' of the component.

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.

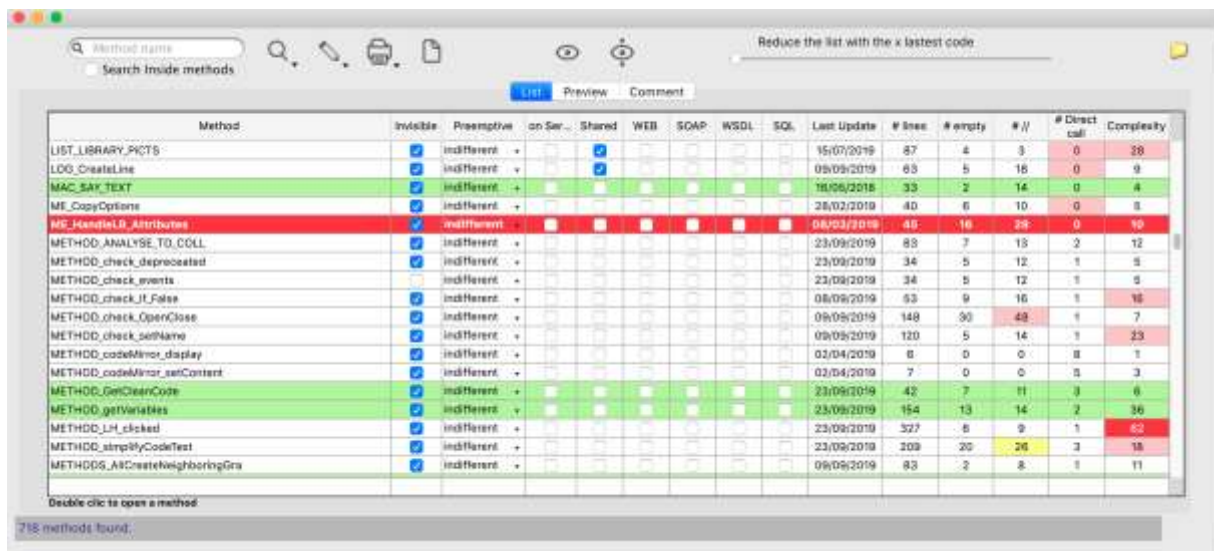
## Method management

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 Analyze: 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	# //	# Direct call	Complexity
LIST_LIBRARY_PICTS	<input checked="" type="checkbox"/>	indifferent		<input checked="" type="checkbox"/>					15/07/2019	87	4	3	0	28
LDG_CreateLine	<input checked="" type="checkbox"/>	indifferent		<input checked="" type="checkbox"/>					09/09/2019	65	5	16	0	9
MAC_SAX_TEXT	<input checked="" type="checkbox"/>	indifferent							18/08/2018	33	2	14	0	4
ME_CopyOptions	<input checked="" type="checkbox"/>	indifferent							28/02/2019	40	6	10	0	8
MC_HandleLib_Attributes	<input checked="" type="checkbox"/>	indifferent							08/03/2018	48	16	28	0	10
METHDD_ANALYSE_TO_COLL	<input checked="" type="checkbox"/>	indifferent							23/09/2019	89	7	13	2	12
METHDD_check_deprecated	<input checked="" type="checkbox"/>	indifferent							23/09/2019	34	5	12	1	5
METHDD_check_events	<input checked="" type="checkbox"/>	indifferent							23/09/2019	34	5	12	1	5
METHDD_check_JIT_False	<input checked="" type="checkbox"/>	indifferent							08/09/2019	53	9	16	1	16
METHDD_check_OpenClose	<input checked="" type="checkbox"/>	indifferent							09/09/2019	148	90	48	1	7
METHDD_check_setName	<input checked="" type="checkbox"/>	indifferent							09/09/2019	120	5	14	1	23
METHDD_codeMinor_display	<input checked="" type="checkbox"/>	indifferent							02/04/2019	8	0	0	0	1
METHDD_codeMinor_setContext	<input checked="" type="checkbox"/>	indifferent							02/04/2019	7	0	0	0	3
METHDD_GetCleanCode	<input checked="" type="checkbox"/>	indifferent							23/09/2019	42	7	11	3	6
METHDD_getVariables	<input checked="" type="checkbox"/>	indifferent							23/09/2019	154	13	14	2	36
METHDD_LH_checked	<input checked="" type="checkbox"/>	indifferent							23/09/2019	327	6	9	1	62
METHDD_stripByCodeTest	<input checked="" type="checkbox"/>	indifferent							23/09/2019	209	20	26	3	18
METHDDS_ArCreateNeighboringGra	<input checked="" type="checkbox"/>	indifferent							09/09/2019	83	2	8	1	11

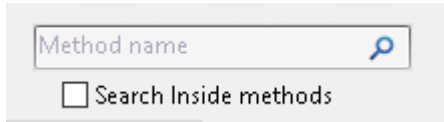
This is the basic module of the component for method management.

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

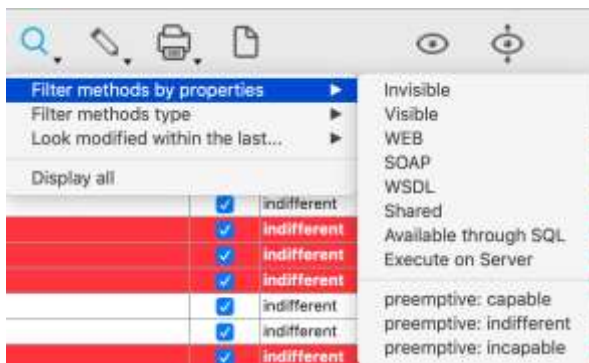
### Search area



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

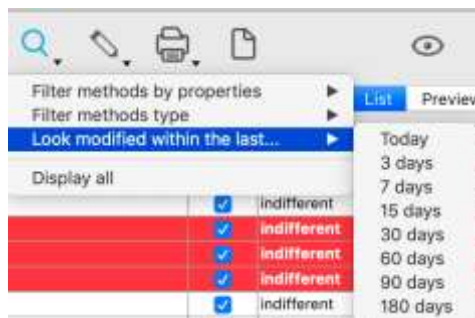
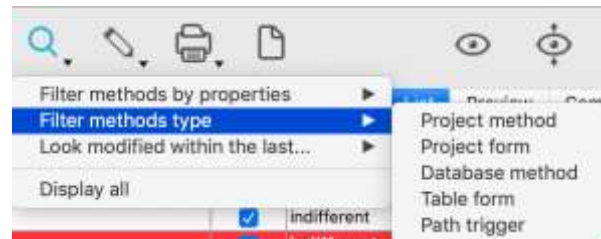
### The magnifying glass

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



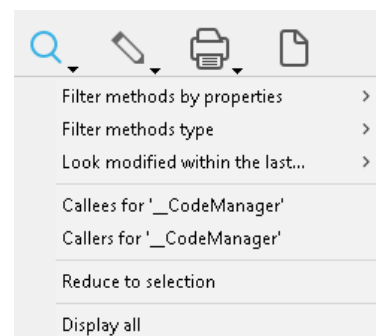
Selection based on method properties

Selection based on the nature of the method



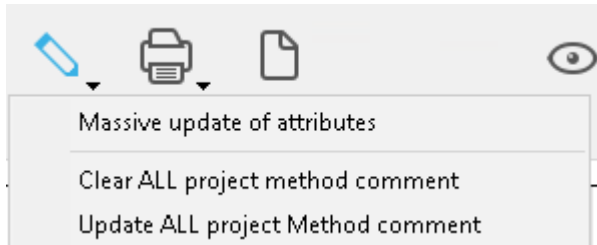
Selection based on the age of the methods

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.



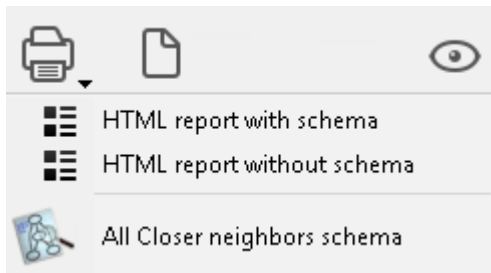
### The pen

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.



- QS\_Toolbox
- Path project method (482)
  - Path database method (2)
  - Path project form (228)
  - Path trigger (0)
  - Path table form (0)

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.





Method name	Attr. Invisibil	Attr. On Sens	Attr. Shared	Attr. Web	Attr. SOAP	Attr. WSDL	Attr. SQL	Last Update	Nb lines	Nb empty	Nb Commen	Nb Nested	Complexity
_Menu_mng	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	218	9	12	16	54
_Start	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	47	5	12	1	3
FE_analyseElementClicked	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	24/09/2019	224	5	8	2	35
QS_PARAM_AFFECTATION	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	22	2	10	2	1
QS_PARAM_SAVE	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	49	6	10	1	5
QST_DB_SetUp	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	FAUX	25/09/2019	32	6	11	3	8
[projectForm]/_FormExplorer/myLH								24/09/2019	53	2	1	0	13
[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	8	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 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 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"/>	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"/>	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.

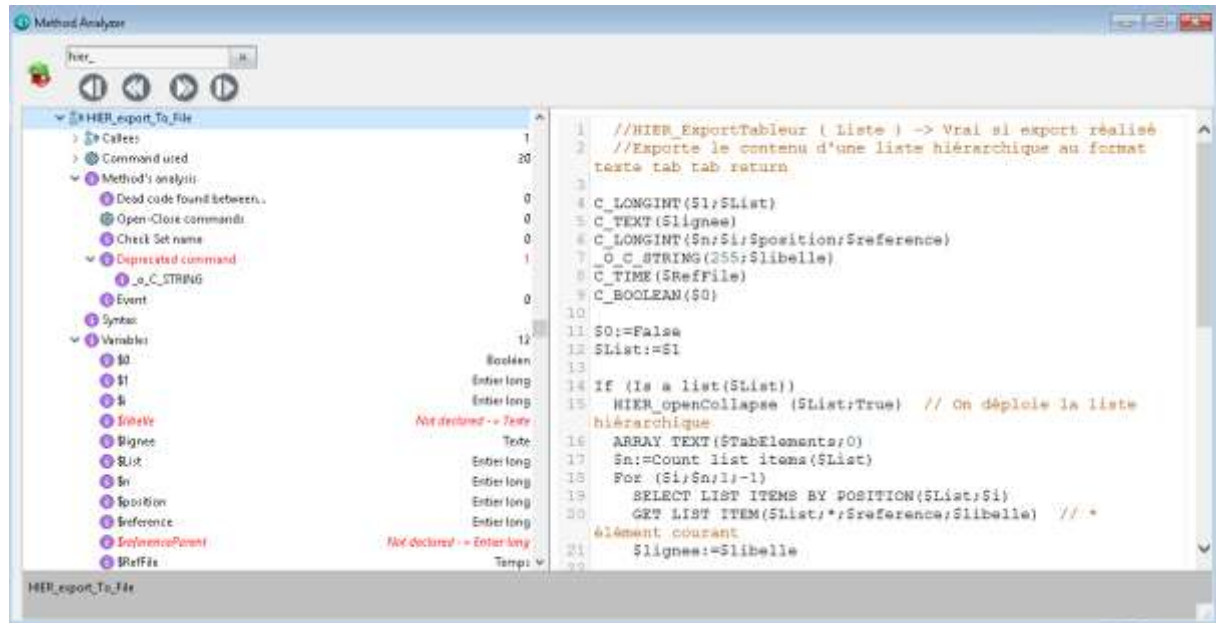
A simple click allows you to modify the desired property.

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



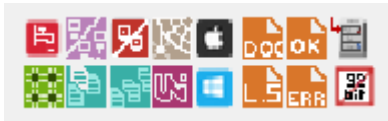
Second module on methods. The latter lists your methods using a hierarchical list, much like the method explorer, with the following differences:

- 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.

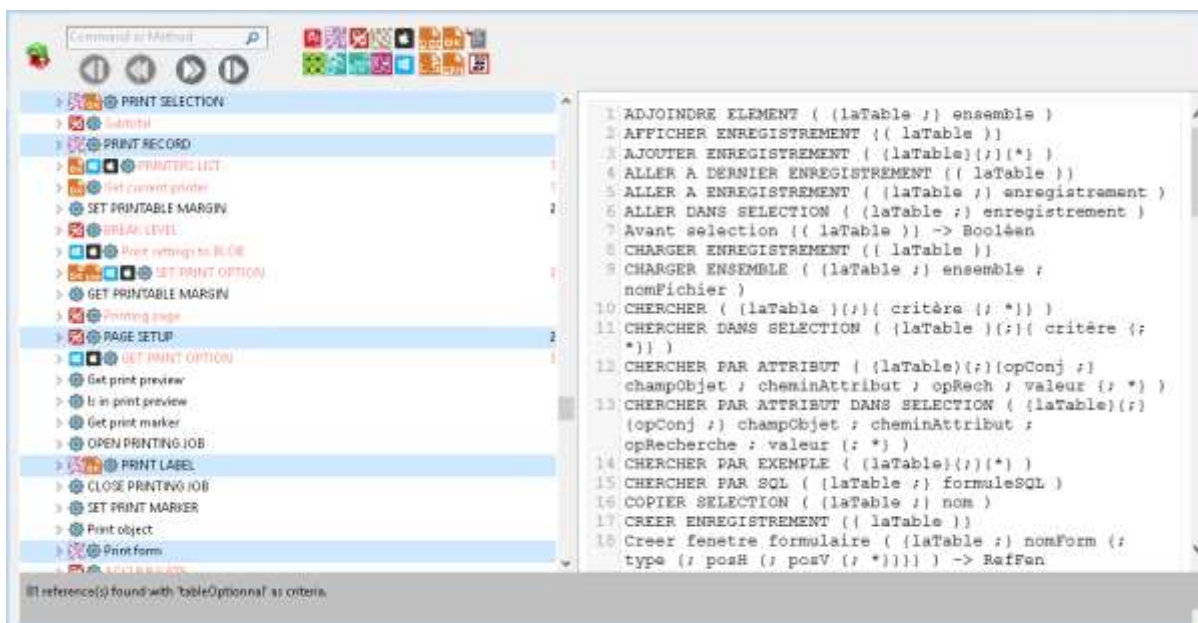
## 4D Command

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, ...

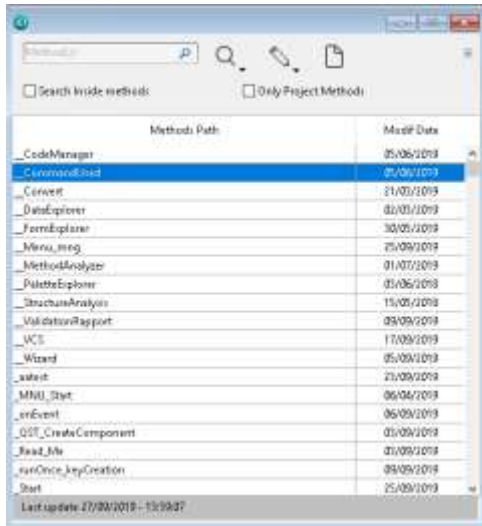


The buttons are "identical" to those found in 4D's documentation (<https://doc.4d.com/> or <https://livedoc.4d.com/>)

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




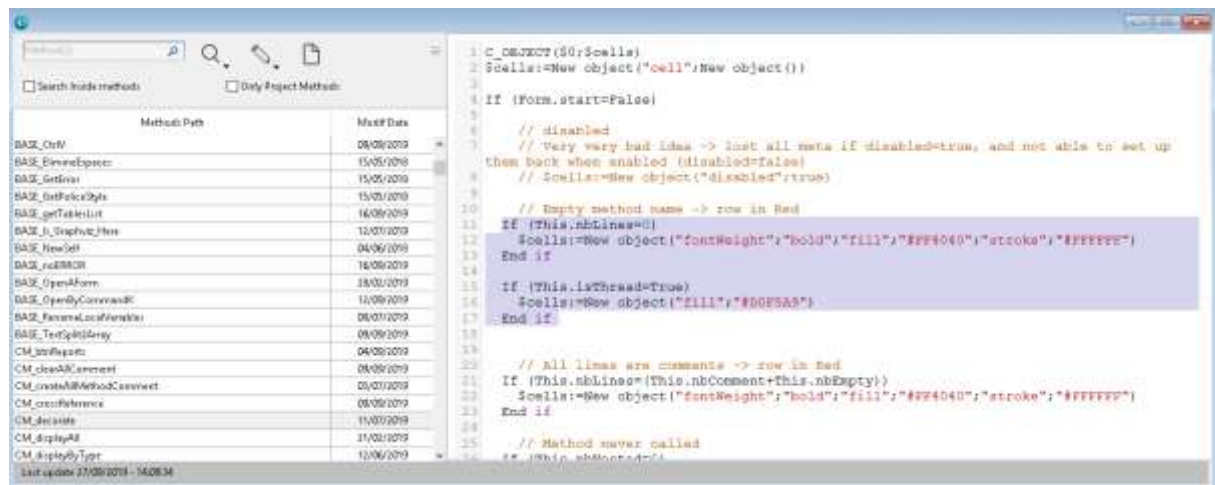
## Methods Pallet



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.

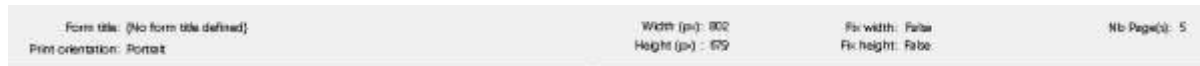
Clicking on it  displays the right panel where you will find the code of the selected method and from which you can extract a portion of the code before copying it.



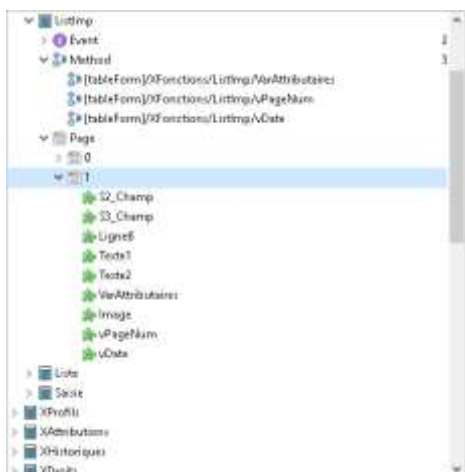
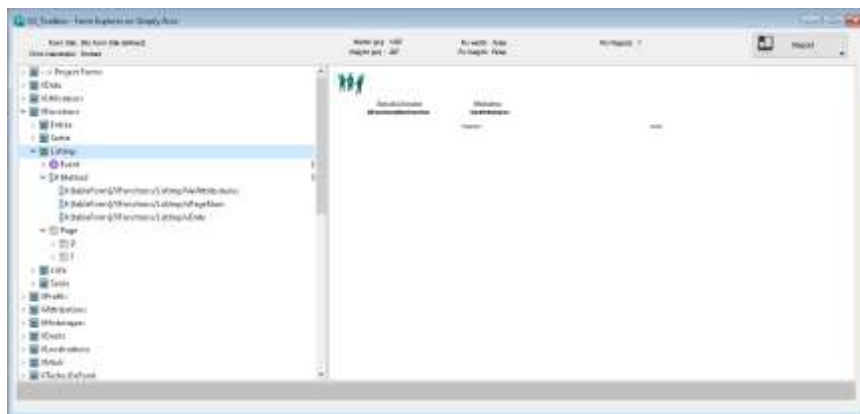
## Form Explorer

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.



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.

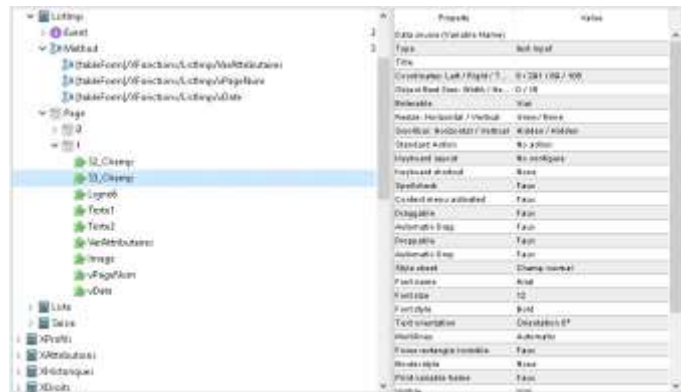


Clicking on the form name displays the screenshot of page 1 of the form.

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.



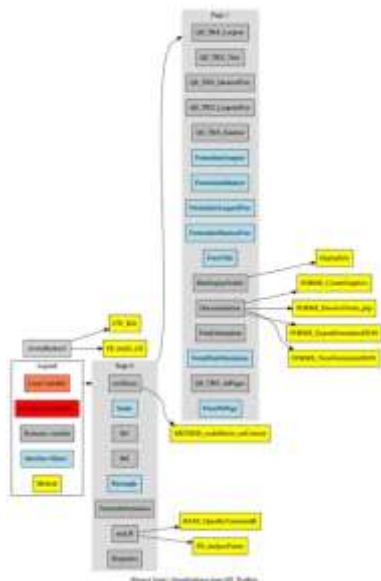
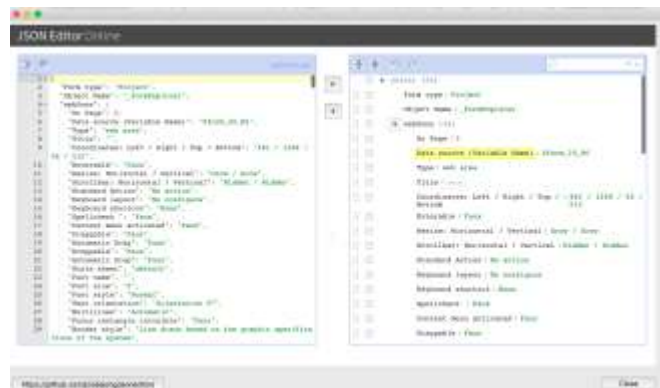
Selecting a widget allows you to display all the properties of the same widget.  
 The list consists of all the properties available using the 4D commands.



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.

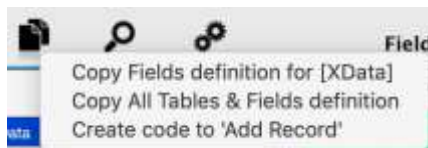




## Specific options

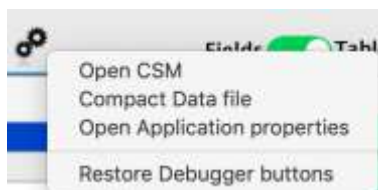
The specific options are available through the following icons:

- 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.



## 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

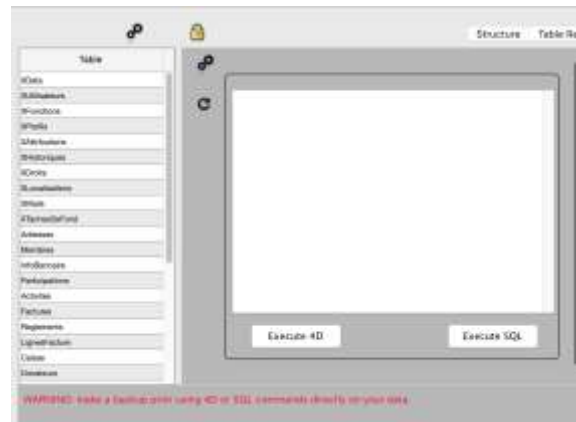
Retrieves information placed in the 'Comments' area and identified by GPDR



## Check & Update

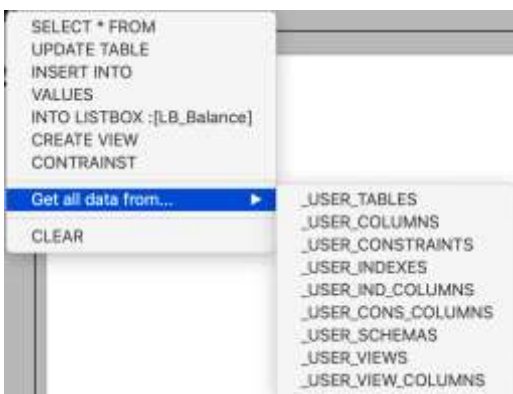
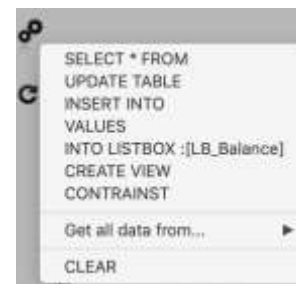
The features in this section may be dangerous for your data if the code executed makes a change to it. An input field is available and allows you to execute either 4D code or SQL code directly from QS\_Toolbox.

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.

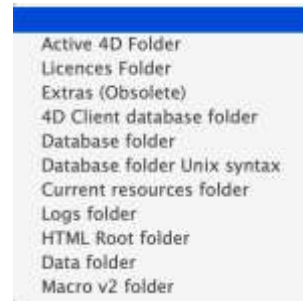
The result of a SQL query of type 'SELECT' is displayed in a listbox.



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

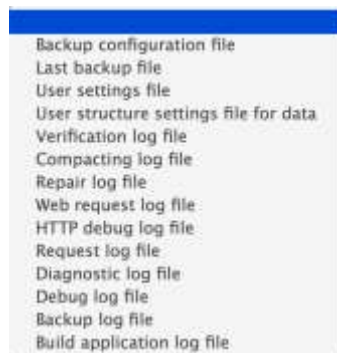
The right part of the Check & Update feature allows you to access the different folders of the system:

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



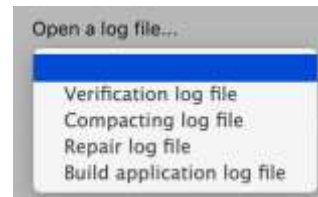
Access to all available folders through the 'System folder' command

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



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

The "Open a log file" part allows you to open a LOG file that can be used by the correct application of "LogStyleSheet.xsl".



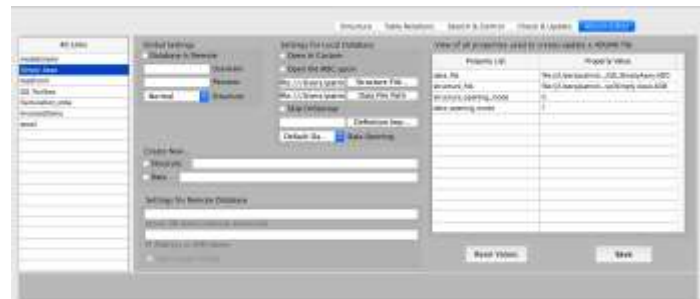
Log opened by double click without the application of "LogStyleSheet.xsl".

Open log with consideration of "LogStyleSheet.xsl".



#### 4D Link Editor

The 4DLink File Editor is a powerful tool because it allows you to create a new 4DLink file manually or modify an existing one.  
 All 4DLink configuration options are available with a single click.

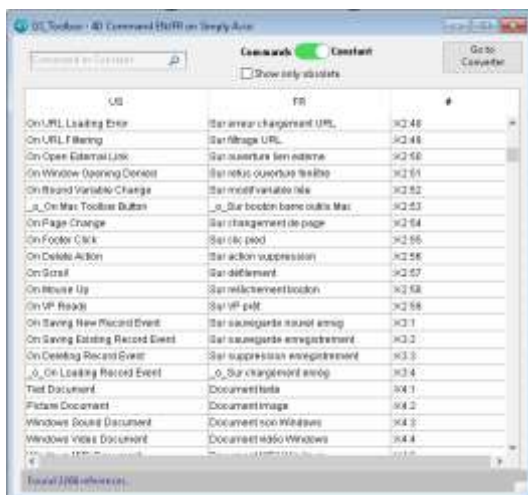
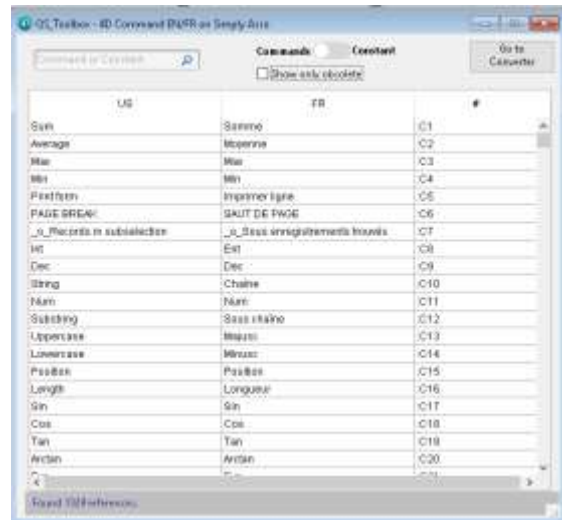


This allows you, for example, to regain control over the start of an application by asking that the next time you open it, the "On Base Opening" method should not be executed.

## 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.

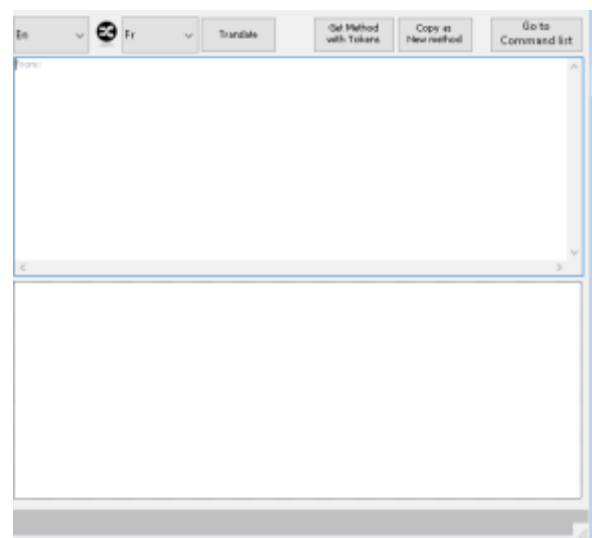
The searches are carried out in the same way as for orders.

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.



Copy as  
New method

The button **Copy as New method** 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.

Get Method  
with Tokens

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



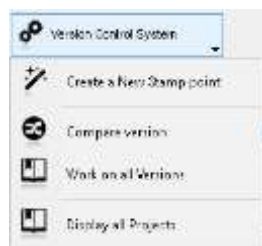
## Version Control System

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'.

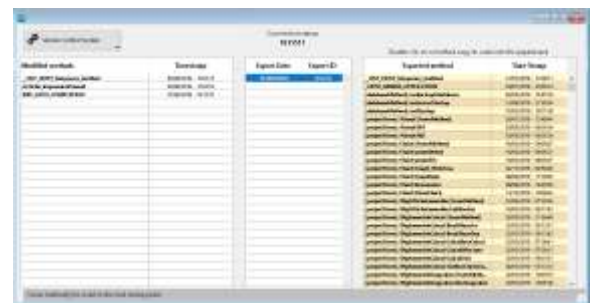
The System Control Version menu allows you to:

- 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



The main screen of the VCS consists of 3 distinct areas:

- List of methods to be saved
- List of exports already made
- - List of methods exported when an export has been selected



Modified method	Timestamp	Export Date	Export ID	Exported method	New Stamp
1001 1001 1001 1001	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1001 1001 1001 1001	101010
1002 1002 1002 1002	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1002 1002 1002 1002	101010
1003 1003 1003 1003	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1003 1003 1003 1003	101010
1004 1004 1004 1004	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1004 1004 1004 1004	101010
1005 1005 1005 1005	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1005 1005 1005 1005	101010
1006 1006 1006 1006	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1006 1006 1006 1006	101010
1007 1007 1007 1007	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1007 1007 1007 1007	101010
1008 1008 1008 1008	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1008 1008 1008 1008	101010
1009 1009 1009 1009	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1009 1009 1009 1009	101010
1010 1010 1010 1010	10/10/2019 10:10:10	10/10/2019 10:10:10	101010	1010 1010 1010 1010	101010

When a click is made on a method that needs to be saved, QS\_Toolbox searches all backups for the previous version of the same method and automatically selects it when it is found.

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

Modified methods	Timestamp	Export Date	Export ID
		30/05/2019	1075510
		17/05/2019	1074012



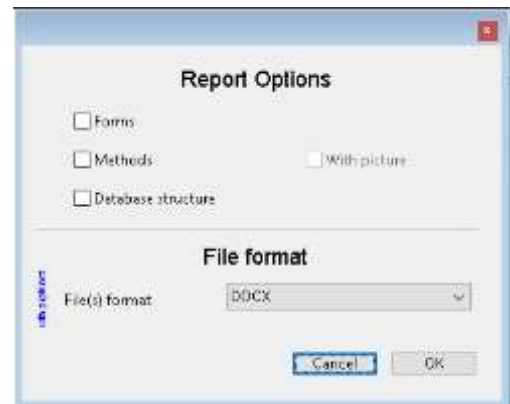
## Reports

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



Item	Modified	Type	Size
Project Form - _Menu.docx	11/01/2019 11:09	Document Micro...	48 Ko
Path project method - 1.docx	01/01/2019 09:59	Document Micro...	102 Ko
Project Form - _Window.docx	14/06/2019 11:01	Document Micro...	100 Ko
Project Form - _KCS.docx	14/06/2019 11:01	Document Micro...	104 Ko
Project Form - _StructureAnalysis.docx	14/06/2019 11:01	Document Micro...	204 Ko
Project Form - _Setup.docx	14/06/2019 11:01	Document Micro...	100 Ko
Project Form - _OutsideCoverage.docx	14/06/2019 11:01	Document Micro...	101 Ko
Project Form - _Methods_Pallet.docx	14/06/2019 11:01	Document Micro...	111 Ko
Project Form - _MethodManager.docx	14/06/2019 11:01	Document Micro...	201 Ko
Project Form - _MethodAnalysis.docx	14/06/2019 11:01	Document Micro...	85 Ko
Project Form - _LinkCreateStructureFinal	14/06/2019 11:01	Document Micro...	35 Ko
Project Form - _LinkCreateDataStructure	14/06/2019 11:01	Document Micro...	31 Ko
Project Form - _LinkOtherLinking.docx	14/06/2019 11:01	Document Micro...	42 Ko
Project Form - _FormOptions.docx	14/06/2019 11:01	Document Micro...	153 Ko
Project Form - _FormGettable.docx	14/06/2019 11:01	Document Micro...	75 Ko
Project Form - _DataOptions.docx	14/06/2019 11:01	Document Micro...	80 Ko
Project Form - _DataOptionsDetail.docx	14/06/2019 11:01	Document Micro...	61 Ko
Project Form - _DataOptions.docx	14/06/2019 11:01	Document Micro...	89 Ko
Project Form - _Copyrights.docx	14/06/2019 11:01	Document Micro...	281 Ko
Project Form - _License.docx	14/06/2019 11:01	Document Micro...	123 Ko
Project Form - _Commands.docx	14/06/2019 11:01	Document Micro...	108 Ko

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.