

DEVELOP 10 TIMES FASTER



SUMMARY

Part 1 - Report editor

1.	Introduction.....	13
2.	How to create a report.....	23
3.	Data sources of a report.....	43
4.	Describing and handling a report.....	51
5.	Blocks of a report.....	59
6.	Report controls	69
7.	Using tables in a report	89
8.	Editable reports	100
9.	Composite reports.....	103
10.	Sequencing reports	106
11.	Nested reports	108
12.	Printing a report by programming.....	113
13.	Printing duplicate copies	128
14.	Printing a text in watermark	131
15.	Programming a report	135

Part 2 - Query editor

1.	Introduction.....	159
2.	How to create a query?.....	168
3.	Description of a query	189
4.	Running the query test and using it	211
5.	Advanced query	217

Part 3 - Reports and Queries

1.	Distributing Reports & Queries	233
2.	Using Reports & Queries	238
3.	Differences compared to the standard report editor of WINDEV	240

Remember to visit the download section of www.windev.com on a regular basis to check whether upgraded versions are available.

Email address of our Free Technical Support: freetechnicalsupport@windev.com.

This documentation is not contractually binding. Modifications may have been made to the software since this guide was published. See the **online help**.

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.
© PC SOFT 2019: This publication may not be reproduced in part or in whole in any form without the express permission of PC SOFT.

DEVELOP 10 TIMES FASTER

PART 1

Report editor

24

REPORT EDITOR - SUMMARY

1.	Introduction	
1.1	What is a report?	13
1.2	The report editor	16
1.2.1	Quick overview	16
1.2.2	Differences between the standard report editor and "Reports & Queries"	18
1.3	Main interface of the report editor	20
1.4	A menu shaped like a ribbon	21
2.	How to create a report	
2.1	Report and project	23
2.2	Creating a report	23
2.2.1	Overview	23
2.2.2	The different types of reports	24
2.2.3	Sources of the report	34
2.2.4	Breaks and sorts	35
2.2.5	Performing calculations in a report	36
2.2.6	Skin template	37
2.2.7	Print format of the report	37
2.3	Saving the report	37
2.4	Report test	37
2.4.1	Running the test	38
2.4.2	Stopping the test	39
2.4.3	Directory of data files used for test	39
2.4.4	Managing passwords (analysis or data file)	39
2.4.5	WINDEV Mobile: Opening a window in test mode in the code of the project	40
2.5	Printing a report	40
2.5.1	Available print modes	40
2.5.2	Print preview	41
3.	Data sources of a report	
3.1	Overview	43
3.2	Report on a data file	43
3.2.1	Overview	43
3.2.2	Characteristics	44
3.3	Report on a query	44
3.3.1	Overview	44
3.3.2	The different types of queries	45
3.4	Report on variable	46
3.5	Report on a Table control	46
3.5.1	Overview	46
3.5.2	Characteristics	46
3.6	Report on Looper control	48
3.6.1	Overview	48
3.6.2	Characteristics	48

3.7	Report on an Organizer control	49
3.7.1	Overview	49
3.7.2	Characteristics	49
3.8	Report on a Scheduler control.....	50
3.8.1	Overview	50
3.8.2	Characteristics	50
4.	Describing and handling a report	
4.1	Overview	51
4.2	Viewing and modifying the report description	51
4.2.1	Managing the data source	51
4.2.2	Display options in the print preview	52
4.2.3	Managing blocks	52
4.2.4	Print format	52
4.2.5	Options	52
4.2.6	Form and background image	53
4.2.7	Language	53
4.3	Opening, closing and saving a report.....	53
4.4	Using a form in a report.....	54
4.4.1	Information specific to the form	56
4.4.2	Including a form in a report	56
4.4.3	Image of the form	57
4.4.4	Multipage report (based on a PDF file)	58
4.5	Changing the skin template associated with the report	58
4.6	Changing the report template associated with the report.....	58
5.	Blocks of a report	
5.1	Overview	59
5.2	The different types of blocks	61
5.2.1	Start of Document block	61
5.2.2	Page Header block	61
5.2.3	Break Header block	61
5.2.4	Body block	62
5.2.5	Body Complement block	62
5.2.6	Break Footer block	62
5.2.7	Page Footer block	62
5.2.8	End of Document block	62
5.2.9	Iteration block	62
5.3	Characteristics of report blocks	63
5.4	Parameters and print options of blocks	63
5.5	Handling the blocks in the report editor	64
5.5.1	Create a block	64
5.5.2	Delete a block	65
5.5.3	Reorganize the blocks	65
5.6	Handle the blocks by programming.....	65
5.6.1	Processes of report blocks	66
5.6.2	Handling a block in a WLanguage process	67
5.6.3	Properties associated with the report blocks	67

6.	Report controls	
6.1	The different types of controls.....	69
6.1.1	Static control	69
6.1.2	Calculated control	71
6.1.3	Image control	72
6.1.4	Bar Code control	72
6.1.5	Preset control	72
6.1.6	Chart control	73
6.1.7	Shape control	75
6.1.8	Check Box control	75
6.1.9	Internal Report control	76
6.1.10	URL Link control	76
6.1.11	Edit control	77
6.2	Characteristics of report controls.....	78
6.2.1	Description window of a control	78
6.2.2	Details of the description window	78
6.3	Multilingual feature in the report controls.....	79
6.4	Group of controls	79
6.5	Handling controls in the editor	80
6.5.1	Creating a control	80
6.5.2	Selecting one or more controls	80
6.5.3	Copying a control	81
6.5.4	Copying the control style	82
6.5.5	Moving a control	82
6.5.6	Deleting a control	83
6.5.7	Using an off-report control	83
6.6	Aligning the report controls	83
6.7	Handling the controls by programming.....	84
6.7.1	Processes of the report controls	84
6.7.2	Using a control in a WLanguage process	84
6.7.3	Properties associated with the report controls	85
7.	Using tables in a report	
7.1	Overview	89
7.2	Creating a table in a report	91
7.2.1	Creating a Table report	91
7.2.2	Creating a Crosstab report	92
7.2.3	Creating a TreeView Table report	93
7.2.4	Inserting a table into a report	93
7.3	Describing the table	94
7.3.1	Characteristics of the table	94
7.3.2	Characteristics of columns	95
7.4	Handling the table cells	95
7.4.1	Selecting a table element	96
7.4.2	Displaying the table description	96
7.4.3	Adding a cell into a table	96
7.4.4	Adding a column into a table	97

7.4.5	Resizing the table cells	97
7.4.6	Moving a table column	98
7.4.7	Removing or deleting a table column	98
8. Editable reports		
8.1	Overview	100
8.2	How to?	100
8.2.1	Creating the editable report	100
8.2.2	Characteristics of edit controls	100
8.2.3	Processes associated with an edit control in a report	101
8.3	Input in the print preview (WINDEV only)	101
8.4	Input in a PDF file	102
8.5	Input in a report based on a PDF form	102
9. Composite reports		
9.1	Overview	103
9.2	Creating a composite report	104
9.3	Characteristics of composite reports and sub-reports	105
9.3.1	Characteristics of composite reports	105
9.3.2	Characteristics of sub-reports	105
10. Sequencing reports		
10.1	Overview	106
10.2	How to?	107
11. Nested reports		
11.1	Overview	108
11.2	Creating a nested report	109
11.2.1	Principle	109
11.2.2	Example	109
11.3	Specific features of secondary reports	110
11.3.1	Blocks in the secondary report	110
11.3.2	Orientation of secondary reports	110
11.3.3	Page break in the secondary report	110
11.3.4	Secondary report based on a query with parameters	110
11.3.5	Secondary report based on a query that is using a specific connection	110
11.3.6	Secondary report printed from a Page Footer block	110
11.3.7	Displaying the total number of pages in the secondary report	110
11.4	Printing a nested report	111
11.5	Differences between nested reports and internal reports	112
12. Printing a report by programming		
12.1	Overview	113
12.1.1	Principle for printing in WINDEV	113
12.1.2	Principle for printing in WEBDEV	113
12.1.3	Principle for printing in WINDEV Mobile	114
12.2	Report on data file	116
12.2.1	Printing a report on a data file	116

12.2.2	Printing a bounded report based on a data file	117
12.2.3	Printing a report with parameters based on a data file	118
12.2.4	Printing a bounded report with parameters based on a data file	120
12.3	Report on query	121
12.3.1	Printing a report on query without parameter	121
12.3.2	Printing a report on a query with parameters	122
12.4	Report on Table control	123
12.5	Report on variable	125
12.5.1	Printing a report on a variable	125
12.5.2	Printing a report with parameters on a variable	126
12.6	Print functions	127
13. Printing duplicate copies		
13.1	Overview	128
13.2	Creating a duplicate copy from the print preview (WinDev only)	128
13.2.1	Print available by default	128
13.2.2	Customizing the created duplicate copy	129
13.3	Creating a duplicate copy by programming	129
13.4	Signing a duplicate copy	130
13.5	Using a duplicate copy	130
14. Printing a text in watermark		
14.1	Overview	131
14.2	Creating a watermark from the print preview	131
14.3	Creating a watermark by programming	132
14.3.1	Method 1: Full configuration: using a Watermark variable	132
14.3.2	Method 2: Quick configuration: direct use	134
15. Programming a report		
15.1	Overview	135
15.2	Sequence of processes associated with a report	135
15.3	Processes associated with the report	136
15.3.1	"Opening" process	136
15.3.2	"Reading data" process	137
15.3.3	"Filter" process	137
15.3.4	"Closing" process	138
15.3.5	"Before re-printing" process	138
15.4	Break process	138
15.5	Sort procedure	139
15.6	Report with parameters	139
15.6.1	Creating a report with parameters	140
15.6.2	Printing a report with parameters	141
15.6.3	Running the test of a report with parameters	142
15.7	Handling a report in a WLanguage process	142
15.8	Keywords associated with reports	142
15.9	Properties associated with reports	143

15.10 WLanguage functions145

15.10.1 WLanguage functions not recommended in the processes145

15.10.2 Print functions of WLanguage that can be used in a report146

1. INTRODUCTION

1.1 What is a report?

A report provides a **custom view of data**: data entered in the database, data found in a text file, data coming from a Table control, ...

Depending on the product used, the data found in the report can be displayed in different formats:

- **In WINDEV:**
 - In a Windows application, a report can be displayed on the screen, saved in an HTML file, printed on a printer, ...
 - In a Linux application, a report can be saved in a PDF file.
- **In WEBDEV**, a report can be saved in a file (HTML or PDF), sent by email, displayed in the browser of the Web user or printed on a printer connected to the server, ...
- **In WINDEV Mobile:**
 - In a **Windows Mobile** application, a report can be printed on a printer (in PCL format) or saved in a PCL file.
 - In an **iOS** application, a report can be saved in a PDF file.
 - In an application **in Universal Windows App mode**, a report can be saved in a PDF file.
 - In an **Android** application, a report can be created in the editor. It is possible to generate the corresponding code and then print this report (in PDF format, for example).

A report includes:

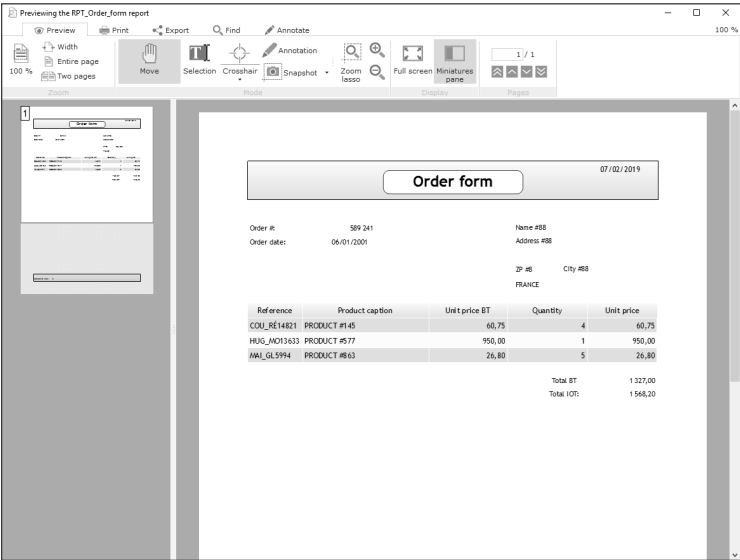
- **blocks** that define the different sections of the report (see "Blocks of a report", page 59 for more details).
- **controls** in which the report data is displayed (see "Report controls", page 69 for more details).

A report can be used to summarize and synthesize data. You can:

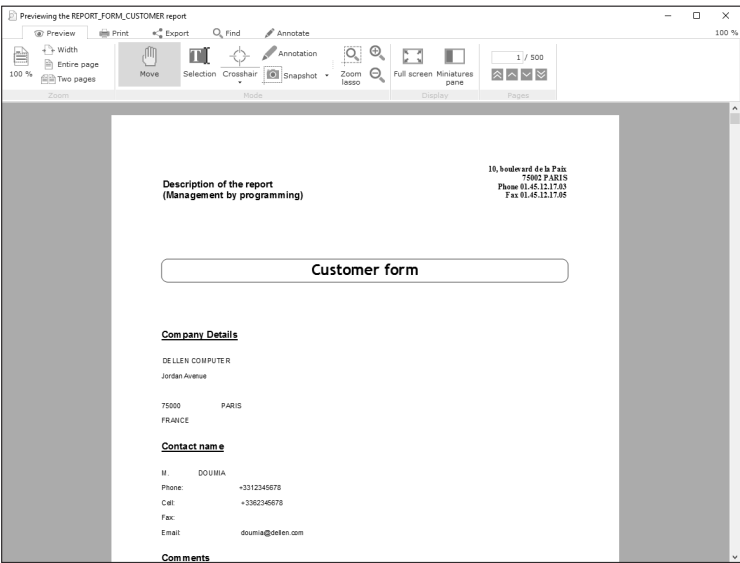
- group data,
- sort data according to any criterion,
- perform calculations (means, statistics) or create charts,
- display the data in several columns.

Just think of a report and the report editor will create it.

Some examples of reports:



Order form (print preview in WINDEV)

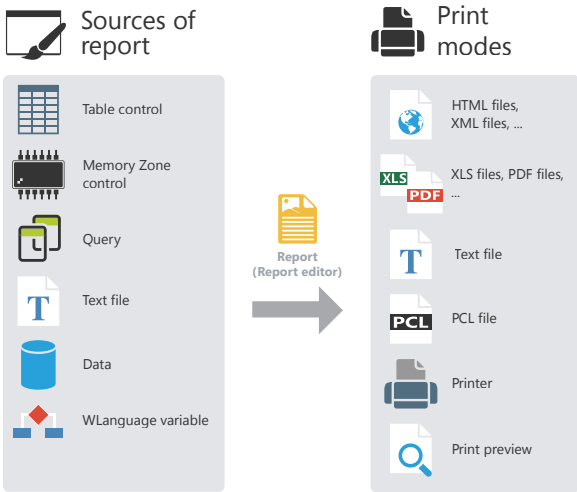


Information form

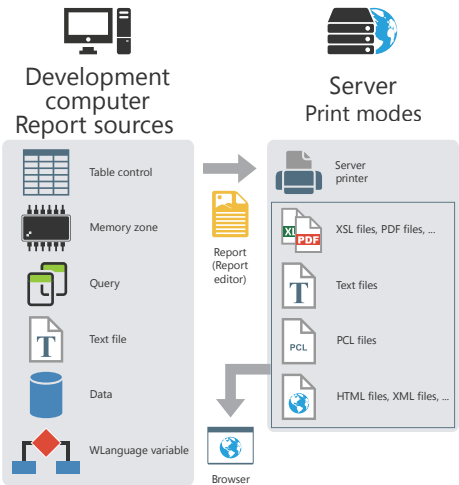
The diagrams below present a simple definition of a report:

- the data to print comes from a data source (data file, HFSQL view, query, memory zone, WLanguage variable or text file, ...).
- the report groups, sorts and formats the data.
- the report execution can be directed to:
 - in WINDEV: the screen, a file (PDF, HTML, RTF), a printer or a fax, ...
 - in WEBDEV: a file (HTML, RTF, PDF), a printer or a fax, ... If the report destination is a file, this file can be sent to the browser.
 - in WINDEV Mobile: a PCL file or a PCL printer (in Windows Mobile) or a PDF file (for the iOS applications or for the applications in Universal Windows App mode).

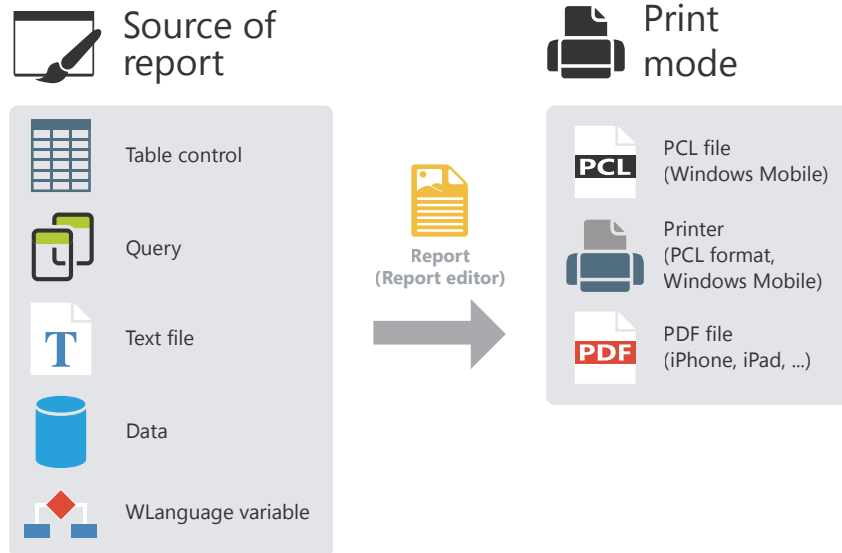
In WINDEV:



In WEBDEV:



In Windows Mobile:



The report editor is easy to use. It proposes several features:

- **Choosing the destination of report execution:**
You can print in a file in PDF, HTML, XML, XLS, RTF or PCL format.
In WEBDEV, this file is saved in a directory accessible in write mode on the server. Then, this file can be displayed on the browser.
In WINDEV, the report can be previewed before it is printed and it can be printed from the preview.
In WINDEV Mobile, the report destination depends on the platform:
- in the Windows Mobile applications, the report can only be saved in PCL format and printed on a PCL printer.
- in the iOS applications, the report can only be saved in PDF format.
- in the UWP applications, the report can only be saved in PDF format.
- **Choosing the items to print:**
You can choose the items that will be printed (no need to print all of them).
- **Ability to print the data of several data files:**
The report can contain items coming from several data files. To simplify the data selection, the queries allow you to easily select the data to print.
- **Ability to print the data coming from variables:**
The report can contain items linked to variables found in the project (global variables) or in the report (local variables). These variables can be simple variables, members of classes, members of structures or members of advanced variables.
- **Selecting records:**
One or more criteria can be defined to select the records that will be printed. Powerful queries allow you to create reports corresponding to complex selection conditions.
- **Sorting records:**
You can select the sort conditions according to which records will be displayed. You can even define a break on the sort arguments.
- **Printing labels:**
You can create and print custom labels.
- **Custom calculations:**
You can add custom calculations to your report (VAT, ...).
- **Totals and counters:**
You can add totals and counters on the controls (calculated controls or controls linked to items). These totals and counters can be general, intermediate, by break, ...
- **Images, borders, background colors:**
You can improve the report layout by adding charts, images, borders and background colors
- **Choosing fonts:**
The controls found in your report can have different fonts, different colors, ...
- **Configuring the page headers and footers:**
Various information (date, page number, total number of pages, ...) can be displayed in the page headers and footers. Different page headers and page footers can be defined in a report (even page footer and odd page footer for example).
- **Printing grouped reports:**
Several methods can be used to print several reports in the same output (in the same PDF file for example):
 - the nesting of reports,
 - the internal reports,
 - the composite reports,
 - the sequence of reports.
- **Choosing the number of columns:**
The data found in your report can be printed in several columns.

1.2 The report editor

1.2.1 Quick overview

WINDEV, WEBDEV and WINDEV Mobile allow you to create reports with the report editor. The reports can be created with the data coming from:

- a data file found in an analysis or any data file,
- a query result,
- a view on a HFSQL data file,
- a memory zone,
- a Table control found in a window or in a page,
- a TreeView Table control found in a window or in a page,
- an Organizer or Scheduler control found in a window,
- a text file,
- a variable (variable global to the project or variable used in the current report),
- ...

The report editor can also be used to create customizable blank reports.


The method for creating a report is straightforward. A wizard helps you create the report. Once it has been described, the report corresponding to your choices is generated. The report can be run:

- from the editor directly.
- from a site or from an application.

You can modify the layout of the report, add, modify or delete controls. You can also perform specific processes in WLanguage (the programming language of WINDEV, WEBDEV and WINDEV Mobile).

1.2.2 Differences between the standard report editor and "Reports & Queries"

Two versions of report editor are available:

- **the standard report editor.**
This version is supplied with WINDEV, WEBDEV and WINDEV Mobile. This version can only be used when developing a WINDEV or WINDEV Mobile application, or a WEBDEV site.
- **the user version of the report editor** (also called **Reports & Queries**). This editor is supplied with WINDEV only.
This version is independent of WINDEV and it can be used by the end users.
Some features of report editor are not available in this version.
The  icon indicates the differences between the user version of report editor and the standard report editor.

Standard report editor

The standard query editor allows you to create reports and to run these reports from the editor or from an application developed with WINDEV, WINDEV Mobile or WEBDEV.

The dongle of the product used is required to use the report editor. It must be plugged into the parallel port of your computer when using the editor.

No dongle is required by the end users to run a report from a WINDEV, WINDEV Mobile or WEBDEV application.

Several examples are supplied with the standard report editor. These examples describe the main reports that can be used in your applications (invoice, list, labels, ...).

To find out the list of examples for using reports:

- Display the home page (Ctrl + <).
- Type the "Report" keyword.
- The list of examples for using reports is displayed.

Check these examples, they will teach you a lot and you will be able to use them when you create your own reports!

User version of report editor (Reports & Queries - WINDEV only)

Reports & Queries allows you to create and run reports from the editor (WINDEV is not required).

The reports cannot contain code lines for modifying the data files (functions used to perform write operations in the files).

The created reports can be run from the report editor or from a WINDEV application (if this application supports the reports created with Reports & Queries).

No dongle is required to use Reports & Queries.

Some operations are required to create new reports or to modify the existing reports with Reports & Queries. See "Distributing Reports & Queries with your WINDEV application", page 233 for more details.

Documentation

Reports & Queries and the standard report editor are supplied with the same online help. The standard report editor is also supplied with printed documentation.

- **Printed documentation**
The printed documentation is intended to help you discover and use the report editor. The second part of this guide describes the query editor.
The "Report editor guide" is supplied with the "Tutorial" and the "Concepts".

- **Online help**
The online help is intended to help you handle the editor. The online help is proposed:
 - in the format of a local help. The necessary files are installed on the development computer.
 - in the format of an Internet help. This help can be accessed from any computer equipped with an Internet access. This help is updated on a regular basis.

To access the online help, use one of the following methods:

- on the "Home" pane, in the "Online help" group, expand "Help" and select "Help index".
- press the F1 key.

To modify the type of online help used:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of XXX".
2. In the "Help" tab, select the access mode to the help database : "Use the online help" or "Use the local help database" as well as the preferred language.
3. Validate.

Setup

- **Installing the standard report editor:** The standard report editor is automatically installed with WINDEV, WEBDEV or WINDEV Mobile.
- **Installing Reports & Queries:** To install Reports & Queries, follow the instructions given when installing the WINDEV application.

Minimum configuration required for Reports & Queries

Reports & Queries must be installed on a computer with the following characteristics:

- 512 MB of RAM (1 GB or more recommended).
- 800 MB of disk space.
- Windows 2000, 2003, XP, Vista, 7, 8 or later.

Register

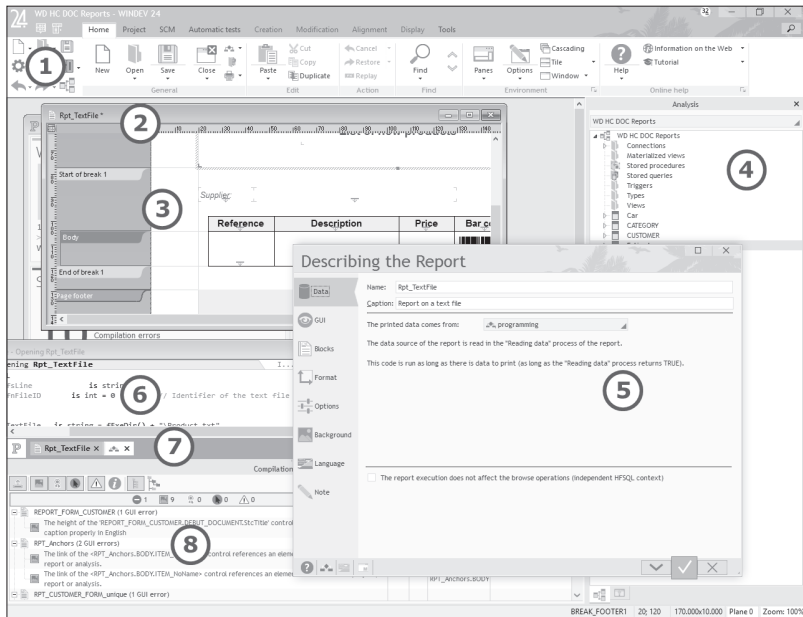
The product setup wizard proposes you to register your product. This registration is very important. It allows you to benefit from the various free services that PC SOFT offers to its customers: free Technical Support, information about new releases, product upgrades, and more.

This registration can also be done directly from the product used: On the "Home" pane, in the "Online help" group, expand "Help" and select "Register the product".

1.3 Main interface of the report editor

To start the report editor, you must:


- create a new report:
 - On the "Home" pane, click "New".
 - The window for creating a new element is displayed: click "Report" then "Report".
- open an existing report (via the "Project explorer" pane for example).



The important elements of report editor are as follows:

- 1 Ribbon**
Gives quick access to the main features of report editor. This ribbon includes panes, groups and options. See "A menu shaped like a ribbon", page 21 for more details.
- 2 Alignment rulers**
Used to align the controls in the report. To display the rulers, press Ctrl + R.
- 3 Report currently edited**
Scrollbars are used to scroll the huge reports. You also have the ability to perform a zoom in the report to better view the report elements (Ctrl + mouse wheel).

- 4 Vertical panes**
Displays:
 - the elements (files, variables, reports, controls, ...) available in the current project ("Project explorer" pane).
 - the description of report elements via the modifier.

**Important**

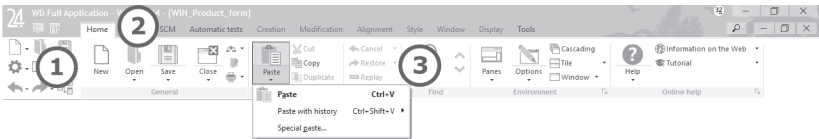
Only the "Project explorer" and "Modifier" panes are available in Reports & Queries.

- 5 Description window**
Allows you to view and/or modify the characteristics:
 - of selected controls.
 - of the current report.
- 6 Code window**
Displays the processes associated with the report, with its blocks and controls.
- 7 Opened documents**
Displays a button for each document opened in the editor. To display the requested element in the foreground, click the corresponding "button".
- 8 Horizontal panes**
Displays:
 - the compilation errors (found in the WLanguage code) or the GUI errors ("Compilation errors" pane),
 - the search tool ("Find - Replace" pane),
 - the debugger during the test of an element ("Debugger" pane),
 - ...

1.4 A menu shaped like a ribbon

WINDEV, WEBDEV and WINDEV Mobile present the different options of the editors via a ribbon. This ribbon includes panes in which the different options of editors are grouped.

The ribbon includes several areas:



- 1. The area of quick access buttons.
- 2. The pane area.
- 3. The area of groups of options.

The quick access buttons (1)

The quick access buttons are used to perform the most usual operations, common to all the editors: save, open, create, ...



The 3 buttons found at the top of this area are specific buttons:

- the logo of the version is used to display:
 - the "About..." window.
 - the custom-menus.
 - the drop-down menus of earlier versions (version 17 and earlier).
- the 2 following pictograms are used to restore the menus and the toolbars found in the earlier versions (version 17 and earlier).

The pane area (2)

The different ribbon panes are used to access the options of different editors.

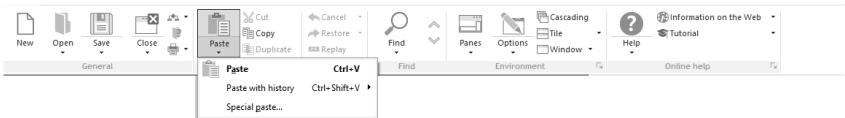


Several types of panes are available:

- the current pane: The pane tab appears in light gray and a colored line is displayed at the top of tab.
- the popup panes, specific to the current element: These panes are related to the current editor or to the selected element. The pane name is displayed:
 - in yellow for WINDEV,
 - in orange for WINDEV Mobile,
 - in blue for WEBDEV.
- the other available panes: The pane name is displayed in white.

The option area (3)

The options are grouped on the panes in groups of options.



The options displayed in the ribbon differ according to the selected pane.

Several types of options are available:

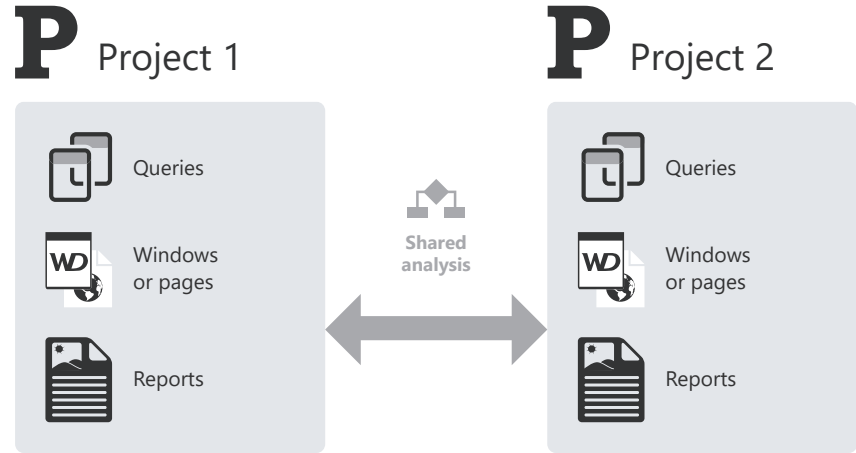
- Options to check.
- Buttons to click.
- Button with arrow used to expand the options. Two types of buttons with arrow are available:
 - the buttons with arrow used to expand a menu.
 - the buttons with arrow used either to expand a menu (click on the arrow), or to perform a default action (click on the button icon).

The options are organized by group. Each group of options has a name and it can also include a group button: . This button is used to perform a specific action according to the current group: display the description of current element, display the help, ...

2.How TO CREATE A REPORT

2.1 Report and project

In most cases, a report is linked to a project. To be able to extract the data from the data files, the report editor must know the description of the data files that is stored in the analysis, the analysis being associated with a project.



In this case, a WINDEV analysis is shared between two projects (WINDEV, WINDEV Mobile or WEBDEV).

To load a project, on the "Home" pane, in the "General" group, expand "Open" and select "Open a project".

2.2 Creating a report

2.2.1 Overview

To create a report:

1. Click on among the quick access buttons.
 - The window for creating a new element is displayed: click "Report" then "Report".
 - The wizard for report creation starts.
2. Specify:
 - the type of report to create (form, table, ...). See "The different types of reports", page 24 for more details.
 - the data source of report (data file, query, ...). See "Sources of the report", page 34 for more details.
 - the report elements (controls, sort conditions, breaks, etc.).
3. Validate the report creation. The report is generated and saved according to the selected options.

These different steps are performed in the wizard for report creation. A report corresponding to the selected options is automatically created at the end of wizard. The created report appears in edit in the report editor. The different blocks and controls of your report have been created. They can be modified according to your own requirements. Most of the options chosen when creating the report can be modified thereafter.

Then, the report can be run:

- from the report editor,
- from a WINDEV application,
- from a WINDEV Mobile application (for Windows Mobile or iOS),
- from a WEBDEV site.

In an Android configuration, a script corresponding to the report can be generated and integrated into the Android application.

2.2.2 The different types of reports

Several types of reports can be created in the report editor. All these types of reports are available in the report editor (supplied with WINDEV, WEBDEV and WINDEV Mobile) and in Reports & Queries.

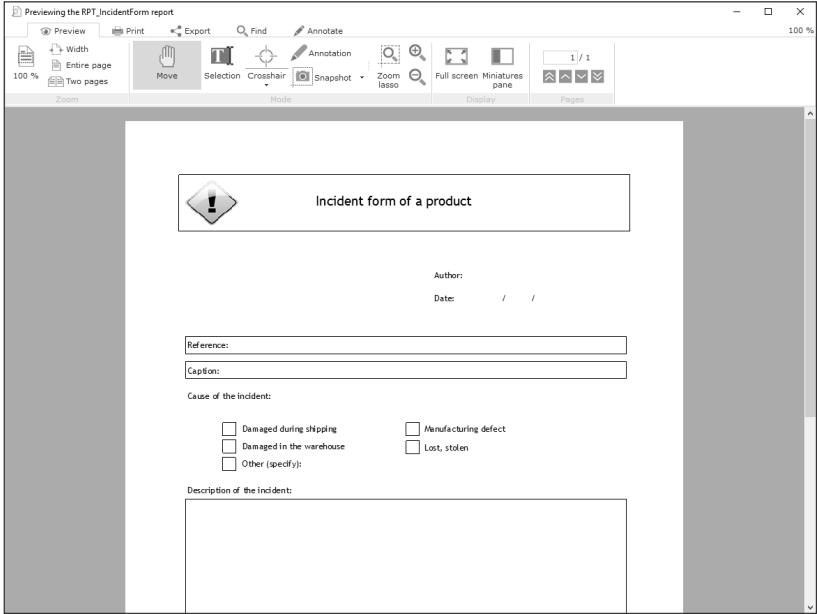
The following types of reports are available:

- Blank report,
- Form report,
- Table report,
- Crosstab report,
- TreeView Table report,
- Gantt Chart report,
- Organizer report (not available in WINDEV Mobile),
- Scheduler report (not available in WINDEV Mobile),
- Custom Mailing report (not available in WINDEV Mobile),
- Label report,
- Report on Form,
- Multicolumn report,
- Composite report.

Let's see the characteristics for each type of report as well as use examples:

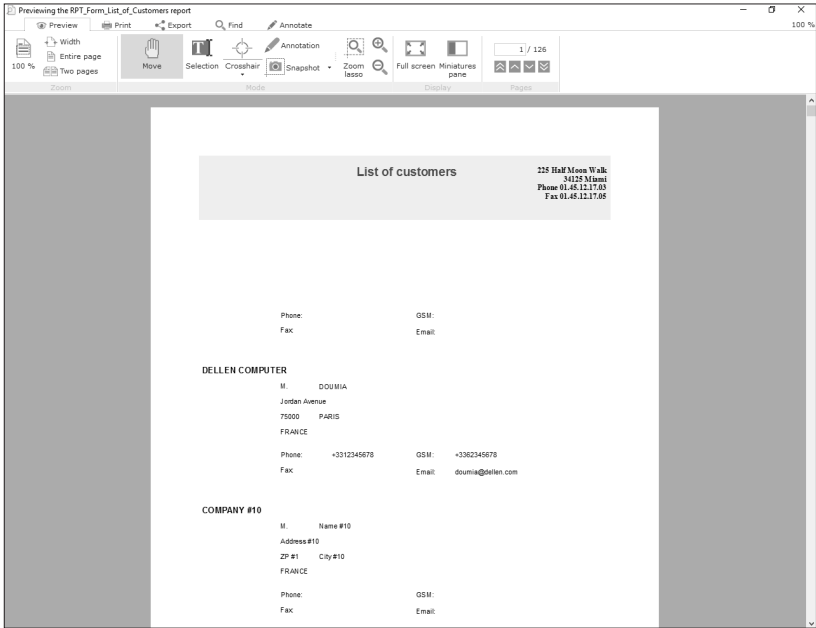
- **Blank report:**
 - **Reading the data source of report**
Each page displays:
 - a specific text.
 - a standard form (filled manually thereafter).
 - the data coming from one or more records.
 - **Use examples:**
 - Form for entering the incoming calls.
 - Form for entering the absenteeism.
 - Form for entering the products received.

• Preview of a blank report.



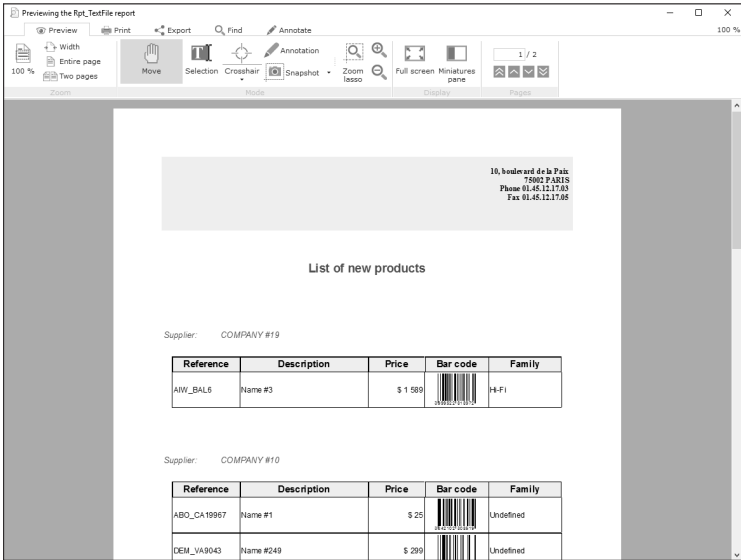
- **Form report:**
 - **Reading the data source of report**
The data read in the data source is displayed in a form: each page displays data from one or several records.
 - **Use examples:**
 - Detailed list.
 - Data grouped according to a criterion.

• Preview of a form report

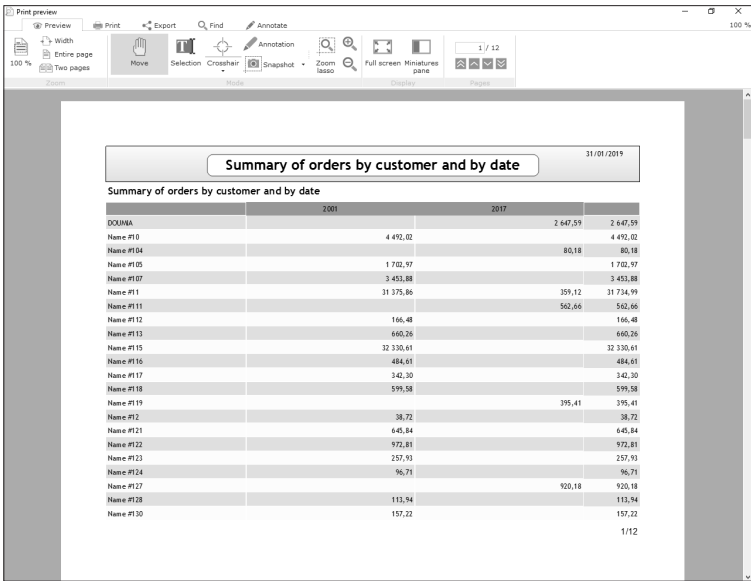


- Table report:
 - Reading the data source of report
Each table row corresponds to a record read in the data source of report (data file, query, ...).
 - Use examples:
 - Summary table,
 - Invoice,
 - Detailed list,
 - Data grouped according to a criterion.

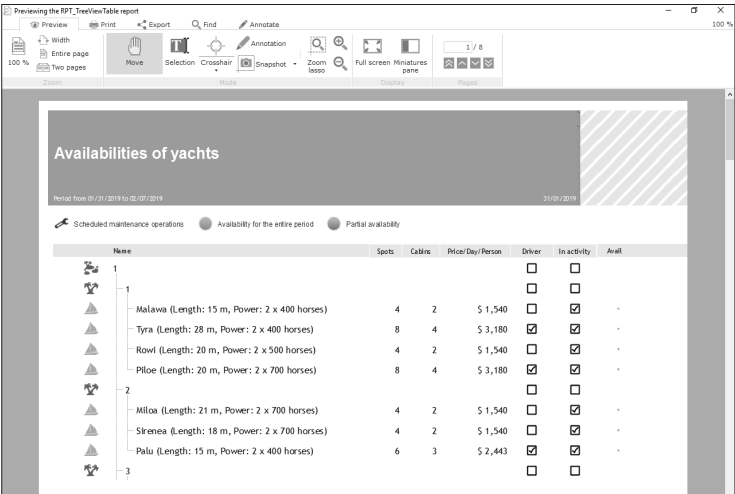
• Preview of table report



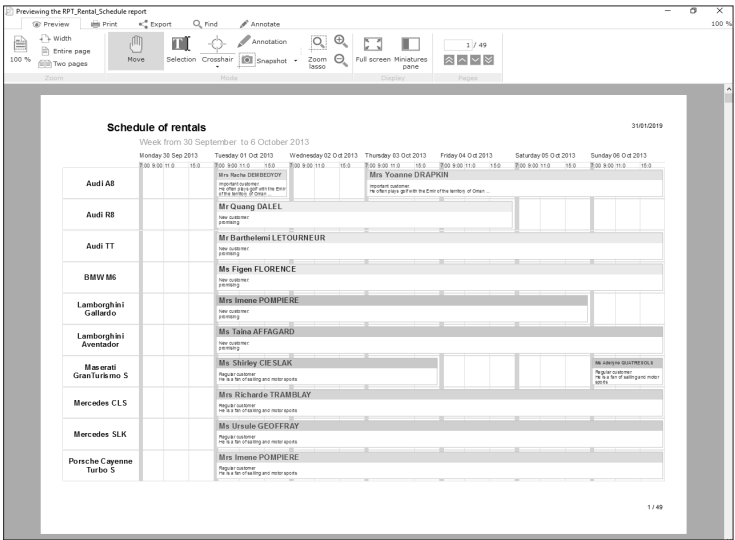
- Crosstab report:
 - Reading the data source of report
The data read in the data source is displayed in a double-entry table.
 - Use example: Table representing the turnover per product and per year.
 - Preview of Crosstab report



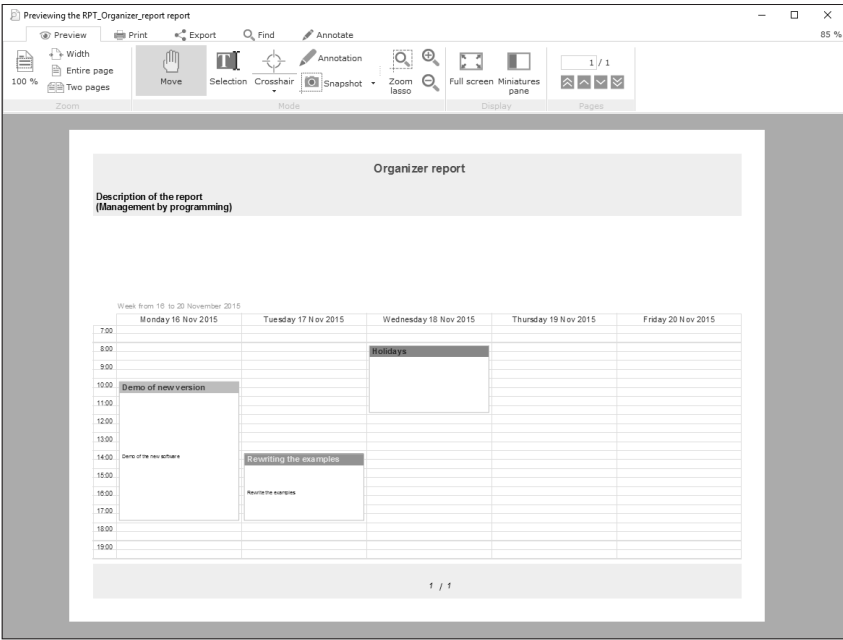
- **TreeView Table report:**
 - **Reading the data source of report**
The data read in the data source is displayed in a treeview table.
 - **Use example:** Table representing the turnover per product and per year.
 - **Preview of TreeView Table report**



- **Scheduler report:**
 - **Reading the data source of report**
The data read in the data source is displayed in schedule format.
 - **Use examples:** Printing a daily or weekly schedule, ...
 - **Preview of Scheduler report**

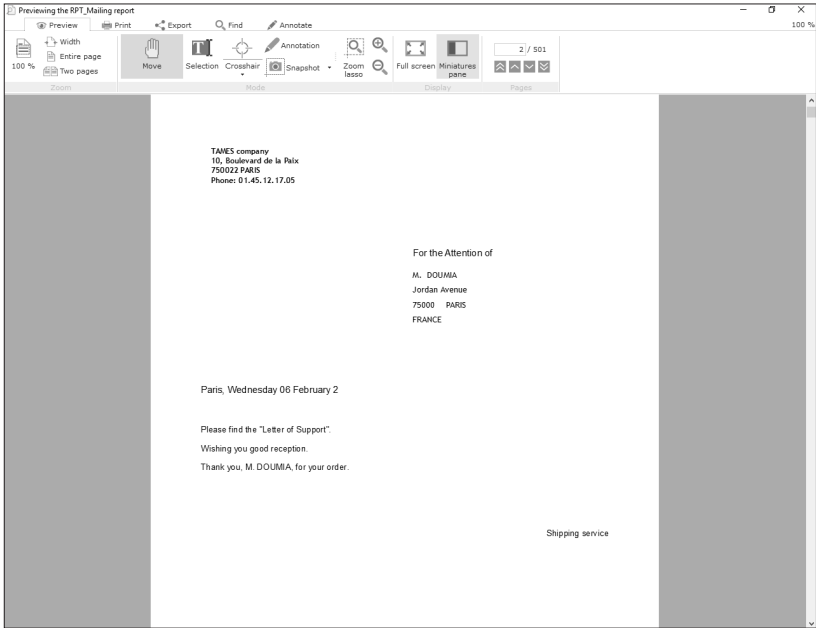


- **Organizer report:**
 - **Reading the data source of report**
The data read in the data source is displayed in organizer format.
 - **Use examples:** Printing a daily or weekly schedule, ...
 - **Preview of Organizer report**



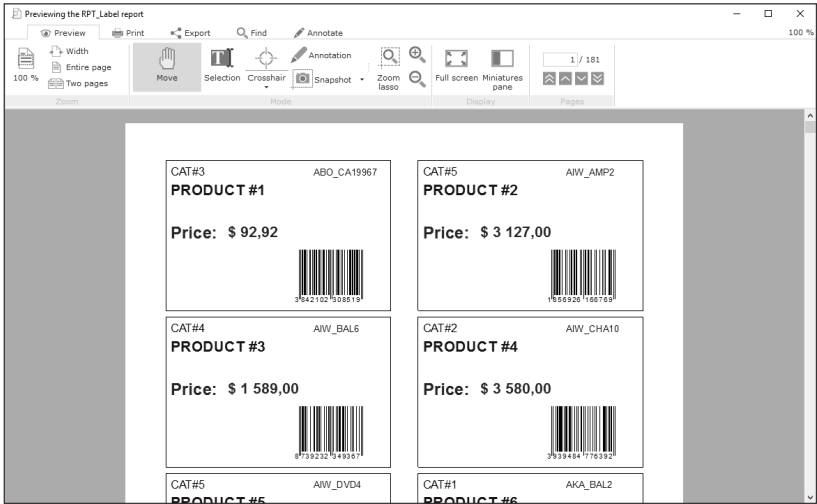
- **Mailing report:**
 - **Reading the data source of report**
Each record read in the data source corresponds to a page.
 - **Use examples:**
 - Mailshot.
 - Advertising letter.

• Preview of mailing report



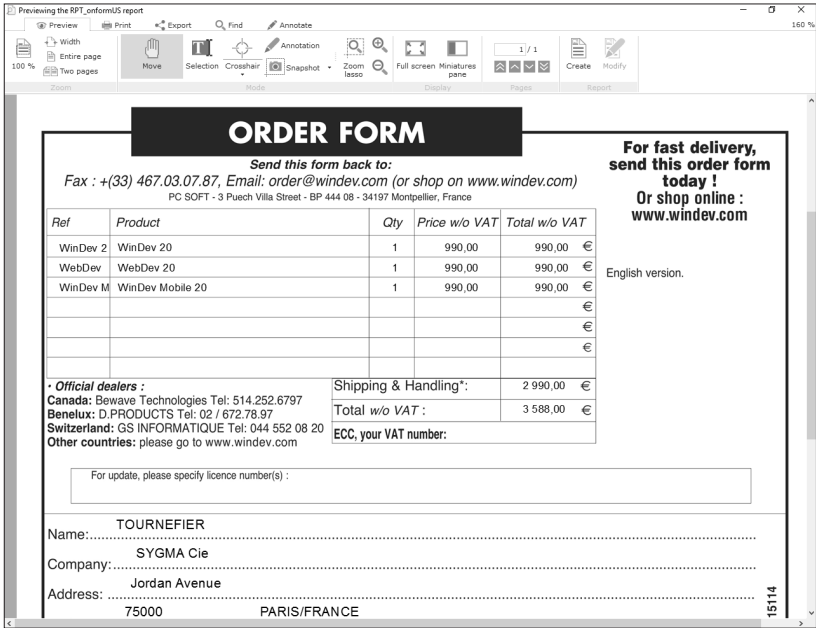
- Label report:
 - Reading the data source of report
Each record read in the data source corresponds to a label.
If you choose this type of report, you can define the number of copies for the same label.
 - Use examples:
 - Price labels.
 - Address labels.

• Preview of label report

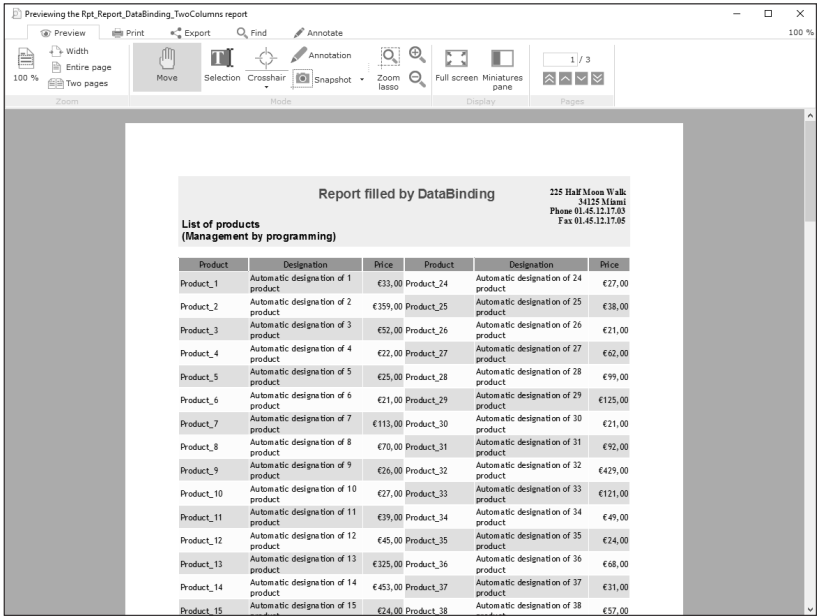


- Report on form:
 - Reading the data source of report
The data read in the data source is displayed in a form: each page displays data from one or several records.
 - Use examples:
 - Postal forms.
 - Check.

• Preview of report on form



• Preview of multicolumn report

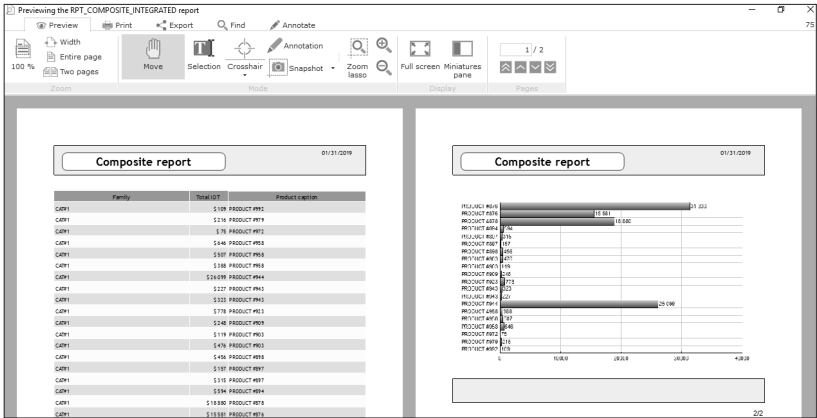


• Multicolumn report:

- **Reading the data source of report**
if the multicolumn report is a Table report, each table row corresponds to a record read in the data source of report (data file, query, ...).
If the multicolumn report is a Form report, the data read in the data source is displayed as a form: each page displays data from one or several records.
- **Use example:** Product catalog

• Composite report:

- **Reading the data source of report**
The composite report includes several sub-reports. Each sub-report can have a specific type. Reading the data source of sub-reports depends on the type of sub-reports.
- **Use examples:** Printing orders and sales agreements.
- **Preview of composite report**



2.2.3 Sources of the report

The data source of report defines the origin of data printed in the report. Several data sources are available. These data sources can be selected in the wizard for report creation or in the description window of report. The available data sources are as follows:

- **Data file:** The data to print comes from a data file described in the project analysis. This data file can have any format (HFSQL Classic, Oracle, ...).

R&Q

Important

The data files and the file items that can be used in the reports created with Reports & Queries are defined by the developer in the data model editor. See "Distributing Reports & Queries", page 233 for more details.

- **Query:** The data to print comes from a query. This query corresponds to:
 - a query created and saved in the query editor.
 - a query created when creating the report.
- **Table, TreeView Table, Organizer or Scheduler control** found in a window or in a page: The data to print comes from a specific control found in one of the application windows or in one of the site pages. The records are selected by the control and they are printed by the report.

R&Q

Important

Reports & Queries cannot be used to create reports on a Table control, a TreeView Table control, an Organizer control or a Scheduler control. However, you have the ability to customize the reports on a Table, TreeView Table, Organizer or Scheduler control created with the standard report editor (color, fonts, size, linked column, ...).

- **Existing variable :** The data to print comes from a WLanguage variable found in the project or in the report. This variable can be a simple array, a class member, a structure member or an advanced variable member.

R&Q

Important

Reports & Queries cannot be used to create or modify reports on variable.

- **SQL query via a connection** (ODBC, Native Accesses, ...) (not available in WINDEV Mobile): The data to print comes from a database accessible via a specific ODBC driver, via a Native Access, ... The SQL query is defined when creating the report.

Note: This type of source is not available in the wizard for report creation.

- **No data source:** The report will be linked to no data source.

Other available data sources

- **Text file:** The data to print is read in a text file.
- **Memory zone (not available in WINDEV Mobile):** The data to print corresponds to the records stored in a memory zone.

R&Q

Important

Reports & Queries cannot be used to create reports on a memory zone. However, you have the ability to customize the reports on a memory zone created with the standard report editor (color, fonts, size, ...).

- **View on a HFSQL Classic data file:** The data to print comes from a HFSQL view.

R&Q

Important

Reports & Queries cannot be used to create reports on a HFSQL view. However, you have the ability to customize the reports on HFSQL view created in the standard report editor (color, fonts, size, ...).

- **Programmed reading of the data source:** To manually program and select the records that will be displayed in the report, choose a report based on a programmed data source.

R&Q

Important

Reports & Queries cannot be used to create reports on a programmed data source.

These data sources are presented in *Data sources*.

Note: If one of the above-mentioned data sources is selected, the records that will be displayed in the report are automatically selected by the report editor.
To manually program and select the records that will be displayed in the report, choose a report with programmed reading of data source.

2.2.4 Breaks and sorts

The breaks and the sorts are used to define the sort order of data printed in the report as well as the options for grouping data.

- If the report is based on a data file, the search items of data file are automatically proposed as sort items.
- If the report is based on a query, the sorts performed in the query are automatically proposed for the breaks.
- If the report is based on a variable, only the breaks can be defined: the sorts must be performed by programming.

The sort

To make the result of a query easier to read, the records can be sorted according to one or more items. Sorting the records means classifying them in an ascending order or in descending order.
For example, the customers can be sorted by name, from A to Z.
If your report is based on a query, the sort order is defined when creating the query. However, you can:

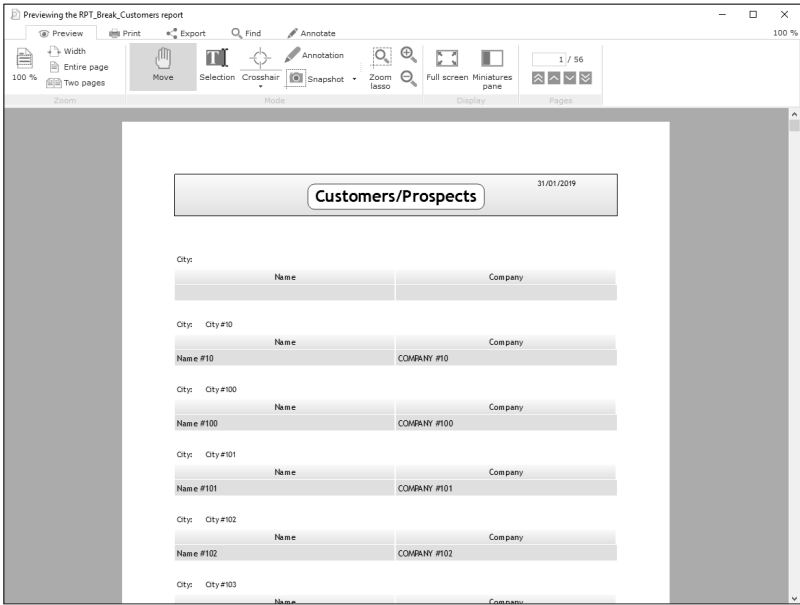
- add or delete a sort item to/from an existing report.
- modify the sort order performed on an item in an existing report.

To improve the sort, you also have the ability to use:

- the WLanguage functions (**Left**, **Right**, ...).
- a WLanguage procedure (server procedure in WEBDEV). This procedure will be automatically called to sort the records displayed in the report. This procedure must be a global procedure of project or a local procedure of report. See "Sort procedure", page 139 for more details.

The breaks

- The break is used to group the records corresponding to the same sort argument.
For example:
- to list the stock price by day, all you have to do is sort the stocks by date and perform a break on the date.
 - to list the customers and the companies by city, sort the customers by city then by company name, and perform a break on the city.



- Adding a break automatically generates the Break Header and Break Footer blocks. These blocks are used to:
- display the content of break item (in the Break header block).
 - perform calculations on the records found in the break (in the Break footer block).

- The report editor proposes two types of breaks:
- **automatic break** (the most common one). The break is performed on an item belonging to the data source of report.
 - **programmed break**. The break is entirely defined by programming in the "Break?" process of Break Header block. See "Break process", page 138 for more details.
- See the online help (keyword: "Break, Break in a report").

2.2.5 Performing calculations in a report

Several calculations can be performed on the numeric data found in a report. For example, in a report used to list the orders per customer, you can calculate and display the average value of the customer orders.

Note: The different calculations performed in a report are defined when the report is created. However, new calculations can be added to the created report via the calculated controls.

2.2.6 Skin template


If the project is associated with a skin template (or with a template), this skin template (or this template) can be selected for your reports. Your printed reports and your application will have the same style. This skin template (or template) can be modified once the report is created.

Caution: Depending on the type of printer, the reports containing images may take quite a long time to print.

2.2.7 Print format of the report

- The format of a report is used to configure:
- the report orientation.
 - the paper format.
 - the print margins.
 - the number of columns in the report. By default, a report contains one column. If the report contains more than one column, the report is said to be "multi-column".
- For a "Label" report, you also have the ability to configure:
- the format of the label sheet.
 - the spacing between the labels.
- This information can be modified after the report creation. See "Describing and handling a report", page 51 for more details.

2.3 Saving the report

- To save a report:
- on the "Home" pane, in the "General" group, click "Save".
 - click on  among the quick access buttons.

The extension of the created report file is ".WDE".
The report is automatically included in the list of project elements.
If you are using the report editor, the reports will be included in the library:

- when creating the executable.
- when deploying the site.

2.4 Report test

The test of a single report allows you to run the current report in WINDEV, WEBDEV or WINDEV Mobile. Like for the project test, the debugger can be started as soon as a problem occurs.

Note: The test of the report is run on the development computer only. The data used for test corresponds to the data found on the development computer.

- The following processes are run during the report test:
- Initialization process of project.
 - Processes associated with the report.
 - Processes associated with the report blocks.
 - Processes associated with the report controls.

Note: The test of the report is available for the reports based on a query, a data file, a text file and for the reports associated with no data source. For the other types of reports, you must run the test of the element (window, page or project) from which the report is printed.


R&Q

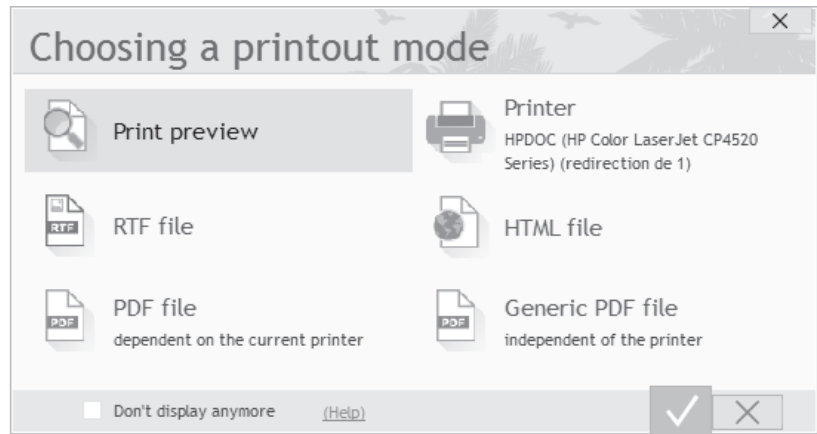
Important

During the test of a report in Reports & Queries, the initialization process of the project is run.

2.4.1 Running the test

To run the test of current report:

- 1. Click  among the quick access buttons (F9 key).
- 2. A screen allows you to choose the print mode of report:



The available options are:

- **Print preview:** used to preview the report as it will be printed.
- **Printer:** the report is printed on the default printer.
- **HTML file:** the report is saved in an HTML file and displayed in the Internet browser installed on the computer.
- **RTF file:** the report is saved in an RTF file and displayed in a text editor (Word for example).
- **PDF file:** the report is saved in a PDF file and displayed in a PDF viewer (Acrobat Reader for example).
- **Generic PDF file:** the report is saved in a generic PDF file (independent of the printer) and displayed in a PDF viewer (Acrobat Reader for example).

See "Printing a report", page 40 for more details.

3. Choose the print mode and validate. If your report is not saved, WINDEV, WEBDEV or WINDEV Mobile automatically proposes to save it under the name given in the wizard for report creation.

4. Specify the additional information if necessary:

- If **the report is bounded**, a window allows you to enter the report bounds. See "Report on a data file", page 43 for more details.
- If **the report is based on a query with parameters**, a window allows you to enter the query parameters. See "Report on a query", page 44 for more details.
- If **the report expects parameters**, give a default value to the parameters in the declaration of parameters. See "Break process", page 138 for more details.

Note: The last test run (window, query, report, ...) can be re-run from any project element via "GO .. Re-run the last test" found in the quick access buttons.

Note: minimising the editor during the test


During a test, the editor is minimized by default. Depending on the hardware configuration (large screen, 2 screens,), this minimize operation is not required.

To keep the editor opened:

- 1. On the "Home" pane, in the "Environment" group, expand "Options" and select "Options of the code editor".
- 2. In the "Debugger" tab, uncheck "Minimize the editor in test mode".
- 3. Validate.

2.4.2 Stopping the test

Several methods can be used to stop the test:

- **1st method:** Close the preview of the report whose test is currently run. The current editor at the time the test was started is displayed.
- **2nd method:** Go back to the editor with the taskbar (or press Alt + Tab) and confirm the end of test. The current editor at the time the test was started is displayed.
- **3rd method:** Go back to the editor via the taskbar or with Alt + Tab and click on  ("End the test") in the "Debugger" pane.

2.4.3 Directory of data files used for test

To specify the data files used when running the application test in the editor:

- 1. Display the project description: on the "Project" pane, in the "Project" group, click "Description".
- 2. In the "Files" tab, select the location of test files:
 - **Directory of the current configuration** (option selected by default): The data files will be found in the sub-directory corresponding to the current configuration (sub-directory of EXE directory of project).
 - **Specific directory:** The data files will be found in the specified directory.

2.4.4 Managing passwords (analysis or data file)

No password is requested when running the test of a report linked to a password-protected data file.

The necessary passwords must be specified by programming in the "Opening" process of the report:

- If the report is linked to one or more password-protected data files, this password must be specified in **HPass**.
- A WLanguage error occurs when running the report test if the passwords are not specified in the "Opening" process of the report.

2.4.5 WINDEV Mobile: Opening a window in test mode in the code of the project

During the test of a report, a WLanguage error is displayed if a window is opened in the initialization or closing code of the project.
To avoid this error, **InTestMode** must be used to configure the window opening. For example:

```
IF InTestMode() = False THEN Open(WIN_MyWindow)
```

However, you have the ability to use the functions for opening the system information boxes (**Info** or **Trace** for example).

2.5 Printing a report

2.5.1 Available print modes

Several print modes are available for a report. These different modes are presented in the table below:

	Printing during the report test in the editor	Printing a report from the preview window (WINDEV only)	Printing a report by programming (iPreview, iDestination and iPrintReport)
Paper print	X	X	X
Print preview	X	-	X (WINDEV only)
Mini-preview			X (WINDEV only)
HTML file (with style sheet)	X	X	X
HTML file (without style sheet)			X
RTF file	X	X	X
XML file		X	X
PDF file	X	X	X
Text file			X
Sending a fax			X
PCL file			X
XLS file			X

WINDEV Mobile: When running a test in the editor, all the options are available. However, when printing a report by programming:

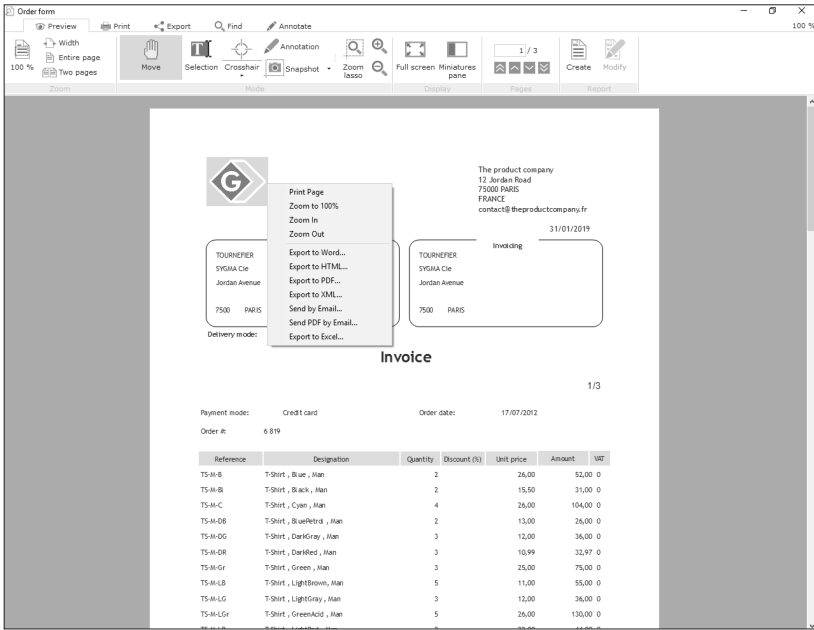
- in a **Windows Mobile** application, the prints can only be performed on a PCL printer or in a PCL file.
- in an **iOS** application or in **Universal Windows App** mode, the prints can only be performed in a PDF file.

2.5.2 Print preview

The print preview allows you to see the report layout without printing it (report preview). The report layout and content can be checked. You also have the ability to print the report.
The print preview is available:

- in the report editor of WINDEV, WEBDEV, WINDEV Mobile.
- in "Reports and Queries".
- when printing reports from a WINDEV application.

The print preview proposes several features:



- A menu shaped like a ribbon that includes the following panes:
 - Preview: used to manage the current preview: Zoom, crosshair, move, move between pages.
 - Print: used to manage the direct print on the printer.
 - Export: used to manage the different export modes of the print (PDF, Word, Excel, ...).
 - Find: used to manage the search in the preview.
 - Annotate: used to annotate the print currently displayed in the preview.
 - This menu is also available during a print preview performed in WLanguage.
 - A popup menu proposing most of the options.
- Note:** To preview a report, a printer must be installed on the current computer.

Other features (WINDEV only)

- **Customizing the preview:** Your application is using a skin template? Why not use this skin template for the preview window. WINDEV allows you to customize the preview window.
- **Previewing a print performed with the WLanguage functions:** If a print is performed with the WLanguage functions, you can also benefit from the features of print preview.
- **Configuring the different options:** To display some panes or some options only:
 - use **iParameterPreview**. This function, used in the initialization code of project, allows you to choose the panes and options that will be displayed for all the application reports.
 - configure the different options that will be displayed for each report in the report description ("GUI" tab).
- **Printing from the preview:** To find out whether the print was performed from the print preview, use **iDocumentPrinted**.

3.DATA SOURCES OF A REPORT

3.1 Overview

We mentioned in the previous chapter that different data sources could be selected when creating a report. This chapter presents these data sources and their benefits:

- Report on a data file.
- Report on a query.
- Report on variable.
- Report on a Table control.
- Report on a TreeView Table control.
- Report on an Organizer control.
- Report on a Scheduler control.

Note: Other data sources are also available but they are rarely used:

- Report on a HFSQL view.
- Report on a memory zone.
- Report on a text file.
- Report on a programmed data source.

See the online help for more details.

3.2 Report on a data file

3.2.1 Overview

A report based on a data file is used to print the records coming from a single data file (HFSQL Classic or Client/Server, accessed via a Native Access, ...). If the records come from several files, we recommend that you use a report on query.

The creation of a report on a data file is performed by the wizard for report creation. This type of report can be created only if the current project is associated with an analysis.

Once the report on a data file is created, you can:

- modify the data source of report ("Data" tab of report description).
- modify the search key ("Data" tab of the report description).
- define the bounds ("Data" tab of the report description).
- add or delete Item controls linked to the data file associated with the report.



Important

The data files and the file items that can be used in the reports created with Reports & Queries are defined by the developer in the data model editor of WINDEV. See "Distributing Reports & Queries", page 233 for more details.

3.2.2 Characteristics

The characteristics of a report on data file are as follows:

- **Search key:** a report on a data file is using a **search key**. This key is used to browse the records that will be printed in the report. This key is defined in the data file.
- **Bounded report:** a report on a data file can be **bounded**. These bounds are used to define the range of records that will be printed in the report. "Report on a data file", page 43 for more details.

3.3 Report on a query

3.3.1 Overview

A query is used to easily select records in one or more data files. A query can also be used to perform calculations on the selected data. You can, for example, select all the orders for the customers living in New York City and calculate the total amount of orders per customer.

Using queries in a report allows you to easily create complex reports. The query pre-selects the records and the report organizes the selected data and performs the different calculations.

Caution: To optimize the report, we advise you to perform the calculations (totals on items for example) in the report (and not in the query).

Several modes can be used to create a report based on a query:

- the query is created and saved in the query editor. The report is based on the file corresponding to the query (.WDR file). The query can be re-used in another report, window, control, ... The query can be modified at any time in the query editor. *The query is said to be independent.*
- the query is created when creating the report. The query is included in the report. No file specific to the query is created. The query can only be modified from the report where it was created. It cannot be re-used in another report, window, control, ... *The query is said to be embedded.*
- *the query is based on data coming from a connection* (not available in WINDEV Mobile).
Note: This type of report on query is available even if no analysis is associated with the WINDEV or WEBDEV project.

Tip: You have the ability to create, from the query editor, a report based on the current select query. To do so, on the "Query" pane, in the "File link" group, click "Create a report".

Note: An independent query or an embedded query can accept parameters.

The report can be easily printed from your WLanguage programs. See "Printing a report on query without parameter", page 121 for more details.

3.3.2 The different types of queries

Report on independent query

The query must be created before the report. When creating the report, all you have to do is specify the name of the query used.

If the query is modified in the query editor (deletion of result items for example), a synchronization screen is displayed when closing the query. This allows you to apply the modifications performed on the query to all elements that use this query (including reports).

Report on embedded query

The query is created when creating the report. This query will be modified from the report editor ("Edit the query" from the popup menu of the report). If modifications are performed in the query, the report editor lists these modifications as well as their incidence on the report.

Note: To switch from the embedded mode to the independent mode (which means to extract a query from a report):


1. Display the report based on the embedded query (in the report editor).
2. Select "Edit the query" from the popup menu of report. Other solution: on the "Modification" pane, in the "Report" group, click "Edit the query". The query is displayed in the query editor.
3. On the "Home" pane, in the "General" group, expand "Save" and select "Export .. To an independent query".
4. Save the new query.

Report on query with parameters

An independent query or an embedded query can accept parameters. In this case, one or more selection conditions depend on a parameter during the execution of the query. This parameter can correspond to the value of an edit control found a window.

For example, to print the invoice for customer X, all you have to do is create a report based on a query with parameters. In this query, the parameter will be the customer name for example. Therefore, the value of the parameter must be specified before the report is printed:

- during the test of the report based on a query with parameters, a window allows you to type the query parameters.
- at run time, the query parameters must be specified by `iInitReportQuery` (before using `iPrintReport`).



Important

If the query is created with Reports & Queries, there is no need to use `iInitReportQuery` to specify the query parameters. Indeed, when running a report on a query with parameters from Reports & Queries, a window allows you to enter the parameters of the query.

Note: See "Query with parameters", page 218 for more details.

3.4 Report on variable

A report on variable is used to directly print the information found in a WLanguage variable. This variable can be:

- a variable global to the project,
- a variable global to a window or to a page.
- a variable global to a set of procedures.

This variable can be a simple array, a class member, a structure member or an advanced variable member.

R&Q

Important

Reports & Queries cannot be used to create or modify reports on variable.

3.5 Report on a Table control

3.5.1 Overview

The report on Table control is used to directly print the information displayed in a Table control (browsing table, memory table or table on source) found:

- in a window or page of application. This window or page must be displayed when printing the report.
- in a window or in an internal page. The window (or the page) that displays the internal window (or the internal page) must be displayed.

R&Q

Important

Reports & Queries cannot be used to create reports on Table control. However, you have the ability to customize the reports on Table control created in the standard report editor (color, fonts, size, ...).

A report on Table control can print:

- all rows displayed in the Table control (ability to filter the data found in the Table control with the magnifier).
- the rows selected by the user (for the multiselection Table controls where the rows are selected by the Ctrl key). This option is not available in WEBDEV.

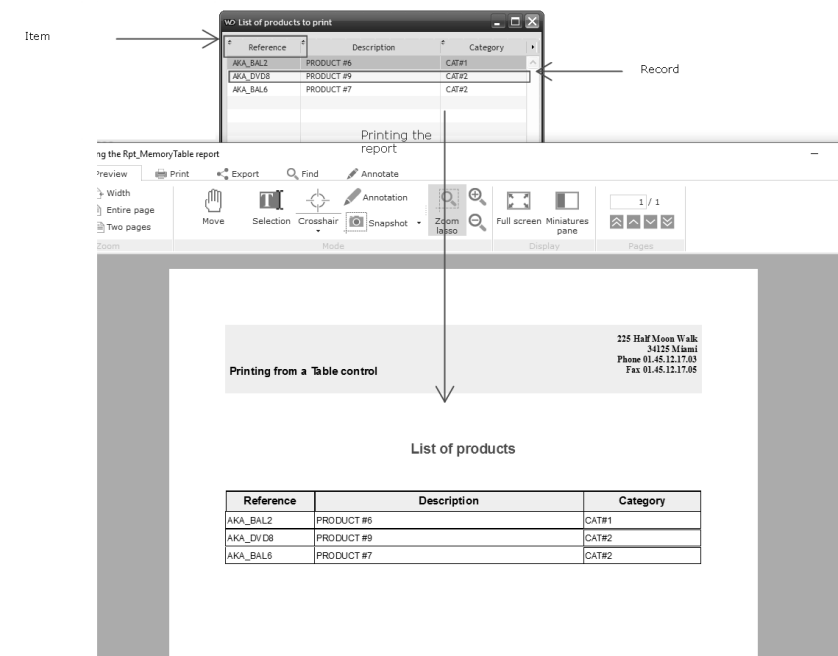
3.5.2 Characteristics

If the report is based on a Table control:

- Each row found in the Table control corresponds to a record.
- Each column of Table control corresponds to a Static control (even the invisible columns).
Note: The columns that must not be printed ("Export and print" option in the "General" tab of column description) are ignored.

For a table report, the order in which the columns are displayed in the report corresponds to the order of columns in the window currently edited. In WINDEV, if the user modifies the order of columns in the window, the order in which the columns are displayed in the report will not be modified dynamically.

Caution: The data printed in the report is sorted according to the sort column defined for the Table control.




A report on a Table control can be created:

- from the window editor. The created report corresponds to a table report.
- from the report editor. The wizard for report creation starts and it allows you to choose the type of report to create (table, form, ...).

Once the report on Table control is created, you can:

- modify the data source of report ("Data" tab of report description).
- add or delete Static controls linked to the Table control associated with the report.

Caution: The test of a report on Table control cannot be run from the report editor because the window or the page containing the Table control must be opened. To run the test of a report on Table control, you must:

- create (if necessary) a button in the window or in the page of Table control. This button is used to print the report on the Table control (iPrintReport).
Case of internal windows: If the Table control is found in an internal window, the button should be created in the internal window (and not in the host window).
- run the test of window (or page) containing the Table control then run the report test (via the  button).
Case of internal windows: the test must be run from the window that is hosting the internal window.

3.6 Report on Looper control

3.6.1 Overview

The report on Looper control is used to directly print the information displayed in a Looper control (browsing looper, memory looper or looper on source) found:

- in a window or page of application. This window or page must be displayed when printing the report.
- in an internal window or in an internal page. The window (or the page) that displays the internal window (or the internal page) must be displayed.

R&Q

Important

Reports & Queries cannot be used to create reports on Looper control. However, you have the ability to customize the reports on Looper control created in the standard report editor (color, fonts, size, ...).

A report on Looper control can print:

- all rows displayed in the Looper control.
- the rows selected by the user (for the multiselection Looper controls where the rows are selected by the Ctrl key). This option is not available in WEBDEV.

3.6.2 Characteristics

If the report is based on a Looper control:

- Each row found in the Looper control corresponds to a record.
- Each attribute of Looper control corresponds to a Static control (even the attributes that do not correspond to a value).

For a table report, the order in which the columns are displayed in the report corresponds by default to the order of attributes in the window currently edited.


A report on Looper control can be created:

- from the window editor. The created report corresponds to a table report.
- from the report editor. The wizard for report creation starts and it allows you to choose the type of report to create (table, form, ...).

Once the report on Looper control is created, you can:

- modify the data source of report ("Data" tab of report description).
- add or delete Static controls linked to the Looper control associated with the report.

Caution: The test of a report on Looper control cannot be run from the report editor because the window or the page containing the Looper control must be opened. To run the test of a report on Looper control, you must:

- create (if necessary) a button in the window or in the page of Looper control. This button is used to print the report on Looper control (iPrintReport).
Case of internal windows: If the Looper control is found in an internal window, the button should be created in the internal window (and not in the host window).
- run the test of the window containing the Looper control then run the report test (via the  button).
Case of internal windows: the test must be run from the window that is hosting the internal window.

3.7 Report on an Organizer control

3.7.1 Overview

A report on an Organizer control is used to directly print the information displayed in an Organizer control found:

- in a window or page of application. This window or page must be displayed when printing the report.
- in an internal window or in an internal page:
 - The window displaying the internal window must be displayed.
 - The page displaying the internal page must be displayed.

R&Q

Important

Reports & Queries cannot be used to create reports on an Organizer control. However, you have the ability to customize the reports on Organizer control created in the standard report editor (color, fonts, size, ...).

Note: Only an Organizer report can be created on an Organizer control.

3.7.2 Characteristics

All the rows and columns found in the Organizer control are displayed in the Organizer report.



When creating the report, you have the ability to specify the number of days to display as well as the display mode of resources (in row or in column).

A report based on an Organizer control can be created from the report editor: to do so, select an Organizer report based on an Organizer control.

Once the report on Organizer control is created, you can:

- modify the data source of report ("Data" tab of report description).
- modify the characteristics of the organizer ("General" tab of the organizer description).

Caution: The test of a report on Organizer control cannot be run from the report editor because the window or the page containing the Organizer control must be opened. To run the test of a report on Organizer control, you must:

- create (if necessary) a button in the window or in the page of the Organizer control. This button is used to print the report on the Organizer control (iPrintReport).
Case of internal windows or internal pages: If the Organizer control is found in an internal window or in an internal page, we advise you to create the button in the internal window/page (and not in the host window/page).
- run the test of the window containing the Organizer control then run the report test (via the  button).
Case of internal windows: the test must be run from the window that is hosting the internal window.
- run the test of the page containing the Organizer control then run the report test (via the  button).
Case of internal pages: the test must be run from the page that hosts the internal page.

3.8 Report on a Scheduler control

3.8.1 Overview

A report on a Scheduler control is used to directly print the information displayed in a Scheduler control found:

- in a window or page of application. This window or page must be displayed when printing the report.
- in an internal window or in an internal page:
 - The window displaying the internal window must be displayed.
 - The page displaying the internal page must be displayed.

Note: Only a Scheduler report can be created on a Scheduler control.

3.8.2 Characteristics



Each row and each column found in the Scheduler control are displayed in the Scheduler report.
When creating the report, you have the ability to specify the number of days to display as well as the display mode of resources (in row or in column).

A report on Scheduler control can be created from the report editor: to do so, select a Scheduler report based on a Scheduler control.

Once the report on Scheduler control is created, you can:

- modify the data source of report ("Data" tab of report description).
- modify the characteristics of the schedule ("General" tab of the schedule description).

Caution: The test of a report on a Scheduler control cannot be run from the report editor because the window or the page containing the Scheduler control must be opened. To run the test of a report on a Scheduler control, you must:

- create a button in the window or page of the Scheduler control. This button is used to print the report on the Scheduler control (**iPrintReport**).
Case of internal windows: If the Scheduler control is found in an internal window, the button should be created in the internal window (and not in the host window).
- run the test of the window containing the Scheduler control then run the report test (via the  button).
Case of internal windows: the test must be run from the window that is hosting the internal window.
- run the test of the page containing the Scheduler control then run the test of the report (via the  button).
Case of internal pages: the test must be run from the page that hosts the internal page.

R&Q

Important

Reports & Queries cannot be used to create reports on a Scheduler control. However, you have the ability to customize the reports on a Scheduler control created in the standard report editor (color, fonts, size, ...).

4. DESCRIBING AND HANDLING A REPORT

4.1 Overview

The steps for creating a report have been explained in the previous chapters. Once a report is created, the entire description of this report can be modified at any time:

- data source,
- controls found in the report,
- layout, ...

Only the report type cannot be modified. Indeed, too many specific parameters are involved. To change the report type, all you have to do is create a new report.

This chapter presents:

- how to view and modify the report description (defined when creating the report).
- how to open, close and save a report.
- the advanced modifications that can be performed on a report (use of a form, modification of the skin template, ...).

4.2 Viewing and modifying the report description

The main options of the report can be accessed in the report description ("Report description" from the popup menu).

Let's take a quick look at the different tabs of report description.

Note: The main options of report can also be accessed via the "Modifier" pane.

4.2.1 Managing the data source

The data source of a report can be configured and modified at any time. In most cases, these operations are performed from the description window of the report.

You have the ability to:

- modify the data source of report ("Data" tab of report description):
 - by changing the type of data source for the report (switching from a report based on a query to a report based on a data file or on a memory zone for example).
 - by changing the parameters of selected data source (modify the query or the sort, implement bounds, ...).
- configure the operating mode of report if the data source is empty ("GUI" tab of report description). If the data source is empty, the report editor gives you the ability to print the report or to display a warning message without printing the report. The warning message can be configured.

Note: If the report is based on a query, select "Edit the query" from the popup menu to directly edit the query in the query editor. Then, this query can be modified.

4.2.2 Display options in the print preview

The configuration of display options in the print preview is performed in the "GUI" tab of report description. You have the ability to:

- specify the export formats proposed by the print preview. The selected formats will be available for the user.
- configure the message displayed if the data source is empty.
- configure the message displayed while selecting and sorting records.

4.2.3 Managing blocks

The report blocks can be managed in the report currently edited or in the description window of report. The blocks can be managed:

- from the options found in the popup menu of blocks (right mouse click on the caption of blocks).
- via the description window of the report ("Report description" from the popup menu), "Blocks" tab.

See "Blocks of a report", page 59 for more details.

4.2.4 Print format

The "Format" tab of the description window of report is used to configure:

- the page format (or the format of label sheet).
- the indent of controls (in order for the control text not to be stuck to the control border).
- the margins.
- the page orientation (portrait or landscape).
- the format of labels if the report is a label report.
- the number of columns.
You also have the ability to create multi-column reports.
- the number of pages in width. In this case, a large report (whose size exceeds the size of a paper sheet) will be printed on several pages. The end user will only have to assemble the pages. See the online help for more details.

4.2.5 Options

The "Options" tab found in the description window of report is used to configure:

- the operating mode of report if the page format is not supported by the printer.
- the position of blocks when printing the report.
- the table style if the report contains a table.
- the list of templates used by the report.
- the style of labels if the report is a label report.

This tab is also used to:

- configure the report for using Reports and Queries. See "Distributing Reports & Queries", page 233 for more details.
- specify whether the compiled code must be included. This option is required in very rare cases only.
- configure some specific parameters:
 - Whether the pages must be sent one by one to the print spooler (print performed page after page). In this case, the print is sent to the print spooler after each printed page.

- Whether the space characters that fill the items must be taken into account. This option affects the self-expanding controls associated with a text item filled with space characters (backward compatibility with WINDEV 5.5).
- Whether the first page is a cover page. In this case, the first page will be displayed on the right in the preview (in side-by-side display mode).
- Whether the code of cloned controls must be run. This option is checked by default. If this option is unchecked, the report behavior will correspond to the one of versions 21 and earlier: the code of cloned controls will not be run.
- Overload the programming standard defined at project level (via the "Programming standard" button).

4.2.6 Form and background image

Most types of reports can be created from a form. The "Background" tab allows you to define the image used as form in the report. This image can be printed (or not) with the report. This option is very useful for the reports that must be printed on a pre-printed form. The form is included in the report, allowing you to easily position the records to print. Printing has now become hassle-free. See "Using a form in a report", page 54 for more details.

4.2.7 Language


A report can be multilingual: this report can be displayed and printed in several languages.

The "Language" tab is used to configure:

- the languages supported by the current report.
 - the language displayed in the report editor: this language will be used by default when the report is displayed in the editor.
- The following information can be typed in several languages:
- all the messages displayed while the report is printed. These messages are translated in the different description windows.
 - the caption of Static controls or RTF controls: To translate these captions, you must:
 - modify the language displayed in the report: on the "Display" pane, in the "Options" group, expand "Language displayed" and select the requested language.
 - enter the translations in the report in edit.
 - the input mask of item controls, calculated controls or preset controls.
 - the image files used in the report.
 - the form of the report.

4.3 Opening, closing and saving a report

To open a report, you have the ability to use one of the following methods:



- on the "Home" pane, in the "General" group, expand "Open" and select "Open". Then, select the "WDE" file to open.
- click  among the quick access buttons and select "Open". Then, select the "WDE" file to open.
- double-click a "WDE" file directly.
- double-click the name of a report in the "Project explorer" pane.

To close the current report, use one of the following methods:

- click the cross found in the top right corner of the title bar of the report currently edited.
- on the "Home" pane, in the "General" group, expand "Close" and select "Close".

If the report contains unsaved modifications, the report editor automatically proposes to save the report.

To save the current report, use one of the following methods:

- on the "Home" pane, in the "General" group, click .
- press Ctrl + S.
- click  among the quick access buttons.

Note: To save a report while renaming it, on the "Home" pane, in the "General" group, expand "Save" and select "Save as".

Automatic retrieval

To avoid problems when a system lock occurs (power outage for example), WINDEV, WEBDEV and WINDEV Mobile propose a system used to automatically retrieve the elements opened in the different editors

When this system is enabled:

- an automatic backup of the opened elements is regularly performed at specific time intervals (every 10 minutes by default).
- when a report is re-opened after a system lock, a window for automatic retrieval is displayed. This window proposes to retrieve:
 - the report as it was during the last backup performed from the report editor.
 - the report automatically saved.
 - The selected version is automatically opened in the editor. The non-selected version is available in the window for managing the versions and it can be opened at any time.

To enable the system for automatic retrieval:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of xxx".
2. Display the "General" tab and check "Backup for automatic retrieval".
3. Specify the time-out for the automatic backup.
4. Validate.

To retrieve a version in the window for version management:

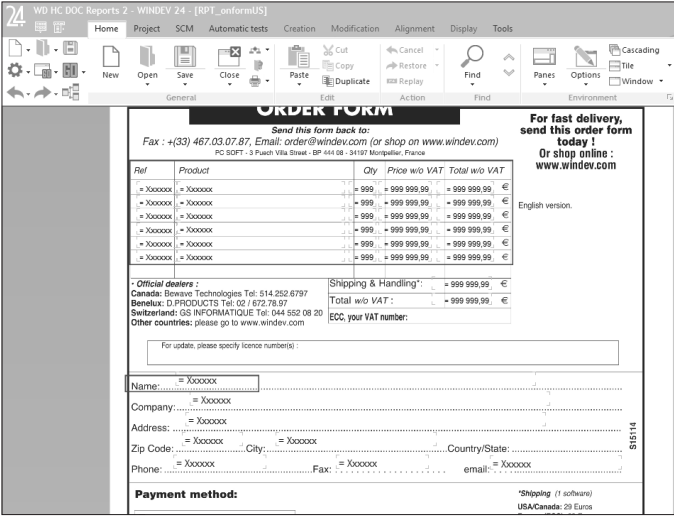
1. On the "Home" pane, in the "General" group, expand "Save" and select "Versions".
2. Select the version to retrieve.
3. Validate.

4.4 Using a form in a report

The report editor gives you the ability to print reports on pre-printed forms (image file or PDF file). For example, the products ordered by the customers can be directly printed on an order form.

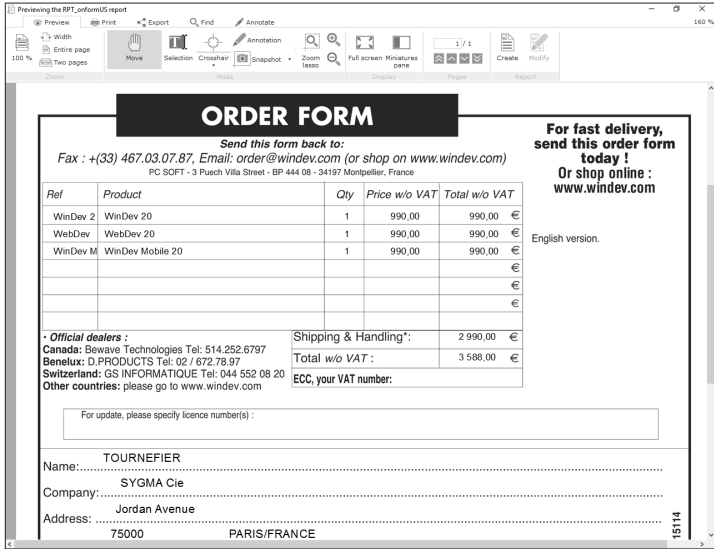
In edit, the form is displayed in background image of the report: therefore, the controls can be positioned to the precision of a millimeter.

For example:



At run time, you can specify whether the image of the form must be printed or not. The image of the form is not printed by default.

For example:



Note: To print reports on pre-printed forms, you can:

- create a "Form" report.
- include a form in an existing report.

4.4.1 Information specific to the form

The information specific to the form is as follows:

- **the file containing the image of the form.** The size of the report is automatically calculated according to the size of the form image.
For the A4 format, the size of an image in 75 DPI must be set to 595 x 842 pixels. For an A4 format, the size of an image in 150 DPI must be set to 1275 x 1753 pixels.
If the image of the form corresponds to a PDF file containing several pages, you can:
 - print all the pages found in the PDF file. A report made of several pages is created in the report editor.
 - specify the number of the page that must be used as report background. The report will print a single page of the PDF file.
- **whether the image of the form must be printed or not.**
Note: If the image of the form is printed, this image will not be printed beyond the physical margins of the printer.
- **the division of the form according to the number of records printed per page:**
 - for example, if the form contains the characteristics of a single customer, a single record must be printed per page.
In this case, the created report will only contain the *Body* block.
 - for example, if the form contains the characteristics of several customers, several records must be printed per page.
In this case, the created report contains a *Page header* block, a *Body* block and a *Page footer* block. The *Body* block will be repeated for each record printed on the page. To print the report properly, you must modify the position of the report controls and the position of the form image in the *Body* block. See "Image of the form", page 57 for more details.

4.4.2 Including a form in a report

Creating (if necessary) the file corresponding to the form.

If you only have a paper copy of the form, we advise you to scan the form at 100% and to save it as an image.

For an A4 format, the size of an image in 75 DPI must be set to 595 x 842 pixels.

For an A4 format, the size of an image in 150 DPI must be set to 1275 x 1753 pixels.

The following image formats are supported in Windows:

- Bitmap (*.BMP),
- Graphics Interchange Format (*.gif),
- Joint Picture Experts Group (*.jpg ; *.jpeg),
- Kodak Photo CD (*.pcd),
- PaintBrush (*.pcx),
- Adobe Photoshop Format (*.psd),
- TrueVision TARGA (*.tga),
- Tagged Image File Format (*.tif ; *.tiff),

Note: the "tiled tiff" option of tiff 6.0 standard is not supported. In this case, we advise you to save the image in tiff 5.0 standard.

- Portable Network Graphics (*.png),
 - Windows MetaFiles (*.emf ; *.wmf),
 - Icons (*.ico ; *.icw),
- Notes:
- HQ display (High Quality) available.
 - Management of sheets in transparent BMP format (32 bits).
- Cursors (*.cur),
 - Portable Document Format (*.pdf),
 - Scalable Vector Graphics (*.svg),
 - WINDEV images (generated by the image editor, *.WDPIC).

To include a form in a report:

1. Display the "Background" tab of the report description ("Report description" from the popup menu of the report).
2. Select the image of the form or the PDF file corresponding to the form.
3. If the form is a PDF file containing several pages, specify:
 - whether all the pages found in the PDF file must be printed. The report editor will display all the pages of the PDF file. The controls required by the report can be positioned on the different pages of the report in the editor.
 - the page containing the form to use. This page number can be modified by programming with **..PageNumber**. The report displayed in the report editor will contain a single page.
4. Specify whether the image of the form must be printed or not.
 - If the image of the form is printed, this image will not be printed beyond the physical margins of the printer. To print the entire image of the form, we advise you to check "Ignore the physical margins of the printer".
 - If the image of the form must not be printed with the report (print on a pre-printed form for example), the option "Ignore the physical margins of the printer" must not be checked. Otherwise, the printed controls will be shifted in relation to the pre-printed form.

4.4.3 Image of the form

The image of the form is displayed in the report background. The image is displayed in the background of each report block.

If several blocks are found in the page, you have the ability to position the image in each one of the blocks.

To move the form image displayed in each block:

1. On the "Modification" pane, in the "Alignment" group, click "Move the background image" . The mouse cursor turns into hand.
2. Click the block where the image of the form must be moved.
3. To move the image of a form for a block:
 - to the top or to the bottom: keep the mouse button down and move the mouse.
 - to the left or to the right: hold the mouse button and the Shift# key down and move the mouse.
4. To disable the mode for moving the image of the form, on the "Modification" pane, in the "Alignment" group, click "Move the background image" again.

Note: The cursor must be moved inside the block area in order to be taken into account.

4.4.4 Multipage report (based on a PDF file)

In order for the report to display all the pages of a PDF file, if the report is based on a data file or on a query, the different pages will be printed for each record of the data source used.

Caution: This type of report can consume a large amount of memory resources (especially during the print preview).

Note: Using a PDF file in page background

- Using a PDF file in page background requires the GDI+ framework.
- In the editor, we advise you to use a zoom less than or equal to 100% to better view the PDF file used in report background.

4.5 Changing the skin template associated with the report

When describing a report, a skin template can be applied to the report. A skin template is used to define the characteristics of the report interface: background image, size, style of controls, ... A skin template can be applied to one or more reports (to harmonize the style of application reports for example).

To change the skin template or to apply a preset skin template to a report:

1. On the "Modification" pane, in the "Templates and skin template" group, click "Apply a skin template".
2. In the window that is displayed, select the requested skin template and validate.

Note: You can create your own report skin templates. See the online help for more details (keyword: "Skin template, Create a report skin template").

R&Q

Important

Reports & Queries cannot be used to create report skin templates.

4.6 Changing the report template associated with the report

When describing a report, a template can be applied to the report. A report template is used to define the characteristics common to a set of reports (presence of different controls, specific code to run, ...). A report template can be applied to one or more reports. The characteristics of the template can be overloaded by the report that is using the template.

To change a template or to apply a template to a report:

1. On the "Modification" pane, in the "Templates and skin template" group, expand "Templates" and select "List of templates used".
2. In the window that is displayed, select the requested template.

Note: See the online help for more details (keyword: "Template, Report template").

R&Q

Important

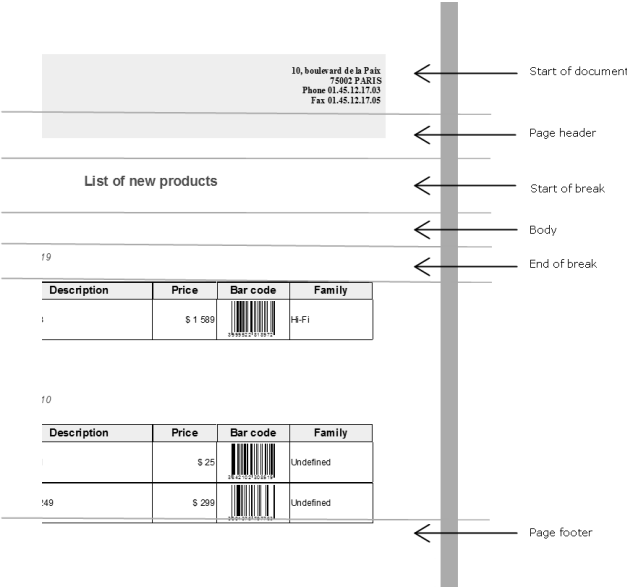
Reports & Queries cannot be used to create or modify report templates.

5.BLOCKS OF A REPORT

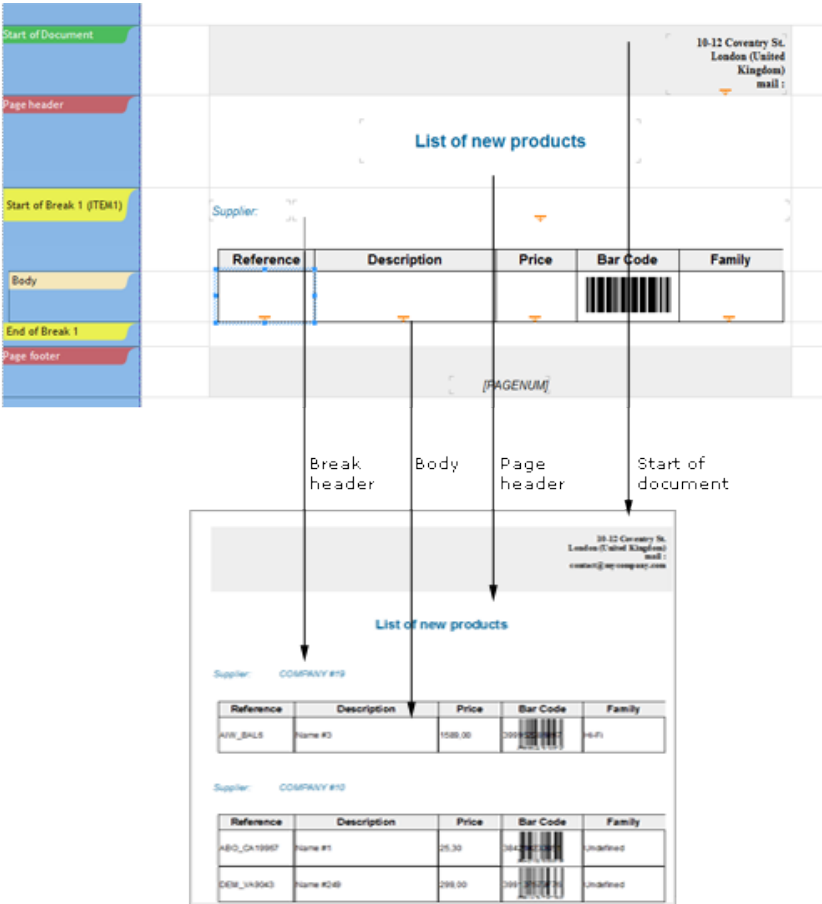
5.1 Overview

A printed report includes several sections:

- a start of document,
- a page header,
- the page body,
- a page footer,
- an end of document,
- one or several breaks, if the report is sorted.



To manage the different sections of the report, the report editor divides the structure of the report into blocks. A WLanguage process can be associated with each block.



Report in edit and preview of report execution

5.2 The different types of blocks

A report includes different types of blocks. Each block:

- can contain all types of controls.
- has specific parameters and print options.
- is associated with different WLanguage processes.

A report contains at least the *Body* block and it can only contain one block of each type (except for the iteration block and for the break blocks).

The different types of blocks are as follows (according to the order in which they appear in the report editor):

- Start of Document block,
- Page Header block,
- Start of Break block,
- Body block,
- Body Complement block,
- End of Break block,
- Page Footer block,
- End of Document block,
- Iteration block (or substitution block).

Notes:

- To select a block, all you have to do is click its caption displayed on the left of the reports.
- By default, the caption of the blocks is entirely displayed on the left of reports. To display the abbreviated captions, on the "Display" pane, in the "Viewing" group, uncheck "Full name of blocks" (F10 key).
- During the click on a block name or in a report section, the delimiters of the block are highlighted.

5.2.1 Start of Document block

The **Start of document** block is printed once at the beginning of the report execution. In most cases, this block contains the title of the document as well as today's date and time.

In the process associated with this type of block, you can display, for example, a window warning of imminent print.

5.2.2 Page Header block

The **Page header** block is a report block created in the report editor. The **Page header** block is printed on each new page. This block is run after the *Start of document* block if it exists.

The **Page header** block contains the information that will be printed at the top of each page, the title or a preset control indicating the page number for example.

5.2.3 Break Header block

The **Start of break** block is optional in the reports created with the report editor. It is created if a break is requested on a sort argument. The *End of break* block necessarily exists if this block exists.

This block is run whenever the value of the sort argument changes, before the first iteration of the *Body* block for a new value of the sort argument.

Note: The *Start and End of break* blocks cannot be dissociated. However, one of these blocks can be made invisible if necessary (in the description window of the block or by **..Visible**).

5.2.4 Body block

The **Body** block corresponds to the main block of the report. This block is necessarily found in each report. This block is printed as many times as the number of records to print.

For the free reports, the **Body** block can be repeated as many times as necessary to fill the page ("Repeated until the end of the page" checked in the "GUI" tab of the block characteristics).

Note: The **Body** block cannot be deleted. However, this block can be made invisible (in the description window of the block or by `..Visible`).

5.2.5 Body Complement block

The **Body complement** block is a block found in a report created with the report editor. The **Body complement** block is linked to the *Body* block. This block is used to fill the print until the page footer if the number of lines issued from the *Body* block is not sufficient.

This block is printed as soon as a free space is found between the *Body* block and the *Page footer* block.

5.2.6 Break Footer block

The **End of break** block is an optional block found in the reports created with the report editor. This block is created if a break is requested on a sort argument. The *Start of break* block necessarily exists if this block exists.

This block is run whenever the value of the sort argument changes, after the last iteration of the *Body* block for a value of the sort argument. In most cases, it contains the counter or total controls for the records corresponding to the sort argument.

Note: The *Start of break* block and the **End of break** block cannot be dissociated. However, one of these blocks can be made invisible (in the description window of the block or by `..Visible`).

5.2.7 Page Footer block

The **Page footer** block is one of the blocks found in a report created with the report editor. The **Page footer** block is run at the end of each page.

The **Page footer** block contains information that must be printed at the bottom of each page, counters and/or totals generated when creating the document or added thereafter, or the number of the current page in relation to the total number of pages for example.

Caution: The self-stretchability of controls is not managed in the "Page footer" block.

5.2.8 End of Document block

The *End of document* block is run once at the end of the report execution. In most cases, the *End of document* block contains the counters and/or the totals (generated when creating the document or added afterwards). In the process associated with this type of block, you have the ability to display a window indicating the end of print.

5.2.9 Iteration block


The **Iteration** block is a free block, created manually in the report editor. It can be used like any other type of block. For example, an **Iteration** block can be used to replace a *Page header* block in order to differentiate between the even pages and the odd pages.

The block substitution is performed by programming with `iSubstBlock`. You also have the ability to print a specific block with `iPrintBlock`.

Note: The planes are available in the reports from version 19. In most cases, using planes in the reports allows you to replace the iteration blocks. Using planes in the reports is easier than using iteration blocks. See the online help for more details.

5.3 Characteristics of report blocks

To display the description window of a block, perform one of the following operations:

- double-click the caption of the block.
- on the "Display" pane, click the group icon of "Options" group () or press the F12 key. In the window that is displayed, select the block and click the "Description" button.
- select "Block description" from the popup menu of the block (right mouse click on the block caption).

For example, description window of a "Break header" block:



5.4 Parameters and print options of blocks

The parameters and the print options of a block are defined in the "GUI" tab of its description (double click on the caption of the block). The parameters and the print options of the report blocks are as follows:

- **Visibility:**
 - **Visible:**
The block can be displayed or not. If a block is invisible, the block and its controls will not be displayed in the report.
This characteristic can be modified by programming with `..Visible`.
 - **Always visible:**
The block and its controls will be displayed on all the report pages.
This option is available for the Page Header and Page Footer blocks.
 - **Always invisible:**
The block and its controls will not be displayed in the report.
This option is available for the Page Header and Page Footer blocks.
 - **Visible on the first page only:**
The block and its controls will be displayed on the first report page only.
This option is available for the Page Header and Page Footer blocks.
 - **Invisible on the first page only:**
The block and its controls will be displayed on all the report pages except for the first page.
This option is available for the Page Header and Page Footer blocks.

- **Breakability:**
 - **Breakable block:**

If a block is breakable and there is not enough space left on a page to print the entire block, the content of the block is printed on several pages.

If a block is unbreakable and there is not enough space on the page to print the entire block, the content of the block is printed on the next page.
- **Filling the page:**
 - **Completion until the end of the page:**

A block can be stretched until it reaches the end of the page.

Note: This option is useful if the block has a background image or a background color.
 - **Extend the table columns until the end of the page** (for the blocks containing a table):

The table columns can be extended until they reach the end of the page.

Note: To extend the table columns until the page footer, you also have the ability to add a *Body complement* block. The *Body complement* blocks are automatically stretched until they reach the end of the page.
 - **Repeated until the end of the page:**

The block can be copied until it reaches the end of the page.

This option is available especially for the Body block.
 - **Keep space for the page footer when it is invisible** (for the Page Footer blocks only):

By default, the space occupied by the Page Footer block will always be kept in the report pages (no matter whether the block is visible or not). With this option, the space occupied by the Page Footer block will be kept only in the pages where this block is visible. This option is used to optimize the method for filling the report pages.
 - **Reinitialize the number of pages after the block print:** By default, the number of pages present in the "Number of pages" and "Number of pages and page number" preset controls are not reinitialized during printing. This option allows you to automatically reinitialize these preset controls once the block is printed (during a break or at the end of the print).

This option can be used when printing a set of invoices for example, to number the pages of each invoice independently.
- **Automatic break:**
 - **Page break after the block:**

A page break can be automatically generated once the block is printed.
 - **Column break after the block:** (only for the Body block of multi-column reports)

A column break can be automatically generated once the block is printed.

5.5 Handling the blocks in the report editor

5.5.1 Create a block

To create a block in a report, perform one of the following operations:

- on the "Creation" pane, in the "Structure" group, expand "Blocks".
- select "Add a block" from the popup menu of the report (right mouse click on the report currently created).
- click on "Add a block" in the "Blocks" tab of the report description.

Note: The *Break header and footer* blocks cannot be dissociated. However, one of these blocks can be made invisible (in the description window of the block or by *..Visible*).

5.5.2 Delete a block

To delete a report block, use:

- "Delete the block/break" from the popup menu of the block to delete (right mouse click on the caption of the block).
- the "Delete a block" button in the "Blocks" tab of the report description.
- the Del key (the block to delete must be selected).

Note: The *Body* block cannot be deleted. However, this block can be made invisible (in the description window of the block or by *..Visible*).

5.5.3 Reorganize the blocks

The report editor allows you to reorganize the blocks found in a report. When printing the report, the blocks will be printed according to the order specified in the report editor. The order in which the processes of the blocks are run corresponds to the order in which the blocks are printed.

To reorganize the report blocks:

1. Display the description window of the report ("Report description" from the popup menu of the report).
2. Display the "Blocks" tab.
3. Modify the print order of the blocks:

Start of Document

Position

☒

BEFORE the "Page header" block

☐

AFTER the "Page header" block

Page header

Body

Page footer

Position

☒

In page footer

☐

Stuck to the previous blocks

End of document

Position

☒

BEFORE the "Page footer" block☐☒☐

Note: The print order of the blocks can also be modified in the description window of each block ("Block description" from the popup menu of the block).

5.6 Handle the blocks by programming

Each report block is associated with different processes. These processes are used to improve the operating mode of the report via the programming functions of WLanguage.

Furthermore, each block can be handled in WLanguage: to do so, use the name of the block in the WLanguage code.

5.6.1 Processes of report blocks

The following processes are associated with the report blocks:

Processes	Runtime condition
Initialization	<p>Run when initializing the block. The order in which the processes are run when printing a report (including during a test) is as follows:</p> <ul style="list-style-type: none">"Opening" process of the report."Initialization" process of report blocks."Initialization" process of controls. <p>This process is used to:</p> <ul style="list-style-type: none">initialize the "Calculated" controls.initialize the global variables declared in the "Opening" process of the report.perform block substitutions.
Before printing	<p>Before printing a block, the following operations are performed:</p> <ul style="list-style-type: none">reading the current record.assigning the values read in the controls of the block.performing calculations on the values of items.running the "Before printing" process of block.performing calculations on the controls of the block. <p>When this process is called, the report editor checks whether there is enough space left on the current page to print the current block (case of non-breakable block). If there is not enough space to print the current non-breakable block, it will be printed on the next page (as well as the Page Footer, Page Header and Body Complement blocks if necessary).</p> <p>If the block is auto-sizable, the "Before printing" process is run before knowing whether the block can be printed on the page.</p> <p>This process is used to:</p> <ul style="list-style-type: none">see and modify the controls of the relevant block.assign images.run iPageEnd in the Break Header block. <p>Caution: This process cannot perform any substitution of the current block. However, it can ask for the substitution of another block.</p>
After printing	<p>Run once the block is printed.</p> <p>This process is used to:</p> <ul style="list-style-type: none">see and modify the controls of the relevant block.assign images.reset to zero the controls that have just been printed.substitute the blocks (taken into account for the next blocks).print an <i>Iteration</i> block (iPrintBlock).
Break? (Break header block only)	<p>Run before the "Before printing" process of the <i>End of break</i> block. See "Break process", page 138 for more details.</p>

Reminder: If the destination of the print is an HTML file, the Page Header and Page Footer blocks will be edited once only.

To view the processes specific to a block, use:

- "Block code" from the popup menu of the block (right mouse click on the header of the block).
- the "Code" icon of the block description.

Some functions and properties are specifically intended for handling the blocks.

Note: The processes associated with a report block are run when this block is printed. A WLanguage code can be used in these processes. See "Sequence of processes associated with a report", page 135 for more details.

5.6.2 Handling a block in a WLanguage process

To handle a report block in a WLanguage process, you must:

- use the name of this block directly. This syntax is valid only if the block is handled from the report to which it belongs.
- use the following notation: "<ReportName>.<BlockName>".

A block can be handled:

- in a WLanguage function.
- with a WLanguage property.
- in direct assignment.

For example:

```
// Print an iteration block
iPrintBlock("ITERATION1")
// Make the "Body" block invisible
Body.Visible = False
```

5.6.3 Properties associated with the report blocks

The following properties can be used with the report blocks:

Anchor	Used to manage the anchor of a control in a report.
AnchorRateHeight	Used to find out and modify the anchor rate in height of a control.
Border	Used to find out and modify the characteristics of the border for a control found in a report or in a report block.
BrushColor	Used to manage the background color of a control or report block.
Height	Used to find out and modify the height of a control or block in a report.
Image	Used to find out and modify the background image of a control or block.
InitialHeight	Used to find out the initial height of a control or block in a report.
Name	Used to find out the name of a control, report or report block.
Note	Used to: <ul style="list-style-type: none">Find out the notes associated with a control, a window, a page or a report.Modify the notes associated with a control, a window a page or a report.
NoteTitle	Used to: <ul style="list-style-type: none">Find out the title of the notes used in the program documentation of a control, window, page or report.Modify the title of the notes used in the program documentation of a control, window, page or report.

Plane	Used to find out and modify the active plane of a report or the plane to which a control belongs.
Process	Used to modify the processes that will be run by the available events on a control or report, ...
State	Used to find out and modify the display status of an element.
Visible	Used to: <ul style="list-style-type: none">Find out whether an element is visible.Make an element visible or invisible.
Width	Used to find out and modify the width of a control or block in a report.

6.REPORT CONTROLS

6.1 The different types of controls

Several types of controls can be handled in a report created by the report editor:

- Static control (whose type is text, currency, RTF, HTML).
- Calculated control.
- Image control.
- Bar Code control.
- Preset control.
- Chart control.
- Shape control.
- Check Box control.
- Internal Report control (advanced type of control).
- URL Link control.
- Edit control.
- Signature control.

Let's now give a short description of these types of controls, as well as their characteristics and the operations that can be performed on these controls. See the online help for more details.

6.1.1 Static control

The "Static" control is used to display different types of data:

- text, RTF, HTML,
- numeric, currency, currency + Euro,
- date,
- time, duration.

To specify the type of data displayed in the control:

1. Display the "General" tab in the description window of Static control ("Description" from the popup menu of control).
2. Select the type of information displayed ("Type" combo box).
3. Validate.

The displayed data can contain:

- the content of an item (or variable) found in the data source of report.
Note: The item or the variable linked to the control is defined in the "Link" tab of the description window of control.
- the content of another control found in the report.
- a value defined by programming.

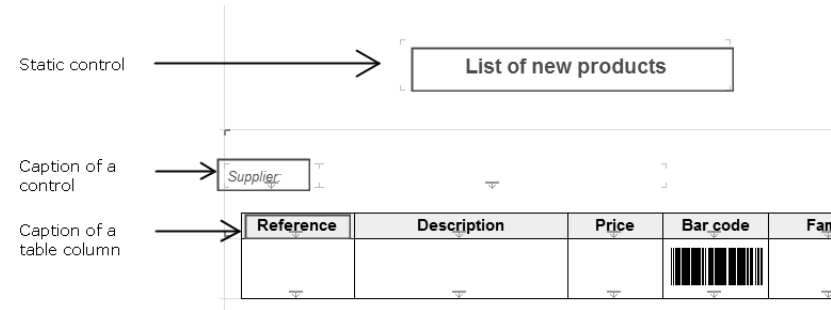
To insert the value of a report item or control into the text, the name of item or control must be typed between the [% and %] characters.

For example, the caption can correspond to: The [%SupplierName%] supplier sells the [%ProductName%] product at [%PriceOT%]. The SupplierName, ProductName and PriceOT items will be replaced by their value when printing the report.

Note: If the Static control must display a text on several lines, all you have to do is choose the "Multiline text" type. In this case, the control size must be large enough.

Reminder: The "Auto-Sizable" option ("GUI" tab in the description window of control) is used to automatically enlarge the control according to its content during the print.

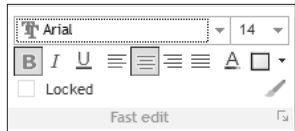
Examples:



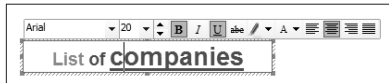
Modifying the caption

The caption of a "Static" control can be modified:

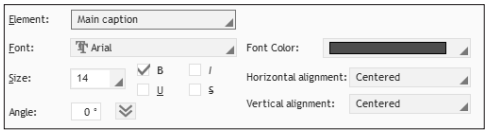
- by programming (**..Caption**).
- in the report editor. In this case, two methods are available:
 - Method 1:
 1. Select the control. The selection handles appear around the control.
 2. Click the control again. The mouse cursor turns into an input cursor.
 3. Type the new caption of control.
 4. Modify (if necessary) the formatting of the caption:
 - for a text caption: via the formatting options found on the "Modification" pane, in the "Fast edit" group of ribbon:



- for an RTF caption: via the formatting toolbar that is displayed above the control:



- Method 2:
 1. Select the control.
 2. Display the description window of control ("Description" from the popup menu of control).
 3. Type the new caption in each language supported by the control.
 4. Modify (if necessary) the formatting in the "Style" tab.



Multilingual captions

If your application or your site is multilingual, a different caption can be typed for each language supported by the report.

The corresponding caption will be printed according to the runtime language of project.

"Currency + Euro" Static control

For a "Currency + Euro" Static control, you have the ability to manage:

- the stored currency: currency of data found in the data source of report. If the control is handled by programming, the currency used for the value will be the stored currency.
- the display currency: currency used to display the value when printing the report. This currency can differ from the stored currency. The conversion is automatically performed.
- the dual display: two "Currency + Euro" controls can be linked. In this case, the report editor distinguishes between two types of controls:
 - the base control: this control contains the value found in the data source of report.
 - the tracker control: this control is used to display the value of base control in the selected currency.

See the online help for more details.

6.1.2 Calculated control

The "Calculated" control is used to perform calculations in a report. These calculations can be made:

- on an item of the data source.
- on a control found in the report.
- by programming.

Examples:

Product caption	Unit price BT	Quantity	Unit price	
PRODUCT #1	92,92	1	15,30	← Calculation performed on an item
			Total BT	269,80
			Total IOT	322,68 ← Calculation performed on the table


Specific "Calculated" control: "Currency + Euro"

For a "Currency + Euro" Calculated control, you have the ability to manage:

- the stored currency: currency of data found in the data source of report. If the control is handled by programming, the currency used for the value will be the stored currency.
- the display currency: currency used to display the value when printing the report. This currency can differ from the stored currency. The conversion is automatically performed.
- the dual display: two "Currency + Euro" controls can be linked. In this case, the report editor distinguishes between two types of controls:
 - the base control: this control contains the value found in the data source of report.
 - the tracker control: this control is used to display the value of base control in the selected currency.

6.1.3 Image control

An "Image" control is used to display an image in the report. The image displayed in the control can:


- be selected from the description window of control ("General" tab) by clicking the  button:
 - The "Browse" option is used to select an image file accessible from your computer.
 - The "Catalog" button is used to choose an image found in the image catalog of WINDEV.
- correspond to the content of an item found in the data source (for example, if the data source is a HFSQL data file, the item can be an Image memo or a character string containing the full image path). The "Image" control is linked to this item.

Note: The item (or the variable) linked to the control is defined in the "Link" tab of the description window of control.

Multilingual images

If your application or your site is multilingual, a different image can be selected for each language supported by the report. This option is useful when the images contain a text for example.

The corresponding image will be printed according to the runtime language of project.

To select an image per language, click the  button and select "Multilingual".

6.1.4 Bar Code control

The "Bar code" control is used to easily print bar codes in your reports.

The value of a "Bar code" control corresponds to an alphanumeric string.

The "Bar code" control can be associated with an item found in the data source of report or it can be assigned by programming. Note: You also have the ability to specify an initial value in the "General" tab of the description window of control.

The report editor proposes several types of bar codes that can be printed horizontally or vertically.

Example:



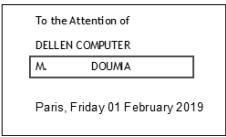
Note: The item or the variable linked to the control is defined in the "Link" tab of the description window of control.

6.1.5 Preset control

The preset control is used to display current information without any programming. A "Preset" control can be inserted into a report to display the following information:

- System date,
- System time,
- Number of the current page,
- Total number of pages in the report,
- Number of the current page in relation to the total number of pages in the report.

Example:



To modify the type of information displayed in a "Preset" control:

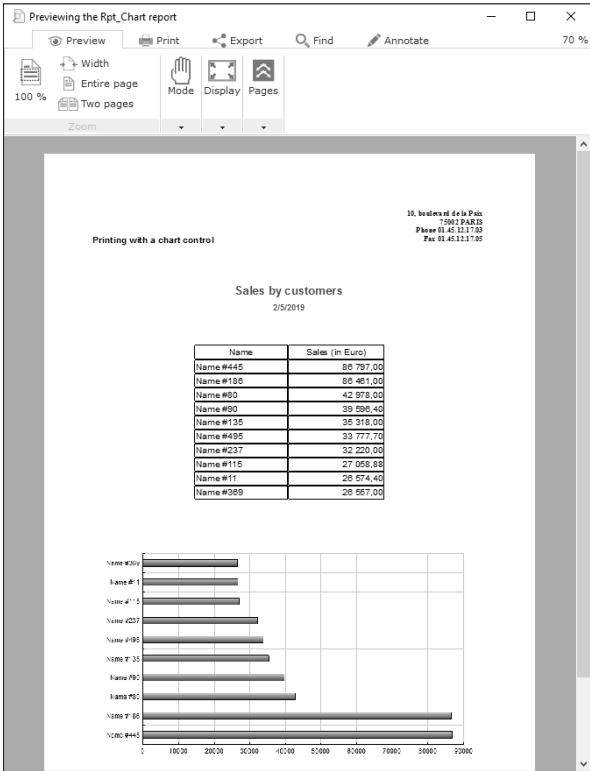
1. Display the "General" tab in the description window of "Preset" control ("Description" from the popup menu of control).
2. Select the type of information displayed ("Type" combo box).
3. Validate.

6.1.6 Chart control

The "Chart" control is used to view and print a chart in your reports. The available types of charts are as follows:

- Pie, donut, semi-circular.
- Column (clustered or stacked, vertical or horizontal).
- Line, area.
- Scatter, radar.
- Bubble chart.
- Stock chart (Candlestick, MinMax or BarChart).

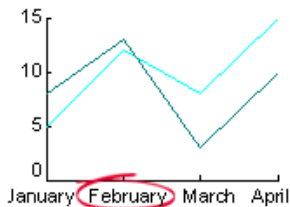
Example:



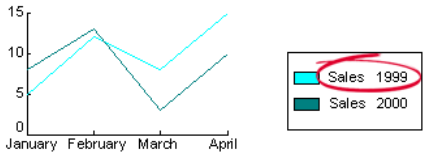
Information specific to chart creation

A wizard for chart creation is automatically started. This wizard is used to enter the information required to create the Chart control. This information is as follows:

- the type of chart (stock, pie, line, ...).
- the chart parameters: title, legend, 3D chart, display of labels, ...
- the parameters of axes: titles, grid, ...
- the characteristics of series:
 - the source and data displayed by the category labels (programming, data file, query, WLanguage array, data source of report).



- the name of series (used for the series labels):



- the source of series (programming, data file, query, WLanguage array, data source of report).
- the data displayed by the series.
- the line thickness, the color and the type of dots.

Note: if the data source corresponds to the data source of report, you have the ability to define:

- the fill block,
- the control or the item corresponding to the series,
- the selection mode of data.

6.1.7 Shape control

A "Shape" control is used to insert geometric shapes into a report. You have the ability to insert:

- horizontal or vertical lines,
- circles,
- rectangles (with straight or rounded borders).

The line style, thickness and color can be entirely configured.

6.1.8 Check Box control

A "Check box" control corresponds to a box to check.

If the control value is set to True, the box is checked: ☒

If the control value is set to False, the box is unchecked: ☐

A "Check box" control can be linked to an item or variable found in the data source of report. For example, if the data source corresponds to a HFSQL file, the item can be a boolean or a check box.

The size, the background color and the color of check box can be configured.

Example:

<input checked="" type="checkbox"/> Ms	<input type="checkbox"/> Mrs	<input type="checkbox"/> Mr
Name DOUMIA		
Company DELLEN COMP		

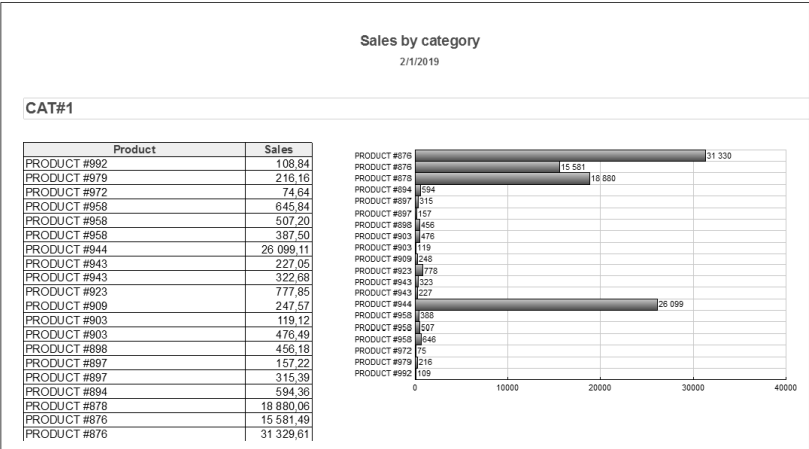
Note: The item or the variable linked to the control is defined in the "Link" tab of the description window of control.

6.1.9 Internal Report control

The "Internal report" controls are used to print several reports (secondary reports) in a single report (main report) without programming.

For example, the following main report is made of two secondary reports printed side-by-side:

- a report displaying the data in a table.
- a report displaying the data in a chart.



See online help for more details.

Benefits of internal reports compared to the nested reports

The report editor can also be used to simultaneously print several reports via nested reports. However, the nested reports are a bit more complicated to use. Indeed, the nested reports require:

- the call to **iPrintReport** to print the secondary report.
- the obligation to print the nested report after the main report. These reports cannot be printed side by side.

The internal reports allow you to print several reports without programming:

- one below another,
- side by side.

See "Nested reports", page 108 for more details.

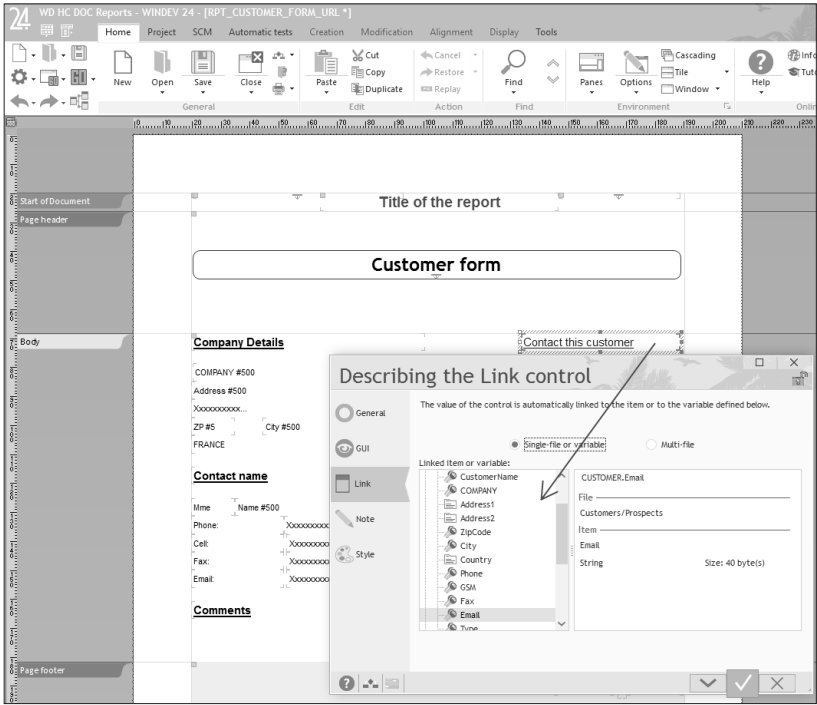
6.1.10 URL Link control

A "URL link" control is used to insert links into a report. You have the ability to insert:

- a link to an email address,
- a link to an Internet site,
- ...

The "URL Link" control is enabled only if the report is printed in an HTML file. Otherwise, the "URL link" control is visible but it is disabled.

Example:



Note: The item or the variable linked to the control is defined in the "Link" tab of the description window of control.

6.1.11 Edit control

An edit control is used to type and display text in a report. You can use an edit control:

- to allow the user to type the information to print,
- to allow the user to modify the information to print,
- ...

The edit control is enabled only if the report is displayed in "Print preview" mode or if it is printed in a PDF file. In the other cases, the edit control is visible but it is disabled.

Characteristics of edit controls

When creating an edit control in a report, you have the ability to define:

- the type of control (text, numeric, date, time),
- the associated mask: the available masks are the ones supported by the PDF format. They are available for the numerics, the dates and the times.
- the input size, which means the maximum number of characters allowed in the control. If this parameter is set to 0, there is no limit.
- the horizontal alignment,
- the initial value,
- the style of caption and input area.

Notes:

- Multi-line edit control: if the users enter more lines than what is visible, the non-visible lines will not be printed. During the input, the edit controls manage a vertical scrollbar.
- Style of input area: In this version, only the "Arial" font can be used.
- The edit control is not linked to an item found in a file or in a query. To display information coming from a file or from a query, the control must be assigned by programming.
- The border of the input area cannot be configured (thickness, color, ...).
- No currency type can be used (the Euro symbol is not supported).

See "Editable reports", page 100 for more details.

6.2 Characteristics of report controls

The characteristics of one or more controls can be viewed and/or modified in:



- the description window of this control.
- the modifier.

The report editor also allows you to:

- modify the control caption.
- modify the control name.

6.2.1 Description window of a control

To display the description window of a control, use one of the following methods:


- double-click the control (recommended).
- select "Description" from the popup menu of control (right mouse click).
- select the control and on the "Modification" pane, click the group icon  of "Fast edit" group (Alt + Enter keys).
- on the "Display" pane, click the group icon  of "Options" group (F12 key). Then, select the control and click the "Description" button.

Note: The description window can be displayed for a selection of controls. Only the characteristics common to the selected controls will be displayed.

6.2.2 Details of the description window

The description window groups the characteristics of selected controls by tab:

Tab	Characteristics
General	Used to specify the control name and the display characteristics of control. For example: <ul style="list-style-type: none">• the control caption (this caption can be multilingual).• the type of information displayed in the control (text, numeric, ...).• the display mask used for the control (the mask can be multilingual). See the online help for more details.• the image displayed in an Image control, the type of bar code used in a Bar Code control, the type of chart used in a Chart control, etc. See the online help about this type of control for more details.

GUI	Used to define: <ul style="list-style-type: none">• the print parameters of control (visible, breakable, ...).• the control size.• the control position in the report and in the block to which the control belongs.• the control anchoring.• the associated group of controls.• the associated plane.
Link	Used to define the item or the variable found in the data source to which the control is linked. The value of this item or variable will be displayed in the control. To link the control to an item or to a variable that belongs to the data source of report, select this item in the "Linked item" list. To link the control to an item that does not belong to the data source of report, select "Multi-file" then select the item linked to the current control.
Note	Used to describe the detailed operating mode of control. This information will be printed in the program documentation ( icon on the "Home" pane, in the "General" group). This information can be completed and found by programming via ..Note and ..NoteTitle .
Style	Used to define: <ul style="list-style-type: none">• the control style (background color, border, ...).• the font used in the control.

Note: All options found in the description window of report editor propose full tooltips, displaying for example, in addition to the help about the option, the corresponding WLanguage property, the availability for each platform, the multilingual values, ...

6.3 Multilingual feature in the report controls

In a multilingual application or site, the multilingual feature is proposed for the following report elements:

- the captions, the display masks,
- the images, the links.

At run time, the report editor will automatically select the caption, the mask or the image corresponding to the runtime language of the project.

Reminder: The default language of the project is defined in the "Languages" tab of the project description. To display the project description, on the "Project" pane, in the "Project" group, click "Description".

See the online help for more details (keyword: "Multilingual, Report controls").

6.4 Group of controls

A group is a set of controls grouped together so that their common characteristics (their visibility status for example) can be modified (or displayed) in a single operation, by programming.

In the report editor, all the controls of all the groups are displayed: therefore, some controls may be overlaid. The same control can be associated with several groups.

Example: To make several controls invisible, all you have to do is make the group invisible. In a health-care form for example, some controls may relate to men while others may relate to women. Grouping the related controls together allows you to make them visible or invisible according to the gender of the user.

See the online help for more details (keyword: "Group of controls").

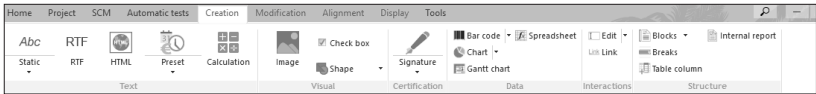
6.5 Handling controls in the editor

This paragraph describes the main operations that can be performed on a control found in a report. See the online help for more details (keyword: "Handle, Handle the report controls").

6.5.1 Creating a control:

To create a control in a report:

- 1. On the "Creation" pane, click the type of control to create:

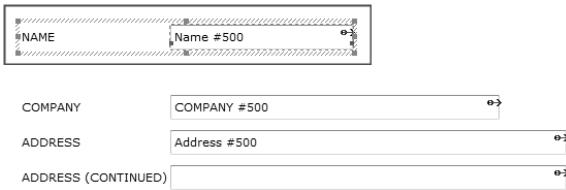


- 2. Click the position where the control will be created in the report.

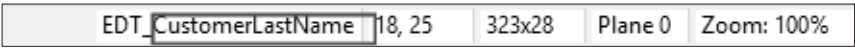
Other creation modes are also available for the controls. See the online help for more details (keyword: "Create, A control in a report").

6.5.2 Selecting one or more controls.

The selected control is displayed with handles.



The name of the selected control appears in the status bar.



Several methods can be used to select several controls at the same time:

- Method 1: with the Ctrl key.
- Method 2: with the lasso.

To select several controls at the same time with the Ctrl key:

- 1. Press the Ctrl key and keep it down.
- 2. Select the controls with the mouse (click each control with the mouse).
- 3. Release the Ctrl key. The selected controls are displayed with handles.

Notes:

- To add a control to an existing selection or to delete a control from an existing selection, keep the Ctrl key down and click the control that must be added or deleted.
- When displaying the 7-tab window ("Description" from the popup menu of the selection), only the options common to the selected controls are displayed.

The lasso of the mouse corresponds to a selection area. This area is represented by a dotted rectangle.



To select several controls at the same time:

- 1. Click beside the first control to select and keep the left mouse button down.
- 2. Move the mouse cursor to select all the controls: the selection area is represented by a dotted rectangle (the lasso of the mouse).
- 3. Release the mouse button when the selection is performed.

Notes:

- By default, all the controls with a section included in the lasso area will be selected. To only select the controls that are entirely included in the lasso area, hold the Shift key down during the selection steps.
- To add a control to an existing selection or to delete a control from an existing selection, keep the Ctrl key down and click the control that must be added or deleted.
- When displaying the description window ("Description" from the popup menu of the selection), only the options common to the selected controls are displayed.

See the online help for more details (keyword: "Select, A control").

6.5.3 Copying a control

One or more controls can be copied into the same report or from a report to another report. The description and the code of copied control will be identical to the ones of initial control.

To copy a control into the same report:

- 1. Select the control to copy.
- 2. On the "Home" pane, in the "Edit" group, click "Copy".
Note: You also have the ability to select "Copy" from the popup menu of control.
- 3. On the "Home" pane, in the "Edit" group, click "Paste".
Note: You also have the ability to select "Paste" from the popup menu of control.
- 4. Position the control with the mouse (the control outline appears in dotted lines and it helps you position the new control).

To copy a control from a report into another report:

- **1st method:**
 - In the first report, select the control to copy.
 - On the "Home" pane, in the "Edit" group, click "Copy".
Note: You also have the ability to select "Copy" from the popup menu of control.
 - Display the destination report in the report editor.
 - On the "Home" pane, in the "Edit" group, click "Paste".
Note: You also have the ability to select "Paste" from the popup menu of control.
 - Position the control with the mouse (the control outline appears in dotted lines and it helps you position the new control).
- **2nd method:** (both reports must be displayed in the editor)
 - In the first report, select the control to copy.
 - Keep the left mouse button down and drag the control toward the destination report.
 - Release the mouse button once the control is positioned in the destination report (the outline control appears in dotted lines and it helps you position the pasted control).

6.5.4 Copying the control style

Copying the style of a control consists in copying the characteristics of a control style into another control. Only the control style is copied (not its content).



This copy can be performed between:

- Two controls of the same type.
- Two controls of different types. In this case, only the style characteristics common to the two controls are copied.

The style copy can be performed between two controls belonging to the same window (to the same page or to the same report) or between two controls belonging to different windows (to different pages or to different reports).

Note: The style of a control corresponds to the characteristics specified in the "Style" tab of the description window of the control.

To perform a style copy between two controls:

1. Select the control containing the style to copy.
2. On the "Modification" pane, in the "Fast edit" group, click on , (Ctrl + Shift + C).
The mouse cursor turns into brush: 
3. Select the control into which the style must be copied. The type of this control can differ from the type of the control whose style is copied.

Note: To apply the style of a control to several controls, keep the Ctrl down when copying the style onto the destination controls.

6.5.5 Moving a control

To move one or more report controls:

1. Select the controls to move.
2. Keep the left mouse button down and move the controls with the mouse. The real-time interface checker allows you to align the controls (see the paragraph below).

Notes:

- A control necessarily belongs to a block.
- The selected controls can also be moved with the arrow keys found on the keyboard. To speed up the move, press the Ctrl key together with the arrow key. To slow down the move, press the Shift key together with the arrow key.

6.5.6 Deleting a control

To delete one or more report controls:

1. Select the controls to delete.
2. Press the Del key.

6.5.7 Using an off-report control

In most cases, the controls belong to a report. Therefore, these controls appear in the report in edit mode and at run time.

A control can also be positioned in the home window of a report: the control is said to be "off report". This control will be visible in edit only. This control will not be printed.



An "off-report" control is used to store a report control without printing it. This allows you to use the control content in programming (like a variable).

To create an "off-report" control:

1. On the "Creation" pane, click the type of control to create.
2. Click the left section of the home window where the control must be created.

6.6 Aligning the report controls

The report editor proposes several methods for aligning the controls found in a report:

- the alignment options.
- the real-time interface checker (automatic positioning).
- the zoom.
- the rulers and markers.
- the snap-on grid.

See the online help for more details (keyword: "Alignment").

6.7 Handling the controls by programming

6.7.1 Processes of the report controls

The following processes are associated with the report controls:

Processes	Runtime condition
Initialization	Run when initializing the report. The order in which the processes are run when printing a report (including during a test) is as follows: <ul style="list-style-type: none">"Opening" process of the report."Initialization" process of report blocks."Initialization" process of controls.
Before printing	Run before printing a control.
Entry in	Run when the input cursor enters in the control. This process is available for the edit controls used in preview mode only.
Exit from	Run when changing control (TAB key, change of control with the mouse, ...). This process is available for the edit controls used in preview mode only.
Modification of	Run whenever the content of the control is modified (typing or erasing a character for example). This process is available for the edit controls used in preview mode only.
Click	Run when clicking the control. The click on the control is available in the print preview only. A star allows the user to easily identify the clickable controls.

Source code in WLanguage can be entered in these processes. See "Sequence of processes associated with a report", page 135 for more details.

To view and/or modify the processes of a control, use one of the following methods:

- select the control and press F2.
- select the control and display the "Context-Sensitive View of Code" pane.
- select "Code" from the popup menu of the control (right mouse click on the control).
- click the "Code" icon of control description.

6.7.2 Using a control in a WLanguage process

To handle a report control in a WLanguage process, you must:

- use the name of this control directly. This syntax is valid only if the control is handled from the report to which it belongs,
- use the following notation: "<ReportName>.<ControlName>".

A control can be handled:

- in a WLanguage function,
- with a WLanguage property,
- in direct assignment.

For example:

```
// Modify the background color of CtTotalBT control
// in the RPT_Order report
CtTotalBT..BrushColor = LightBlue
// equivalent to
// RPT_Order.CtTotalBT..BrushColor = LightBlue
```

6.7.3 Properties associated with the report controls

The following properties can be used with the report controls:

AdjustmentMode	Used to find out and modify the adjustment mode of a Spreadsheet control in a report.
Anchor	Used to manage the anchor of a control in a report.
AnchorRateRight	Used to: <ul style="list-style-type: none">• Find out the anchor rate when a control moves on the horizontal axis.• Modify the anchor rate when a control moves on the horizontal axis.
AnchorRateWidth	Used to: <ul style="list-style-type: none">• Find out the anchor rate when a control (or a table column) stretches on the horizontal axis.• Modify the anchor rate when a control (or a table column) stretches on the horizontal axis.
Border	Used to find out and modify the characteristics of the border for a control found in a report or in a report block.
BrushColor	Used to manage the background color of a control or report block.
Caption	Used to find out and modify the text displayed in a Static control or Link control.
CollapsedImage	Used to find out and/or modify the image of a collapsed row in a treeview item found in a "TreeView Table" report.
Color	Used to find out and modify the font color for a text displayed in a Static control or in a calculated control of a report.
DisplayCurrency	Used to find out and modify the currency used when a display and/or an input is performed in a "Currency + Euro" control.
DisplayedItem	Used to find out and modify the item or the variable displayed by the control.
Empty	Used to find out whether a report control is empty (control linked to an item).
EndDatePeriodDisplayed	Used to find out the end date of the period currently printed in a Scheduler or Organizer control found in a report.
ExpandedImage	Used to find out and/or modify the image of an expanded row in a treeview item found in a "TreeView Table" report.

Font	Used to: <ul style="list-style-type: none">Find out the font used in a report control.Modify the font used in a report control.
FontAppointmentTitle	Used to find out and modify the characteristics of the font used for the title of appointments in the Scheduler and Organizer reports.
FontBold	Used to: <ul style="list-style-type: none">Find out whether a text is bold or not.Modify the weight of a text.
FontCharset	Used to: <ul style="list-style-type: none">Find out the character set used by the screen fonts of a control.Modify the character set used by the screen fonts of a control.
FontCondensed	Used to: <ul style="list-style-type: none">Find out whether the characters in a text are condensed or not.Condense (or not) the characters in a text.
FontExtended	Used to: <ul style="list-style-type: none">Find out whether the characters in a text are extended or not.Extend (or not) the characters displayed in a text.
FontItalic	Used to: <ul style="list-style-type: none">Find out whether the italic attribute is assigned to a text.Modify the italic attribute of a text.
FontLarge	Used to: <ul style="list-style-type: none">Find out whether the characters in a text are enlarged or not.Enlarge (or not) the characters in a text.
FontName	Used to: <ul style="list-style-type: none">Find out the font used in a control.Modify the font used in a control.
FontSize	Used to: <ul style="list-style-type: none">Find out the size of the font used in a control.Modify the size of the font used in a control.
FontStrikeOut	Used to: <ul style="list-style-type: none">Find out whether a text is struck out or not.Strike out (or not) the text displayed in a control.
FontUnderlined	Used to: <ul style="list-style-type: none">Find out whether a text is underlined or not.Underline (or not) a text.
FutureHeight	Used to find out the height of an auto-sizable control in a report when this control is enlarged.
GranularityGridlines	Used to find out and modify the size of the grid used to display the gridlines in a Scheduler or Organizer control found in a report.
Group	Used to: <ul style="list-style-type: none">find out whether the control belongs to a group of controls.associating a control with a group of controls.

Height	Used to find out and modify the height of a control or block in a report.
HeightInPixel	Used to find out the height of a report control in pixels.
HierarchicalDepth	Used to find out and modify the depth of a TreeView Item control found in a report.
HorizontalAlignment	Used to find out and modify the horizontal alignment of a control.
HTML	Used to: <ul style="list-style-type: none">find out whether the current report is printed in HTML mode.manage the HTML type of a Static control found in a report.
Image	Used to find out and modify the background image of a control or block.
InitialHeight	Used to find out the initial height of a control or block in a report.
InitialWidth	Used to find out the initial width of a report control.
InputMask	Used to find out and modify the input mask.
InternalReportName	Used to: <ul style="list-style-type: none">Find out the name of the report printed in an “Internal report” control.Change the report printed in an “Internal report” control.
MemoryCurrency	Used to find out and modify the currency used when handling a “Currency + Euro” control by programming.
Multiline	Used to: <ul style="list-style-type: none">find out whether a report control is multi-line.modify the “Multi-line” option of a report control.
Name	Used to find out the name of a control, report or report block.
Plane	Used to: <ul style="list-style-type: none">Find out the active plane of a report or the plane to which a control belongs.Modify the active plane of a report or associate a control with another plane.
PrintArea	Used find out and modify the area to print for a Spreadsheet control found in a report.
PrintedSheet	Used find out and modify the sheet to print for a Spreadsheet control found in a report.
Process	Used to modify the processes that will be run by the available events on a control, a report, ...
RichEdit	Used to manage the RTF type (Rich Text Format): <ul style="list-style-type: none">of an edit control or Static control found in a window.of a Static control found in a report.
RightToLeft	Used to find out the actual writing direction, effective on a control, a column, a window or a report.
RTFWithImagesAndTables	Used to define how the images and tables are supported in a Static control in RTF format found in a report.
Size	Used to find out and modify the size of an element.

StartDatePeriodDisplayed	Used to find out the start date of the period currently printed in a Scheduler or Organizer control found in a report.
State	Used to find out and modify the display status of an element.
TextDirection	Used to find out and modify the writing direction used in a window, in a report or in a control (useful for managing languages such as Arabic or Hebrew).
Value	Used to find out and modify the value of a report control.
VerticalAlignment	Used to find out and modify the vertical alignment of a control.
Visible	Used to: <ul style="list-style-type: none">Find out whether an element is visible.Make an element visible or invisible.
Width	Used to find out and modify the width of a control or block in a report.
WidthInPixel	Used to find out the width of a report control in pixels.
X	Used to find out and modify the X coordinate of a control.
XInitial	Used to find out the X coordinate of a control in relation to its block (position on the X axis in millimeters) when opening the report.
Y	Used to find out and modify the Y coordinate of a control.
YInitial	Used to find out the Y coordinate of a control in relation to its block (position on the Y axis in millimeters) when opening the report.
ZOrder	Used to manage the z-order position of a control.

See the online help for more details.

7.USING TABLES IN A REPORT

7.1 Overview

In a report, a table includes a set of controls called "Table cells".

In most cases, a report contains a single table. To display several tables in the same report, you must use the internal reports or the nested reports.

A table includes three sections:

- the **header row**, which displays the name of the table's columns. This row is optional. If it is displayed, this row must be the first table row and it is displayed for each new table.
In most cases, this row is displayed in a *Page header* block or in a *Break header* block.
- the **data rows**, which display the information from the data source. For each record found in the data source, a new row is displayed when the report is run. The data of a given record is displayed on a single row. In most cases, the data row is associated with the *Body* block.
- the **total row** used to calculate the totals on the table columns.
In most cases, these rows are displayed in the *Break footer* block or in the *Page footer* block.

Remark: The different total cells do not necessarily have to be positioned on the same row.

For example:

Reference	Product caption	Unit price BT	Quantity	Unit price	
COU_RÉ14821	PRODUCT #145	60.75	4	60.75	← Header row Data rows Total rows
HUG_MO13633	PRODUCT #577	950.00	1	950.00	
MAI_GL5994	PRODUCT #863	26.80	5	26.80	
		Total BT		1,327.00	Total rows
		Total IOT:		1,568.20	

Benefits of tables

In edit mode and at run time, tables are a simple way for aligning the report controls. Furthermore, the controls are automatically resized.

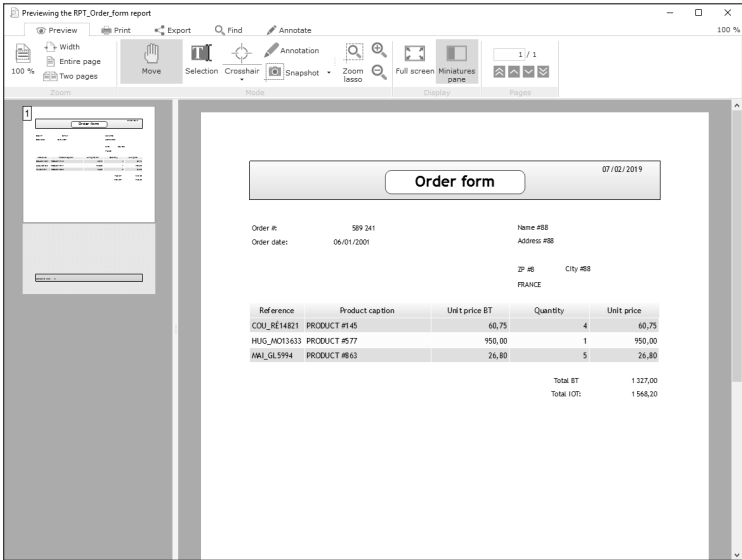
If a table cell is expanded in the editor, the other table cells are automatically shifted.

At run time, if a value cannot be entirely displayed in a cell of an auto-sizable table, all the cells found on the row will take the same height in order for the value to be entirely displayed.

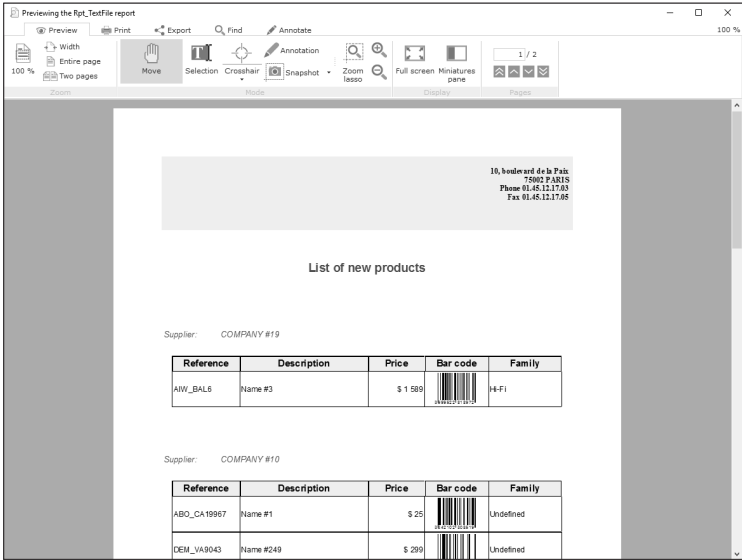
The report editor proposes two types of tables:

- simple tables.
- the crosstab tables (double-entry tables).

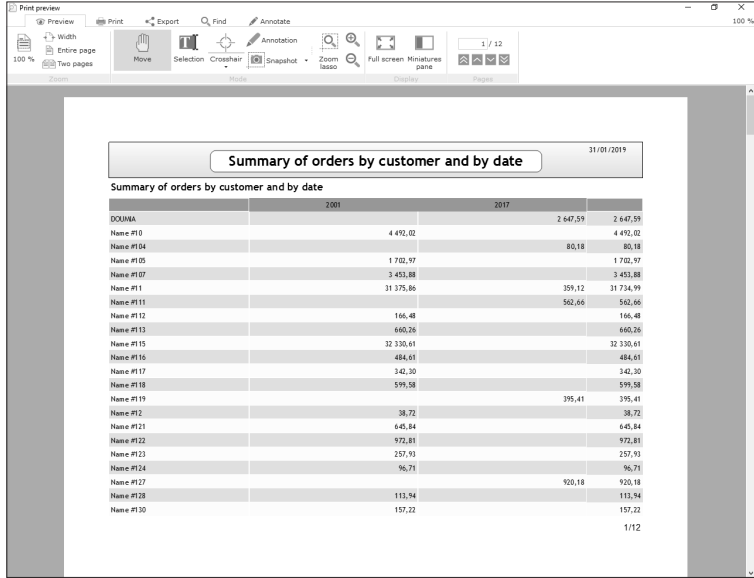
- Some examples of reports with table:
- Order form (simple table):



- List of products (simple table):



- Sales per year and per customer (crosstab):




7.2 Creating a table in a report

The tables can be created in a report:

- when creating a report:
 - Table report.
 - Crosstab report.
 - TreeView Table report.
- by inserting a table into an existing report.

7.2.1 Creating a Table report

To create a Table report:

- Click  among the quick access buttons.
- The window for creating a new element is displayed: click "Report" then "Report".
- The wizard for report creation starts.
- In the wizard, select the type of report to create: Array.

The main steps for creating a Table report are as follows:


- selecting the source of the report. See "Data sources of a report", page 43 for more details.
- selecting the sort items (if necessary). See "Breaks and sorts", page 35 for more details.
- selecting the break items. See "The breaks", page 36 for more details.
- selecting the data to display in the report. This data will be displayed in the data row of the table.

- choosing the calculations performed in the report. These calculations will be displayed in the total row of the table.
- selecting the page format (A4, A3, landscape, ...).
- selecting the skin template. See "Skin template", page 37 for more details.

7.2.2 Creating a Crosstab report

The report editor allows you to create "Crosstab" reports. In this case, the report contains a double-entry table only.

To create a Crosstab report:

- Click  among the quick access buttons.
- The window for creating a new element is displayed: click "Report" then "Report".
- The wizard for report creation starts.
- In the wizard, select the type of report to create: Crosstab.

The main steps for creating a Crosstab report are as follows:


- The item corresponding to the captions of columns.
Note: If this item is a "Date" item, the wizard proposes to:
 - keep only the day or the year for these dates.
 - define the period to take into account (start date and end date).
- The item corresponding to the captions of the rows.
Note: If this item is a "Date" item, the wizard proposes to:
 - keep only the day or the year for these dates.
 - define the period to take into account (start date and end date).
- The information that will be displayed in the table cells. This information can correspond to:
 - the number of records corresponding to the intersection between rows and columns.
For example: the number of products ordered per product and per year. The customer named "Smith" has placed 2 orders in 2000 and 3 orders in 2001.
 - the sum, the average, the maximum or the minimum of an item.
For example: the number of products ordered per product and per year. In 2000, 279 sofas were ordered.
 - an item.
For example: the score given by each judge to a skater in competition. In the European championships 2002, the Italian referee gave a score of 5.2 to skater "A".
- The calculation of totals for the table rows and columns (if necessary).
- The format of the page on which the report will be printed. The report is printed in A4 format by default.
- The skin template of the report if necessary.
- The name and title of the report. This name ("WDE". file corresponding to the report) will be used to identify the report in your applications or in your sites.

See the online help for more details.

7.2.3 Creating a TreeView Table report

The report editor allows you to create "TreeView Table" reports. The report allows you to see a hierarchy in the table.

To create a TreeView Table report:

- Click  among the quick access buttons.
- In the window that is displayed, click "Report" then "Report".
- The wizard for report creation starts.
- In the wizard, select the type of report to create: TreeView table.

The main steps for creating a Treeview Table report are as follows:

- selecting the source of the report.
- selecting the data to display in the report. This data will be displayed in the data row of the table.
- selecting the page format (A4, A3, landscape, ...).
- selecting the skin template.

See the online help for more details.

7.2.4 Inserting a table into a report

A table can be inserted into any report that does not already contain a table (except for the Label reports).

Note: Only a simple table can be inserted. A crosstab cannot be inserted into an existing report.

This report must contain at least two blocks: the *Body* block and a header block (*Page header*, *Break header* or *Start of document*).

To insert a table into a report:

- On the "Creation" pane, in the "Structure" group, click "Table".
- The wizard for table creation starts.

Depending on the data source of the current report, you can:

- select the items that will be displayed in the table (report based on data file, query, ...). The selected controls are automatically included in the table.
- specify the headers of the table (report without data source, report based on text file).

In all cases, the header row of the table is created in the header block and the data row is inserted into the *Body* block of the report.

7.3 Describing the table

To display the table description, select "Table description" from the popup menu of a table element (cell belonging to the header, to the data or to the total of table). The following window is displayed:



This window is used to view:

- in the top section, the name of the table as well as the columns found in the table.
- in the bottom section, either the characteristics of the table if the table name is selected, or the characteristics of the selected column.

7.3.1 Characteristics of the table

Three tabs are displayed in the bottom section if the name of the table is selected. These tabs are used to configure the entire table.

"General" tab: Configuring the table

The "General" tab is used to define:

- Whether the table rows are breakable and/or auto-sizable.
 - If the rows are breakable, the lines can "overlap" over 2 pages (if the text displayed in the column is very long for example).
 - If the rows are auto-sizable, the cells can be expanded in order to display their entire content. If a cell is expanded, all the cells found in the row will also be expanded.
- The position of the table header (block for displaying the header) as well as its characteristics.
- The size and position of the table in the page.

"Border" tab: Configuring the table gridlines

The "Border" tab is used to define the table gridlines. For a custom setting, you have the ability to configure the gridlines of each cell. To do so, display the description of each table cell.

"Style" tab: Configuring the table

The "Style" tab is used to view and modify the style of the table. Only the values common to all the table cells are displayed. For a custom setting, you have the ability to configure the style of each cell. To do so, display the description of each table cell.

7.3.2 Characteristics of columns

To display the characteristics of a column, select this column in the top section of the description window.

For each column, the following elements are displayed in the top section:

- The **caption of the column**. This caption will be displayed in a Static control, in the header row of the table. This caption can be left empty.
- The **item associated with the column**. The corresponding control will be displayed in the data row of the table. This control can be associated with no item in order to be filled by programming for example. The type is automatically updated according to the selected item.
- The **type of data** displayed in the column.
- The **calculation associated with the column**. All you have to do is select the type of calculation made on the items displayed in the column. The corresponding calculated control will be displayed in the total row of the table.

Two tabs are displayed in the bottom section if the name of the column is selected. These tabs are used to configure the characteristics of the column:

• "Column" tab: Configuring the columns

The "Column" tab is used to configure the display characteristics of the column: type, mask, ...

• "Link" tab: Configuring the link of the column

The "Link" tab is used to configure the element of data source (variable or item) linked to the table column.

7.4 Handling the table cells

The table cells can be handled individually or via the table description.

The individual operations that can be performed on the table cells are as follows:

- Selecting a table element.
- Displaying the table description.
- Adding a cell into a table.
- Adding a column into a table.
- Resizing the table cells.
- Moving the table columns.
- Deleting cells from the table.


Note: All types of controls available in the report editor can be displayed in a table column.

7.4.1 Selecting a table element

- To select:
- a **table cell**: click the cell once. The selection handles appear around the column cells.
 - a **table column**:
 - click a column cell twice successively. The selection handles appear around the column cells.
 - hover the header of the column and click when the cursor turns into a vertical black arrow (pointing down). The selection handles appear around the column cells.
 - a **table row**: hover the left side of the row and click when the cursor turns into an horizontal black arrow (pointing to the right). The selection handles appear around the row cells.
 - **all the table cells**: click a cell three times successively. The selection handles appear around all the table cells.

7.4.2 Displaying the table description

By default, when a double click is performed on a cell belonging to a table, the description window of cell is displayed.

- To display the table description, perform one of the following actions:
- select a cell and select "Table description" from the popup menu.
 - select a cell then, on the "Modification" pane, click the group icon  of "Table" group.
 - click a cell three times successively (all table cells are selected) and double-click.

7.4.3 Adding a cell into a table

- All types of controls available in the report editor can be added into a table. These controls can:
- replace an existing cell in the table.
 - be added to a column (a calculation added at the bottom of column for example).

- To add or replace a cell in a table:
1. Select the control that must be added to the table.
 2. Select "Add to table" from the popup menu of control. A wizard starts.
 3. Select "Add to an existing column".
 4. Select the column where the addition will be performed as well as the control position in the column (header, data or total).
 5. Validate the cell addition.

7.4.4 Adding a column into a table

You have the ability to add a column to a table. This new column may (or may not) correspond to an existing control.

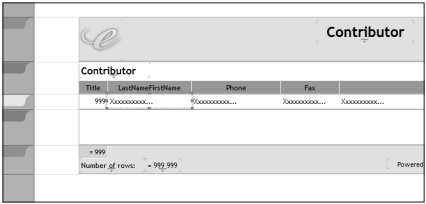
- To add an existing control into a table as column:
1. Select the control.
 2. You can:
 - Drag and Drop the control to the table. After confirmation, a new column is created after the existing columns. The description of the cells found in this new column corresponds to the initial control.
 - select "Add to table" from the popup menu of control. A wizard allows you to specify the characteristics of this new column.

- To insert a new column:
1. You can:
 - select "Insert a column" from the popup menu of the table.
 - on the "Creation" pane, in the "Structure" group, click "Table column".
 2. Type the header caption of new column.
 3. Select (if necessary) the item linked to the new column and validate the addition of the new column.

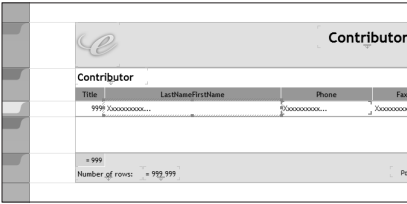
7.4.5 Resizing the table cells

The entire table is resized when a table cell is resized in the report editor.

- Two cases may occur:
- **Modifying the width of a cell:**
All the cells found in the column will also be resized.
All the columns found after the column currently resized will be moved.

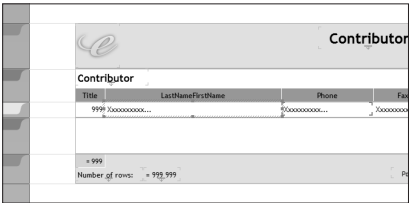


Before the resize operation

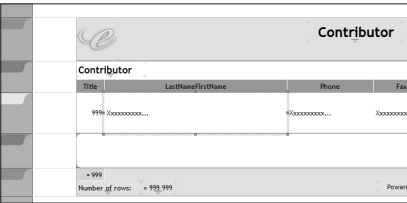


After the resize operation

- Modifying the height of a cell:**
All the table cells found on the same row will be resized.



Before the resize operation



After the resize operation

To resize a cell:

1. Select the corresponding cell in the editor. Black handles appear around the cell.
2. Hover the black handle corresponding to the side to resize. When the mouse cursor turns into a double arrow, keep the left mouse button down and resize the cell.
3. Release the left mouse button. The resizing is performed.

Note: Border of the block and border of the cell

A cell can be stuck to a report block. In this case, when the cell is resized, the block will be automatically resized or not according to the cursor displayed.

The mouse cursor differs according to the element to resize:

- a small double black arrow indicates that the cell is currently resized.
- a large double blue arrow indicates that the block and the cell are currently resized.

7.4.6 Moving a table column

Moving the table columns consists in modifying the order in which the columns are displayed in the table.

Several methods can be used to move the table columns:

- in the report currently modified: select a column cell and use the keyboard arrows (or select "Move to..." from the popup menu of the cell).
- from the description window of the table: the arrow buttons are used to move the selected column.
- "Drag and Drop" a column cell in the editor.

7.4.7 Removing or deleting a table column

A column found in a table can be:

- **removed from the table**: the column cells are no longer associated with the table. These cells remain in the report and they can be handled independently of the table.
- **deleted**: the cells are no longer present in the table nor in the report.

To remove a column from the table:

1. Select one of the cells in the column to delete (See "Selecting a table element", page 96 for more details).
2. Use one of the following methods:
 - select "Remove from table" from the popup menu of the cell.
 - on the "Modification" pane, in the "Table" group, click "Remove from table".

Note: If the column to remove is not located on the left extremity of the table, the report editor proposes to automatically move the column at the beginning of the table before removing it.

3. The different cells found in the column are independent and they can be moved in the report.

To delete a column from the table:

1. Select one of the cells in the column to delete: Column header or data control.
2. Use one of the following methods:
 - press the Del key on the keyboard.
 - select "Delete" from the popup menu of the cell.
 - on the "Modification" pane, in the "Table" group, click "Delete the column".
 - select "Delete" in the description window of the table.
3. The column containing the cell is deleted from the table.

Note: If you select a cell that belongs to the total row, the column will not be deleted when this cell is deleted (only the cell will be deleted).

8.EDITABLE REPORTS

8.1 Overview

The report editor allows you to create reports that include edit controls. These reports can be generated in PDF format.

This gives you the ability to distribute reports (in PDF format if necessary) allowing the user to perform a direct input in the specified areas: administrative form (tax return, registration, ...).

The data can be printed or stored in a database (when the input is performed from the print preview). Then, the report becomes an entry form.

The edit controls can be used on all types of reports. In most cases, the edit controls are used on the Form reports, using a PDF in report background for example.

8.2 How to?

8.2.1 Creating the editable report

- To create an editable report:
1. Create a report according to your requirements (see "Creating a report", page 23 for more details).
 2. Create the edit controls in the areas where the input will be performed: on the "Creation" pane, in the "Interactions" group, click "Edit".
 3. Save the report.
 4. During a print, the user will be able to directly enter data in the edit controls. The input is available:
 - in the print preview.
 - in the reports printed in PDF format.

8.2.2 Characteristics of edit controls

- When creating an edit control in a report, you have the ability to define:
- the type of control (text, numeric, date, time),
 - the associated mask: the available masks are the ones supported by the PDF format. They are available for the numerics, the dates and the times.
 - the input size, which means the maximum number of characters allowed in the control. If this parameter is set to 0, there is no limit.
 - the horizontal alignment,
 - whether there is a border around the input area.
 - the initial value,
 - the style of caption and input area.

- Notes:
- Multi-line edit control: if the users enter more lines than what is visible, the non-visible lines will not be printed. During the input, the edit controls manage a vertical scrollbar.
 - Style of input area: In this version, only the "Arial" font can be used.
 - The edit control is not linked to an item found a data file or in a query, nor to a variable. To display information coming form a data file, a query or a variable, the control must be assigned by programming.
 - The border of the input area cannot be configured (thickness, color, ...). You only have the ability to make it visible or invisible.
 - No currency type can be used (the Euro symbol is not supported).
 - It is not possible to make calculations between the different input controls.

8.2.3 Processes associated with an edit control in a report

The following processes are associated with an edit control found in a report:

Process	Runtime condition
Initializing	Run when opening the report.
Before printing	Run before printing a control.
Entry	Run when the input cursor enters in the control.
Exit	Run when changing control (TAB key, change of control with the mouse, ...).
Modification	Run whenever the content of the control is modified (typing or erasing a character for example).

8.3 Input in the print preview (WINDEV only)

When a report is displayed in print preview, the edit controls are enabled.

In this case, the user will be able to enter the requested information and print the report in one of the available formats. A message is displayed in the print preview to inform the user of the ability to enter information.

By default, the user can browse the edit control via the TAB key: the available controls are browsed from top to bottom and from left to right.

- Notes:
- The processes associated with the edit controls are run when the controls are handled in the print preview.
 - The data coming from the database can be displayed by programming. To do so, link the report to a data file or to a query and assign the requested item to the edit control in the "Pre-print" process of the edit control.
 - The data modified by the user in the preview window can be saved by programming. Let's take a simple example, the modification of a record displayed in the report. In each edit control whose value must be modified, you must:
 - assign the entered value to the item,
 - modify the record.

These actions are performed in the "Exit from control" process.

For example, to modify the customer name, the corresponding code is as follows:

```
Customer.Name = EDT_Name
HModify(Customer)
```

8.4 Input in a PDF file

When a report is printed in a PDF file, the edit controls are enabled. In this case, the user will be able to enter the content of the different edit controls and to print the PDF file with the entered values.

Notes:

- The edit controls are used to enter information in the PDF file. This information will be used during the print. If the end user owns a full version of Adobe Acrobat, the data entered can be saved in PDF format (Acrobat Reader cannot be used to save the PDF files).
- The data coming from the database can be displayed by programming. To do so, link the report to a data file or to a query and assign the requested item to the edit control in the "Initializing" or "Before printing" process of edit control.

8.5 Input in a report based on a PDF form

The edit controls are well suited to the "Form" reports. The "Form" reports are based on an image file or on a PDF file.

For a PDF form containing several pages, you have the ability to create a report displaying the different pages of the PDF file ("Print all the pages found in the PDF" option in the "Background" tab of the report description).

These pages are visible and they can be modified in the report editor: the edit controls can be positioned on any report page.

See "Using a form in a report", page 54 for more details.

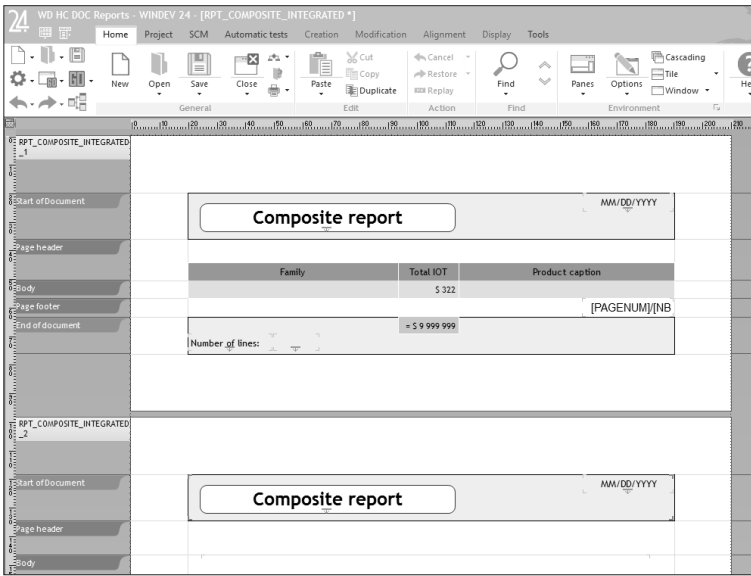
9.COMPOSITE REPORTS

9.1 Overview

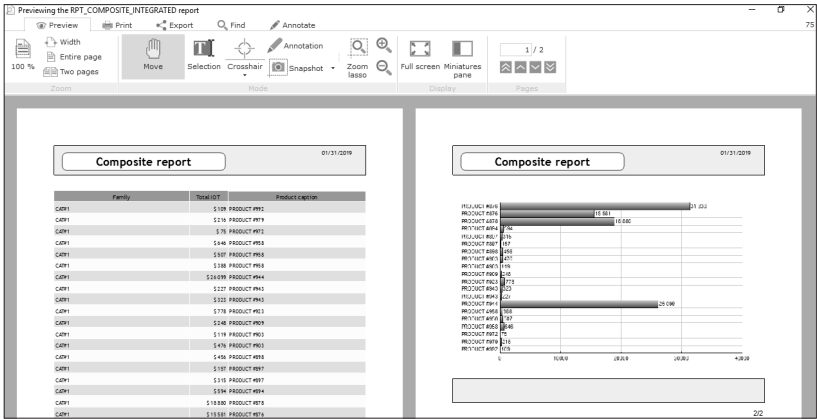
The report editor gives you the ability to create "Composite" reports. In this case, the report includes several sub-reports. Each sub-report can display a different set of data and use a specific orientation. The different sub-reports are printed one after another.

For example:

- In the editor:




- At runtime:



- Note:** The report editor proposes several methods for printing several reports in a single document:
- the internal reports, that are used to print several reports side by side. See "Internal Report control", page 76 fore more details.
 - the sequencing of reports that can be performed by programming. This method can be easily implemented. See "Sequencing reports", page 106 for more details.
 - the nested reports, that are used to create sequences of complex reports, with management of page footers. See "Nested reports", page 108 for more details.

9.2 Creating a composite report

To create a "Composite" report:

1. Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Report" then "Report".
 - The wizard for report creation starts.
2. Select the type of report: "Composite report". Go to the next wizard step via the arrow buttons.
3. Specify in the wizard the number of sub-reports that will be included in the composite report. Go to the next wizard step.
4. Give a name to your report and validate.
5. The composite report appears in creation in the report editor. You must define the characteristics of each sub-report.


For each sub-report:

Click the link "<Click here to define the sub-report>". The wizard for defining the sub-report is displayed. You can:


- **Create a new sub-report.**
 1. Select:
 - the type of sub-report to create (form, table, ...). See "The different types of reports", page 24 for more details.
 - the data source of the sub-report (data file, query, ...). See "Sources of the report", page 34 for more details.
 - the elements of the sub-report (controls, sort conditions, breaks, ...).
 2. Give a name to the sub-report and validate the creation of the sub-report.
- **Use an existing report.**
 1. Select the report to display in the sub-report.
 2. Give a name to the sub-report and validate the creation of the sub-report. By default, the sub-report is named: "S<ReportName>".

9.3 Characteristics of composite reports and sub-reports

9.3.1 Characteristics of composite reports

- The composite report includes several sub-reports. All the sub-reports are directly viewed in the report editor.
- The composite report has a name, as well as each sub-report. The different sub-reports of the composite report are listed in the "Project explorer" pane.
- Several methods can be used to view the description window of the composite report:
 - On the "Modification" pane, click the group button  of the "Composite report" group.
 - Select "Description of composite report" from the popup menu of the report.
- The description window of the composite report is used to:
 - Define the caption of the report and its name ("General" tab).
 - Define the export buttons found in the preview ("GUI" tab).
 - Define the behavior if the sub-reports are empty ("GUI" tab).
 - Make the report modifiable in Reports & Queries ("Details" tab).
 - Define the languages supported by the report ("Language" tab).
 - Manage the sub-reports ("Sub-reports" tab): you have the ability to add or delete sub-reports, to modify their print order, ...

9.3.2 Characteristics of sub-reports

- The sub-reports can be accessed in the editor by opening the composite report.
- To view the description window of the sub-report:
 1. Click in the sub-report.
 2. On the "Modification" pane, click the group button  of the "Report" group. You also have the ability to select "Sub-report description" from the popup menu of the report.
- The description window of the sub-report is used to select the report to display.
- The sub-reports can be reports with parameters. In this case, to initialize the parameters of the sub-report, you must use **ilInitSubReport**.
- The report displayed by a sub-report can be modified by programming via **iChangeSubReportSource**.

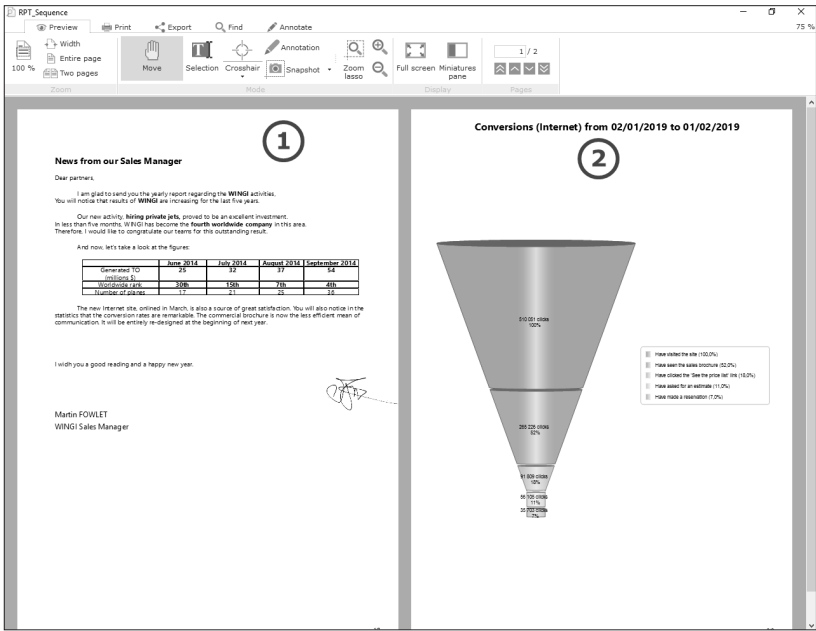
10. SEQUENCING REPORTS

10.1 Overview

In some cases, you may want to print several reports one after another, and view them in the same print preview.

The composite reports can be use for this purpose when the sequence of reports is known in advance.

If the sequence is not known in advance and if it must be programmed, the sequence of reports can be used by programming.



Note: You also have the ability to add into the sequence:

- a Word processing document.
- a PDF document.
- an XLS or XLSX document.

10.2 How to?

To sequence the reports by programming:

1. To add:
 - a report to the sequence, use **iSequencingAdd**. This function must be used for each report that will be added to the sequence. This function is used to pass parameters to the report to print.
 - a Word Processing document to the sequence, use **iSequencingAddDoc**. This function is used to specify the Word Processing document to print.
 - a PDF document to the sequence, use **iSequencingAddPDF**. This function is used to specify the PDF document to print.
 - an XLS or XLSX document to the sequence, use **iSequencingAddXLS**. This function is used to specify the XLS or XLSX document to print.
2. To end the sequence and to start the print, use **iSequencingPrint**.

Example:

```
MyDocument is Document = "c:\temp\generalconditions.docx"
MySpreadsheet is string = "c:\temp\prices.xls"

iPreview()
// Add reports into the sequence
iSequencingAdd(RPT_Report_Portrait)
iSequencingAdd(RPT_Report_Landscape, 3)
// Add the general conditions in document format
iSequencingAddDoc(MyDocument)
// Add prices in the format of an XLS worksheet
iSequencingAddXLS(MySpreadsheet)
iSequencingPrint()
```

Notes:

- The pages of the reports found in the sequence are numbered sequentially. The numbering can be modified by **iPageNum**.
- The reports found in the sequence can have different orientations (portrait and/or landscape).
- The reports are not run when running **iSequencingAdd**: only their name and their parameters are stored. These reports are actually run during the call to **iSequencingPrint**.
Caution: If the reports found in the sequence are based on a query, the query must be initialized by **ilnitReportQuery** in the initialization code of the report (and not before the call to **iSequencingAdd**).

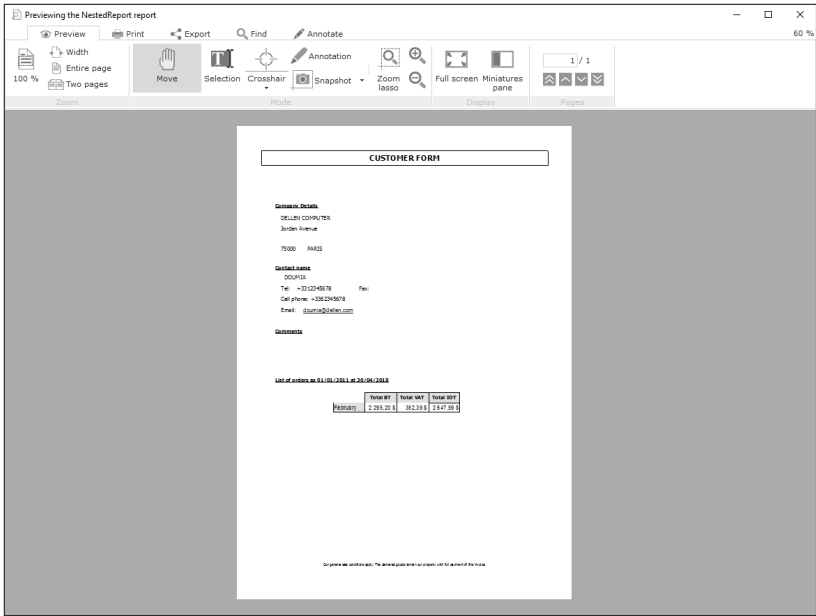
11. NESTED REPORTS

11.1 Overview

The **nested reports** are used to print several reports (secondary reports) in a single report (main report). These reports can have a different orientation (portrait/landscape).

For example, the following main report contains:

- a report displaying the characteristics of a customer.
- a report displaying the summary of orders for this customer for the last months.



11.2 Creating a nested report

11.2.1 Principle

The creation of a main report containing several secondary reports is performed in several steps:

1. Creating a main report.
2. Creating the secondary reports. The orientation of these reports can differ from the orientation of main report.
3. Printing secondary reports in the main report with **iPrintReport**. This function must be used:
 - in the "Before printing" process of a block: the secondary report will be printed **before** this block.
 - in the "After printing" process of a block: the secondary report will be printed **after** this block.**Caution:** If the nested reports have different orientations, **iParameter** must be called in order to change the print orientation accordingly.
4. Configuring the print of page headers and footers for the nested report (**iNestedHeaderFooter**). This function can be used:
 - in the opening code of main report: it will be taken into account by all the nested reports that will be printed from the main report.
 - before printing the nested report: it will be taken into account by the following nested report.
5. Printing the main report. See "Printing a nested report", page 111 for more details.

Note: To print reports with different orientations (portrait/landscape), we recommend that you use the sequencing of reports. This method can be easily implemented. See "Sequencing reports", page 106 for more details.

11.2.2 Example



The "RPT_Orders" report is based on the "Orders" data file. This report is used to print the characteristics of each order (date, customer number, total amount, ...). A break on the "OrderNum" item is used to separate the different orders.

The "QRY_ProductOrdered" query selects all the products of a specified order. This query expects an order number in parameter.

The "RPT_ProductOrdered" report is based on the "QRY_ProductOrdered" query. This report only prints the products selected by the "QRY_ProductOrdered" query.

The "RPT_ProductOrdered" report corresponds to the secondary report. This report will be printed after each break.

Implementation

1. Create the "RPT_Orders" report: this report corresponds to the main report.
 - Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Report" then "Report".
2. Create the "QRY_ProductOrdered" query. See "How to create a query?", page 168 for more details.
3. Create the "RPT_ProductOrdered" report: this report corresponds to the secondary report.
 - Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Report" then "Report".**Note:** The Page footer block of this report will not be printed. See "Blocks in the secondary report", page 110 for more details.

4. In the "After printing" process of the *Break footer* block of the "RPT_Order" report, enter the following code:

```
// Initialize the query and
// pass in parameter the number of the order currently printed
iInitReportQuery(RPT_ProductOrdered, MySource.OrderNum)
// Print the nested report
iPrintReport(RPT_ProductOrdered)
```

11.3 Specific features of secondary reports

11.3.1 Blocks in the secondary report

By default, when printing a secondary report (with **iPrintReport**), all the blocks found in the secondary report are printed except for the following blocks:

- the *Page header* block.
- the *Page footer* block.

To force these blocks to be printed, use **iNestedHeaderFooter**.

Note: If the secondary report contains a table whose header is found in the *Page header* block, only the *Page footer* block will not be printed (all the other blocks will be printed). In this case, the *Page header* block will be printed before the first row of the table (found in the secondary report).

11.3.2 Orientation of secondary reports

The secondary report and the main report can have different orientations. In this case, **iParameter** must be called to modify the print orientation accordingly.

Note: To print reports with different orientations (portrait/landscape), we recommend that you use the sequencing of reports. This method can be easily implemented. See "Sequencing reports", page 106 for more details.

11.3.3 Page break in the secondary report

The page breaks defined in a secondary report will be taken into account when printing the main report.

11.3.4 Secondary report based on a query with parameters

If the secondary report is based on a query with parameters, use **iInitReportQuery** to pass parameters to the query. This function must be used before **iPrintReport**.

11.3.5 Secondary report based on a query that is using a specific connection

If the secondary report is based on a query that handles a specific connection, use **iInitReportQueryConnection** to run this query (and to pass parameters if necessary). This function must be called before **iPrintReport**.

11.3.6 Secondary report printed from a Page Footer block

If the secondary report is printed from a *Page Footer* block, its *Page Footer* blocks will not be printed.

11.3.7 Displaying the total number of pages in the secondary report

A page counter must be added into the main report in order for the pages counter to operate in the nested report. However, the counter found in the main report can be made invisible if necessary.

11.4 Printing a nested report

To create a main report containing several secondary reports:

1. Configure the print destination of main report with **iPreview** or **iDestination**.
2. Specify the name of main report to print in **iPrintReport**.

Notes:

- If additional parameters must be passed to the main report, these parameters must be specified in **iPrintReport**, after the report name. See "Report with parameters", page 139 for more details.
- **iListNestedReports** returns the list of secondary reports nested in a main report.
- To print reports with different orientations (portrait/landscape), we recommend that you use the sequencing of reports. This method can be easily implemented. See "Sequencing reports", page 106 for more details.

Example:

The "RPT_OrderReport" report includes a main report and a secondary report.

The "RPT_OrderReport" report is printed during a click on the "Print" button.

In this example, the click code of "Print" button is as follows:

WINDEV code:

```
// Open the preview window
iPreview()
// Print the RPT_OrderReport report
iPrintReport(RPT_OrderReport)
```

WEBDEV code:

```
// Configure the destination of the print
iDestination(iHTML, fWebDir() + "\Report.htm")
// Print the RPT_OrderReport report
iPrintReport(RPT_OrderReport)
// Send the file to the browser
FileDisplay(fWebDir() + "\Report.htm")
```

WINDEV Mobile code:

```
// Configure the destination of the print
iDestination(iPCL, "\\MyNetwork\MyPrinter")
// Print the RPT_OrderReport report
iPrintReport(RPT_OrderReport)
```

11.5 Differences between nested reports and internal reports

The **internal reports** can also be used to print several secondary reports in a main report.

- Position of the secondary reports:
In the **nested reports**, the secondary reports are printed one below another. The reports can have different orientations. A report in portrait mode can be nested in a report in landscape mode.
In the **internal reports**, the secondary reports can be printed:
 - one below another.
 - side by side.
- Parameters of the secondary reports:
In a **nested report**, the secondary reports accept parameters. The parameters are passed when printing the nested report (**iPrintReport**). See "Report with parameters", page 139 for more details.
In an **internal report**, the secondary report cannot accept parameters.

Note: To simplify the use of secondary reports, the report editor allows you to create:



- internal reports.
- reports based on queries.

12. PRINTING A REPORT BY PROGRAMMING

R.Q

Important

If you are using Reports & Queries, you have the ability to print a report:

- by starting a print ().
- by running the report test ().

Therefore, no programming is required to print the report.

In order for the reports created in Reports & Queries to be directly printed from a WINDEV application, this feature must be implemented by the provider of the WINDEV application. The provider can for example include a window allowing the end user:

- to select a report created with Reports & Queries.
- and to run this report.

12.1 Overview

When creating a report, the test of this report can be run from the report editor.
You also have the ability to include this report in your applications or in your sites. The user or the Web user will have the ability to print the report from a button, a menu option, ... for example.

For a WEBDEV site, the server will prepare the report in the selected format (PDF, HTML, ...). Then, you will have the ability to:

- display this report on the browser of the Web user,
- save it on the server and propose to download it or to display it, ...

In an Intranet site, the report can also be printed on a local printer or on a network printer connected to the server.

12.1.1 Principle for printing in WINDEV

Regardless of the type of report to print, the principle for printing a report is as follows:

1. Specify the print mode of the report (preview, direct print, generation of an HTML page, ...) with **iPreview** or **iDestination**.
2. Start the print with **iPrintReport**.

This principle must be adapted according to the data source used in each report.

12.1.2 Principle for printing in WEBDEV

Regardless of the type of report to print, the principle for printing a report is as follows (in server code):

1. Specify the print mode of the report (direct print, generation of an HTML page, generation of a PDF file) with **iDestination**.
2. Start the print with **iPrintReport**.
3. Then, you will have the ability to:
 - display the generated file on the computer of the Web user with **FileDisplay**.
 - send an email to the Web user allowing him to download the created file.
 - ...

Caution: The following points must be checked if the destination of the print is a file in HTML, PDF, XML or RTF format:

- You have the rights to write into the directory where the document will be generated (the document is generated in the site directory by default).
If you generate a report in PDF format, this file can be saved in the data directory (path returned by `fDataDir`),
If you generate a report in HTML format, this file must be saved in the `_WEB` directory of the site (path returned by `fWebDir`).
- A specific name must be used for each document generated on the server. Otherwise, several simultaneous prints would overwrite the prints already performed.
Use `ConnectionCount` or `GetIdentifier` to name your reports (for example: `FileName = "CustomerReport" + ConnectionCount() + ".HTML"`).

This principle must be adapted according to the data source used in each report.

12.1.3 Principle for printing in WINDEV Mobile

Principle for printing a report in iOS or in Universal Windows App

The principle for printing a report is straightforward: a PDF file is created when printing the report. Then, all you have to do is open this PDF file in the appropriate software and print it on the printer configured for the computer used.

Regardless of the type of report to print, the principle for printing a report is as follows:

1. Specify the print mode of report (generation of a PDF file) with `iDestination`.
2. Start the print in PDF format with `iPrintReport`.
3. Then, you will have the ability to:
 - display the generated file on the user computer (`ShellExecute`).
 - print the generated PDF file via the iOS print system (`iPrintPDF`).
 - ...

Caution: Check the following points:

- You must have the rights to write into the directory where the document will be generated.
If you generate a report in PDF format, this file can be saved in the data directory (path returned by `fDataDir`).
- Use a specific name for each generated document. Otherwise, several simultaneous prints would overwrite the prints already performed.

This principle must be adapted according to the data source used in each report.

Principle for printing a report in WINDEV Mobile

The principle for printing a report from a Mobile Device (Pocket PC) is straightforward: a PCL file is created when the report is printed and this file is sent to the destination printer. The content of the PCL file is specific to the printer used.

The steps for printing a report created in the report editor are as follows:

1. Configuring the print parameters:
 - Printing on the printer selected by the user.
 - Printing on a printer selected during the development.
2. Printing the report.

Note : The result of the print differs according to the selected printer. See the online help for more details.

Case 1: Printing on a printer selected by the user

A specific window allows the user to choose the printer that will be used.
The user can add a new printer if no printer is listed.

To perform this type of print, you must:

1. Use `iConfigure` or `iParameter` (without specifying any parameter).
2. Use `iPrintReport` and specify the name of the report to print.

Example: The `RPT_Customer` report will be printed on the printer selected by the user in the configuration window. Then, this printer will be automatically used for all the other prints performed by this application.

```
// Select the printer
iConfigure()
// Print the report
iPrintReport(RPT_Customer)
```

Note: In order for the printer to become the default printer for all the applications, use the command:

```
iConfigure(False)
```

Case 2: Direct print on a printer

WINDEV Mobile proposes 2 modes for printing on a printer directly:

- direct print by defining the type of printer used.
- print on one of the printers referenced on the Mobile Device.

Printing on a type of printer

You have the ability to directly print a report on a printer associated with a Mobile Device. In this case, you must:

1. Configure the destination of the print with `iDestination`. This allows you to specify whether the print is performed in a PCL file or on a printer (infrared, WiFi, Bluetooth or network).
2. Use `iParameter` with the "PrinterType" parameter to specify the type of printer to use. The printer used must correspond to one of the printers recognized by WINDEV Mobile.
3. Print the report with `iPrintReport`.

Example: The `RPT_Customer` report will be printed on the DeskJet 350 type printer, connected via infra-red to the Mobile Device (Pocket PC).

```
// Print on an infrared printer
iDestination(iPCL, "IRDA:")
// Specify the type of printer
iParameter("PrinterType=Deskjet 350")
// Print the report
iPrintReport(RPT_Customer)
```


Printing on a printer referenced on the Mobile Device (Pocket PC)

To print on one of the printers referenced on the Mobile Device (without opening the window for printer selection), you must:

- 1. Configure the printer to use with **iConfigure**. All you have to do is specify the name of the printer to use. This name must correspond to a name specified in the window for printer selection.
- 2. Print with **iPrintReport**.

Example: The RPT_Customer report will be printed on the "InfraPnt" printer configured on the Mobile Device. Then, this printer will be automatically used for all the other prints performed by this application.

```
// Select the printer by programming
iConfigure("InfraPnt")
// Print the report based on the "TABLE_Customer" control
iPrintReport(RPT_Customer)
```

Tip: The printers referenced on a Mobile Device are stored in the registry of the Mobile Device. To add a new printer configuration by programming, all you have to do is modify the registry of the Mobile Device. See the online help for more details.

12.2 Report on data file

The method for printing a report based on a data file is the same as the above-mentioned method. However, specific information (bounds, parameters, ...) can be taken into account by a report based on a data file. In this case, the report print must adapt. The following paragraphs present:

- printing a report on a data file (general case).
- printing a bounded report based on a data file.
- printing a report with parameters based on a data file.
- printing a bounded report with parameters on a data file.

12.2.1 Printing a report on a data file

To print a report on a data file, you must:

- 1. Configure the destination of report print:
 - In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
 - In WEBDEV, with **iDestination** (print in an HTML file, ...).
 - In Windows Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
- 2. Specify the name of report to print with **iPrintReport**.

Additional parameters must be specified in the following cases:

- Printing a bounded report based on a data file.
- Printing a report with parameters based on a data file.
- Printing a bounded report with parameters based on a data file.

WINDEV and WINDEV Mobile examples

The "RPT_Customer" report is based on the "Customer" file. This report is used to print the characteristics of customers.

The "RPT_Customer" report is printed during a click on the "BTN_Customer" button. In this example, the click code of "BTN_Customer" is as follows:

```
• WINDEV code:

// Open the preview window
iPreview()
// Print the RPT_Customer report
iPrintReport(RPT_Customer)
```

```
• Windows Mobile code:

// Select the printer
iConfigure()
// Print the RPT_Customer report
iPrintReport(RPT_Customer)
```

WEBDEV example

The "RPT_Customer" report is based on the "Customer" file. This report is used to print the characteristics of customers.

The "RPT_Customer" report is displayed in a PDF file when clicking the "LINK_Customer" link. Then, the PDF file is displayed in the browser. In this example, the server click code of the "LINK_Customer" link is as follows:

```
// Generates a unique PDF file name
UniqueFileName is string = fWebDir() + "\" + "RPT_Customer_" + ...
DateSys() + TimeSys() + ".pdf"
// Configures the print destination
iDestination(iPDF, UniqueFileName)
// Prints the RPT_Customer report
iPrintReport(RPT_Customer)
// Sends the file to the browser
FileDisplay(UniqueFileName, "application/pdf")
// Deletes the file
fDelete(UniqueFileName)
```

Note: **..SourceName** used in the report code allows you to find out and change the name of the data file associated with the report.

12.2.2 Printing a bounded report based on a data file

To print a bounded report based on a data file, you must:

- 1. Configure the destination of report print:
 - In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
 - In WEBDEV, with **iDestination** (print in an HTML file, ...).
 - In Windows Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
- 2. Specify the following parameters with **iPrintReport**:
 - the name of the report to print.
 - the value of the lower bound for the report.
 - the value of the upper bound for the report.

WINDEV and WINDEV Mobile example

The "RPT_Customer" report is a bounded report based on the "Customer" file. This report is used to print the characteristics of the customers whose name starts with the letter "D".

The lower bound corresponds to "D". The upper bound corresponds to "D"+hMaxVal.

The "RPT_Customer" report is printed when clicking the "BTN_CustomerD" button.

In this example, the click code of "BTN_CustomerD" is as follows:

- WINDEV code:

```
// Open the preview window
iPreview()
// Print the RPT_Customer report
iPrintReport(RPT_Customer, "D", "D" + hMaxVal)
```

- Windows Mobile code:

```
// Select the printer
iConfigure()
// Print the RPT_Customer report
iPrintReport(RPT_Customer, "D", "D" + hMaxVal)
```

WEBDEV example

The "RPT_Customer" report is a bounded report based on the "Customer" file. This report is used to print the characteristics of the customers whose name starts with the letter "D". The lower bound corresponds to "D". The upper bound corresponds to "D"+hMaxVal. The "RPT_Customer" report is printed in an HTML file and sent to the browser when the "BTN_CustomerD" button is clicked.

In this example, the click code of the "CustomerD" button is as follows:

```
// Generates a unique file name
UniqueFileName is string = fWebDir() + "\" + "RPT_Customer_" + ...
DateSys() + TimeSys() + ".pdf"
// Configures the print destination
iDestination(iPDF, UniqueFileName)
// Prints the RPT_Customer report
iPrintReport(RPT_Customer, "D", "D" + hMaxVal)
// Sends the file to the browser
FileDisplay(UniqueFileName, "application/pdf")
// Deletes the file
fDelete(UniqueFileName)
```

12.2.3 Printing a report with parameters based on a data file

To print a report with parameters, parameters must be passed to the report.

To print a report with parameters based on a data file, you must:

1. Configure the destination of report print:

- In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
- In WEBDEV, with **iDestination** (print in an HTML file, ...).
- In Windows Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.

2. Specify the following parameters with **iPrintReport**:

- the name of the report to print.
- the parameters expected by the report (in the order specified in the report).

WINDEV example

The "RPT_Param" report corresponds to a mailshot sent to the customers. This report expects in parameter:

- the text of this mailshot entered in the "EDT_MailshotText" edit control.
- the date for sending this mailshot entered in the "EDT_SendDate" edit control.

This report is printed during a click on the "BTN_Print" button. The "RPT_param" report is based on the "Customer" file.

In this example:

- the click code of the "BTN_Print" button is as follows:

```
// Open the preview window
iPreview()
// Print the report while passing parameters
iPrintReport(RPT_Param, EDT_MailshotText, EDT_SendDate)
```

- the code of the report is as follows:

```
// Retrieve the parameters passed to the report and
// pass the default parameters for the report test
PROCEDURE RPT_Param(sText = "Default", sDate = Today())
```

Windows Mobile example

The "RPT_Param" report corresponds to an invoice. This report expects in parameter the delivery date entered in the edit control named "EDT_DelDate".

This report is printed during a click on the "BTN_Print" button. The "RPT_param" report is based on the "Customer" file.

In this example:

- the click code of the "BTN_Print" button is as follows:

```
// Select the printer
iConfigure()
// Print the report while passing parameters
iPrintReport(RPT_Param, EDT_DelDate)
```

- the code of the report is as follows:

```
// Retrieve the parameters passed to the report and
// pass the default parameters for the report test
PROCEDURE RPT_Param(sDate = Today())
```

WEBDEV example

The "RPT_Order" report is a report with parameters used to print the order of a customer identified by his customer number. The "RPT_Order" report is displayed in a PDF file. This PDF file is sent to the browser when the "BTN_Order" button is clicked.

In this example:

- the click code of the "BTN_Order" button is as follows:

```
// Generates a unique file name
UniqueFileName is string = fWebDir() + "\" + "Invoice_" + ...
DateSys() + TimeSys() + ".pdf"
// Configures the print destination
iDestination(iPDF, UniqueFileName)
// Generates the order form
iPrintReport(RPT_Order, CUSTOMER.CUUNIQUEKEY)
```

- the code of the report is as follows:

```
// Retrieve the parameters passed to the report and
// pass the default parameters for the report test
PROCEDURE RPT_Order(nCustomerID = 450)
```

12.2.4 Printing a bounded report with parameters based on a data file

To print a bounded report with parameters based on a data file, you must:

- 1. Configure the destination of report print:
 - In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
 - In WEBDEV, with **iDestination** (print in an HTML file, ...).
 - In Windows Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
- 2. Specify the following parameters with **iPrintReport**:
 - the name of the report to print.
 - the value of the lower bound for the report.
 - the value of the upper bound for the report.
 - the parameters expected by the report (in the order specified in the report).

WINDEV example

The "RPT_Param" report corresponds to a mailshot sent to the customers whose name starts with the letter "D" or "E".

The lower bound corresponds to: "D" and the upper bound to: "E".

This report expects in parameter:

- the text of this mailshot entered in the "EDT_MailshotText" edit control.
- the date for sending this mailshot entered in the "EDT_SendDate" edit control.

This report is printed during a click on the "BTN_Print" button. The "RPT_param" report is based on the "Customer" file.

In this example, the click code of "BTN_Print" button is as follows:

```
// Open the preview window
iPreview()
// Print the report while passing bounds and parameters
iPrintReport(RPT_Param, "D", "E", EDT_MailshotText, EDT_SendDate)
```

Windows Mobile example

The "RPT_Param" report corresponds to an invoice sent to the customers whose name starts with the letter "D" or with the letter "E".

The lower bound corresponds to: "D" and the upper bound to: "E".

This report expects in parameter the delivery date entered in the edit control named "EDT_DelDate".

This report is printed during a click on the "BTN_Print" button. The "RPT_Param" report is based on the "Customer" file.

In this example, the click code of "BTN_Print" button is as follows:

```
// Select the printer
iConfigure()
// Print the report while passing bounds and parameters
iPrintReport(RPT_Param, "D", "E", EDT_DelDate)
```

WEBDEV example

The "RPT_Param" report is used to print the invoices of a given customer for the month of January 2004.

The lower bound corresponds to: "01/01/2004" and the upper bound to: "31/01/2004".

This report expects the customer number in parameter. This report is displayed in a PDF file when the "BTN_Print" button is clicked.

In this example, the click code of "BTN_Print" button is as follows:

```
// Generates a unique file name
UniqueFileName is string = fWebDir() + "\" + "RPT_Param_" + ...
DateSys() + TimeSys() + ".pdf"
// Configures the print destination
iDestination(iPDF, UniqueFileName)
// Prints the report while passing bounds and parameters
iPrintReport(RPT_Param, "01012003", "31012003", CustNum)
// Sends the file to the browser
FileDisplay(UniqueFileName, "application/pdf")
```

12.3 Report on query

To print a report based on a query, you must differentiate between two cases:

- Report on query without parameter.
- Report on a query with parameters.

Notes:

- If parameters must be passed to the report, these parameters must be specified in **iPrintReport**, after the name of report to print.
- **..SourceName** used in the report code allows you to find out and modify the name of the query associated with the report.

12.3.1 Printing a report on query without parameter

To print a query without parameter, you must:

- 1. Configure the destination of report print:
 - In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
 - In WEBDEV, with **iDestination** (print in an HTML file, ...).
 - In Windows Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
- 2. Specify the name of report to print with **iPrintReport**.

For example:

- In WINDEV: to print the "RPT_Query" report:

```
// Open the preview window
iPreview()
// Print the RPT_Query report
iPrintReport(RPT_Query)
```

- In Windows Mobile: to print the "RPT_Query" report:

```
// Select the printer
iConfigure()
// Print the RPT_Query report
iPrintReport(RPT_Query)
```

- In WEBDEV: to print the "RPT_Query" report in HTML format:

```
// Generates a unique file name
UniqueFileName is string = fWebDir() + "\RPT_Query_" + ...
    DateSys() + TimeSys() + ".htm"
// Configures the print destination
iDestination(iHTML, UniqueFileName)
// Prints the RPT_Query report
iPrintReport(RPT_Query)
// Extracts the name and extension of the generated file
FileName is string = fExtractPath(UniqueFileName, ...
    fFileName + fExtension)
// Sends the file to the browser
FileDisplay(FileName)
```

12.3.2 Printing a report on a query with parameters

To print a query with parameters, you must:

1. Configure the destination of report print:
 - In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
 - In WEBDEV, with **iDestination** (print in an HTML file, ...).
 - In Windows Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
2. Pass the parameters to the query with **iInitReportQuery**.
3. Specify the name of report to print with **iPrintReport**.

For example: The "RPT_Query" report is based on a query with parameters. The parameters expected by the query are the name and type of the company. These parameters have been entered in the window for starting the report print.

- WINDEV code:

```
// Open the preview window
iPreview()
// Pass the parameters to the query associated with the report
iInitReportQuery(RPT_Query, CompanyName, CompanyType)
// Print the list of companies
iPrintReport(RPT_Query)
```

- Windows Mobile code:

```
// Select the printer
iConfigure()
// Pass the parameters to the query associated with the report
iInitReportQuery(RPT_Query, CompanyName, CompanyType)
// Print the list of companies
iPrintReport(RPT_Query)
```

- WEBDEV code:

```
// Generates a unique file name
UniqueFileName is string = fWebDir() + "\RPT_Query_" + ...
    DateSys() + TimeSys() + ".htm"
// Configures the print destination
iDestination(iHTML, UniqueFileName)
// Pass the parameters to the query associated with the report
iInitReportQuery(RPT_Query, CompanyName, CompanyType)
// Prints the RPT_Query report
iPrintReport(RPT_Query)
// Extracts the name and extension of the generated file
FileName is string = fExtractPath(UniqueFileName, ...
    fFileName + fExtension)
// Sends the file to the browser
FileDisplay(FileName)
```

Note: If some of the parameters expected by the query are not specified (value corresponding to NULL for example), the selection conditions corresponding to these parameters will be ignored.

12.4 Report on Table control

A report on Table control must be printed from the window displaying the Table control to print. If the button was not created automatically, you must:

1. Create a button in the window (or page) containing the Table control.
2. Configure the destination of report print:
 - In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
 - In WEBDEV, with **iDestination** (print in an HTML file, ...).
 - In Windows Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
3. Specify the name of report to print with **iPrintReport**.

Notes:

- If additional parameters must be passed to the report, these parameters must be specified in **iPrintReport**, after the name of the report to print.
- If the window containing the Table control, source of report, is opened several times (non-modal opening by **OpenChild**), the full name of Table control to print must be passed in parameter to the report. For example:

```
// The ..FullName property will contain the window name
// that must be opened with an alias
// to distinguish between each window
iPrintReport(RPT_Customer_Table, TABLE_Customer..FullName)
// Opening code of report:
PROCEDURE RPT_Customer_Table(NameSourceTable)
MyReport..SourceName = NameSourceTable
```

- **..SelectionOnly** is used to find out and modify the records that will be printed in the report: printing all records or the selected records only.

R&Q

Important

In Reports & Queries, a report can be printed:

- by starting a printing (🖨).
- by running the report test (🧪).

Therefore, no programming is required to print the report.

WINDEV and WINDEV Mobile example

The information to print is in the Table control named "TABLE_CustomerTable" found in the "WIN_Customer" window.

The report to print is named "RPT_Customer" and it is printed during a click on the "BTN_Print" button.

In this example, the click code of "BTN_Print" button is as follows:

In WINDEV:

```
// Open the preview window
iPreview()
// Print the report based on "TABLE_CustomerTable" control
iPrintReport(RPT_Customer)
```

In Windows Mobile:

```
// Select the printer
iConfigure()
// Print the report based on "TABLE_CustomerTable" control
iPrintReport(RPT_Customer)
```

WEBDEV example

The information to print is in the Table control named "TABLE_CustomerTable" found in the "PAGE_Customer" page.

The report to print is named "RPT_Customer" and it is displayed in a PDF file during a click on the "BTN_Print" button.

In this example, the click code of "BTN_Print" button is as follows:

```
// Generates a unique file name
UniqueFileName is string = fWebDir() + "\" + ...
    "RPT_Customer_" + DateSys() + TimeSys() + ".pdf"
// Configures the print destination
iDestination(iPDF, UniqueFileName)
// Prints the report based on TABLE_CustomerTable control
iPrintReport(RPT_Customer)
// Displays the report in PDF format
FileDisplay(UniqueFileName, "application/pdf")
// Deletes the file
fDelete(UniqueFileName)
```

12.5 Report on variable

12.5.1 Printing a report on a variable

To print a report on variable, you must:

1. Initialize the value of variable used.
2. Sort the array if necessary (with **ArraySort** for example).
3. Configure the destination of report print:
 - In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
 - In WEBDEV, with **iDestination** (print in an HTML file, ...).
 - In Windows Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
4. Specify the name of report to print with **iPrintReport**.

Notes

- **..SourceName** used in the report code allows you to find out and change the name of the variable associated with the report.
- **..DisplayedItem** used in the report code allows you to find out and change the name of the variable associated with the item.

WINDEV and WINDEV Mobile example

The "RPT_gglEvent" report is based on the MyEvents variable (gglEvent type). This report is used to print the list of events found in a Google calendar.

The "RPT_gglEvent" report prints during a click on the "BTN_Event" button. In this example, the click code of "BTN_Event" button is as follows:

• WINDEV code:

```
// Open the preview window
iPreview()
// Print the RPT_gglEvent report
iPrintReport(RPT_gglEvent)
```

• Windows Mobile code:

```
// Select the printer
iConfigure()
// Print the RPT_gglEvent report
iPrintReport(RPT_gglEvent)
```

WEBDEV example

The "RPT_gglEvent" report is based on the MyEvents variable (gglEvent type). This report is used to print the list of events found in a Google calendar.

The "RPT_gglEvent" report is displayed in a PDF file during a click on the "LINK_Event" link. Then, the PDF file is displayed in the browser. In this example, the server click code of the "LINK_Event" link is as follows:

```
// Generates a unique PDF file name
UniqueFileName is string
UniqueFileName = fWebDir() + "\" + "RPT_gglEvent_" + ...
    DateSys() + TimeSys() + ".pdf"
// Configures the print destination
iDestination(iPDF, UniqueFileName)
```



```
// Prints the RPT_gglEvent report
iPrintReport(RPT_gglEvent)
// Sends the file to the browser
FileDisplay(UniqueFileName, "application/pdf")
// Deletes the file
fDelete(UniqueFileName)
```

12.5.2 Printing a report with parameters on a variable

To print a report with parameters based on a variable, you must:

- 1. Initialize the value of array variable used.
- 2. Sort the array if necessary (with **ArraySort** for example).
- 3. Configure the destination of report print:
 - In WINDEV, with **iPreview** (print preview, print in an HTML file, ...).
 - In WEBDEV, with **iDestination** (print in an HTML file, ...).
 - In WINDEV Mobile, with **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
- 4. Specify the following parameters via **iPrintReport**:
 - the name of the report to print.
 - the parameters expected by the report (in the order specified in the report).

WINDEV and WINDEV Mobile example

A window lists the organizers and their events. The "RPT_gglEvent" report is printed during a click on the "Print the organizer events" button. This report expects in parameter the organizer for which the events will be listed.

WINDEV code:

- the click code of the "Print the organizer events" button is as follows:

```
// Open the preview window
iPreview()
// Print the report while passing parameters
iPrintReport(RPT_gglEvent, Organizer[1])
```

- the code of the report is as follows:

```
// Retrieve the parameters passed to the report
PROCEDURE RPT_gglEvent(MyOrganizer)
```

Windows Mobile code:

- the click code of the "Print the organizer events" button is as follows:

```
// Select the printer
iConfigure()
// Print the report while passing parameters
iPrintReport(RPT_gglEvent, Organizer[1])
```

- the code of the report is as follows:

```
// Retrieve the parameters passed to the report
PROCEDURE RPT_gglEvent(MyOrganizer)
```

WEBDEV example

The "RPT_gglEvent" report is displayed in a PDF file during a click on the "Print the organizer events" link. Then, the PDF file is displayed in the browser. This report expects in parameter the organizer for which the events will be listed.

In this example:

- the click code of the "Print the organizer events" button is as follows:

```
// Generates a unique file name
UniqueFileName is string = fWebDir() + "\" + "gglEvent_" + ...
DateSys() + TimeSys() + ".pdf"
// Configures the print destination
iDestination(iPDF, UniqueFileName)
// Generates the report
iPrintReport(RPT_gglEvent, MyOrganizer)
// Sends the file to the browser
FileDisplay(UniqueFileName, "application/pdf")
// Deletes the file
fDelete(UniqueFileName)
```

- the code of the report is as follows:

```
// Retrieve the parameters passed to the report
PROCEDURE RPT_gglEvent(MyOrganizer)
```

12.6 Print functions

The following functions can be used to easily print reports:

iPreview/iDestination	Configures the destination of the print.
iPrintReport	Prints a report created with the report editor.
iInitReportQuery	Initializes the query linked to the report (for a query with parameters).
iNestedHeaderFooter	Prints (or not) the page headers and footers of nested report.

See the online help for more details.

Note: These functions are the functions required to print the report. Other functions are presented in the "WLanguage functions", page 145.

13. PRINTING DUPLICATE COPIES

13.1 Overview

For legal reasons, it may be useful to keep an identical copy of a report (an invoice for example). Furthermore, due to the large amount of data to process (statistical reports for example), some reports may take quite a long time to generate. The Duplicate feature automatically proposes a solution to fix this problem.

The principle is straightforward: When editing a report (report created in the report editor or by programming), its identical copy can be generated and stored in a file in ".dpl" format. This allows you to re-edit a duplicate copy of the report. The data is not processed again so it is protected against the modifications and this re-edit operation is performed immediately.

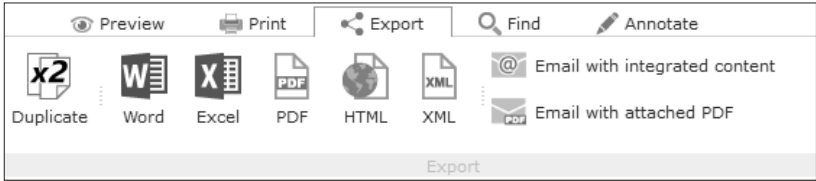
- The creation of a duplicate copy can be performed:
- from the print preview (WINDEV only).
 - by programming.
- You have the ability to sign a duplicate copy via a certificate:
- from the print preview.
 - by programming.

- Notes:**
- Printing a report with duplicate copies slows the print down. Indeed, the duration for creating the duplicate copy is included in the print duration. We advise you to run performances tests.
 - Printing a page of a report with duplicate copy triggers the creation of the duplicate copy for the entire report.

13.2 Creating a duplicate copy from the print preview (WINDEV only)

13.2.1 Print available by default

By default, the "Export" pane of the print preview proposes the "Duplicate" option allowing the use to create a duplicate copy of the report currently displayed.



- When selecting this option, a file picker allows the user to:
- enter the name of the duplicate file that will be created.
 - specify the certificate that will be used to sign the duplicate. The signature of the duplicate is optional.
- By default, the duplicate:
- is saved in the directory of the application.
 - has no password.
 - is named: Dupli_YYYYMMDD_HHMMSS.dpl.

- Notes:**
- To avoid proposing the print of duplicate copies, use **iParameterPreview** (associated with the **iButtonDuplicate** constant).
 - To modify the default parameters of the duplicate copy, use **iParameterDuplicate** before displaying the print preview.

13.2.2 Customizing the created duplicate copy

To modify the default parameters for the duplicate copies created from the print preview, you have the ability to use **iParameterDuplicate** associated with the **iDplPrintout** constant. Example:

```
// Configure the duplicate copy of invoice
iParameterDuplicate(iDplPrintout, "C:\Temp", ...
    "MyDuplicate_%d_%t", "DDDD DD MMMM YYYY", "HH.MM.SS")

// Print the duplicate
iPreview()
iPrintReport(RPT_Invoice)
```

- iParameterDuplicate** allows you to:
- customize the duplicate options proposed by default by the "Duplicate" option found in the print preview.
 - automate the print of a duplicate copy from the print preview. Indeed:
 - the duplicate copy will be generated during the print (performed from the print preview).
 - if an annotation or an edit control is modified in the print preview, a new duplicate copy will be generated during the next printout.
 - if the print is performed on several printers, a duplicate copy will be generated for each printer.

Note: The list of duplicate copies created during the last print is returned by **iLastFile**.

13.3 Creating a duplicate copy by programming

- To create a duplicate copy by programming:
1. Use **iParameterDuplicate** (associated with the **iDplPrintout** constant) to configure the options for creating the duplicate copies.
 2. Use **iDestination** and specify that the print must be performed on a printer. The duplicate copy will be created while the print is performed.
- To stop printing the duplicate copies, use **iParameterDuplicate** (associated with the **iDplNone** constant).

13.4 Signing a duplicate copy

The duplicate copies can be signed via a certificate. This signature ensures the integrity of the reports stored. The signature is included in the duplicate.

This signature automatically perform a timestamp. If may be required legally for some uses.

The signature standard used is PKCS7.

You have the ability to sign a duplicate copy via a certificate:

- from the print preview.
- by programming:
 - The signature of duplicate copies is performed by the functions:

iDestination	Configures the print destination.
iParameterDuplicate	Configures the automatic creation of the duplicate copy generated during the next print.

- The signature of a duplicate file is checked by the function:

CertificateCheckDuplicate	Checks the signature of a duplicate.
---------------------------	--------------------------------------

Note: If the report includes a Signature control (linked to a certificate) or if **iPrintSignature** is used, the duplicate will be automatically signed during its creation. See the online help for more details.

13.5 Using a duplicate copy

To use an existing duplicate copy, you can:

- drop the ".dpl" file into the editor of WINDEV or WEBDEV: the duplicate copy is automatically opened in the print preview and it can be printed.
- use **iPrintDuplicate**. This function is used to print the specified ".dpl" file. To display the duplicate copy in the print preview, all you have to do is use **iPreview**. For example:

```
iPreview()  
iPrintDuplicate("C:\My Projects\My_Project\Exe\Dupli_20101021_112712.dpl")
```

Viewer of duplicate copies

Installed with the application on the end-user computer, this viewer is used to easily view and print the content of a duplicate file. To do so, double-click the ".dpl" file.

To install this tool, select the optional "DplViewer" tool in the wizard for generating the setup program.

Note: Colored barsWhen displaying a duplicate copy in the print preview, several bars may appear at the bottom of the preview to inform the user of the duplicate file used:

Viewer of duplicate copies

These color bars are also available in the viewer of duplicate copies.

14. PRINTING A TEXT IN WATERMARK

14.1 Overview

A report (and a duplicate copy) can include an additional text in the format of a watermark, such as "Confidential" or "Duplicate" printed across the report.

The watermark can be created:

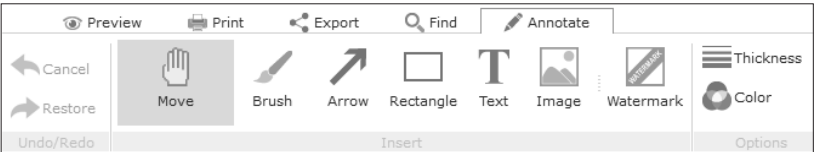
- from the print preview (WINDEV only).
- by programming.

Notes:

- If a watermark is defined for a report, it is defined for all the pages of the report.
- A single watermark can be defined for each report. You cannot define several watermarks.
- The watermark is printed on the pages containing information only. It is not printed on empty pages.

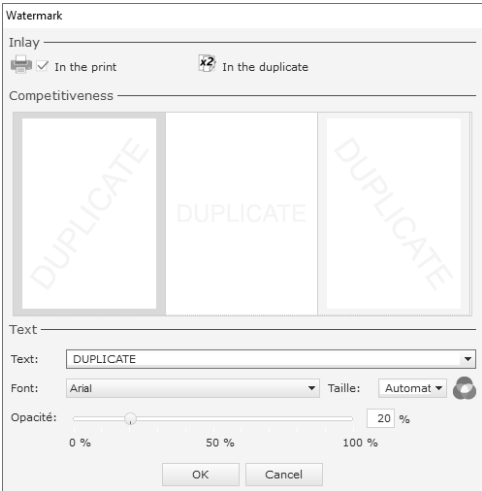
14.2 Creating a watermark from the print preview

On the "Annotate" pane of the print preview, the "Watermark" option allows the user to create a watermark in the report currently displayed.



When selecting this option, a specific window allows the user to define:

- the destination of the watermark: print and/or duplicate copy.
- the positioning of the watermark: ascending diagonal, descending diagonal or horizontal.
- the text of the watermark, its font, its size, its color and its opacity.



14.3 Creating a watermark by programming

Different methods can be used to create a watermark by programming:

- Method 1: Full configuration: using a Watermark variable.
- Method 2: Quick configuration: direct use.

14.3.1 Method 1: Full configuration: using a Watermark variable

This method is as follows:

1. Define a Watermark variable. This variable contains all the characteristics of the watermark to print.
2. Use `iParameterWatermark` to add the watermark into your report.
3. Start printing the report.

For example:

- Code used to print a watermark in background of a report created with the report editor:

```
// Print a watermark in report background
MyWatermark is Watermark

// Text of the watermark
MyWatermark..Text = "Confidential"
// Font used
MyWatermark..Font..Name = "Arial"
// The watermark will be horizontally centered at bottom
MyWatermark..Position = iCenterH + iBottom

// Print preview
iPreview()

// Indicates that the watermark will be printed during the print
// and in the duplicate copies
iParameterWatermark(iWatermarkPrinting + ...
iWatermarkDuplicate, MyWatermark)

// Starts printing the report
iPrintReport(RPT_Commercial)
```

- Code used to print a watermark in background of a report created with the print functions:

```
MyWatermark is Watermark

// Text of the watermark
MyWatermark..Text = "Confidential"

// Print preview
iPreview()

// Implement a watermark in document background only
iParameterWatermark(iwatermarkPrinting, MyWatermark)

// Prints several pages with watermark
FOR i = 6 TO 10
iPrint(" Page " + i)
iSkipPage()
END
// The last page is empty and it has no watermark

// Ends the print
iEndPrinting()
```

14.3.2 Method 2: Quick configuration: direct use

- This method is as follows:
1. Use `iParameterWatermark` to specify the text of the watermark that will be added into your report. In this case, the "Arial" font is used, with an automatic size and with an opacity set to 20%.
 2. Start printing the report.

For example:

- Code used to print a watermark in background of a report created with the report editor:

```
// Print preview
iPreview()

// Indicates that the watermark will be printed during the print
// and in the duplicate copies
iParameterWatermark("Confidential")

// Starts printing the report
iPrintReport(RPT_Commercial)
```

- Code used to print a watermark in background of a report created with the print functions:

```
// Print preview
iPreview()

// Implement a watermark in document background only
iParameterWatermark(iWatermarkPrinting, "Confidential")

// Prints several pages with watermark
FOR i = 6 TO 10
  iPrint(" Page " + i)
  iSkipPage()
END
// The last page is empty and it has no watermark

// Ends the print
iEndPrinting()
```

15. PROGRAMMING A REPORT

15.1 Overview

This chapter presents the processes that can be run in a report and in its elements (controls and blocks) as well as their operating mode and sequence.

Several WLanguage functions and properties can be used in these processes. Some functions and properties specific to the reports are presented in this chapter.

See the online help for more details (keyword : "Report processes").

15.2 Sequence of processes associated with a report

Several operations are performed when printing a report. These operations are performed in the following order:

1.	Running the "Opening" process of report.
2.	Running the "Before re-printing" process of the report (for a direct print from the print preview or from an export from the print preview).
3.	Running the "Initialization" processes of blocks and the "Initialization" processes of controls.
4.	Initializing the data source.
5.	For the reports based on a data file, a HFSQL view or a memory zone: <ul style="list-style-type: none">• Running the "Filter" process of the report.• Running the sort procedures of the report.
6.	Initializing the margins and defining the Portrait/Landscape mode.
7.	Resizing the report (if necessary) according to the specified format.
8.	Printing the Start of document block and running its processes (*).
9.	Printing the Page Header block and running its processes for each report page (*).
10.	Print loop: <ul style="list-style-type: none">• Reading the record.• Running the "Filter" process of the report (for the reports based on a query or on a table). If not OK, goes to the next record.• Performing calculations on the values of items.• Running the "Read" process of the report.• Test of the break:<ul style="list-style-type: none">• on the break item for the automatic breaks.• on the value returned by the "Break" process for the programmed breaks.If OK, change of break and print one or more Break Header and Break Footer blocks.• Printing the Body block.

	<div>Note: When printing a Break Header, Break Footer or Body block, the following operations are performed:</div> <ul style="list-style-type: none">Running the "Before printing" process of the block and the "Before printing" process of block controls.Performing calculations on the controls of the block.Enlarging the controls and the auto-sizable blocks.Printing and running the "After printing" process of the block.
11.	Printing the End of Document block and running its processes (*).
12.	Printing the Page Footer block and running its processes for each report page (*).
13.	Running the "Closing" process of report.

(*) The order in which the Start of document, Page Header, Page Footer and End of Document blocks are printed can be modified in the "Options" tab of the report description ("Report description" from the popup menu of the report).

Reminder : If the destination of the printout is an HTML or XML file, the Page Header and Page Footer blocks are generated once only.

15.3 Processes associated with the report

The following processes are associated with the reports (order of appearance in the code editor):

- Opening the report.
- Reading data.
- Filtering (taking the record into account?).
- Closing the report.
- Before printing the report again.

Several processes are associated with the reports. These processes are run when printing the report.

15.3.1 "Opening" process

The "Opening" process of the report is run when starting the report print.

Reminder: The order in which the processes are run when printing a report (including during a test) is as follows:

- "Opening" the report.
- "Initializing" the report blocks.
- "Initializing" the controls.

Use examples

This process is used to:

- declare the global variables of the report.
- modify the report bounds with `..MinValue` and `..MaxValue`.
- initialize the passwords of data files if necessary.
- configure the destination of the print (**iPreview**).
- pass parameters to the query linked to the report (**ilnitReportQuery**).
- retrieve the parameters passed to the report.
- change the data source of the report.

Note: **iPrint** must not be used in the opening process of the report.

Password of the data file used as source for the report

If the data file used as source for the report is password protected, this password must be specified before opening the data file (by **HPass** for example).

If the data file was not opened before printing the report, the password can be specified in the "Opening" process of the report.

Initializing the variables

When printing a report from the print preview, the report is automatically re-run when:

- the user changes printer.
- the user uses one of the toolbar buttons (print in PDF mode, in HTML mode, in an email, ...).

In this case, the data printed in the report can differ from the data displayed in the preview.

The variables used by the report must be initialized in the "Report opening" process rather than in the code of the button that starts the print.

15.3.2 "Reading data" process

This process is run when browsing the records to print (in the print loop), once the record was read.

Use examples

This process is used to:

- perform block substitutions (as the next record to print is known).
- select records.
- perform specific calculations.

Note: This code returns True if there are still records to print and False if there is no record to print. To avoid printing a record, use the process named "Filter (take the record into account?)".

15.3.3 "Filter" process

This process is run:

- when the data source is initialized (for the reports based on a file, a view or a memory zone).
- in the print loop, once the record is read (for the reports based on a query or on a table).

This process is used to filter the records that must be displayed in the report.

Use examples

This process is used to:

- select records.
- perform advanced calculations (sub-total, ...) for the reports based on files, views or memory zones.

Using the process

If this process contains code lines, these code lines must return a value:

- Cancel to stop reading the records (and to stop the print).
- True to specify that the record matches the filter and that it must be printed.
- False to specify that the record does not match the filter and that it must not be printed.

15.3.4 "Closing" process

This process is run when the report print is ended, just before closing the report.
This process must not contain any print functions.
Note: **iCurrentFile** is used to find out the name of the file generated for the report (for the exports in PDF, HTML, ...).

Use examples

This process is used to:

- find out the status of the current print (**iReportPrintingStatus**).
- open a dialog box.

15.3.5 "Before re-printing" process

This process is run during a print performed from the print preview (direct print or export in PDF, XLS or RTF format).
This process is run after the "Opening" process of report.
If this process returns:

- True, the re-print will be performed.
- False, the re-print will be canceled.

Notes:

- **iDestination** used in this process allows you to find out the destination of the report (HTML, PDF, ...).
- **iCurrentFile** used in this process allows you to find out the name of the file generated for the report (for the exports in PDF, HTML, ...).

Use examples

This process is used to:

- find out the quality of the generated PDF before the export in PDF format.
- add a watermark.
- ask for a password before the print or the export.

15.4 Break process

If your report contains one or more programmed breaks, the break process is associated with the *Break header* block.
This process allows you to use a programmed break. Two types of programming can be implemented for the breaks:

- the break code returns True or False (True to perform the break, False not to perform the break).
- the break code returns the break value. If the value changes, the break is automatically performed by the report (see the example below).
The break value can correspond to the value of an item, a report control, a variable, ...


Note: If the break value corresponds to the value of a report control, the returned value will be false if this block is found in the *Body* block of report. Indeed, the break process is run before filling the controls found in the *Body* block. In this case, you must return the value of item linked to the control.

```
// -- Break code
nSupplier is int = 0
// Retrieve the supplier identifier
nSupplier = ExtractString(gfsLine, 1, TAB)
// Find the supplier name
HReadSeekFirst(SUPPLIER, SupplierNum, nSupplier)
IF HFound() = True THEN
    // Initialize the control
    SupplierName = SUPPLIER.Company
END
// Return the supplier identifier
RESULT nSupplier
```

Important: The code of the break is run **only** if the corresponding break is a programmed break (linked to no sort criterion).

15.5 Sort procedure

A sort procedure can be implemented in a report based on a data file or in a report based on a memory zone.
This procedure can be used to modify the sort criteria dynamically.
To implement a sort procedure:

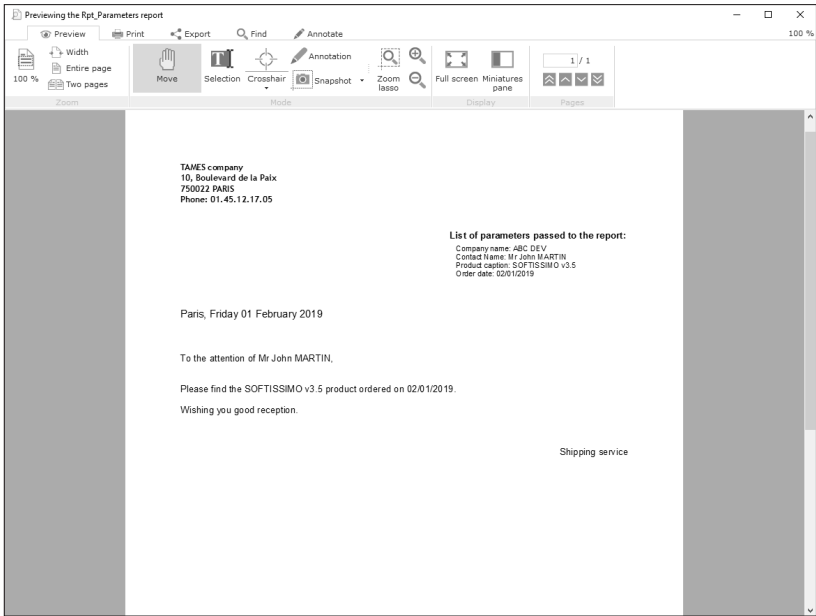
1. On the "Modification" pane, in the "Report" group, click "Edit the sorts". The list of sort criteria is displayed.
2. Click the "New sort criterion" button ().
3. Check "Sort on WLanguage code".
4. Enter the name and value of the sort procedure parameters (for example: "SortVal("City")").
5. Validate the addition of the new sort criterion.
6. Create the sort procedure. This procedure must return the sort argument.

For example:

```
PROCEDURE SortVal(Criterion)
// Sort by NAME or by CITY
SWITCH Criterion
    Case "City": RESULT Customer.City
    Case "Name": RESULT Customer.Name
END
```

15.6 Report with parameters

When printing a report, you have the ability to pass parameters to this report. The report is called "Report with parameters".
For example, your report corresponds to a mailshot sent to all the customers. The text of the mailshot is entered in a control before running this report. This text corresponds to a parameter expected by the report.



15.6.1 Creating a report with parameters

The creation of a report with parameters is performed in several steps:

- 1. Creating a report.
- 2. Modifying this report to support the parameters:
 - Retrieving the parameters in the "Opening" code of the report via the following syntax:

```
PROCEDURE <ReportName>(<NameParam1>, ...  
    <NameParam2>= "<DefaultValueParam2>", ...  
    ...  
    <NameParamN> = "<DefaultValueParamN>")
```

Reminder: Some of the parameters passed to a procedure can be optional parameters. When declaring the procedure, the optional parameters must be described in last position (on the right), with a default value preceded by the "=" sign. See the online help for more details.

- Using these parameters in the report controls.
- 3. Modifying (if necessary) the application in order to specify the different parameters.
- 4. Printing the report while passing parameters.

Example:
The program required to create a report with parameters is illustrated by the following example:

- The "RPT_ParamReport" report corresponds to a mailing sent to the customers.
- This report expects in parameter the text of the mailing entered in the "EDT_MailingText" edit control.

- Implementation:**
- Create the "RPT_ParamReport" report:
 - Click among the quick access buttons.
 - The window for creating a new element is displayed: click "Report" then "Report".
 - The wizard for report creation starts.
 - In the wizard, select "Mailshot" and validate.
This report is linked to the "Customer" file of your analysis.
 - Customize this report.
 - Enter the following code in the "Opening" process of the "RPT_ParamReport" report:

```
// Retrieve the parameter passed to the report and  
// pass a default parameter for the report test  
PROCEDURE RPT_ParamReport(sText = "Default value")
```

- In the *Body* block of this report, add the "STC_MailingText" static control used to display the text of the mailing.
- In the "Before printing" process of "STC_MailingText" control, type the following code:

```
// Retrieve the mailing text  
STC_MailingText = sText
```

- In an application window, add:
 - the "EDT_MailingText" edit control allowing the user to type the text of the mailing.
 - the "BTN_Print" button used to print the report.

15.6.2 Printing a report with parameters

To print a report with parameters, you must:

- Configure the destination of the report print with **iPreview** (print preview, print in an HTML file, ...).
- Specify the following parameters with **iPrintReport**:

- the name of the report to print.
- the parameters.

- Notes:**
- If the report is based on a HFSQL view, the name of this HFSQL view must be specified in **iPrintReport**, before the parameters. See the online help for more details.
 - If the report is based on a memory zone, the name of this memory zone must be specified in **iPrintReport**, before the parameters. See the online help for more details.
 - If the report is bounded, the maximum and minimum values of the bounds must be specified in **iPrintReport**, before the parameters. See the online help for more details.

Example:

The "RPT_ParamReport" report corresponds to a mailing sent to the customers. This report expects in parameter the text of the mailing entered in the "EDT_MailingText" edit control.

The "RPT_ParamReport" report is printed during a click on the "BTN_Print" button.

In this example, the code of the "BTN_Print" button is as follows:

```
// Open the preview window
iPreview()
// Print the report while passing parameters
iPrintReport(RPT_ParamReport, EDT_MailingText)
```

15.6.3 Running the test of a report with parameters

To run the test of a report with parameters from the report editor, a default value should be given to the parameters in the parameter declaration.

For example, to run the test of the report found in the previous example, enter the following code in the "Opening" process of the "RPT_ParamReport" report:

```
// Retrieve the parameter passed to the report and
// pass a default parameter for the report test
PROCEDURE RPT_ParamReport(sText = "Default value")
```

15.7 Handling a report in a WLanguage process

To handle a report in a WLanguage process, all you have to do is use the name of this report. This name can be used:

- in a WLanguage function,
- with a WLanguage property,
- in direct assignment.

For example:

```
// Print the "RPT_ParamReport" report
iPrintReport(RPT_ParamReport)
// Modify the number of copies for the "RPT_Customer" report
RPT_Customer..NbCopy = 10
```

15.8 Keywords associated with reports

The following keywords can be used in the reports:

Keyword	Effect
MySource	Allows you to handle the current data source.
MyReport	Allows you to handle the current report.

See the online help for more details.

15.9 Properties associated with reports

The following properties can be used on the reports:

AllModifiableWithReportsAndQueries	Used to find out whether a report can be entirely modified in Reports & Queries.
BackgroundImage	Used to find out and modify the background image of an element: window, page, control or report.
BackgroundImagePrinted	Used to manage the print mode for the background image of a report.
BottomMargin	Used to find out and modify the bottom margin of a report.
BrowsedItem	Allows you to find out and modify the item used for the automatic browse a report on a data file.
HSpace	Used to find out and modify the horizontal spacing between the report labels. This spacing was defined when creating the Label report.
HTML	Used to find out whether the current report is printed in HTML mode.
InterfaceModifiableWithReportsAndQueries	Used to find out whether a report can be modified in Reports & Queries.
KeepVisibleHierarchy	Used to manage the print mode of the hierarchy: printing the entire hierarchy or printing the hierarchy visible in the control.
LabelHeight	Used to find out and modify the height of labels in a Label report.
LabelNumberToStart	Used to find out and modify the number of the label on the label sheet from which the print will be performed.
LabelWidth	Used to find out and modify the width of labels in a "Label" report.
LeftMargin	Used to find out and modify the left margin of a report.
MaxValue	Used to find out and modify the upper bound of a report based on a data file.
MemZoneName	Allows you to find out and modify the name of the memory zone used as data source of a report.
MinValue	Used to find out and modify the lower bound of a report based on a data file.
MsgSortSel	Used to find out and modify the wait message displayed when sorting and selecting the records found in the report.
Name	Used to find out the name of a control, report or report block.
NbColumnn	Used to find out the number of report columns.
NbCopy	Used to find out and modify the number of report copies that must be printed.
NbLabel	Used to find out and modify the number of labels printed across the page width.
NbLabelCopy	Used to find out and modify the number of copies of identical labels that will be printed.
NbPrintedRec	Used to find out the number of records already printed.

NbRecRead	Used to find out the number of records read: selected both by the source of the report AND by the filter code (if it exists).
ODBCConnectionName	Allows you to find out and specify the name of the ODBC connection used when printing a report based on a “query performed on a connection”.
ODBCDatabaseName	Allows you to find out and specify the name of the database used when printing a report based on a “query performed on a connection”.
ODBCPassword	Allows you to find out and specify the password used to connect to a database via an ODBC driver or via a Native Access.
ODBCSQLCode	Used to find out and modify the SQL code of the “query performed on a connection”, source of the report.
ODBCUserName	Allows you to find out and specify the user name used to connect to a database via an ODBC driver or via a Native Access.
PageByPagePrint	Used to manage the print mode of the report (page by page or for the entire report).
PageEndStick	Used to manage the position of the Page Footer block.
PageNumber	Used to find out and modify the page number of a PDF file displayed in report background.
Plane	Used to find out and modify the active plane of report or the plane to which a control belongs.
Process	Used to modify the processes that must be run by the events available on a control, window, page, report, ...
ReturnedValue	Used to find out and modify the value returned by a report.
RightMargin	Used to find out and modify the right margin of a report.
RightToLeft	Used to find out the actual writing direction, effective on a control, a column, a window or a report.
SelectionOnly	Used to find out and modify the mode for printing the selected rows in a report on table or treeview table.
SourceName	Used to find out the name of the data source of the report and to change the data source of the report.
TextDirection	Allows you to find out and modify the writing direction used in a window or in a control (useful for managing languages such as Arabic or Hebrew).
TopMargin	Used to find out and modify the top margin of a report.
TotalNbRec	Used to find out and modify the number of records to print.
ViewName	Used to find out and modify the name of the HFSQL view, data source of a report.
VSpace	Used to find out and modify the vertical spacing between the report labels.. This spacing was defined when creating the Label report.

15.10 WLanguage functions

Several WLanguage functions can be used in the processes of a report (created with the report editor). However, some of them are not recommended such as the print functions specific to the print in WLanguage or the functions for accessing the records used by the report.

15.10.1 WLanguage functions not recommended in the processes

The following functions should not be used in the processes of a report, blocks or controls. If these functions are used, the report execution may be incorrect.

Print functions: iEllipse, iEndPrinting, ilmageHeight, iPrint, iPrintBarCode, iPrintImage, iPrintWord, iPrintZone, iPrintZoneRTF, ilmageWidth, iLink, iNewLine, iXPos, iYPos, iSkipLine, iLine, iHLine et iVLine.

Functions for accessing a record: All the HFSQL functions that modify a record in memory or the current record in the base file of the report are not recommended in the processes of a report, blocks or controls. Indeed, these functions may modify the browse performed in the base file of the report.

These functions are mainly: HAdd, HCreation, HCreationIfNotFound, HLast, HWrite, HReadxxxx, HModify, HPrevious, HSetPosition, HFirst, HSeek, HBackward, HCross, HNext, HDelete, ...

R&Q

Important

The following functions are not allowed in Reports & Queries:

HAddUser	HStopServer	HChangeLogDir
HCopyFile	HCreation	HCreationIfNotFound
HDBCcreation	HWrite	HImportHF55
HImportText	HImportText_55	HFree
HMigrateLinkedCompositeKey	HModify	HModifyDatabaseRights
HModifyFileRights	HModifyServerRights	HModifyGroup
HModifyDatabaseProperty	HModifyFileProperty	HModifyServerProperty
HCross	HRegenerateFile	HDelete
HDeleteDatabase	HDeleteFile	HDeleteGroup
HDeleteLink	HDeleteUser	HViewToFile

15.10.2 Print functions of WLanguage that can be used in a report

The following print functions can be used in a report:

iBorder	Prints a border at specified coordinates.
iCloseReport	Stops printing the current report immediately.
iConfigure	Configures the printer for a print performed in WLanguage: <ul style="list-style-type: none">• by opening the configuration window.• by modifying the printer used for the current print.
iCreateFont	Creates a new print font.
iDidotFont	Selects the unit that will be used for the height of the fonts (created by iCreateFont): DIDOT point or PICA point.
iDocumentCanceled	Used to find out whether the current print was canceled by the user.
iDocumentPrinted	Allows you to find out whether the user has requested to print the document from the print preview.
iEndReport	Forces the report print to stop.
iEscape	Sends an ESCAPE command or a data command to a printer.
iFont	Selects the default font.
iForceComplement	Forces the printing of a Body Complement block.
iInfoPrinter	Retrieves the characteristics of the current or default printer.
iInitReportQuery	Initializes the query linked to the report (for a query with parameters).
iInitReportQueryConnection	Initializes the query linked to the report.
iLastFile	Returns: <ul style="list-style-type: none">• the full name of the last file generated during the print (PDF, RTF, text, ...).• the list of created duplicate files.
iListNestedReports	Returns: <ul style="list-style-type: none">• the list of nested reports currently printed (reports run by iPrintReport in one of the processes of the report currently printed).• the list of sequenced reports currently printed (iSequenceAdd).
iListPrinter	Returns the list of printers installed on the current computer.
iMargin	Defines the "logical" print margins.
iMMToPica	Converts the coordinates of a point (expressed in millimeters) into system coordinates.
iNestedHeaderFooter	Prints (or not) the page headers and footers of nested report.
iPageEnd	Forces the move to the next page during the report print.
iPageHeight	Calculates the height of printable page (in millimeters) while taking the top or bottom margins into account.
iPageHeightLeft	Calculates the available height (in millimeters) left on the current page while taking into account the defined margins (top or bottom).
iPageNum	Returns or initializes the number of the page currently printed.

iPageWidth	Calculates the width (in millimeters) of the printable page while taking the left or right margins into account.
iParameter	Retrieves the printer configuration and configures the print.
iParameterPDF	Defines the options for the generated PDF file.
iParameterPreview	Configures the print preview: <ul style="list-style-type: none">• display a message regarding the resources of current computer.• display or not the ribbon panes (as well as the buttons found).• the export options available in the mini preview window.
iPicaToMM	Converts the coordinates of a point (expressed in system coordinates) into millimeters.
iPreview	Configures the print destination.
iPrintBlock	Forces a report block to print.
iPrintBodyComplement	Forces a body complement block to print while specifying the height of complement block.
iPrintReport	Prints a report created with the report editor.
iReportPrintingStatus	Returns the status of the current print.
iReset	Re-initializes the print settings (margins, font, paper format, orientation, ...) stored in the print library.
iRoundBorder	Prints a border with rounded outlines at specified coordinates.
iSelectFont	Creates a new print font from the standard window for font selection...
iSkipPage	Generates a page break: the corresponding code is sent to the printer queue.
iSubstBlock	Substitutes a report block for another block while the report is printed.
iTextHeight	Calculates the height of font for the text to print (en millimeters).
iTextWidth	Calculates the width (in millimeters) of the text to print, according to the specified font.
iTransparentMagenta	Modifies the management mode of Magenta color in order for this color to be considered as being transparent (or not) in all the forthcoming prints.
iWindowCancel	Configures the display of the window used to cancel the current print...
iZoneHeight	Calculates the necessary height of a fixed-width area in order for the entire text using the current font to be printed in this area.

INDEX

A			
Add		Bound	44
Cell into a table	96	Bounded report	44
Column into a table	97	Break	35
Align		Break process	138
Alignment options	83	Coded	138
Align controls	83	Break footer	62
Associated properties		Break header	61
Report controls	85	C	
Associated properties		Calculated	71
Report blocks	67	Calculations in a report	36
Reports	143	Cell	95
Automatic retrieval	53	Add	96
B		Height	97
		Resize	97
Bar code	72	Selection	96
Before re-printing (process)	138	Width	97
Blank report	24	Characteristics	
Blank (type of report)	24	Report blocks	63
Block	59	Report controls	78
Associated processes	66	Table	94
Associated properties	67	Chart	
Body	62	Type of control	73
Body complement	62	Check box	75
Break footer	62	Circle	75
Break header	61	Close a report	53
Characteristics	63	Closing (process)	138
Create a block	64	Column	
Delete a block	65	Add	97
Description	63	Delete	98
End of document	62	Move	98
Iteration	62	Remove	98
Page footer	62	Select	96
Page header	61	Composite report	24, 103
Parameters	63	Composite (Report)	103
Position	65	Composite (type of report)	24
Print options	63	Control	69
Programming	65	Align	83
Reorganize	65	Associated processes	84
Setting	52	Associated properties	85
Start of document	61	Bar code	72
Types of blocks	61	Calculated	71
Body	62	Characteristics	78
Body complement	62	Chart	73
		Check box	75

L		PDF	40
		Print preview	40
Label report	24	Report	113
Label (type of report)	24	Report on query	121
Line	75	Report on Table control	123
Looper	48	Report with parameters	141
M		RTF	40
		Text file	40
		XML	40
Main editor window	20	Print a report	116
Move		Print margins	37
Column (table)	98	Print options of blocks	63
Control	82	Print preview	40
Multicolumn report	24	Process	
Multicolumn (type of report)	24	Before re-printing	138
Multilingual feature of report controls	79	Break	138
N		Closing	138
		End of print	138
		Filter	137
Nested report	108	Opening	136
Nest reports	108	Reading data	137
Number of pages	72	Report	136
O		Report blocks	66
		Report controls	84
		Start of print	136
Off-report control	83	Programming	
Open a report	53	Report	135
Opening (process)	136	Report blocks	65
Organizer	49	Q	
Organizer report	24		
Organizer (type of report)	24	Query	44
P		Embedded query	45
		Independent query	45
Page footer	62	Print the report	121
Page format	37	Query with parameters	45
Page header	61	Query with parameters	45
Page Header block	61	R	
Page number	72		
Parameter		Reading data (process)	137
Report	139	Rectangle	75
Report blocks	63	Registration card	19
PDF	40	Reorganize the blocks	65
PDF file	40	Report	13
Position of blocks	65	Associated keywords	142
Preset	72	Associated properties	143
Print		Available types	24
Duplicate copy	128	Blocks	59
HTML	40	Bound	44
On a form	53	Calculations	36
Paper	40		

Close	53	RTF file	40
Composite report	103	Run the test	
Creation	23	Report	37
Data source	43	Report with parameters	142
Description	51		
Form	53, 54		
Format	37		
Functions	145		
Languages	53		
Margins	37		
Nested report	108		
Open	53		
Print	40, 113, 116		
Processes	136		
Programming	135		
Report controls	69		
Retrieval	53		
Run the test	37		
Save	37		
Skin template	37		
Structure	13		
Using tables	89		
With parameters	139		
Report editor	16		
Developer Version	18		
Documentation	18		
Main window	20		
Minimum configuration	19		
Registration card	19		
Setup	19		
User Version	18		
Report languages	53		
Report on file			
Print	116		
Report on form	24		
Report on query			
Print	121		
Report on Table control			
Print	123		
Report on variable			
Print	125		
Reports and Queries	18		
Report template	58		
Report with parameters	139		
Create	140		
Print	141		
Run the test	142		
Resize the table cells	97		
Ribbon (menu)	21		
Ribbon menu	21		
RTF	40		

S

T

TreeView Table (type of report)	
Create	93
Type	
Type of blocks	61
Type of controls	69
Type of report	24

U

URL link	76
----------	----

W

Wizard	
Report creation	23
WLanguage functions	145
WLanguage variable	46

X

XML	40
XML file	40

PART 2

Query editor

24

QUERY EDITOR - SUMMARY

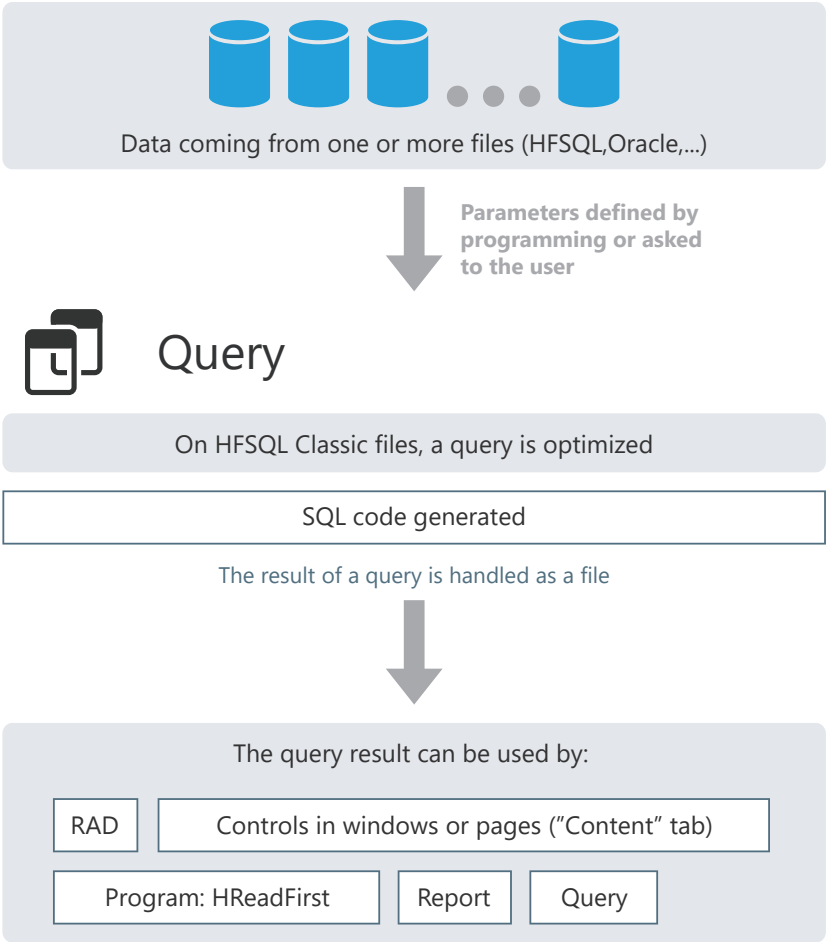
1.	Introduction	
1.1	What is a query?	159
1.2	Query editor	161
1.2.1	Overview	161
1.2.2	Differences between the standard query editor and "Reports & Queries"	161
1.3	Main interface of query editor	163
1.4	A menu shaped like a ribbon	165
1.5	Query and project	167
2.	How to create a query?	
2.1	Overview	168
2.2	Select query	169
2.2.1	Overview	169
2.2.2	Creating a select query	171
2.3	Insert query	172
2.3.1	Overview	172
2.3.2	Creating an insert query	173
2.4	Update query	174
2.4.1	Overview	174
2.4.2	Creating an update query	175
2.5	Delete query	176
2.5.1	Overview	176
2.5.2	Creating a delete query	177
2.6	Union query	178
2.6.1	Overview	178
2.6.2	Creating a union query via the description window	180
2.7	Query in SQL code	181
2.7.1	Overview	181
2.7.2	Creating a query in SQL code	181
2.7.3	Reverse engineering of query in SQL code	181
2.7.4	Runtime modes of a query in SQL code	182
2.7.5	Available SQL commands	182
2.8	Creation mode of queries	188
3.	Description of a query	
3.1	Overview	189
3.2	Description window	189
3.2.1	Displaying the description window	189
3.2.2	Description window of select query	189
3.2.3	Description window of union query	194
3.2.4	Description window of query	195
3.3	Characteristics of a query	196
3.3.1	Items used in a select query	196
3.3.2	Sorting the records selected by a Select query	198

3.3.3	Operations that can be performed in a select query	200
3.3.4	Modifying the values displayed in the result of a select query	201
3.3.5	Selection conditions	201
3.3.6	Sequence of selection conditions	204
3.3.7	Displaying the records corresponding to a given period	205
3.3.8	Limiting the number of records displayed in a Select query	206
3.3.9	Keeping (or not) the duplicates	207
3.3.10	Joins between data files	207
3.3.11	Consistency with the analysis	208
3.4	Opening, closing and saving a query	208
3.5	Query documentation	209
4.	Running the query test and using it	
4.1	Query test	211
4.1.1	Exporting the query result to a Word file	213
4.1.2	Exporting the query result to an Excel file	213
4.1.3	Exporting the query result to an XML file	213
4.1.4	Exporting the query result to a HFSQL data file	213
4.1.5	Printing the query result	214
4.2	Using the query result.....	214
4.2.1	Query in a WINDEV window or in a WEBDEV page	214
4.2.2	Query in a control found in a WINDEV window or in a WEBDEV page	215
4.2.3	Query in a report	216
4.3	Running the query and handling the result with the WLanguage functions	216
5.	Advanced query	
5.1	Overview	217
5.2	Sub-query	217
5.2.1	Sub-query as selection condition	217
5.2.2	Sub-query used as data source of a query	218
5.3	Query with parameters	218
5.3.1	Creating a query with parameters	219
5.3.2	Test of a query with parameters	219
5.3.3	Using a query with parameters	219
5.3.4	Two methods for passing parameters to a query	222

1. INTRODUCTION

1.1 What is a query?

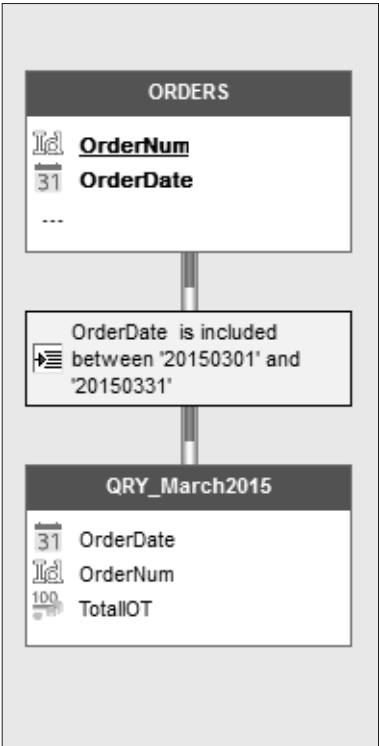
A query is used to interrogate a database in order to view, insert, modify or delete data. The query structure defines the data used.
A query can interrogate one or more data files.
The query editor allows you to easily create queries without programming.



For example, regarding the Sales Management application, you want to find out:

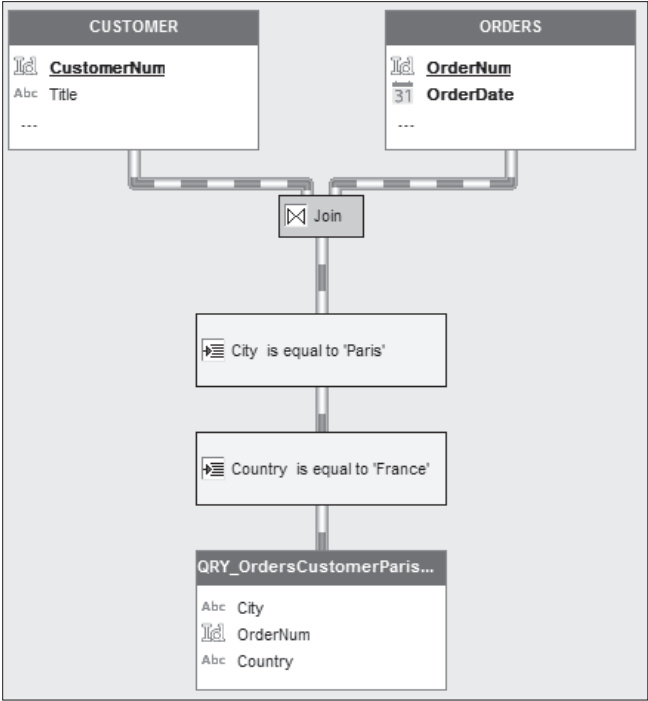
- the orders placed in March 2015. You are going to create a query used to select the orders placed between 03/01/2015 and 03/31/2015 in the ORDERS file.

The QRY_March2015 query interrogates the ORDERS file.



- the numbers of the orders placed by customers who live in Paris (France). You are going to create a query that will interrogate:
 - the CUSTOMER file to select the customers who live in Paris (France).
 - the ORDERS file to display the order numbers.

The QRY_OrdersCustomerParisFrance query interrogates two data files (CUSTOMER and ORDERS).



1.2 Query editor


1.2.1 Overview

The query editor is used to automatically create queries on the data files. Creating a query is simple and intuitive. The query is automatically generated (in optimized SQL code) and the query test can be immediately run.

No programming hassle: windows, pages, tables, combo boxes and reports found in your project can be based on queries.

1.2.2 Differences between the standard query editor and "Reports & Queries"

Two versions of the query editor are available:

- the standard query editor included in the product.**
This version is supplied with WINDEV, WEBDEV or WINDEV Mobile. This version can only be used when developing a WINDEV or WINDEV Mobile application, or a WEBDEV site.
- the user version of query editor (also called Reports & Queries).** This editor is supplied with WINDEV only. This version is independent of WINDEV and it can be used by the end users.
Some features of query editor are not available in this version. The  icon indicates the differences between Reports & Queries and the standard query editor.

Standard query editor

The standard query editor allows you to create queries and to run these queries from the editor or from an application developed with WINDEV, WINDEV Mobile or WEBDEV.

The dongle of WINDEV, WINDEV Mobile or WEBDEV is required to use the query editor. It must be plugged into the parallel port of your computer when using the editor.

However, no dongle is required by the end user (to run a query from a WINDEV, WINDEV Mobile or WEBDEV application).

Several examples are supplied with the integrated query editor. In these examples, you will discover the main queries that can be used in your applications.

To find out the list of examples for using reports:

- Display the home page (Ctrl + <).
- Type the "Report" keyword.
- The list of examples for using reports is displayed.

Check these examples, they will teach you a lot and you will be able to use them when you create your own queries!

User version of query editor (Reports & Queries - WINDEV only)

Reports & Queries allows you to create and run queries from the editor (WINDEV is not required).

The created queries can be run from the query editor or from a WINDEV application (if this application supports the queries created by Reports & Queries).

No dongle is required to use Reports & Queries.

Some operations are required to create new queries or to modify the existing queries with Reports & Queries. See "Distributing Reports & Queries", page 233 for more details.

Documentation

Reports & Queries and the standard query editor are supplied with the same online help. The standard query editor is supplied with a printed documentation.

- **Printed documentation:**
The printed documentation was intended to help you discover and use the query editor. The first part of this guide is dedicated to the report editor.
The "Query editor guide" complements the "Tutorial" and the "Concepts".
- **Online help**
The online help is intended to help you handle the editor. The online help is proposed:
 - in the format of a local help. The necessary files are installed on the development computer.
 - in the format of an Internet help. This help can be accessed from any computer equipped with an Internet access. This help is updated on a regular basis.

To access the online help, use one of the following methods:

- on the "Home" pane, in the "Online help" group, expand "Help" and select "Help index".
- press the F1 key.

To modify the type of online help used:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of XXX".
2. In the "Help" tab, select the access mode to the help database: "Use the help database by Internet" or "Use the local help database" as well as the preferred language.
3. Validate.

Setup

- **Installing the standard query editor:** The query editor is automatically installed with WINDEV, WEBDEV or WINDEV Mobile.
- **Installing Reports & Queries:** To install Reports & Queries, follow the instructions given when installing the WINDEV application.


Minimum configuration required for Reports & Queries

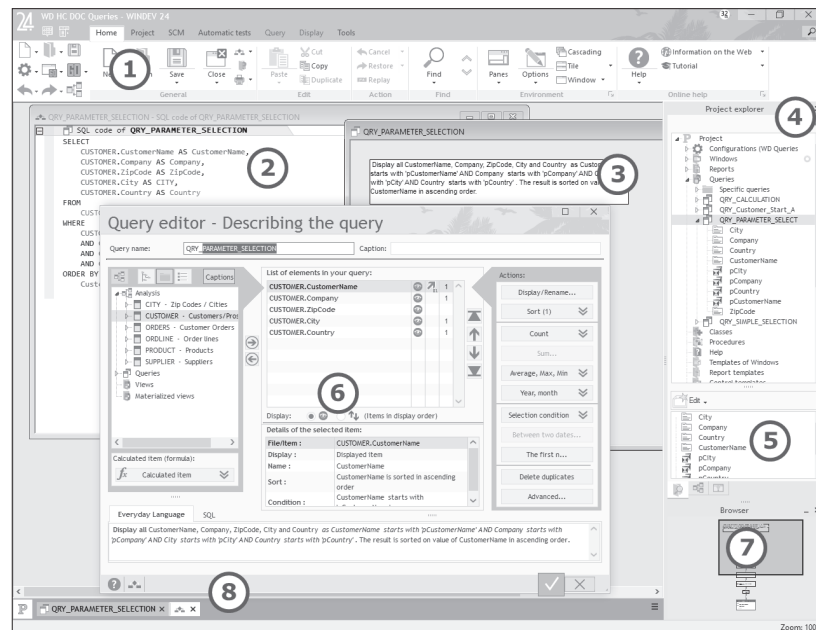
Reports & Queries must be installed on a computer with the following characteristics:

- 512 MB of RAM (1 GB or more recommended).
- 800 MB of disk space.
- Windows 2000, 2003, XP, Vista, 7, 8 or later.

1.3 Main interface of query editor

To display the query editor, you must:

- create a new query:
 - click  among the quick access buttons.
 - the window for creating a new element is displayed: click "Queries".
- open an existing query (via the "Project explorer" pane for example).



The main elements of query editor are as follows:


- 1 Ribbon**
Gives quick access to the main features of query editor.
- 2 SQL code of current query**
- 3 Graphic representation of current query**
Scrollbars are used to scroll the query representation if the size of this representation is quite huge. You also have the ability to perform a zoom (via the mouse for example: Ctrl + Mouse wheel).
- 4 Dockable panes**
Displays:
 - the shortcuts to the main features available in the current editor ("Commands" pane),
 - the most common features ("Commands" pane),
 - the available elements (files, items, reports, controls, ...) in the current project ("Project explorer" pane).

Important

In Reports and Queries, only the accessible elements are listed in the "Project explorer" pane.

- 5** **List of items and parameters of the selected query**

If the query is run by programming (**HExecuteQuery**), the order in which the parameters are passed (for a query with parameters) corresponds to the order specified in this list.

**Important**

In Reports & Queries, the list of items and parameters is not available.
 - 6** **Description window of the current query.**

Used to find out and modify all the characteristics of the query.
 - 7** **Browser**

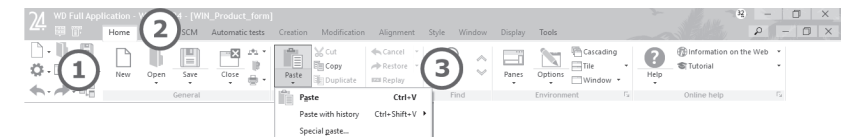
When a query cannot be entirely displayed in the workspace, the browser allows you to easily move within this query.
 - 8** **Opened documents**

Displays a button for each document (reports, queries, ...) opened in the editor. To display the requested element in the foreground, click the corresponding "button".

1.4 A menu shaped like a ribbon

WINDEV, WEBDEV and WINDEV Mobile present the different options of the editors via a ribbon. This ribbon includes panes in which the different options of editors are grouped.

The ribbon includes several areas:



1. The area of quick access buttons.
2. The pane area.
3. The area of groups of options.

The quick access buttons (1)

The quick access buttons are used to perform the most usual operations, common to all the editors: save, open, create, ...



The 3 buttons found at the top of this area are specific buttons:

- the logo of the version is used to display:
 - the "About..." window.
 - the custom-menus.
 - the drop-down menus of earlier versions (version 17 and earlier).
- the 2 following pictograms are used to restore the menus and the toolbars found in the earlier versions (version 17 and earlier).

The pane area (2)

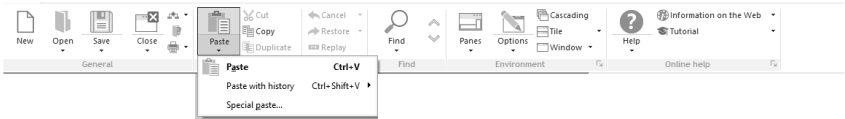


The different ribbon panes are used to access the options of different editors. Several types of panes are available:

- the current pane: The pane tab appears in light gray and a colored line is displayed at the top of tab.
- the popup panes, specific to the current element: These panes are related to the current editor or to the selected element. The pane name is displayed:
 - in yellow for WINDEV,
 - in orange for WINDEV Mobile,
 - in blue for WEBDEV.
- the other available panes: The pane name is displayed in white.

The option area (3)


The options are grouped on the panes in groups of options.



The options displayed in the ribbon differ according to the selected pane.

Several types of options are available:

- Options to check.
- Buttons to click.
- Button with arrow used to expand the options. Two types of buttons with arrow are available:
 - the buttons with arrow used to expand a menu.
 - the buttons with arrow used either to expand a menu (click on the arrow), or to perform a default action (click on the button icon).

The options are organized by group. Each group of options has a name and it can also include a group button: . This button is used to perform a specific action according to the current group: display the description of current element, display the help, ...

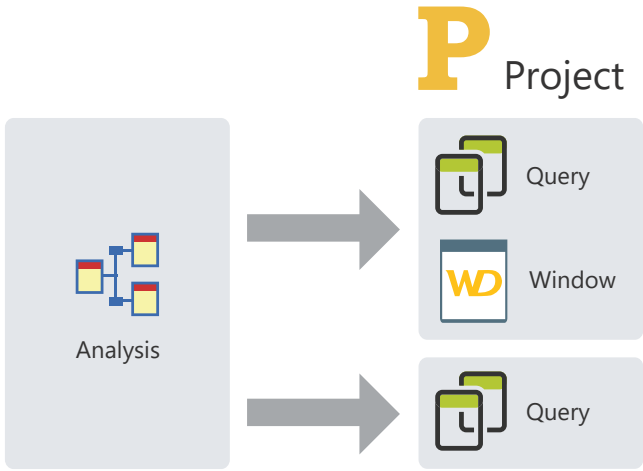
1.5 Query and project

A query is not necessarily attached to a project. However, a query must be included in a project in order to be called from a process of the project.

On the contrary, a query is necessarily attached to a WINDEV, WEBDEV or WINDEV Mobile analysis. A query can handle the data coming from an analysis other than the creation analysis. In this case, the structure of the files and items used by the query must be identical.

The WINDEV, WEBDEV or WINDEV Mobile analysis contains the description of the files that will be interrogated by the query.

If the analysis is modified, an automatic synchronization is performed between the analysis and the queries.



When creating the query:

- if a project associated with an analysis is opened, the query will be automatically attached to the analysis of the project.
- if no project is opened, the analysis used must be specified in the wizard for query creation.

2. HOW TO CREATE A QUERY?

2.1 Overview

Two methods can be used to create a query:

- creating the query with a wizard or with the description window, and generating the SQL code automatically.
- typing the SQL code. See "Creating a query in SQL code", page 181 for more details.

The query editor allows you to create several types of queries:

- **select query:** this type of query allows you to view a selection of records from one or more files in a database. This is the most common type of query.
See "Creating a select query", page 171 for more details.
- **insert query:** this type of query is used to add records to a database file.
See "Creating an insert query", page 173 for more details.
- **update query:** this type of query is used to update the records found in a database file.
See "Creating an update query", page 175 for more details.
- **delete query:** this type of query is used to delete records from a database file.
See "Creating a delete query", page 177 for more details.
- **union query:** this type of query is used to combine select queries.
See "Creating a union query via the description window", page 180 for more details.

R&Q

Important

Reports & Queries cannot be used to create insert queries, update queries or delete queries.

Note: All types of queries can be based on items belonging:

- to a data file described in the analysis.
- a query that was previously created with the query editor.

Once the query is created, the characteristics of this query can be modified at any time. See "Description of a query", page 189 for more details.

Insert query, Update query and Delete query:

The ability to create insert, update and delete queries is supported for compatibility with the SQL standard.

The use of native WINDEV commands (**HReadFirst**, **HReadNext**, **HAdd**, **HModify**, **HDelete**, ... of WLanguage) is much better suited to programming this type of process.

SQL is a query language used to interrogate data but it is not well suited for custom modifications of this data. To add, modify or delete records in a database file, the power and simplicity of WINDEV makes it the ideal and recommended solution.

R&Q

Important

Reports & Queries cannot be used to create insert queries, update queries or delete queries.

Principle:

To create a query, you must:

- specify the method used to create the query (wizard, description window or SQL code).
- specify the type of query (insert, update, ...).
- specify (if necessary) the WINDEV, WINDEV Mobile or WEBDEV analysis containing the description of data files on which the query will be based.
- select the items to use.
- enter the characteristics of the query (selection conditions, sort, ...).

2.2 Select query

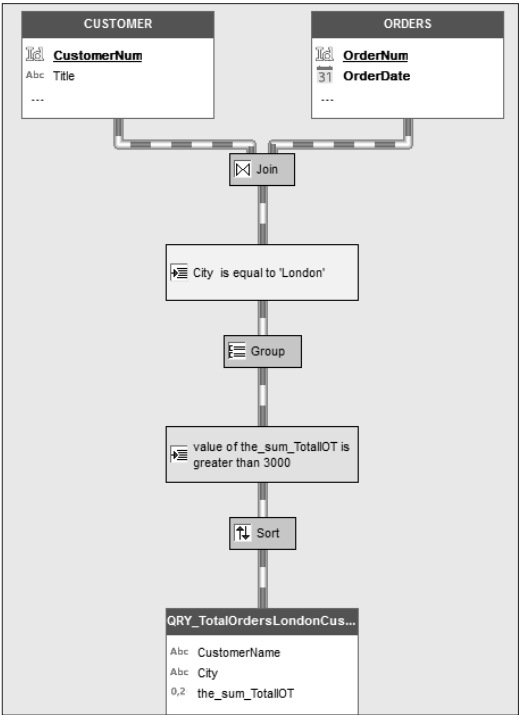
2.2.1 Overview

A **select query** (corresponding SQL statement SELECT) is used to view a selection of records in a database. This is the most common type of query.

To perform this selection, the main query characteristics are as follows:

- the type of join between the data files used (if several data files are used),
- the selection conditions of records (customers who live in Seattle, products whose price is greater than 40 Dollars, ...),
- the grouping of item values (grouping the products by type, the customers by age, ...),
- the operations performed on the content of items (total amount of orders for a customer, average sum spent per customer, ...),
- the sort of query result (result sorted by zip code, by price, ...).

For example, you want to display the total amount of orders per customer living in London. Only the customers with a total amount of orders exceeding 3000 Dollars will be considered. The result will be sorted by total amount in ascending order. To perform this query, the result will be automatically grouped by customer name and by city.



The query editor automatically generates the SQL code of this query.

To see the SQL code of a query:

- 1. Display the popup menu of query graph.
- 2. Select "SQL code".

You also have the ability to press the F2 key.

```
SELECT
  CUSTOMER.CustomerName AS CustomerName,
  CUSTOMER.CITY AS CITY,
  SUM(ORDERS.TotalIOT) AS the_sum_TotalIOT
FROM
  CUSTOMER,
  ORDERS
WHERE
  CUSTOMER.CustomerNum = ORDERS.CustomerNum
  AND
  (
    CUSTOMER.CITY = 'London'
  )
GROUP BY
  CUSTOMER.CustomerName,
  CUSTOMER.CITY
HAVING
  SUM(ORDERS.TotalIOT) > 3000
ORDER BY
  the_sum_TotalIOT ASC
```

2.2.2 Creating a select query

To create a new select query with the description window of a select query:

- 1. Click among the quick access buttons.
 - In the window that is displayed, click "Query".
 - The wizard for query creation starts.
- 2. Specify that you want to create a select query ("Select" option). Go to the next wizard step.
- 3. If your project is linked to no analysis, specify the analysis to which the query will be attached. Validate.
- 4. The description window of the query is displayed.
- 5. Type the query name (name of ".WDR" file corresponding to the query). This name will be used to identify the query in your programs.
- 6. Specify the caption of the query by briefly describing the purpose of query. This caption will be used in the project documentation.
- 7. In the left section, select the different items used by the query.
Note: To simplify the selection of items, these items can be displayed by associated data file, by associated query or in alphabetical order.
See "Items used in a select query", page 196 for more details.
- 8. In the middle section, select the query onto which an action will be performed (selection condition, sort, ...).
- 9. In the right section, define the actions that must be performed on the selected item (selection condition, sort, ...).
 - the selection conditions of a query,
 - the operations that can be performed in a query,
 - the sort order of query records.
- 10. Delete (if necessary) the "duplicate" records from the query result ("Delete the duplicates" button). In this case, each record found in the query result will be unique. For example, if a query is used to display all customers who placed at least one order, the customers who placed several orders will appear once only in the query result.
- 11. Validate the window. The graphic representation of the query is automatically displayed.
- 12. On the "Home" pane, in the "General" group, expand "Save" and click "Save" to save the query.

Notes

- You also have the ability to create a new Select query with a simplified wizard: check "Use the simplified wizard" in the wizard for query creation.
Note: The simplified wizard cannot be used to manage the "full-text" indexes.
- To modify the characteristics of a query, on the "Query" pane, in the "Edit" group, click "Description". See "Description of a query", page 189 for more details.
- To generate the SQL code corresponding to this query, select "SQL code" from the popup menu of the query. If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.
- You also have the ability to create a query from one or more data files described in the data model editor: to do so, "Drag and Drop" the files selected in the "Analysis" pane to the query editor.

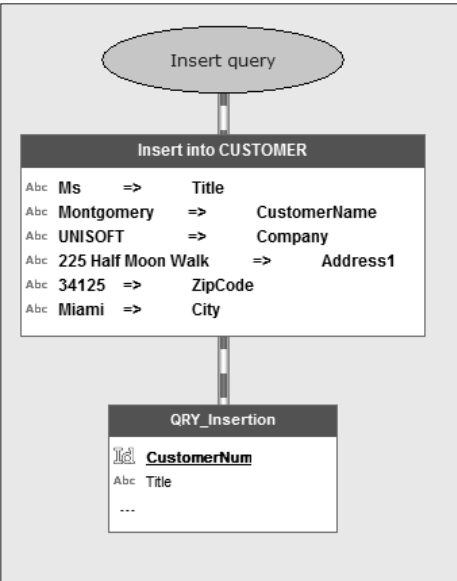
2.3 Insert query

2.3.1 Overview

An **insert query** (corresponding SQL statement INSERT) is used to add records into a database file.
The following information is required to perform an insert query:

- the data file into which the data will be inserted.
- the items into which new values will be inserted.

For example, creating a new customer in the "Customer" data file:



The query editor automatically generates the SQL code of this query.

To see the SQL code of a query:

1. Display the popup menu of query graph.
2. Select "SQL code".

You also have the ability to press the F2 key.
In our example:

```
INSERT INTO CUSTOMER
(
    Title,
    CustomerName,
    Company,
    Address1,
    PostalCode,
    City
)
VALUES
(
    'Miss',
    'Montgomery',
```

```
'UNISOFT',
'Accacia Avenue',
'EC4',
'London'
)
```



Important

Reports & Queries cannot be used to create insert queries.

2.3.2 Creating an insert query

To create a new insert query:

1. Click among the quick access buttons.
 - The window for creating a new element is displayed: click "Query".
 - The wizard for query creation starts.
2. Specify that you want to create an insert query ("Insert" option).
3. If your project is linked to no analysis, specify the analysis to which the query will be attached. Go to the next wizard step.
4. Select the data file into which the data will be inserted. Go to the next step.
5. The new values will be inserted into the different items of the data file. Click the "Value/Parameters" column for each relevant item. A window allows you to select the value or the parameter to insert.
6. Specify the value or the parameter containing the value to insert. The new values to insert can correspond to:
 - the default value defined in the analysis.
 - a value that must be specified. Specify this value.
 - a value contained in a parameter. Specify the name of the parameter.
 - a null value.
7. Validate your choice.
8. Repeat the operations 5 to 7 for each item into which a value must be inserted. Go to the next step.
9. Type the query name (name of ".WDR" file corresponding to the query). This name will be used to identify the query in your programs.
10. Specify the caption of the query by briefly describing the purpose of query. This caption will be used in the project documentation.
Note: The query name is automatically defined from the caption typed.
11. Validate the description of the query. The graphic representation of query is automatically displayed in the query editor.

Notes

- To modify the characteristics of a query: on the "Query" pane, in the "Edit" group, select "Description". See "Description of a query", page 189 for more details.
- To generate the SQL code corresponding to this query, select "SQL code" from the popup menu of the query. If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.

- You also have the ability to create a query from one or more data files described in the data model editor: to do so, "Drag and Drop" the files selected in the "Analysis" pane to the query editor.
- The test of the created query can be run from the query editor. See "Query test", page 211 for more details.

2.4 Update query

2.4.1 Overview

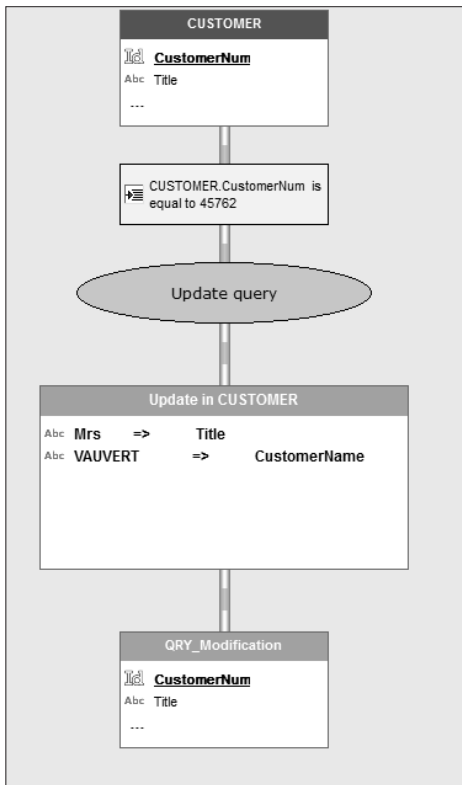
An Update query (corresponding SQL statement UPDATE) is used to modify records in a database file.

To perform this modification, the main characteristics of the query are as follows:

- the data file in which the data will be modified.
- the items in which the values will be modified.
- the selection conditions of the records (customers who live in Seattle, products whose price is greater than 40 Dollars, ...).

See "Selection conditions", page 201 for more details.

For example, customer #45762 has got married. Her title and her last name have been modified.



The query editor automatically generates the SQL code of this query.

To see the SQL code of a query:

1. Display the popup menu of query graph.
2. Select "SQL code".

You also have the ability to press the F2 key.

In our example:

```
UPDATE
  CUSTOMER
SET
  Title = 'Mrs',
  CustomerName = 'VAUVERT'
WHERE
  Customer.CustomerNum = 45762
```




Important

Reports & Queries cannot be used to create update queries.

2.4.2 Creating an update query

To create a new update query:

1. Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Query".
 - The wizard for query creation starts.
2. Specify that you want to create an update query ("Update" option).
3. If your project is linked to no analysis, specify the analysis to which the query will be attached. Go to the next wizard step.
4. Select the data file in which the data will be modified. Go to the next wizard step.
5. For each item whose value must be modified, click the "Value/Parameter" column. A window allows you to select the value or the parameter to modify.
6. Specify the value or the parameter containing the value to modify. The values to modify can correspond to:
 - the default value defined in the analysis.
 - a value that must be specified. Specify this value.
 - a value contained in a parameter. Specify the name of the parameter.
 - a null value.
7. Validate your choice.
8. Repeat the operations 5 to 7 for each item containing a value that must be modified.
9. Define the selection conditions. Click the "Condition" column of relevant items. See "Selection conditions", page 201 for more details. Go to the next wizard step.
10. Specify (if necessary) the sequence of the selection conditions with the conditional AND and OR AND operators:
 - *Conditional AND operator*: the selected records will match both conditions. For example, the customers whose name starts with the letter "A" AND who live in Denver.

- **Conditional OR operator:** the selected records will match one of the two conditions or both conditions. For example, the query selects the customers who live in Seattle OR in Denver.
Go to the next wizard step.
- 11. Type the query name (name of ".WDR" file corresponding to the query). This name will be used to identify the query in your programs.
- 12. Specify the caption of the query by briefly describing the purpose of query. This caption will be used in the project documentation.
Note: The query name is automatically defined from the caption typed.
- 13. Validate the description of the query. The graphic representation of query is automatically displayed in the query editor.

Notes

- All the characteristics of a query can be modified: on the "Query" pane, in the "Edit" group, click "Description". See "Description of a query", page 189 for more details.
- To generate the SQL code corresponding to this query, select "SQL code" from the popup menu of the query. If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.
- You also have the ability to create a query from one or more data files described in the data model editor: to do so, "Drag and Drop" the files selected in the "Analysis" pane to the query editor.
- The test of the created query can be run from the query editor. See "Query test", page 211 for more details.

2.5 Delete query

2.5.1 Overview

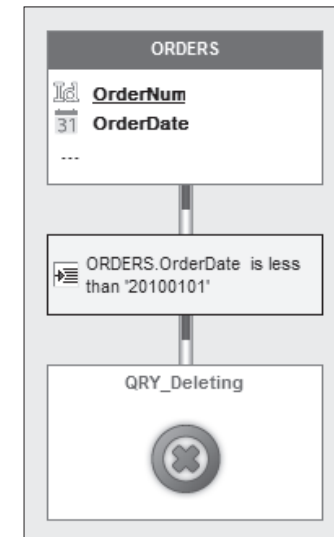
A **delete query** (corresponding SQL statement DELETE) is used to delete records from a file found in a database.

To perform this deletion, the main characteristics of the delete query are as follows:

- the data file from which the data will be deleted.
- the selection conditions of the records to delete (customers who live in Seattle, products whose price is greater than 40 Dollars, etc).

See "Selection conditions", page 201 for more details.

For example, delete the orders placed before January 1st 2010:



The query editor automatically generates the SQL code of this query.

To see the SQL code of a query:

1. Display the popup menu of query graph.
2. Select "SQL code".

You also have the ability to press the F2 key.

```
DELETE FROM
ORDERS
WHERE
ORDERS.OrderDate < '20100101'
```




Important

Reports & Queries cannot be used to create delete queries.

2.5.2 Creating a delete query

To create a new delete query:

1. Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Query".
 - The wizard for query creation starts.
2. Specify that you want to create a delete query with the wizard ("Delete" option).
3. If your project is linked to no analysis, specify the analysis to which the query will be attached.
4. Go to the next wizard step.
5. Select the data file from which the data will be deleted.
Go to the next step.

6. Define the selection conditions by clicking the "Condition" column of relevant items. See "Selection conditions", page 201 for more details.
Go to the next step.
7. Specify (if necessary) the sequence of the selection conditions with the conditional AND and OR AND operators:
- *Conditional AND operator*: the selected records will match both conditions. For example, the customers whose name starts with the letter "A" AND who live in Denver.
 - *Conditional OR operator*: the selected records will match one of the two conditions or both conditions. For example, the query selects the customers who live in Seattle OR in Denver.
Go to the next step.
8. Type the query name (name of ".WDR" file corresponding to the query). This name will be used to identify the query in your programs.
9. Specify the caption of the query by briefly describing the purpose of query. This caption will be used in the project documentation.
10. Validate the description of the query. The graphic representation of query is automatically displayed in the query editor.

- Notes**
- To modify all the characteristics of a query: on the "Query" pane, in the "Edit" group, click "Description". See "Description of a query", page 189 for more details.
 - To generate the SQL code corresponding to this query, select "SQL code" from the popup menu of the query. If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.
 - You also have the ability to create a query from one or more data files described in the data model editor: to do so, "Drag and Drop" the files selected in the "Analysis" pane to the query editor.
 - The test of the created query can be run from the query editor. See "Query test", page 211 for more details.

2.6 Union query

2.6.1 Overview

A *Union query* (corresponding SQL statement: UNION) is used to combine several select queries. Each select query must be based on data files of identical structure.

For example, select the customers whose last name starts with the letter 'A' and the prospects who live in Paris. To perform this selection, the two following queries must be combined:

- selecting the customers whose last name starts with the letter 'A'. This query is based on the "Customer" data file.
- selecting the prospects who live in London. This query is based on the "Prospect" data file.

The "Customer" and "Prospect" data files have the same structure in the analysis.

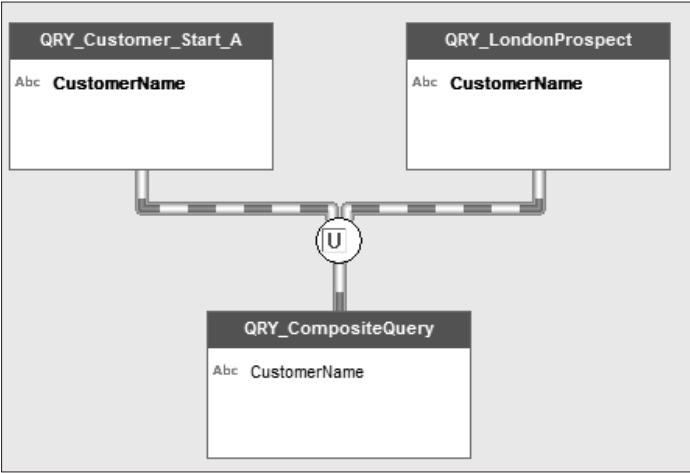
The records that are common to both data files ("Customer" and "Prospect") can:

- appear once only in the query result (UNION operator).
- appear twice in the query result (UNION ALL operator).

To combine queries, you must know:

- the name of the queries to combine.
- whether the common records must appear once or several times in the query result.

Note: The items displayed in the result of union query correspond to the items of the first query run. Therefore, in our example, the people who are both customer and prospect, whose name starts with the letter 'A' and who live in London will only appear once in the result of the query.



The query editor automatically generates the SQL code of this query.

To see the SQL code of a query:

1. Display the popup menu of query graph.
2. Select "SQL code".

You also have the ability to press the F2 key.

For example:

```
SELECT
  CUSTOMER.CustomerName AS CustomerName
FROM
  CUSTOMER
WHERE
  CUSTOMER.CustomerName LIKE 'A%'

UNION

SELECT
  PROSPECT.CustomerName AS CustomerName
FROM
  PROSPECT
WHERE
  PROSPECT.CITY = 'London'
```

In order for the people who are both customer and prospect, whose last name starts with the letter 'A' and who live in London to appear several times in the query result, the SQL code must be:


```
SELECT
  CUSTOMER.CustomerName AS CustomerName
FROM
  CUSTOMER
WHERE
  CUSTOMER.CustomerName LIKE 'A%'

UNION ALL

SELECT
  PROSPECT.CustomerName AS CustomerName
FROM
  PROSPECT
WHERE
  PROSPECT.CITY = 'London'
```

2.6.2 Creating a union query via the description window

To create a new union query:

1. Click  among the quick access buttons.
 - In the window that is displayed, click "Query".
 - The wizard for query creation starts.
2. Choose "Union" and validate.
3. If your project is linked to no analysis, specify the analysis to which the query will be attached.
4. The description window of the query is displayed.
5. Enter the name and caption of the query.
6. Select the different queries to combine (left box).
7. Choose to keep or delete the records common to the combined queries ("Keep the duplicates/Delete the duplicates" button).
8. Sort the result of the query if necessary ("Sort" button).
9. Validate the description window of query. The graphic representation of query is automatically displayed in the query editor.

Notes

- To modify the characteristics of a query, on the "Query" pane, in the "Edit" group, click "Description". see "Description of a query", page 189 for more details.
- To generate the SQL code corresponding to this query, select "SQL code" from the popup menu of the query. If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.

2.7 Query in SQL code

2.7.1 Overview


The query editor also allows you to create a query by typing SQL code (if you are familiar with this language).

Important

Reports & Queries cannot be used to create queries in SQL code using the INSERT, UPDATE or DELETE commands.

2.7.2 Creating a query in SQL code

To create a query in SQL code:

1. Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Query".
 - The wizard for query creation starts.
2. Specify that you want to create a query by entering the SQL code directly ("SQL code" option).
3. If your project is linked to no analysis, specify the analysis to which the query will be attached. Validate the wizard screen.
4. Type the query name (name of ".WDR" file corresponding to the query). This name will be used to identify the query in your programs.
5. Specify the caption of the query by briefly describing the purpose of query. This caption will be used in the project documentation.
Note: The query name is automatically defined from the caption typed.
6. Validate the query creation. A code window is displayed.
7. Type the SQL code of your query. You can insert into the query:
 - **comments** by writing the "-" or "/" character in front of them. Only comments about the selection conditions will be taken into account during the reverse engineering of the query: the other comments will be automatically deleted. The comments about the selection conditions with parameters will not be proposed when running a GO of query.
 - **WLanguage functions.** The WLanguage functions must be prefixed by "WL.". Assisted input is available. The WLanguage functions can be nested. Example:

```
SELECT Customer.CustomerNum FROM Customer
WHERE WL.Left (Customer.ZipCode,2)=75
```

8. Save the SQL code ().

2.7.3 Reverse engineering of query in SQL code

Retroanalysis of the query in SQL code allows the query to be graphically represented from the data files found in the SQL code and in the project analysis.

To perform the reverse engineering of a query in SQL code:

1. Display the SQL code of the query.
2. To represent the query graphically, on the "Query" pane, in the "Analyze" group, click "Reverse Engineering".

- Limits:
- This feature is available for the SELECT, UPDATE, DELETE and INSERT queries.
 - The sub-queries are not imported.
 - The queries of queries are not supported.
 - The UNION queries are not supported.
 - This feature is available only if the SQL commands used in the query are recognized by the HFSQL engine.

2.7.4 Runtime modes of a query in SQL code

The query editor allows you to easily run queries in SQL code. These queries can be:

- typed in the query editor. Then, the query is run by **HExecuteQuery**.
- typed in **HExecuteSQLQuery** in the code editor.

Two runtime modes are available for the SQL queries:

- **Running a SQL query while checking the SQL code:** This option is available by default for all the SQL queries run from WINDEV. The SQL code of the query must use the commands listed below. In this case, the HFSQL engine checks the SQL code of query.
- **Running an SQL query without checking the SQL code** (with the *hQueryWithoutCorrection* constant): This option is recommended for the queries run via a native access or OLE DB. In this case, the HFSQL engine does not check the query. This option must be used if the query contains commands specific to a connection type (Oracle, SQL Server, ...). All the SQL commands supported by the accessed database can be used.

Note: The *hQueryWithoutCorrection* constant is also available when the query test is run from the query editor:

- **for a Select Query:** click the "Advanced" button of the description window and check "Run with hQueryWithoutCorrection in test mode" in the "hQueryWithoutCorrection" tab.
- **for an Insert query, an Update query or a Delete query:** check "Run hQueryWithoutCorrection in Test Mode" in the "General" tab in the description window.

2.7.5 Available SQL commands

The "SQL commands" (statements, functions, clauses, ...) that can be used in a SQL query checked by the HFSQL engine are as follows:

ADD_MONTH	Adds months to the specified date.
ALL	<ul style="list-style-type: none">• Used with the SELECT statement Selects all records.• Used with the UNION operator Selects the common records.• Used in the sub-queries.
AND	Combines several selection conditions (both).
ANY	Used in the sub-queries.
AS	Creates an alias for each item and for each file used.
ASC	Used with the ORDER BY statement Defines the ascending order.
ASCII	Returns the ASCII code: <ul style="list-style-type: none">• of a character.• of the first character in a string.

ATAN2	Returns the arctangent of X and Y variables.
AVG	Calculates the mean for a set of values.
BEGIN	Starts a transaction.
BEGIN WORK	Starts a transaction.
BEGIN TRANSACTION	Starts a transaction.
BETWEEN	Selects the records for which the value of an item belongs to a range of values.
BIN	Returns the character string representing "integer" in binary.
BOTTOM	Returns the last n records found in the result of a query.
BTRIM	Deletes a string found in another string from the beginning or from the end.
CASE	Returns the value corresponding to the condition.
CAST	Used to convert a number into another type.
CBRT	Returns the cube root of a number.
CEILING CEIL	Returns the rounded value of a number.
CHAR_LENGTH	Returns the size (number of characters) of an expression.
CHARACTER_LENGTH	Returns the size (number of characters) of an expression.
COALESCE	Returns the first not-null expression among its arguments.
COMMIT	Validates a transaction.
COMMIT TRANSACTION	Validates a transaction.
COMMIT WORK	Validates a transaction.
CONCAT	Concatenates several strings together.
CONVERT	Converts a character string from a character set to another one.
COUNT	Returns the number of records found in a file or in a group of records.
CREATE TABLE	Describes and creates a file/table.
CURRENT_TIMESTAMP	Returns the date and the local time of server.
CURRENT_USER	Returns the user name for the current connection.
DAY	Returns the day of the month, which means a number included between 1 and 31.
DAYOFMONTH	Returns the day in the month (included between 1 and 31).
DAYOFWEEK	Returns the day in the week (1 for Sunday, 2 for Monday, etc).
DAYOFYEAR	Returns the day in the year (included between 1 and 366).
DECODE	Returns the value corresponding to the condition.
DEGREES	Converts an angle expressed in radians into degrees.
DELETE	Deletes records from a file.
DESC	Used with the ORDER BY statement Defines the descending order.
DISTINCT	Used with the SELECT statement Deletes the duplicated records (duplicates) from the result of the query.

DIV	Performs an entire division.
DROP TABLE	Deletes a file/table physically.
ELT	Returns the nth string of a list of strings.
EVERY	Returns True if all the arguments are checked and true.
EXISTS	Used in the sub-queries.
FIELD	Returns the index of the string that must be found in the list.
FIND_IN_SET	Returns the position of a string in a list of values.
FROM	Used with the SELECT statement Defines the files used. Used with the DELETE statement Defines the files used.
FULL OUTER JOIN	Performs an outer join.
GETDATE	Returns the date and the local time of server.
GETUTCDATE	Returns the date and the UTC time of server.
GREATEST	With two arguments or more, returns the greatest value.
GROUP BY	Classifies a selection of records by group.
HAVING	Used to specify one or more conditions on groups of records generated by the GROUP BY statement.
HEX	Returns the hexadecimal value of an integer (expressed in base 10).
IN	Selects the records whose values correspond to a list of values. Used in the sub-queries.
INITCAP	Returns a string containing the first letter of each word in uppercase characters.
INNER JOIN	Performs an inner join.
INSERT	Adds a record into a data file.
INSTR	Returns the position of a character string.
INTO	Used with the INSERT statement Defines the file used.
IS NULL / IF NULL	Selects the records for which the value of an item is null.
LAST_DAY	Calculates the date of last day for the specified month.
LAST_INSERT_ID	Returns the last automatic identifier that was calculated.
LCASE	Returns the str string with all the character in lowercase, according to the current set of characters.
LEAST	With two arguments or more, returns the lowest value.
LEFT	Extracts the first characters from the content of an expression.
LEFT OUTER JOIN	Performs an outer join.
LEN	Returns the size (number of characters) of an expression.
LENGTH	Returns the size (number of characters) of an expression.


LIKE	Selects the records for which the value of an item corresponds to a specified value (with wildcard characters).
LIMIT	Returns part of the query result.
LOWER	Converts the content of an expression into lowercase characters.
LPAD	Returns a string of a given size. To reach the requested size, the string is completed to the left: <ul style="list-style-type: none">by space characters.by a character or by a given string.
LTRIM	Returns a character string: <ul style="list-style-type: none">without the space characters on the left.without a list of characters.
MAX	Returns the greatest value of an item for all the records selected in the file.
MD5	Calculates the MD5 check sum of the string.
MID	Extracts a sub-string from a given position.
MIN	Returns the lowest value of an item for all the records selected in the file.
MONTHS_BETWEEN	Returns the number of months between two dates.
NEW_TIME	Returns the date after time zone conversion.
NEXT_DAY	Returns the first day of the week following the specified date or day.
NOT	Reverses the meaning of the logical operator used. Used in the sub-queries.
NVL	Replaces the null values of the column by a substitution value.
OCT	Returns the octal value of an integer (expressed in base 10).
OCTET_LENGTH	Returns the size (number of characters) of an expression.
ON	Used when creating a join.
OR	Combines several selection conditions (one or the other).
ORDER BY	Sorts the selected records.
OVERLAY	Replaces a string in another one.
PATINDEX	Returns the position of the first occurrence of a character string.
POSITION	Returns the position of a character string in an expression.
RADIANS	Converts an angle expressed in degrees into radians.
RANDOM/RAND	Returns a random number included between 0.0 and 1.0 (inclusive).
REPEAT	Returns a character string containing the repetition of count times the str string.
REPLACE	Returns a character string: <ul style="list-style-type: none">by replacing all the occurrences of a word found in a string by another word.by replacing all the occurrences of a word found in a string.
REVERSE	Returns a string for which the order of characters is the reversed order of the str string.
RIGHT	Extracts the last characters from the content of an expression.

RIGHT OUTER JOIN	Performs an outer join.
ROLLBACK	Cancels a transaction.
ROLLBACK TRANSACTION	Cancels a transaction.
ROLLBACK WORK	Cancels a transaction.
ROUND	Returns the date rounded to the specified format.
RPAD	Returns a string of a given size. To reach the requested size, the string is completed to the right: <ul style="list-style-type: none">• by space characters.• by a character or by a given string.
RTRIM	Returns a character string: <ul style="list-style-type: none">• without space characters on the right.• without a list of characters.
SELECT	Find records in one or more files.
SET	Used with the UPDATE statement Defines the name of the items used.
SET ISOLATION	Defines the isolation level of transactions.
SHA/SHA1	Calculates the 1-bit SHA160 check sum of the string, as described in the RFC 3174 (Secure Hash Algorithm).
SOME	Used in the sub-queries.
SOUNDEX, SOUNDEX LIKE	Returns the phonetic value.
SOUNDEX2, SOUNDEX2 LIKE	Returns the phonetic value (adapted to French).
SPACE	Returns a string containing N spaces.
SPLIT_PART	Divides the string according to a separator and returns the nth part.
START TRANSACTION	Starts a transaction.
STDDEV	Used to find out the standard deviation of a series of values representing a sample of data.
STDDEV_POP	Used to find out the standard deviation of a series of values representing the full set of data.
STDDEV_SAMP	Used to find out the standard deviation of a series of values representing a sample of data.
STRING_AGG	Used to concatenate non-null strings from a list of values.
SUBSTR	Extracts a sub-string from a given position.
SUBSTRING	Extracts a sub-string from a given position.
SUM	Returns the total sum of all the item values for all the records selected in the file.
SYSDATE	Returns the date and the current time.
SYSTEM_USER	Returns the user name for the current connection.
TO_CHAR	Formats a date or a number.

TOP	Returns the first n records found in the result of a query.
TRANSLATE	Returns a character string with all the specified characters replaced by other characters.
TRIM	Returns a character string: <ul style="list-style-type: none">• without space characters on the left and on the right.• without a character string found at the beginning and at the end of string.• without a character string found at the beginning of string.• without a character string found at the end of string.
TRUNC	Returns the date truncated to the specified format.
UCASE	Returns the str string in uppercase characters, according to the current set of characters.
UNHEX	Contrary of HEX. Each pair of hexadecimal digits is interpreted as numbers and converted into a character represented by the number.
UNICODE	Returns the integer value defined by the Unicode standard of the first character in the specified expression.
UNION	Performs union queries.
UPDATE	Updates the file records.
UPPER	Converts the content of an expression into uppercase characters.
UUID	Generates a UUID (Unique Universal Identifier). A UUID is an integer on 128 bits represented in hexadecimal as a character string in the following format: aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee. A UUID is guaranteed as being unique for each call to the function.
USER_NAME	Returns the user name for the current connection.
VALUES	Used with the INSERT statement Defines the added values.
VARIANCE	Returns the variance for a series of values (data sample).
VAR_POP	Returns the variance for a series of values (full data set).
VAR_SAMP	Returns the variance for a series of values (data sample).
WHERE	Used to specify the selection conditions of a query.
XOR	Logical XOR (exclusive OR).

2.8 Creation mode of queries

The query editor proposes two methods for creating a query:

- creating the query in the query editor:
 - Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Queries" and select the type of query to create.
This query corresponds to a project element (".WDR" file) and it can be used as data source for a report, a window control or a page control (List Box, Combo Box or Table control), by programming, in another query, ... The **query is independent**.
- creating the query when creating the ListView or Table control or when creating the report based on this query ("A query that I want to define now" option in the wizard for creating the control or the report). No specific file is created. The **query is embedded**.

An embedded query cannot be used as data source for an element (report, ...) other than the one where this query was created.

Control in a window or in a page: Switching from the embedded mode to the independent mode

To extract a query from a window control or from a page control:

1. Display the window or the page in the editor.
2. Select the control based on an embedded query.
3. Select "Embedded query .. Open the associated query" from the popup menu of the control.
Caution: the window or the page must have been saved at least once.
4. The query editor is opened and the query graph is displayed. On the "Home" pane, in the "General" group, expand "Save" and select "Export .. To an independent query".
5. Save the new query. The query now is an independent query. It can be used in all the windows, pages and reports of your project.

Report: Switching from the embedded mode to the independent mode

To extract a query from a report:

1. Display the report on embedded query (in the report editor).
2. Select "Edit the query" from the popup menu of report. The query is displayed in the query editor.
Caution: the report must be saved at least once.
3. On the "Home" pane, in the "General" group, expand "Save" and select "Export .. To an independent query".
4. Save the new query. The query now is an independent query.

3.DESCRPTION OF A QUERY

3.1 Overview

Once a query is created, all the characteristics of this query can be modified at any time:

- files and/or items used,
- items displayed in the query result,
- selection conditions,
- sort of records,
- ...

Only the query type cannot be modified. Indeed, too many specific parameters are taken into account. To change the query type, all you have to do is create a new query.

This chapter presents:

- the description window of a query.
- the characteristics of a query.
- how to open, close and save a query.
- the management of query versions.

3.2 Description window

The main characteristics of a query can be accessed in the description window of the query. The query editor proposes three description windows:

- description window of a select query.
- description window of an insert query, update query, delete query or in SQL code.
- description window of a union query.

3.2.1 Displaying the description window

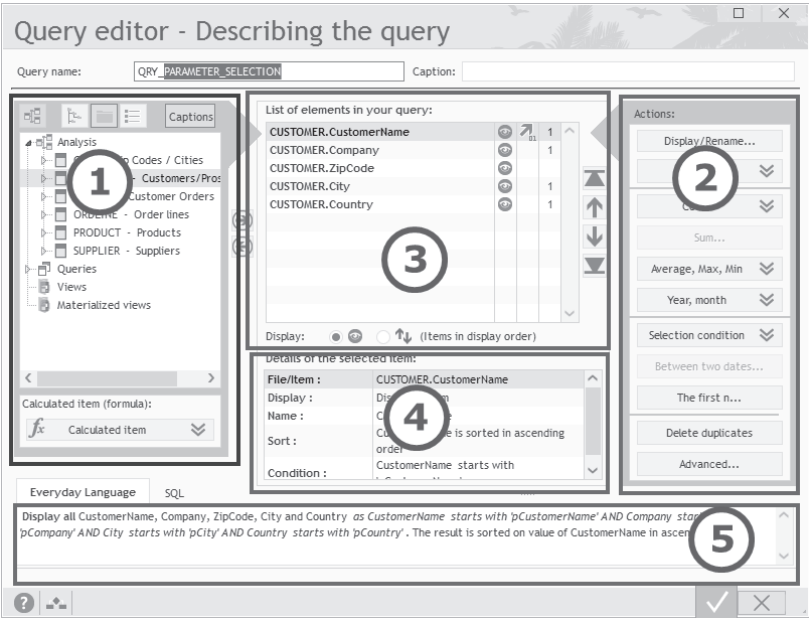
To display the description window of a query:

- double-click the graphic representation of the query.
- select "Query description" from the popup menu of the query editor.
- on the "Query" pane, in the "Edit" group, click "Description".

3.2.2 Description window of select query

The description window of a select query includes several sections. This description window is used to:

- modify the items used in the query.
- manage the selection conditions.
- view the query in SQL code or in natural language.
- manage the "duplicate" records.
- ...

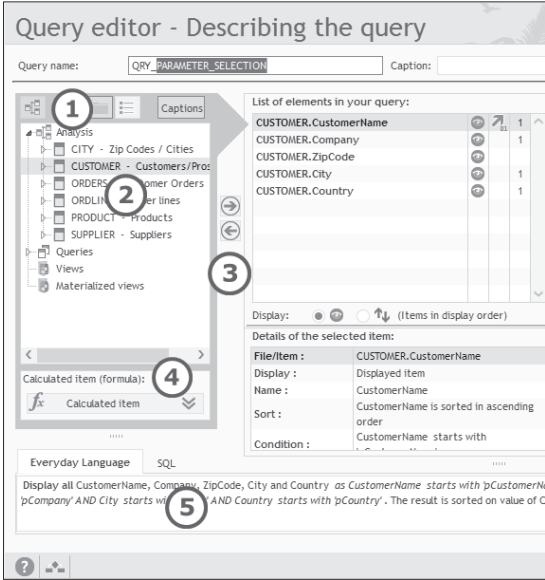


In this window:

- 1
- List of items that can be used in the query.
- 2
- Actions that will be applied to each query item.
- 3
- List of items taking part in the query.
- 4
- Description of the selected item.
- 5
- Query displayed in SQL code or in natural language.

This description window includes three sections:

- Left section: This section contains the following elements:



- 1
- List of items that can be used in the query. These items belong to:
- a file of the analysis associated with the current application.
 - a query of current application.
- Note:
- To select several items, keep the Ctrl key down.

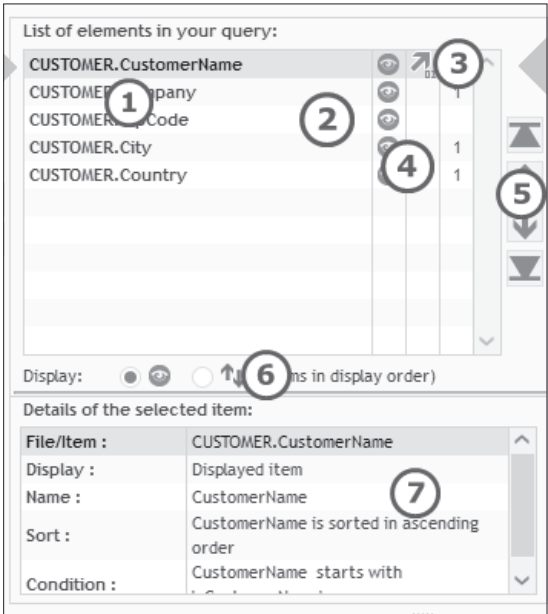
R&Q






Important

The data files and the file items that can be used in Reports & Queries are defined in the data model editor of WINDEV. See "Distributing Reports & Queries", page 233 for more details.

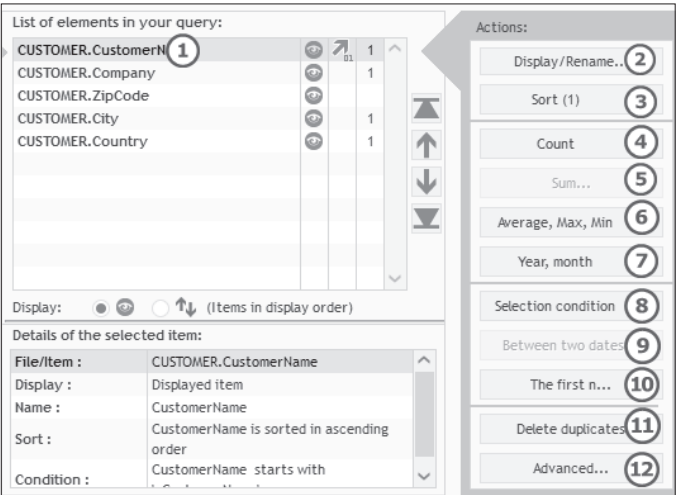
- 2
- Used to add or delete the selected items to/from the query.
- 3
- The items can be displayed:
- by associated file or by associated query.
 - in alphabetical order.
- 4
- Adds a new item to the query result. This item corresponds to a calculation performed on some analysis items.
- For example, the "PriceIOT" item corresponds to the "PriceBT" item multiplied by the "VATRate" item.
- 5
- Displays:
- the query in natural language.
 - the SQL code corresponding to the query.

- Middle section: This section contains the following elements:



- 1
- List of items used in the query.
- 2
- The values of this item will be displayed in the result of the query ().
- 3
- Sort direction for the values of this item ( for ascending,  for descending).
- 4
- Number of selection conditions applied to this item (2 in this example).
- 5
- Modifies the display order and/or the sort order of items in the query result.
- 6
- Modifies the display order in the list of query elements:
-  : corresponds to the display order of elements in the query result.
 -  : corresponds to the sequence of sorts in the query result.
- 7
- Details of options defined for the selected item.

- Right section: This section is used to define the different actions that will be applied to the query items.



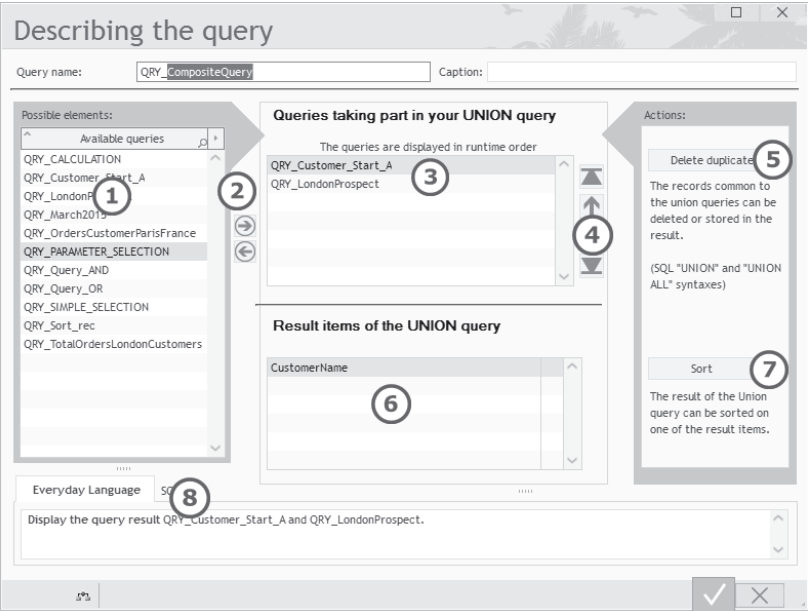
- 1
- Item onto which the actions will be applied.
- 2
- Allows you to:
- display (or not) the values of selected item in the query result.
 - rename the selected item in the query result (create an alias).
- 3
- Sorts the values of selected item in the query result.
- 4
- Adds a new item to the query result. This item corresponds to:
- the number of records found in a group of records.
 - the number of records found in a data file.
- 5
- Adds a new item to the query result. This item corresponds to the sum of a group of records.
- 6
- Adds a new item to the query result. This item corresponds to:
- the mean for a group of records.
 - the minimum value for a group of records.
 - the maximum value for a group of records.
- 7
- Modifies the values displayed in the query result. You have the ability to display:
- the year from a date item.
 - the month from a date item.
 - the state from an item containing a zip code.
- 8
- Defines the selection conditions. It is possible to:
- apply a selection condition to the selected item.
 - modify and/or delete an existing condition.
 - modify the sequence of existing conditions.

- 9
- Selects the records corresponding to a given period.
- 10
- Selects the first n records or the last n records displayed in the query result.
- 12
- Deletes (or not) the "duplicate" records from the query result.
- Allows you to:
- specify the type of database used (HFSQL, Oracle or other).
 - manage the joins (add, modify, delete, re-initialize).
 - check (or not) the SQL syntax used (in test mode only).
 - specify the runtime option (optimizing the memory used, checking the file format, ...).
 - specify whether the query is visible in "Reports & Queries".

3.2.3 Description window of union query

The description window of a union query includes several sections. This description window is used to:

- change the queries used in the query.
- modify the runtime order of queries.
Reminder: The items displayed in the result of union query correspond to the items of the first query run.
- view the query in SQL code or in natural language.
- manage the "duplicate" records.
- sort the records.



- 1
- List of select queries that can be used in the union query.
Note: To select several queries, keep the Ctrl key down.
- 2
- Used to add or delete the selected queries to/from the union query.
- 3
- List of queries used in the union query. These queries are displayed in their runtime order.
- 4
- Modifies the runtime order of queries.
- 5
- Used to keep or delete the duplicated records (duplicates) from the result of the union query.
- 6
- List of result items for the union query.
Reminder: These items correspond to the items from the first query run.
- 7
- Used to perform or cancel the sort on a query item.
- 8
- Displays:
 - the union query in natural language.
 - the SQL code corresponding to the union query.

3.2.4 Description window of query

The description window of a query (insert, update or delete) or the description window of a query in SQL code is used to:

- find out and/or modify the name and caption of the query.
- specify whether the SQL commands used must be checked (or not) by the HFSQL engine ("Run with hQueryWithoutCorrection in test mode").
- specify the connection used during the test of the query.
- find out and/or modify the values to insert or to modify in the query result (insert query or update query).
- find out and/or modify the selection conditions (update query or delete query).
- find out and/or modify the sequence of selection conditions (update query or delete query).
- ...



Important

Reports & Queries cannot be used to create insert queries, update queries or delete queries.

3.3 Characteristics of a query

3.3.1 Items used in a select query

A query created in the query editor can contain several types of items:

- **items of the analysis** to which the query is attached.
These items correspond to the items of the data files described in the analysis. You have the ability to apply calculations to these items.
Note: You have the ability to use a full-text index item: in this case, the query will perform a full-text search.
- **items of a query** found in the current project. These items correspond to the items used in the queries included in the application.
- **calculation items.** This type of item is used to perform a calculation on the values of an item: sum, average, maximum and minimum.
- **count items.** This type of item is used to count the number of records found in the query or the number of records found in a grouping of records.
- **"calculated" items.** This type of item is used to add to the query result an item whose value corresponds to a calculation formula on the items. For example, PriceBT * VATRate.

The items taking part in a query are selected when creating the query. However, you have the ability to add, modify or delete the items used in a query. These operations are described in the following paragraphs.

Note: The items whose value is displayed in the result of the query are called *result items*.

To add an item into the current query:

1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
2. In the left section, double-click the item to add. The added item is automatically displayed in the middle section.
3. Validate.

Note: An item can also be added into the current query from one or more items described in the data model editor: to do so, Drag and Drop the selected items from the "Analysis" pane to the query editor.

To perform a full-text search:

1. Select "Query description" from the popup menu of the query. The description window of query is displayed.
2. In the left section of the window, double-click the "FullText_Index" item to add. The description window of the search is displayed.
3. Specify the parameters of the "full-text" search to perform:
 - the index is taken into account.
 - the search value (value or parameter). If the value is described in the query, a wizard allows you to enter all the necessary conditions on the search value.
 - the sort of query result according to the relevance search subscript for each record.
Note: only the records whose relevance subscript is greater than 0 are stored in the result of the query.

To add a count item into the current query:

1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
2. In the right section, expand the "Count" button.
3. To count:
 - the number of values for the selected item in the query result: select "On the selected item".
For example, this option can be used to find out the number of customer who have a cell phone ("CellPhone" item filled).
 - the number of values for another item in the query result: select "On a new item" and the corresponding item.
 - the number of records found in a file or in a query (according to the selection conditions of the current query): select "On a file or query".
For example, this option can be used to find out the number of products found in the "Product" file.

Note: Regardless of the count operation performed, you can specify whether the duplicates must be taken into account (or not) by the count operation (Count(Distinct) in SQL.

To add a "calculated" item into the current query:

1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
2. In the left section, expand the "Calculated item" button and select "New calculated item". The description window of a calculated item is displayed. This window allows you to enter the SQL code corresponding to the calculation to perform. In this window, you can:
 - handle the items found in the files of the current analysis and the items found in the queries of the current project.
 - enter SQL statements. The list of available SQL statements is displayed. All you have to do is double-click a statement in order to display it in the code section.
 - enter WLanguage functions. The list of available WLanguage functions is displayed. All you have to do is double-click a function in order to display it in the code section. To use a WLanguage function, its name must be preceded by "WL".
 - nest the WLanguage functions and the SQL code used.
3. Specify the mathematical formula of the item and validate. The item is automatically added.

Note: Using parameters in the code of the calculated item.

To use parameters in the code of the item, the name of the parameter must be enclosed in braces. For example:

```
LEFT(MyItem, {ParameterName})
```

Caution: The result may differ in test mode and at run time. Indeed, in test mode, the value of the parameter is always passed in string format.

To modify a "calculated item" in the current query:

1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
2. In the middle section, select the calculated item to modify.
3. In the left section, click the "Calculated item" button and select "Modify the calculated item". The description window of a calculated item is displayed.
4. Modify the formula and validate.

Note: Using parameters in the code of the calculated item.
To use parameters in the code of the item, the name of the parameter must be enclosed in braces. For example:

```
LEFT(MyItem, {ParameterName})
```

Caution: The result may differ in test mode and at run time. Indeed, in test mode, the value of the parameter is always passed in string format.

To rename a result item in the current query (alias creation):

- 1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
- 2. In the middle section, select the item to rename.
- 3. In the right section, click the "Display/Rename" button.
- 4. Enter the new name of the item and validate. The item is automatically renamed.

To display/hide an item in the result of the query:

- 1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
- 2. In the middle section, select the item to display/hide.
- 3. In the right section, click the "Display/Rename" button.
- 4. Answer "No" to the question "Display this item?".

Note: To display/hide an item in the result of the query, you can also:

- display the popup menu of the requested item (right mouse click).
- select "Display the item" or "Don't display".

To delete an item from the current query:

- 1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
- 2. In the middle section, select the item to delete and press the Del key. The item is automatically deleted from the query.

Notes:


- Deleting an item also deletes the selection conditions defined for this item.
- To delete an item, you also have the ability to select "Delete the item" from the popup menu of the item.

3.3.2 Sorting the records selected by a Select query

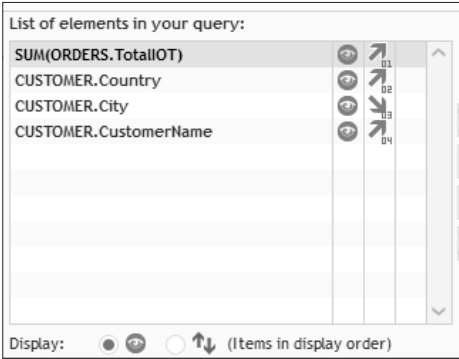
To make the result of a query easier to read, the records can be sorted according to one or more items. Sorting the records means classifying them in an ascending order or in descending order.
For example, the customers can be sorted by name, from A to Z.

Note: Only the items displayed in the query result can be sorted.
The sort order is defined when creating the query. However, you can:

- add or delete a sort item.
- modify the sort order performed on an item.

When several sort items are defined, the sorts are nested. To find out the sequence of the different items, click  found in the middle section. The sort starts from the first item found in the table of sorted items.
For example, in the window below, the query result is sorted:

- on the total amount of orders first,
- then on the country of customers,
- then on the city of customers,
- finally on the name of customers.



The arrows found to the right of the window are used to modify the sequence of sort items.

Note: The numbers found beside the arrows also allow you to find out the sequence of the different sort items.

To add a sort item into the current select query:

- 1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
- 2. In the middle section, select the requested item.
- 3. Display the popup menu of the item, select "Sort the item" and select the sort order.

Notes:

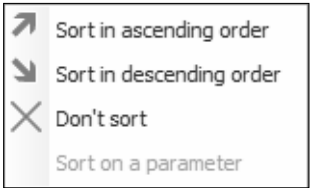
- To add a sort item, you also have the ability to click the "Sort" button found in the right section and to select "Sort on the selected item".
- The query editor can also be used to perform a sort on a parameter. All you have to do is click the "Sort" button and to select "Sort on a parameter". In this case, you must specify the name of the parameter that will be used for the sort as well as the default sort order. This sort order can be modified at run time if the content of the parameter contains the sort order. For example: "Customer.CustomerName ASC". This parameter must be passed to the query when it is run. See "Two methods for passing parameters to a query", page 222 for more details.

To delete a sort item from the current select query:

- 1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
- 2. In the middle section, select the requested item.
- 3. Display the popup menu of this item (right mouse click).
- 4. Select "Sort the item .. Don't sort".

- To modify the sort order of an item in the current select query:
1. Display the description window of query: on the "Query" pane, in the "Edit" group, click "Description".
 2. In the middle section, select the requested item.
 3. Display the popup menu of this item (right mouse click).
 4. Select "Sort the item" and select the new sort direction for the item.

Tip: To add, modify or delete a sort, click the column used to sort the query items and select the operation to perform in the following menu:



3.3.3 Operations that can be performed in a select query

The query editor allows you to perform operations on the values of an item. The possible operations are:

- the sum of item values.
- the mean of item values.
- the calculation of the minimum value for all the item values.
- the calculation of the maximum value for all the item values.

These operations are performed:

- on **all the records** selected by the query.
- on **sub-sets of records** (automatically grouped by the query).

For example, you want to find out:

- the total amount of orders placed by the customers living in Seattle.
- the order with the smallest amount.
- the average sum spent by the customer named Darwin.
- ...

The operations are defined when creating the query. However, you have the ability to add, modify or delete the operations performed in a query.

Caution: If the query result is displayed in a report, the operations to perform on the item values must be defined during report description.

To add an operation into the current select query:

1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
2. In the middle section, select the item on which an operation must be performed.

3. To calculate:
 - the sum of the values in the item, click on "Sum...".
 - the average of the values in the item, click on "Average, Max, Min" and select "Average".
 - the maximum value of the values in the item, click on "Average, Max, Min" and select "Maximum".
 - the minimum value of the values in the item, click on "Average, Max, Min" and select "Minimum".
4. An item corresponding to the requested calculation is automatically added to the query items.
5. Validate. The operation is automatically added into the query.

To delete an operation from the current select query:

1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
2. In the middle section, select the calculation item to delete.
3. Press the Del key.

3.3.4 Modifying the values displayed in the result of a select query

In the result of a select query, you have the ability to only display:

- the year from a date item.
- the month from a date item.

To modify the values displayed in the result of the current Select query:

1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
2. In the description window, select a date item or an item containing a zip code.
3. Click the "Year, month" button and select the option:
 - "Month" to display the month in the query result.
 - "Year" to display the year in the query result.
4. Validate.

3.3.5 Selection conditions

The selection conditions of records are applied to the items used in the queries. These conditions are used to select the records that match one or more specific criteria.

For example:

- customers whose name starts with "C".
- products whose price exceeds 500 Euros.
- orders placed between "01/01/2000" and "06/01/2000".
- ...


Several selection conditions can be defined for a single item.

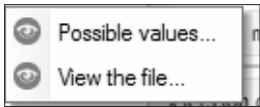
The selection conditions are defined when creating the query. However, you have the ability to add, modify or delete the selection conditions used in a query. These operations are described in the following paragraphs.

When at least two selection conditions are found in a query, the sequence of the selection conditions must be specified. See "Sequence of selection conditions", page 204 for more details.

How to add a selection condition into a query?

To add a selection condition into a select query:

- 1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
 - 2. In the middle section, select the item onto which the selection condition must be added.
 - 3. Click the "Selection condition" button and select "New condition". The wizard for creating a selection condition is displayed.
 - 4. Choose the selection criteria (equal to, greater than, ...) of the condition. This condition can be applied to a given value, a parameter, the records selected by another query, ... For a date item, the selection condition can be applied to a preset date (the current year, the current month, today).
- Note:** You have the ability to select a given value via . The following menu is displayed:



You can:

- select one of the possible values among the values found in the file
 - view the entire file.
- Caution:** This option is available only if Live Data is enabled.

- 5. Indicate (if necessary) a comment on the condition. This comment will be displayed in the SQL code of query (preceded by the "-" or "/" character). If the condition is applied to a parameter, during a GO of the query with parameters, the comment will be displayed in the window for entering parameters.
- 6. Validate. The selection condition is automatically added to the query.

To add a selection condition into an update query or into a delete query:

- 1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
- 2. Display the "Query" tab.
- 3. Select the item onto which a selection condition must be added and click the "Condition" column corresponding to the selected item. The wizard for creating a selection condition is displayed.
- 4. Select "I want to add a condition" (for an item onto which at least one condition is performed).
- 5. Choose the selection criteria (equal to, greater than, ...) of the condition. This condition can be applied to a given value, a parameter, a sub-query, ...
- 6. Indicate (if necessary) a comment on the condition. This comment will be displayed in the SQL code of query (preceded by the "-" or "/" character). If the condition is applied to a parameter, during a GO of the query with parameters, the comment will be displayed in the window for entering parameters.
- 7. Validate. The selection condition is automatically added to the query.

How to modify a selection condition in a query?

To modify a selection condition in a select query:

- 1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
- 2. In the middle section, select the item on which a condition must be modified.
- 3. Click the "Selection condition" button and select "Modify the condition".
- 4. Select (if necessary) the selection condition to modify (for an item on which several conditions are performed).
- 5. Modify the selection condition.
- 6. Validate. The selection condition is automatically modified.

To modify a selection condition in an update query or in a delete query:

- 1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
- 2. Display the "Query" tab.
- 3. Select the item on which a condition must be modified and click the corresponding "Condition" column. The wizard for selection conditions is displayed.
- 4. Select "Modify an existing condition".
- 5. Select (if necessary) the selection condition to modify (for an item on which several conditions are performed).
- 6. Modify the selection condition.
- 7. Validate. The selection condition is automatically modified.

How to delete a selection condition from a query?

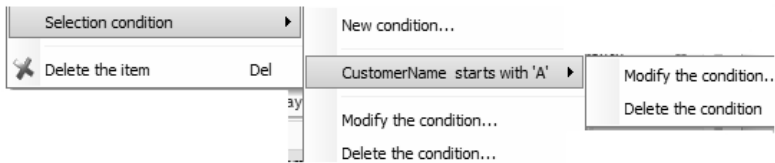
To delete a selection condition from a select query:

- 1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
- 2. In the middle section, select the item from which a condition must be deleted.
- 3. Click the "Selection condition" button and select "Delete a condition".
- 4. Select (if necessary) the selection condition to delete (for an item on which several conditions are performed).
- 5. Validate. The selection condition is automatically deleted from the query.

To delete a selection condition from an update query or from a delete query:

- 1. On the "Query" pane, in the "Edit" group, click "Description". The description window of query is displayed.
- 2. Display the "Query" tab.
- 3. Select the item for which a condition must be deleted and click the corresponding "Condition" column. The wizard for selection conditions is displayed.
- 4. Select "Delete an existing condition".
- 5. Select (if necessary) the selection condition to delete (for an item on which several conditions are performed).
- 6. Validate. The selection condition is automatically deleted from the query.

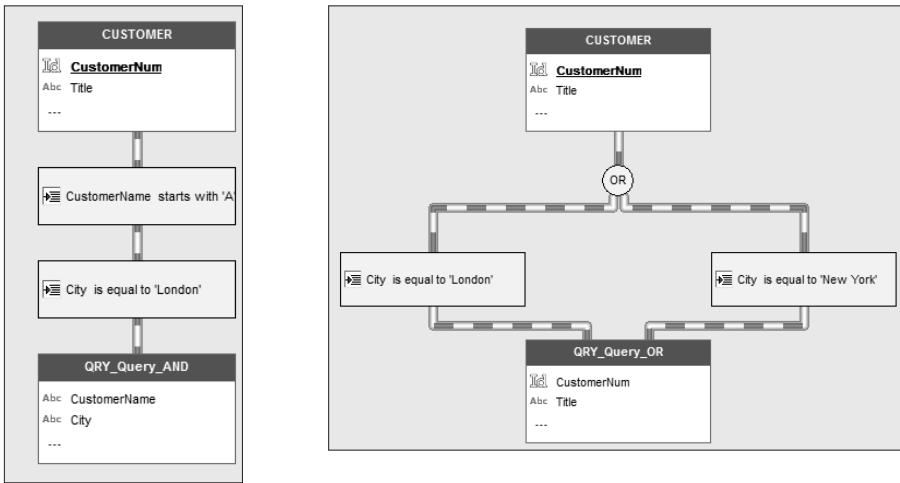
Tip: To add, modify or delete a selection condition in a select query, click the condition column of the query items and select the operation to perform in the following menu:



3.3.6 Sequence of selection conditions

When at least two selection conditions are found in a query, the sequence of selection conditions must be specified by the conditional operators AND and OR:

- **Conditional AND operator:** the selected records will match both conditions. For example, select the customers whose name starts with the letter "A" AND who live in London.
- **Conditional OR operator:** the selected records will match one of the two conditions or both conditions. For example, select the customers who live in London OR in New York.



The sequence of selection conditions is defined when creating the query. However, this sequence can be modified.

Two methods can be used to modify the sequence of selection conditions in the current query:

- **1st method:** via the description window of query.
- **2nd method:** from the graphic representation of query.

1st method: Via the description window of query

To modify the sequence of selection conditions in the current select query via the description window of the query:

1. Display the popup menu of the query (right mouse click) and select "Query description". The description window of query is displayed.
2. Click the "Selection condition" button and select "Existing conditions".
3. Modify the sequence of selection conditions.
4. Validate. The modification is automatically performed in the query.

To modify the sequence of selection conditions in the current update or delete query via the description window of the query:

1. Display the popup menu of the query (right mouse click) and select "Query description". The description window of query is displayed.
2. Select the "And/Or" tab.
3. Modify the sequence of selection conditions.
4. Validate. The modification is automatically performed in the query.

2nd method: From the graphic representation of the query

To modify the sequence of selection conditions in the current query via the graphic representation of the query:

1. Select the selection condition whose sequence must be modified in the graphic representation of the query.
2. Select "Reverse AND/OR" from the popup menu (right mouse click) of the current selection condition.
3. The new sequence of selection conditions is automatically applied to the query.

3.3.7 Displaying the records corresponding to a given period

When a "Date" item is found in the query, you have the ability to only select the records corresponding to a given period.

For example:

- the customers born between "01/01/1950" and "01/01/1960".
- the orders placed last month.
- the products sold yesterday.
- ...

To display the records corresponding to a given period:

1. Select "Query description" from the popup menu of the query. The description window of query is displayed.
2. In the middle section, select a "Date" item.
3. In the right section, click the "Between two dates" button.
4. Specify the start date and the end date, then validate.

Note: For the "DateTime" items, when the start and end dates correspond to values, the query editor proposes preset periods (current month, previous week, calendar year, ...). To use one of these preset periods, click the "Preset period" button and select the requested period.

3.3.8 Limiting the number of records displayed in a Select query

The editor of select queries is used to limit the number of records that will be displayed in the query result. It allows you to:

- select the first n records found in the query result.
- select the last n records found in the query result.
- select the n records from a given record found in the query result.

The number of records to select can be defined:

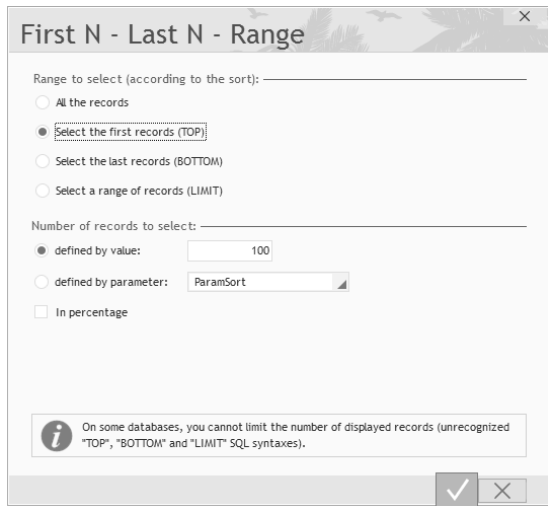
- by value.
- in percentage.
- by a parameter.

For example, you want to find out:

- the ten best customers: the customers with the greatest turnover.
- the 15 worst selling products.
- the 20 products found after the record XXX.
- ...

To limit the number of records found in the result of a query:

1. Display the description window of a Select query: display the popup menu of the query (right mouse click) and select "Query description".
2. Click the "The first n..." button.
3. In the window that is displayed:



- Specify the range of records to select: the first ones, the last ones or a group of records.
- Specify the number of records that will be taken into account. This number can correspond to:
 - a value.If "In percentage" is checked, the value must correspond to a percentage of all the records.

- a parameter. The parameter will have to be specified when running the query. See "Two methods for passing parameters to a query", page 222 for more details.

4. Validate.

Note: When selecting a range of records, specify:

- the subscript of the first record to select.
- the number of records to select.

3.3.9 Keeping (or not) the duplicates

The query editor allows you to keep (or not) the duplicates in the result of a Select query (corresponding SQL keyword: DISTINCT).

The duplicates are kept by default.

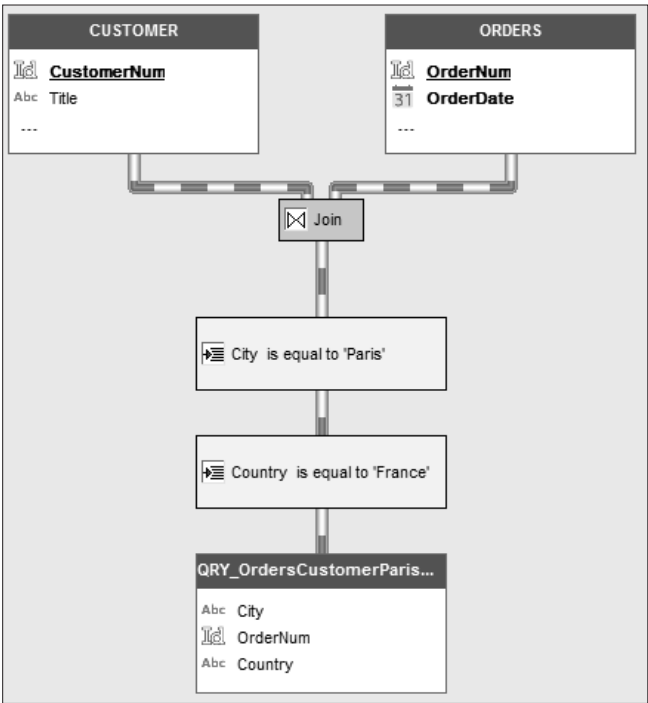
To delete the duplicates, click the "Delete the duplicates" button in the description window of query.

To store the duplicates, click the "Keep the duplicates" button in the description window of query.

3.3.10 Joins between data files

When several files are used in a query, WINDEV or WEBDEV automatically searches for the links described in the analysis between these files. These links are called joins in the query editor.

For example, the following query selects the numbers of the orders placed by the customers who live in Paris. The ORDERS file and the CUSTOMER file are joined by the CustomerNum item. The join between these files was automatically defined by the query editor.



Note: The links between the files appear in the graphic representation of the query only if the two linked files are used by the query.

By default, the joins are automatically created by the query editor. However, these joins may not correspond to the requested result. Therefore, you have the ability to add or delete joins. These operations are described in the following paragraphs.

To add a join into the current select query:

1. Select "Edit the join" from the popup menu of join (right mouse click). The description window of query joins is displayed.
2. Click the "Add a join" button. A window used to define a new join is displayed.
3. Specify the characteristics of join to add.
4. Validate. The join is automatically added into the query.

This allows you to create multiple joins between two files.

To delete a join from the current select query:

1. Select "Edit the join" from the popup menu of join (right mouse click). The description window of query joins is displayed.
2. Select the join to delete.
3. Click the "Delete the join" button. The join is automatically deleted. The joins between files are automatically optimized in order for the query to operate properly.

3.3.11 Consistency with the analysis

Checking the consistency with the analysis is used to update the items handled by the query according to the modifications performed in the analysis.

When generating the analysis, the modifications performed can be directly applied to the relevant elements: windows, queries, pages, ... If this operation is not performed, the consistency with the analysis can be checked in the query editor.

To perform the consistency with the analysis of current query:

1. On the "Query" pane, in the "Analyze" group, click "Consistency with the analysis".
Note: this option is available only if modifications have been made in the analysis and have not been applied to the query.
2. In the window that is displayed, select the action to perform for each modification of the analysis.
3. Validate. The modifications are automatically performed.

3.4 Opening, closing and saving a query

To open a query in the editor:

- double-click its name in the "Project explorer" pane.
- on the "Home" pane, in the "General" group, expand "Open" and select "Open". Select one or more ".WDR" files and validate.

The query is immediately displayed in the query editor.

To close a query:

- click the cross found in the top right corner of the window for editing the query.
- on the "Home" pane, in the "General" group, click "Close". If the query contains unsaved modifications, you have the ability to save the query. To save the query, on the "Home" pane, in the "General" group, click "Save" or expand "Save" and select "Save as" to save and rename the query.

Automatic retrieval

To avoid problems when a system lock occurs (power outage for example), the query editor proposes a system used to automatically retrieve the elements opened in the different editors:

- on the "Home" pane, in the "Environment" group, expand "Options" and select "General options of XXX".
- in the "General" tab, check "Backup for automatic restore".

When this system is enabled:

- an automatic backup of opened elements is performed at specific time intervals (every 10 minutes by default).
- when re-opening a query after a system lock, a window for automatic restore is displayed. This window proposes to retrieve:
 - the query as it was during the last backup performed from the query editor.
 - the query automatically saved by the query editor.
- the selected version is automatically opened in the query editor. The unselected version is available in the window for version management: on the "Home" pane, in the "General" group, expand "Save" and select "Versions". The unselected version can be opened at any time.

3.5 Query documentation


The query editor automatically creates a program documentation for each query. This documentation contains:

- the graphic representation of the query.
- the general information about the query (logical name and physical name of the query, type of query, analysis associated with the query, ...).
- the information specific to the query results (result items, query parameters, sort criteria, ...).
- the information about the query characteristics (selection conditions, groups, ...).
- the SQL code corresponding to the query.


The content of this documentation can be customized, it can contain:

- the full information about the query.
- a selection of information.

To edit the program documentation of current query in the editor:

1. On the "Home" pane, in the "General" group, expand  and select "Print".
2. Specify the information found in this documentation.
3. Choose the type of edit for the documentation.

Note: The print of the query representation can be configured according to:

- print areas.
- a specific layout. To define this layout, expand  and select "Documentation layout".

To specify the print areas of graphic representation:

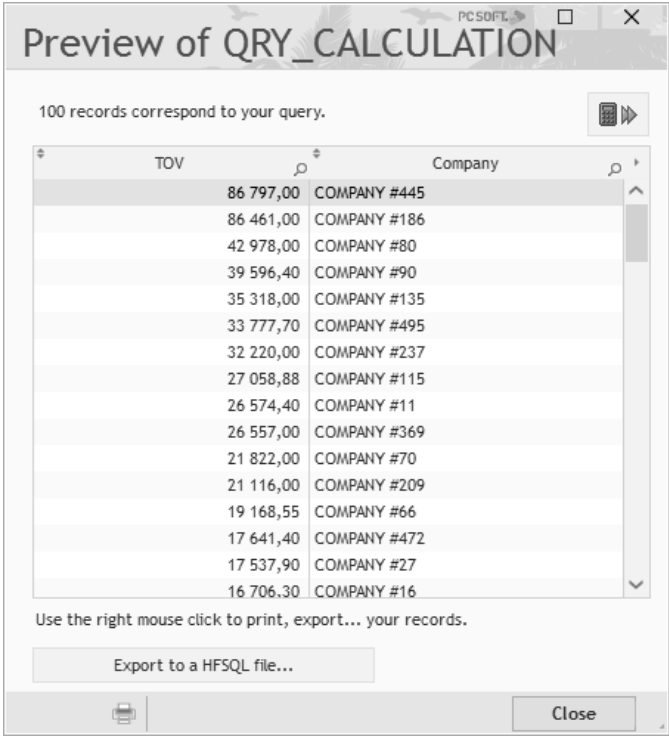
1. On the "Display" pane, in the "Help for edit" group, check "Print areas". Borders representing the areas that can be printed in the documentation appear in the query editor.
2. Reduce the display of graphic representation: on the "Display" pane, in the "Help for edit" group, modify the "Zoom" value (or Ctrl + Mouse wheel) to see the entire query.
3. With the mouse:
 - move these borders to the requested location by keeping the mouse button down (the mouse cursor is black).
 - choose the number of pages on which the graphic representation must be printed (the mouse cursor turns into a North-West/South-East double arrow).

4.RUNNING THE QUERY TEST AND USING IT

4.1 Query test

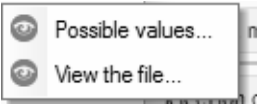
The query editor allows you to run the test of query result (or "Run the query test" from the popup menu of query).

During the test of the query, the initialization code of the project is run.



Notes:


- Several query tests can be run at the same time.
- The last test run (window, query, report, ...) can be re-run from any project element via "GO .. Re-run the last test" found in the quick access buttons.
- During the test of a query with parameters, a window allows you to specify the different parameters. In this window, you have the ability to select an existing value via . The following menu is displayed:



You can:

- select one of the possible values among the values found in the file,
- view the entire file.

Caution: This option is available only if Live Data is enabled. See the online help for more details.

- The  button is used to automatically display the mean, the total, the minimum and maximum values for the numeric result items.

R&Q

Important

Reports & Queries cannot be used to check:

- insert queries,
- update queries,
- delete queries.

During the test of a query in the user version of query editor, the initialization process of the project is run.

The test result depends on the type of query:

- The result of a *select query* corresponds to the records selected by the query.
- The result of an *insert query* corresponds to the record that will be inserted. During the test, the record is directly inserted into the file.
- The result of an *update query* corresponds to the records that will be modified. During the test, you can:
 - see the records that will be modified,
 - modify the records in the file.
- The result of a *delete query* corresponds to the records that will be deleted. During the test, you can:
 - see the records that will be deleted,
 - delete the records from the file.

Result of the test

Then, the query result can be:

- Exported to a text file.
- Exported to an XML file.
- Exported to an Excel file.
- Exported to a HFSQL file.
- Printed.

A record found in the query result can also be copied into the clipboard.

Password

If one of the data files taking part in the query is password protected, during the test of the query:

- If the password is specified in the initialization code of the project (**HPass**), the query is run.
- If the management of HFSQL errors was not customized (**HOnError**) and if the password is not specified by programming, the standard window for password input is displayed.

Note: if the management of errors is customized in the initialization code of project, this management of errors will be used.

4.1.1 Exporting the query result to a Word file

For the next operation, we consider that the query result is displayed.

To export the query result to a text file:

1. Select "Export the table to Word" (or "Export the table to OpenOffice.org Writer") from the popup menu (right mouse click) of the query result. The window for selecting the "RTF" file is displayed.
2. Specify the name and location of the "RTF" file.
3. Validate. The records found in the query result are automatically exported to the "RTF" file.

4.1.2 Exporting the query result to an Excel file

For the next operation, we consider that the query result is displayed.

To export the query result to an Excel file:

1. Select "Export the table to Excel" (or "Export the table to OpenOffice.org Calc") from the popup menu (right mouse click) of the query result. The window for selecting the "XLS" file is displayed.
2. Specify the name and location of the "XLS" file.
3. Validate. The records found in the query result are automatically exported to the "XLS" file.

4.1.3 Exporting the query result to an XML file

For the next operation, we consider that the query result is displayed.

To export the query result to an XML file:

1. Select "Export the table to XML" from the popup menu (right mouse click) of the query result. The window for selecting the XML file is displayed.
2. Specify the name and location of the XML file.
3. Validate. The records found in the query result are automatically exported to the XML file.

4.1.4 Exporting the query result to a HFSQL data file

For the next operation, we consider that the query result is displayed.


To export the query result to a HFSQL data file:

1. Click the "Export to a HFSQL file" button or select "Export into a HFSQL file" from the popup menu (right mouse click) of the query result. The window for selecting the HFSQL file is displayed.
2. Specify the name and location of the HFSQL data file.
3. Validate. The records found in the query result are automatically exported to the HFSQL data file. The result can be viewed with WDMaP for example.

4.1.5 Printing the query result

For the next operation, we consider that the query result is displayed.

To print the query result:


1. Select "Print" from the popup menu (right mouse click) of the query result or click . The window for selecting the items to print is displayed.
2. Indicate the items to print and validate.
3. A print preview showing the query result is displayed.

4.2 Using the query result

The result of a select query can be used in:

- a WINDEV application, a WINDEV Mobile application or a WEBDEV site,
- a window (created by the wizard in WINDEV or WINDEV Mobile),
- a page (created by the wizard in WEBDEV),
- a window control (table, list box, combo box, ... in WINDEV or WINDEV Mobile),
- a page control (loop, table, list box, combo box, ... in WEBDEV),
- a report.

The result of an insert query, update query or delete query can only be used in a program.




Important

Reminder: Reports & Queries cannot be used to create: Insert queries, Update queries, Delete queries.

See "Running the query and handling the result with the WLanguage functions", page 216 for more details.

4.2.1 Query in a WINDEV window or in a WEBDEV page


A query can be used as data source for a window or for a page. The window or the page will display the records selected by the query.




Important

The queries created with Reports & Queries can be run from a WINDEV application only if this application supports the queries created with Reports & Queries

To use a query in a window:

1. Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Window" then "Window".
 - The wizard for window creation starts.
2. In the RAD tab, choose the RAD pattern to use and the type of window to create.
3. Specify (if necessary) the skin template of the window and validate the creation. The wizard for window creation starts.
4. In the creation wizard, select the query that will be associated with the window and the items that will be displayed in the window.
5. Validate. The created window will display the information coming from the associated query.


To use a query in a page:

1. Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Page" then "Page".
 - The wizard for page creation starts.
2. In the RAD tab, choose the RAD pattern to use and the type of page to create.
3. Specify (if necessary) the page template and validate the creation. The wizard for page creation starts.
4. In the creation wizard, select the query that will be associated with the page and the items that will be displayed in the page.
5. Validate. The created page will display the information coming from the associated query.

4.2.2 Query in a control found in a WINDEV window or in a WEBDEV page

All the controls that can be linked to a data source can be associated with one or more result items of the query (edit control, list box, table, ...).

A query can be used as data source for a control. The control will display the records selected by the query.



Important

The queries created with Reports & Queries can be run from a WINDEV application only if this application supports the queries created with Reports & Queries

To display the query result in a List Box, Combo Box, Loop or Table control:

1. Open the window or the page into which the List Box, Combo Box, Loop or Table control will be added.
2. Insert the control associated with the query (via the "Creation" pane of ribbon). The wizard for control creation starts.
3. Specify that the information displayed in the control comes from an existing query.
4. Select the query that will be associated with the control and the items that will be displayed in the control.
5. Validate. The created control will display the information coming from the associated query.
6. Type (if necessary) the following code line in the initialization process of the control. This code line is used to initialize and run the query, and to display the query result in the current control.

```
HExecuteQuery(<Name of Associated Query>)
```

To use a result item of the query in a control (other than a List Box, Combo Box, Loop or Table):

1. Open the window or the page into which a control associated with a query will be added.
2. Create the control associated with the query.
3. On the "Modification" pane, in the "Help for edit" group, click "Description". The description window of control is displayed.
4. Display the "Link" tab in the description window of the control.
5. Select the item to display in the "Linked item" option.
6. Validate. The created control will display the information coming from the associated query.
7. Type (if necessary) the following code line in the initialization process of the control. This code line is used to initialize and run the query, and to display the query result in the current control.

```
HExecuteQuery(<Name of Associated Query>)
```

Note: When creating a Table control associated with a query, you can:

1. Select the query in the "Project" pane directly.
2. Perform a Drag and Drop toward the window. A table associated with the query is automatically created in the window or in the page.

4.2.3 Query in a report

A query can be used as data source for a report. The report will display the records selected by the query. See "Report on query", page 121 for more details.

Tip: On the "Query" pane, in the "File Link" group, the "Create a report" option is used to automatically create a report based on the current Select query.

4.3 Running the query and handling the result with the WLanguage functions

The selection of records performed by query is not necessarily displayed in a report, in a window, ... This selection can also be used in programming.

Most of the HFSQL functions (browse, search, read, ...) can be used both on data files and on the query result. To use these functions on a query, the query must be run beforehand (**HExecuteQuery**).

For example, the *Customer34* query selects all the customers whose zip code starts with 34. The *PrintCustomer* procedure is used to print a proforma invoice for each customer. The following code is used to browse the result of the *Customer34* query and to print an invoice for all the selected customers.

```
// Run the query
ResExecute = HExecuteQuery(Customer34)
// Error while running the query?
IF ResExecute = True THEN
  // Browse the result of the query
  FOR EACH Customer34
    // Print the invoice for each record
    PrintCustomer(Customer34.CustomerLName,
                  Customer34.Customer
FName, ...
                  Customer34.CustomerAddress,
                  Customer34.CustomerZipCode,
...
                  Customer34.CustomerCity)
  END
ELSE
  // Display the error message
  Error("The query was not run.")
END
```

Password

- **Analysis password:** If the query is associated with an analysis protected by password at run time, this password must be specified when opening the analysis (**HOpenAnalysis**) in the initialization process of the project.
- **File password:** If the query is based on one or more password-protected files, these passwords must be specified via **HPass** before running the query (**HExecuteQuery**).

5.ADVANCED QUERY

5.1 Overview

The steps for creating "Simple" queries with the query editor have been presented in the previous chapters. The query editor can also be used to create advanced queries. You have the ability to:

- use the result of a query in another query. It is the concept of sub-queries.
- use parameters in a query. The value of these parameters will be specified when running the query. It is the concept of query with parameters.

5.2 Sub-query

A sub-query is a query whose result is used by another query.

For example, you want to select the orders whose total amount is greater than the average amount of orders. The average amount of orders is not directly known. This average amount will be returned by another query (called sub-query).

A sub-query can be used:

- as selection condition.
- as file of the main query.

Limits:

- The ORDER BY statement is not allowed in the sub-queries.
- The sub-queries cannot be created in the query editor.

5.2.1 Sub-query as selection condition

A sub-query can be used as selection condition.

The authorized syntaxes are as follows:

```
SELECT Item1 [, Item2]
FROM File1 [, File2]
WHERE Item1 Operator (SELECT Item1[, Item2]
  FROM File1 [, File2]
  [WHERE ...])

SELECT Item1 [, Item2]
FROM File1 [, File2]
WHERE [NOT] EXISTS (SELECT Item1[, Item2]
  FROM File1 [, File2]
  [WHERE ...])
```

In this syntax, <Operator> can correspond to:

- **ALL, ANY (or SOME), IN or NOT IN**
 - **ALL:** compare the value selected in the main query with all the values selected in the sub-query.
 - **NOT IN:** checks whether the value selected by the main query does not belong to the list of values selected in the sub-query.

- **IN**: checks whether the value selected by the main query belongs to the list of values selected in the sub-query.
- **ANY (or SOME)**: compare the value selected by the main query with any value selected in the sub-query.
- **ALL**: compare the value selected in the main query with all the values selected in the sub-query.
- **a comparison operator** (=, <, <=, >, >=, <> or !=).
- **the combination between ALL, ANY (or SOME) and a comparison operator** (=, <, <=, >, >=, <> or !=).
 - **ALL**: compare the value selected in the main query with all the values selected in the sub-query.
 - **ANY (or SOME)**: compare the value selected by the main query with any value selected in the sub-query.
 - **ALL**: compare the value selected in the main query with all the values selected in the sub-query.

Example: The following SQL code is used to select the number of the orders whose total amount is greater than the average amount of the orders:

```
SELECT OrderNum
FROM ORDERS
WHERE Amount > (SELECT AVG (Amount)
FROM ORDERS)
```

5.2.2 Sub-query used as data source of a query

A sub-query can be used as file of the main query.
The authorized syntax is as follows:

```
SELECT Item1 [, Item2]
FROM (SELECT Item1 FROM File1 [WHERE ...])
```

5.3 Query with parameters

A query with parameters is a query for which the sort items and the selection criteria are not fixed. The values of parameters are initialized by programming. At run time, the result will depend on the value given to the parameters.

For example, a query with parameters can be used to select the orders placed by a specific customer. The user will only have to select the customer name. The query will use this name in parameter.

To include a query with parameters in a WINDEV/WINDEV Mobile application or in a WEBDEV site, you must:

- Create a query (in the query editor or by programming).
- Create a condition and/or a sort.
- Define the parameter.
- Modify the application in order to use the query with parameters.

The query with parameters can be used with:

- a window or page control (Table, List Box or Combo Box control),
- a report,
- by programming.

5.3.1 Creating a query with parameters



Regardless of the query type (Select, Insert, Delete, Update) and regardless of its creation mode (with the wizard or by SQL directly), you have the ability to use parameters. The name of the parameter is specified when creating the query.

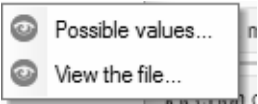
To create a query with parameters:

1. Create a query. See "How to create a query?", page 168 for more details.
2. When defining the selection conditions:
 - specify that the selection condition corresponds to a parameter,
 - give the name of this parameter.

5.3.2 Test of a query with parameters

To run the test of a query with parameters:

1. Open the query with parameters in the query editor.
2. Click  among the quick access buttons.
3. Specify the value of the parameters that will be used for the query test. In this window, you have the ability to select an existing value via . The following menu is displayed:



4. You can:
 - select one of the possible values among the values found in the file.
 - view the entire file.

Caution: This option is available only if Live Data is enabled. See *Live Data* for more details.
 5. Validate. The query result is displayed according to the specified values.
- Note:** If no parameter was selected (unchecked parameter), the conditions that use the parameters will be ignored.

5.3.3 Using a query with parameters

A query with parameters can be used:

- As data source of control (Table, List box, Combo box, Looper, ...).
- As data source of report.
- By programming.

Using a query with parameters in a Table, Looper, List Box or Combo Box control

When a browsing Table control is linked to a query with parameters (embedded or not), parameters must be passed to the query before it is run.

If these parameters are not specified before the first execution of the query, the query will not be run and the Table control will remain empty.

To pass parameters to the query, you can use:

- the following notation:

```
<Query Name>.<Name of Parameter1> = xxx
<Query Name>.<Name of Parameter2> = xxx
<Query Name>.<Name of Parameter3> = xxx
HExecuteQuery(<Query Name>)
```

In this case, the query is automatically run when initializing the Table control.

- HExecuteQuery** by specifying the parameters in the function.

Special case: the query parameters can be directly initialized in the initialization code of control. In this case, there is no need to use HExecuteQuery to calculate the query result. The syntax will be:

```
MySource.<Name of Parameter1> = xxx
MySource.<Name of Parameter2> = xxx
MySource.<Name of Parameter3> = xxx
```

See the online help for more details.

Using a query with parameters in a report

Principle

To use a report based on a query with parameters, the query must be run before opening the report. The query can be run by **iInitReportQuery**. This function is used to run the query and to specify the parameters of this query. This function can be called:

- in the initialization code of report,
- before **iPrintReport**.

Example

The *WIN_CustomerForm* window is used to browse the characteristics of customers. For each customer, the user can click the *BTN_PrintOrder* button to print the *RPT_CustomerOrdReport* report. This report is associated with the *QRY_CustomerOrd* query that selects the orders placed by a given customer.

The following code is used to run the *QRY_CustomerOrd* query according to the current customer and to print the report:

```
// -- Click process of BTN_PrintOrder button
// Run the query according to the current customer
iInitReportQuery(RPT_CustomerOrderReport, CustomerNum)
// Print the report
iPrintReport(RPT_CustomerOrdReport)
```

Using a query with parameters by programming

Principle

When running the query with parameters by programming, the parameters can be passed:

- in the WLanguage **HExecuteQuery** function.
- by using the following notation *<Query Name>.<Parameter Name>*, then by calling **HExecuteQuery**.

See "Two methods for passing parameters to a query", page 222 for more details.

Example

To display the list of orders placed in 2018:

```
// Run the query
HExecuteQuery(QRY_ParamOrder, hQueryDefault, 2018)
// Similar code:
// QRY_ParamOrder.Year = 2018
// HExecuteQuery(QRY_ParamOrder, hQueryDefault)
```

Tip

When a query parameter is initialized with the NULL value, the condition associated with this parameter is ignored when running the query.

For information: If the parameter value comes from an edit control, the option "NULL if empty" ("Details" tab of control description) is used to directly assign the NULL value if the control is not filled.

The SQL code run is:

```
SELECT NameOfItems
FROM NameOfFiles
WHERE Item = {Param1}
```

For this example, the table below describes the use of "NULL if empty":

Value entered by the user	"NULL if empty" option	SQL code run
No value	Option checked	SELECT ItemNames FROM NameOfFiles
No value	Option unchecked	SELECT ItemNames FROM NameOfFiles WHERE Item = ''
A value is entered	Option checked or unchecked	SELECT ItemNames FROM NameOfFiles WHERE Item = 'ValueEntered'

5.3.4 Two methods for passing parameters to a query

Two methods can be used to pass parameters to a query with parameters used in a window or in a control (List Box, Combo Box or Table control for example):

- pass the parameters in **HExecuteQuery**.
- use the `<Query Name>.<Parameter Name>` notation, then **HExecuteQuery**.

This help page presents the operating mode and the benefits of each one of these methods.

Note: The method for passing parameters differs if the query with parameters is used in a report. See "Using a query with parameters in a report", page 220 for more details.

Using HExecuteQuery to pass parameters to a query

When a query is run by **HExecuteQuery**, the last parameters passed to the function correspond to the parameters of the query.

The following syntax is used:

```
<Result> = HExecuteQuery(<Query Name> [, <Connection>] [, <Mode>
[ , <Parameters> [, ...]]])
```

Caution: In this case:

- The values of the query parameters must be specified in their initialization order (visible in the query editor). Each parameter must correspond to the expected type. The different parameters are separated by a comma.
- To specify the parameters of the query, you must specify the initialization mode of the query.

<Query name>.<Parameter name> notation

To pass parameters to a query, all the query parameters can be specified to the HFSQL engine before running the query. This method allows you to specify the parameters in any order (useful for the queries that require several parameters).

The following syntax must be used:

```
<Query Name>.<Name of Parameter 1> = <Value of Parameter 1>
<Query Name>.<Name of Parameter 2> = <Value of Parameter 2>
...
<Query Name>.<Name of Parameter N> = <Value of Parameter N>

<Result> = HExecuteQuery(<Query Name>
[ , <Connection>] [, <Mode>])
```

Caution:

- The name of the parameter must be the name specified when the query was created in the query editor. This name must differ from an item name.
- The indirection operators or the dynamic compilation (**Compile**) cannot be used with this syntax.
- The parameters are reinitialized (reset to 0) whenever the query is run by **HExecuteQuery**.
- If at least one parameter is specified in **HExecuteQuery**, all the parameters specified beforehand are ignored.
- The value of the parameter can correspond to NULL.
- If one of the parameters is not specified, the corresponding condition is ignored.

Avoid specifying all the parameters

Regardless of the method used to pass parameters to the query, all the query parameters do not necessarily have to be specified. The query conditions that use unspecified parameters will be ignored.

Example: Let's consider the "QRY_Customer_LastName_FirstName" query whose SQL code is as follows:

```
SELECT * FROM CUSTOMER WHERE LASTNAME = {Param1} AND FIRSTNAME =
{Param2}
```

- The 2 parameters are specified:

```
HExecuteQuery(QRY_Customers_LName_FName, hQueryDefault, ...
"Smith", "John")
```

will run the query:

```
SELECT * FROM CUSTOMER WHERE LASTNAME = 'Smith'
AND FIRSTNAME = 'John'
```

- Only the name is specified:

```
HExecuteQuery(QRY_Customer_LastName_FirstName, hQueryDefault,
"Smith" )
```

will run the query:

```
SELECT * FROM CUSTOMER WHERE LASTNAME= 'Smith'
```

- Only the first name is specified. However, the name must be specified and it must correspond to Null.

```
sName is Variant = Null
// Use of variant required
// for the Null variable (not filled)

HExecuteQuery(QRY_Customers_LName_FName, hQueryDefault, ...
sLastName, "John")

// OR

HExecuteQuery(QRY_Customers_LName_FName, hQueryDefault, ...
Null, "John")
```

will run the query:

```
SELECT * FROM CUSTOMER WHERE FIRSTNAME= 'John'
```


INDEX

A

Add	
Item	196
Join	207
Operation	200
Selection condition	202
Sort on an item	198
Automatic retrieval	208

C

Calculation	200
Characteristics of a query	196
Close a query	208
Combine the queries	178
Consistency with the analysis	208
Count	196
Creation	
Query	168

D

Delete	
Item	196
Join	207
Operation	200
Records	176
Selection condition	203
Sort on an item	198
DELETE	176
Delete query	176
Creation	177
Description window	195
Selection condition	201
Test	211
Description	
Description window	189
Display	
First N records	206
Item	196
Last N records	206
Documentation	209
Duplicates	207

E

Embedded query	188
Excel	213

F

File	
Excel	213
XLS	213
XML	213
Full-text	196
Full-text search	196

H

HExecuteQuery	216
HExecuteSQLQuery	216
hQueryWithoutCorrection	182

I

Independent query	188
INSERT	172
Insertion of records	172
Insert query	172
Creation	173
Description window	195
Test	211

Interface	
Query editor	163
Item	196
Count	196
Operation	200
Selection condition	201
Sort	198

J

Join	207
Add	207
Delete	207

L		Execution	216	R		Test	211
M		First N records	206	S		Sequence of selection conditions	204
Looper		Independent query	188	T		Sequende of selection conditions	
Using a query	220	Insert	172	U		Modify	204
O		Insert query	172	W		Sort	198
P		Item	196	X		Add	198
Q		Join	207	Y		Delete	198
Menu		Last N records	206	Z		Sort order	198
Ribbon	165	Open	208	AA		Specific period	205
Modify		Operation	216	BB		SQL	181
Displayed values	201	Overview	159	CC		SQL commands	182
Selection condition	203	Print	214	DD		SQL command	182
Sequence of conditions	204	Print area	209	EE		Start the query editor	163
Open		Rename	208	FF		Sub-Query	217
Query	208	Reverse engineering	181	GG		System lock	208
Operation	200	Save	208	HH		Table	
Add	200	Select	169	II		Using a query with parameters	220
Delete	200	Selection condition	201	JJ		Type of queries	168
Parameters		Select query	169	KK		UNION	
Pass parameters	222	SQL code	181	LL		178	
Print		Sub-query	217	MM		Union query	
Documentation	209	Test	211	NN		178	
Result	214	Union	178	OO		Creation	
Print area	209	Union query	178	PP		180	
Project	167	Update	174	QQ		Description window	
Query and project		Update query	174	RR		194	
Query editor		Use	214	SS		Update	
Developer version		Use in a control	215	TT		Records	
Documentation		Use in a report	216	UU		174	
Interface		Use in a window	214	VV		UPDATE	
Main window		Use with the WLanguage functions	216	WW		174	
Minimum configuration		With parameters	218	XX		Update query	
Registration card		Query and project	167	YY		174	
Setup		Query editor	161	ZZ		Creation	
User version		Developer version	161	AAA		175	
Query in SQL code		Documentation	162	BBB		Description window	
Available SQL commands		Interface	163	CCC		195	
Creation		Main window	163	DDD		Selection condition	
Execution mode		Minimum configuration	163	EEE		201	
Query mode		Registration card	163	FFF		Test	
Query with parameters		Setup	163	GGG		211	
Create		User version	161	HHH		Use the result of a query	
Pass parameters		Query in SQL code	181	III		214	
Run the test		Available SQL commands	182	JJJ		Window	
Use		Creation	181	KKK		Main editor window	
Use in a control		Execution mode	181	LLL		163	
Use in a report		Query mode	188	MMM		Use a query in a window	
Use in a window		Query with parameters	218	NNN		214	
Use with the WLanguage functions		Create	219	OOO		Window control	
With parameters		Pass parameters	222	PPP		Use a query in a control	
Record		Run the test	219	QQQ		215	
Delete				RRR		WLanguage	
First N records				SSS		216	
Insert				TTT		XML	
Last N records				UUU		213	
Select				VVV			
Specific period				WWW			
Update				XXX			
Registration card				YYY			
Rename				ZZZ			
Item				AAA			
Query				BBB			
Report				CCC			
Use a query in a report				DDD			
Result				EEE			
Export to an Excel file				FFF			
Export to an XML file				GGG			
Modify the displayed values				HHH			
Print				III			
Use				JJJ			
Reverse engineering				KKK			
Ribbon				LLL			
Run the test				MMM			
Save				NNN			
Query				OOO			
SELECT				PPP			
Selection condition				QQQ			
Add				RRR			
Delete				SSS			
Modify				TTT			
Sequence				UUU			
Use a sub-query				VVV			
Selection of records				WWW			
Select query				XXX			
Description window				YYY			
Duplicate				ZZZ			
Item				AAA			
Join				BBB			
Modify the displayed values				CCC			
Possible operations				DDD			
Selection condition				EEE			
Sort records				FFF			
Specific period				GGG			

Reports and Queries

24

REPORTS AND QUERIES - SUMMARY

1.	Distributing Reports & Queries	
1.1	Overview	233
1.2	Distributing Reports & Queries with your WINDEV application	233
1.3	Step 1: Configuring the project	234
1.4	Step 2: Configuring the analysis	234
1.4.1	Visible data files and items	234
1.4.2	Translating the names of data files and items	235
1.5	Step 3: Configuring reports	235
1.6	Step 3 bis: Configuring queries	236
1.7	Step 4: Creating the executable and distributing the application	236
1.8	Step 5: Installing the WINDEV application on the end-user computers	237
2.	Using Reports & Queries	
2.1	Starting Reports and Queries	238
2.2	Principle for using reports	238
2.2.1	Modifying a report	238
2.2.2	Using a modified report	238
2.2.3	Using a new report	239
2.3	Principle for using queries	239
2.3.1	Modifying a query	239
2.3.2	Using a new query	239
3.	Differences compared to the standard report editor of WINDEV	

1. DISTRIBUTING REPORTS & QUERIES

1.1 Overview

This chapter presents the operations that must be performed in order for a WINDEV application to be handled by Reports & Queries.

Reports & Queries replaces "WDEtat User". This module is now free and it can be distributed with your WINDEV applications. See the license for more details about the distribution and use conditions.

1.2 Distributing Reports & Queries with your WINDEV application

To distribute Reports & Queries with your own WINDEV application, the provider of the WINDEV application must:

- Specify in the project that the application allows Reports & Queries to be run.
- Define in the analysis the files and the items that can be handled by Reports & Queries and (if necessary) the names of files and items available in the different languages.
- Define the reports that can be modified in Reports & Queries.
- Define the queries that are visible in Reports & Queries.
- Create the executable and the setup program of WINDEV application.
- Install the WINDEV application on the end-user computers.

Note: To reduce the size of the setup program, "Reports & Queries" can be downloaded from your application:

- via the automatic menu,
- via the print preview of a report,
- when using **RunReportsAndQueries**.

Translate the names of data files and items

"Reports & Queries" is distributed in different countries (it is available in English and in French). The languages used are different: Spanish, Romanian, Chinese, Swedish, ...

If the end user wants to create reports or queries, he must handle the names of data files and items. These names are used in the analysis by the application developer. In most cases, these names are in English, which is not suitable for a Chinese or Romanian user.

You now have the ability to specify a translation for the names of data files and items in the analysis.

1.3 Step 1: Configuring the project

To use Reports & Queries from your WINDEV application, you must configure the project associated with your application.

1. Display the description window of project: on the "Project" pane, in the "Project" group, click "Description".
2. In the "Advanced" tab, check "Allow to start "Reports and Queries"".
3. Validate.

Notes:

- By default, the WINDEV projects allow Reports & Queries to be run.
- The initialization code of the project is run during the test of a report or query from Reports & Queries. In this case, `InTestMode` returns `False`.

1.4 Step 2: Configuring the analysis

1.4.1 Visible data files and items

By default, all data files and their items can be handled by Reports & Queries. The end user can use this data in his custom reports and queries.

To limit the data that can be used in Reports & Queries (if your application contains sensitive data for example), the WINDEV analysis must be modified: all you have to do is make the requested data files and/or items "invisible" in Reports & Queries.

Note: The reports and the queries created from WINDEV and based on invisible data files and/or items can be run and/or modified from Reports & Queries.

To define an invisible data file in Reports & Queries:

1. In the data model editor of WINDEV, select the requested data file.
2. Display the description window of the file items ("Description of data file" from the popup menu).
3. In the "Reports and Queries" tab, uncheck "Visible by the end user, in "Reports and Queries"".
4. Validate.

Note: All file items will automatically become invisible.

To define an invisible file item in Reports & Queries:

1. In the data model editor of WINDEV, select the data file containing the requested item.
2. Display the description window of the file ("Description of items" from the popup menu).
3. Select the requested item.
4. In the "Reports and Queries" tab, uncheck "Visible by the end user, in "Reports and Queries"".
5. Validate.

Note: `..VisibleEndUser` can also be used to find out or modify the visibility of an item in Reports & Queries.

Caution: To take into account the modifications performed in the WINDEV analysis, you must regenerate the WINDEV analysis: on the "Analysis" pane, in the "Analysis" group, expand "Generation" and select "Generation".

1.4.2 Translating the names of data files and items

By default, when the data files and the items are handled by the end user when creating a report or a query, all the names used correspond to the ones defined by the developer in the analysis. In most cases, these names do not correspond to the language of the user. You have the ability to propose a translation for the following elements in the WINDEV analysis:

- Name of data file,
- Caption of data file,
- Name of item,
- Caption of item.

Note: If your project is going to be translated, don't forget to translate this information. This information is automatically checked out and checked in by WDMMSG.

To define the multilingual parameters of a data file in Reports & Queries:

1. In the data model editor of WINDEV, select the requested data file.
 2. Display the description window of the file items ("Description of data file" from the popup menu).
 3. In the "Reports and Queries" tab, specify the name of data file and its caption in the different languages supported by the analysis.
- Note:** The name used must not correspond to a file name already found in the analysis.
4. Validate.

To define the multilingual parameters of a file item in Reports & Queries:

1. In the data model editor of WINDEV, select the data file containing the requested item.
2. Display the description window of the file ("Description of items" from the popup menu).
3. Select the requested item.
4. In the "Reports and Queries" tab, type the name of item and its caption in the different languages supported by the analysis.
5. Validate.

Caution: To take into account the modifications performed in the WINDEV analysis, you must regenerate the WINDEV analysis: on the "Analysis" pane, in the "Analysis" group, expand "Generation" and click "Generation".

1.5 Step 3: Configuring reports

To make a report of your application modifiable in Reports & Queries:

1. Open the report in the report editor.
2. Display the description window of report ("Report description" from the popup menu of report).
3. In the "Options" tab, check "Report modifiable in Reports & Queries".
4. Check:
 - "Modifiable interface" in order for the report interface to be modifiable in Reports & Queries.
 - "All modifiable (interface and code)" in order for the entire report to be modifiable in Reports & Queries.
5. Validate and save the report.

Notes:

- `..InterfaceModifiableWithReportsAndQueries` is used to find out whether the report interface can be modified in Reports & Queries.
- `..AllModifiableWithReportsAndQueries` is used to find out whether a report can be entirely modified in Reports & Queries.

1.6 Step 3 bis: Configuring queries

By default, a query is neither visible nor modifiable in Reports & Queries.
It can only be made visible in Reports & Queries. It cannot be made modifiable.

To make a query of your application visible in Reports & Queries:

1. Open the query in the query editor.
2. Display the description window of query ("Query description" from the popup menu of query).
3. Click the "Advanced" button.
4. In the "Reports & Queries" tab, check "Visible in Reports & Queries".
5. Validate and save the query.

Note: The queries can only be viewed in Reports & Queries. They cannot be modified. Reports & Queries allows you to perform the following operations on a query:

- run and view the query result.
- use the query as base for building other queries.
- perform a copy of query ("Save as" option).
- use the query as data source of report.

1.7 Step 4: Creating the executable and distributing the application

The method for creating and distributing a WINDEV application that is using Reports & Queries is the same as the method for creating and distributing a standard application. The parameters specific to Reports & Queries can be specified when creating the setup procedure.

Once your WINDEV application is entirely described (windows, reports, queries, ...), create the corresponding executable.

Then, create the setup program. In the wizard for creating the setup program, in the "Additional modules" step:

1. Check "Include the setup of Reports and Queries". The setup package of Reports and Queries is automatically sought. This setup package is found in the "Install\Reports and Queries" sub-directory of WINDEV setup directory.

Caution: Including Reports and Queries significantly increases the size of your setup (about 160 MB). The setup package of Reports and Queries can also be distributed separately. If Reports and Queries is not included in the setup, it can be downloaded from the application (via the automatic menu, from the print preview of a report or by **RunReportsAndQueries**). By default, the software is downloaded on an FTP site of PC SOFT. You have the ability to customize the download address of software (site of a company for example). See "Customizing the download address" for more details.

2. Specify (if necessary) the directories corresponding to the reports and queries modified or created by the end user:

- the directory of custom reports and queries corresponds to the directory of the reports and queries visible by the user who created them.
- the directory of shared reports and queries corresponds to the directory of the reports and queries visible by all application users.

3. Allow (or not) these directories to be modified when installing the application.

Note: **fReportsAndQueriesDir** is used to find out the directory of custom and/or shared reports and queries.

Customizing the download address

If Report & Queries was not included in the setup program, this software can be downloaded when using the automatic menu, when previewing a report or when starting the software.

By default, the download address corresponds to an FTP site of PC SOFT.

To use another site:

1. Create a text file named LoginFTP.INI. This file and the application executable must be found in the same directory.
2. This file must contain the following elements:

```
[FTP]
Login = Login of the user on the FTP server
PSW = Password of the user on the FTP server
Serv = FTP server (ftp.windev.com for example)
Dir = Download directory (WX22/latest for example)
```

Notes:

- If the download package is found at the root, write "." into the Dir entry.
- The file to download is named "WD24OPACKER.exe".

1.8 Step 5: Installing the WINDEV application on the end-user computers

The method for installing an application that is using Reports & Queries is the same as the method for installing a standard application.

During the setup, if "Reports & Queries" was included in the setup program, the user will only have to specify whether he wants to install Reports & Queries.


If Report & Queries was not included in the setup program, this software can be downloaded when using the automatic menu, when previewing a report or when starting the software.

2.USING REPORTS & QUERIES

2.1 Starting Reports and Queries

To allow the end users to customize the reports of your application or to create their own reports and their own queries, all you have to do is start Reports & Queries.

To allow the end users to start Reports & Queries:

- display your reports in the preview window. The end users will be able to start Reports & Queries by clicking .
- add the "?" menu into the main window of your application: on the "Window" pane, in the "Bars and menus" group, expand "Main menu" and select "Add the '?' menu". In the wizard for creating this menu, check "Create, modify a report" and "Create, modify a query".
- use `RunReportsAndQueries` in your application.

If "Report and Queries" is not installed, a window allows you to download the software.

2.2 Principle for using reports

2.2.1 Modifying a report

When modifying a report, the corresponding file (".WDE" file) is automatically checked out from the application library and copied into the directory of "Custom reports and queries" (defined when creating the setup procedure of WINDEV application).

This is the report that will be modified by the end user.

2.2.2 Using a modified report

The test of the modified report can be directly run in "Reports & Queries". In this case, the initialization process of project will be run.

By default, only the user who modified a report will use this modified report in the WINDEV application.

In order for a modified report to be used by the other application users, the end user must select "File .. Share the element with the other users". The file corresponding to the modified report (".WDE" file) will be automatically deleted from the directory of "Custom reports and queries" and copied into the directory of "Shared reports and queries".

When a report is run from the WINDEV application, the report file is sought:



- first, in the directory of "Custom reports and queries".
- then, in the directory of "Shared reports and queries".
- finally, in the library of the WINDEV application.

Note: A mention displayed in the title of preview window informs the users that the report used corresponds to a modified report.

To restore the report to its initial version, the end users will have to physically delete the file of modified report (".WDE" file) from the "Custom reports and queries" directory and/or from the "Shared reports and queries" directory.

2.2.3 Using a new report

The end users will be able to print their new reports from Reports and Queries:

- by starting a print ().
- by running the report test ().
In this case, the initialization process of project is run.

Therefore, no programming is required to print the report.

In order for the reports created by Reports & Queries to be directly printed from a WINDEV application, this feature must be implemented in your application.

You can for example include a window allowing the end user to:

- select a report created in Reports & Queries.
- run this report.

2.3 Principle for using queries


2.3.1 Modifying a query

The application queries can only be viewed in Reports & Queries. They cannot be modified. Reports & Queries allows you to perform the following operations on a query:

- run and view the query result.
- use the query as base for building other queries.
- perform a copy of query ("Save as" option). A new query will be created.
- use the query as data source of report.

2.3.2 Using a new query

The end users will be able to run their new queries:

- from Reports & Queries directly by clicking .
In this case, the initialization process of project will be run.
- from the WINDEV application.

In order for the queries created by Reports & Queries to be directly run from a WINDEV application, this feature must be implemented in your application.

You can for example include a window allowing the end user to:

- select a query created in Reports & Queries.
- start the execution of this query and display its result in a table (with `BuildBrowsingTable` for example).

3. DIFFERENCES COMPARED TO THE STANDARD REPORT EDITOR OF WINDEV

Reports & Queries cannot be used to:

- create reports on a HFSQL view.
- create reports on a memory zone.
- create reports on a table control.
- create reports on variable.
- create reports on a programmed data source.
- create skin templates of reports.
- create Insert queries.
- create update queries.
- create delete queries.
- use the functions for modifying records (HAdd, HModify, HDelete, ...).

However, Reports & Queries allows you to customize most of the types of reports created with the standard report editor of WINDEV (color, size, font, ...).