

DEVELOP 10 TIMES FASTER

Reports and Queries



PFSOFT

Remember to visit our Web site (www.windev.com) on a regular basis to find out whether upgraded versions are available.

Email address of 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 2010: This publication may not be reproduced in part or in full without the express consent of PC SOFT.



SUMMARY

PART 1 - REPORT EDITOR

1. INTRODUCTION	13
2. HOW DO I CREATE A REPORT ?	28
3. DATA SOURCES OF A REPORT	47
4. DESCRIBING AND HANDLING A REPORT	55
5. REPORT BLOCKS	65
6. REPORT CONTROLS	75
7. USING TABLES IN A REPORT	99
8. EDITABLE REPORTS	109
9. NESTED REPORTS	113
10. PRINTING A REPORT BY PROGRAMMING	117
11. PROGRAMMING A REPORT	139

PART 2 - QUERY EDITOR

1. INTRODUCTION	161
2. HOW DO I CREATE A QUERY ?	170
3. DESCRIBING A QUERY	187
4. RUNNING THE TEST OF A QUERY AND USING IT	208
5. ADVANCED QUERY	215



PART 3 - REPORTS AND QUERIES

1. DISTRIBUTING REPORTS & QUERIES	231
2. USING REPORTS & QUERIES	236
3. LIMITS COMPARED TO THE STANDARD REPORT EDITOR OF WINDEV	239



DEVELOP 10 TIMES FASTER

PART 1

Report editor



PC SOFT



REPORT EDITOR - SUMMARY

1. INTRODUCTION	13
1.1 What is a report ?	13
1.2 The report editor	17
1.2.1 Quick overview.....	17
1.2.2 Differences between the standard query editor and Reports & Queries.....	18
1.3 Main interface of the report editor	20
1.4 Drop-down menus of the report editor	21
1.5 The main toolbars of the report editor	25
1.5.1 Toolbar of controls.....	25
1.5.2 Toolbar for aligning controls.....	26
1.5.3 Toolbar for the border of controls	26
1.5.4 Keyboard shortcuts	26
2. HOW DO I CREATE A REPORT ?	28
2.1 Report and project	28
2.2 Creating a report	29
2.2.1 Overview.....	29
2.2.2 The different types of reports	29
2.2.3 Source of the report	37
2.2.4 Breaks and sorts.....	38
2.2.5 Performing calculations in a report.....	40
2.2.6 Skin template.....	40
2.2.7 Printout format of the report.....	40
2.3 Saving the report	40
2.4 Running the report test	40
2.4.1 Run the test	41
2.4.2 Stopping the test	42
2.4.3 Directory of the data files used for test	42
2.4.4 Managing the passwords (analysis or data file).....	42
2.4.5 WinDev Mobile: Opening a window in test mode in the project's code.....	42
2.5 Printing a report	43
2.5.1 Available print modes.....	43
2.5.2 Print preview	43
3. DATA SOURCES OF A REPORT	47
3.1 Overview	47
3.2 Report based on a data file	47
3.2.1 Overview	47
3.2.2 Characteristics	48
3.3 Report based on a query	48
3.3.1 Overview	48
3.3.2 The different types of queries.....	49
3.4 Report based on a variable	50
3.5 Report based on a HyperFileSQL Classic view	50



3.5.1 Overview	50
3.5.2 Characteristics	50
3.6 Report based on a memory zone	51
3.6.1 Overview	51
3.6.2 Characteristics	51
3.7 Report based on a Table control	52
3.7.1 Overview	52
3.7.2 Characteristics	52
3.8 Report based on a text file	53
3.8.1 Overview	53
3.8.2 Characteristics	54
3.9 Report based on a programmed data source	54
4. DESCRIBING AND HANDLING A REPORT	55
4.1 Overview	55
4.2 Viewing and modifying the report description	55
4.2.1 Managing the data source	55
4.2.2 Display options in the print preview	56
4.2.3 Managing the blocks	56
4.2.4 Options	56
4.2.5 Form and background image.....	57
4.2.6 Language.....	57
4.3 Opening, closing and saving a report	57
4.4 Using a form in a report	58
4.4.1 Information specific to the form	59
4.4.2 Including a form in a report	60
4.4.3 Image of the form	60
4.4.4 Report based on a PDF file	60
4.5 Viewing the report in 3D	61
4.6 Changing the skin template associated with the report	63
4.7 Change the report template associated with a report	63
4.8 Managing the versions	64
5. REPORT BLOCKS	65
5.1 Overview	65
5.2 The different types of blocks	66
5.2.1 Start of Document block	67
5.2.2 Page Header block	67
5.2.3 Break Header block.....	67
5.2.4 Body block	67
5.2.5 Body Complement block	68
5.2.6 Break Footer block	68
5.2.7 Page Footer block	68
5.2.8 End of Document block.....	68
5.2.9 Iteration block	68
5.3 Characteristics of report blocks	68
5.4 Parameters and print options of the blocks	69
5.5 Handling the blocks in the report editor	70
5.5.1 Create a block.....	70





5.5.2 Delete a block	70
5.5.3 Rearranging the blocks	71
5.6 Handling the blocks by programming	71
5.6.1 Processes of the report blocks	71
5.6.2 Handling a block in a WLanguage process.....	72
5.6.3 Properties associated with the report blocks	73
6. REPORT CONTROLS	75
6.1 The different types of controls	75
6.1.1 Static control	75
6.1.2 Item control	77
6.1.3 Calculated control.....	77
6.1.4 Image control	78
6.1.5 Bar Code control	79
6.1.6 RTF control	79
6.1.7 Preset control.....	81
6.1.8 Chart control	81
6.1.9 Shape control.....	83
6.1.10 Check Box control	83
6.1.11 Internal Report control	84
6.1.12 URL Link control	84
6.1.13 Edit control.....	85
6.2 Characteristics of report controls	86
6.2.1 Description window of a control	86
6.2.2 Details of the description window	87
6.3 Multilingual feature in a report controls	87
6.4 Group of controls	88
6.5 Handling the controls in the editor	88
6.5.1 Creating a control	88
6.5.2 Selecting one or more controls.....	89
6.5.3 Copying a control	90
6.5.4 Copying the style of a control.....	90
6.5.5 Moving a control	91
6.5.6 Deleting a control	91
6.5.7 Using an off-report control	91
6.6 Aligning the report controls	92
6.6.1 Alignment options.....	92
6.6.2 Real-time interface checker.....	92
6.6.3 Zoom.....	93
6.6.4 Rulers and markers	93
6.6.5 Snap-on grid	94
6.7 Handling the controls by programming	95
6.7.1 Processes of the report controls	95
6.7.2 Handling a control in a WLanguage process.....	95
6.7.3 Properties associated with the report controls	96
7. USING TABLES IN A REPORT	99
7.1 Overview	99
7.2 Creating a table in a report	101



7.2.1 Creating a Table report	101
7.2.2 Creating a crosstab report	102
7.2.3 Inserting a table into a report	102
7.3 Describing the table	103
7.3.1 Characteristics of the table.....	103
7.3.2 Characteristics of columns	104
7.4 Handling the cells of the table	104
7.4.1 Selecting a table element	105
7.4.2 Displaying the table description	105
7.4.3 Adding a cell to a table.....	105
7.4.4 Adding a column to a table.....	105
7.4.5 Resizing the table cells	106
7.4.6 Moving a table column.....	107
7.4.7 Removing or deleting a column from a table	107
8. EDITABLE REPORTS	109
8.1 Overview	109
8.2 How do I proceed ?	109
8.2.1 Creating the report with edit controls	109
8.2.2 Characteristics of edit controls.....	109
8.2.3 Processes associated with an edit control.....	110
8.3 Input in the preview window (WinDev only)	110
8.4 Input in a PDF file	111
8.5 Input in a report based on a PDF form	111
9. NESTED REPORTS	113
9.1 Overview	113
9.2 Creating a nested report	113
9.2.1 Principle.....	113
9.2.2 Example.....	114
9.3 Specific features of secondary reports	115
9.3.1 Blocks in the secondary report.....	115
9.3.2 Orientation of secondary reports.....	115
9.3.3 Page break in the secondary report.....	115
9.3.4 Secondary report based on a query with parameters	115
9.3.5 Secondary report based on a query that uses a specific connection.....	115
9.4 Printing a nested report	115
9.5 Differences between the nested reports and the internal reports	116
10. PRINTING A REPORT BY PROGRAMMING	117
10.1 Overview	117
10.1.1 Principle for printing in WinDev.....	117
10.1.2 Principle for printing in WebDev.....	118
10.1.3 Principle for printing in WinDev Mobile	118
10.2 Report based on a data file	121
10.2.1 Printing a report based on a data file.....	121
10.2.2 Printing a bounded report based on a data file	122
10.2.3 Printing a report with parameters based on a data file	123
10.2.4 Printing a bounded report with parameters based on a data file	125





10.3 Report based on a query	126
10.3.1 Printing a report based on a query without parameter	126
10.3.2 Printing a report based on a query with parameters.....	127
10.4 Report based on a table control	128
10.5 Report based on a variable	130
10.5.1 Printing a report based on a variable	130
10.5.2 Printing a report with parameters based on a variable.....	131
10.6 Report based on a HyperFileSQL Classic view	132
10.7 Report based on a memory zone	134
10.8 Report based on a text file	136
10.9 Printout functions	137
11. PROGRAMMING A REPORT	139
<hr/>	
11.1 Overview	139
11.2 Sequence of processes associated with a report	139
11.3 Processes associated with the report	140
11.3.1 "Opening" process	140
11.3.2 "Reading data" process.....	141
11.3.3 "Filter" process.....	141
11.3.4 "Closing" process.....	141
11.4 Break process	142
11.5 Sort procedure	142
11.6 Report with parameters	143
11.6.1 Creating a report with parameters.....	143
11.6.2 Printing a report with parameters.....	144
11.6.3 Running the test of a report with parameters.....	145
11.7 Handling a report in a WLanguage process	146
11.8 Keywords associated with reports	146
11.9 Properties associated with the reports	146
11.10 WLanguage functions	148
11.10.1 WLanguage functions not recommended in the processes	148
11.10.2 Print functions of WLanguage that can be used in a report.....	148







1. INTRODUCTION

1.1 What is a report ?

A report provides a custom view of the data: data entered in the database, data found in a text file, etc.

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

- In WinDev, a report can be displayed on the screen, saved in an HTML, PDF, PCL file, printed directly on a printer, etc.
- In WebDev, a report can be saved in an HTML or PDF file, sent by email, displayed in the browser of a Web user or printed on a printer connected to the server, etc.
- In WinDev Mobile, a report can be printed on a printer (in PCL format) or saved in a PCL file.

A report contains:

- **blocks** dividing the report into different parts (See “Report blocks”, page 65, for more details).
- **controls** where the report’s data is displayed (See “Report controls”, page 75, for more details).

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

- group the data,
- sort the data according to any criterion,
- perform calculations, averages, statistics, and even create charts,
- display the data in several columns.

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

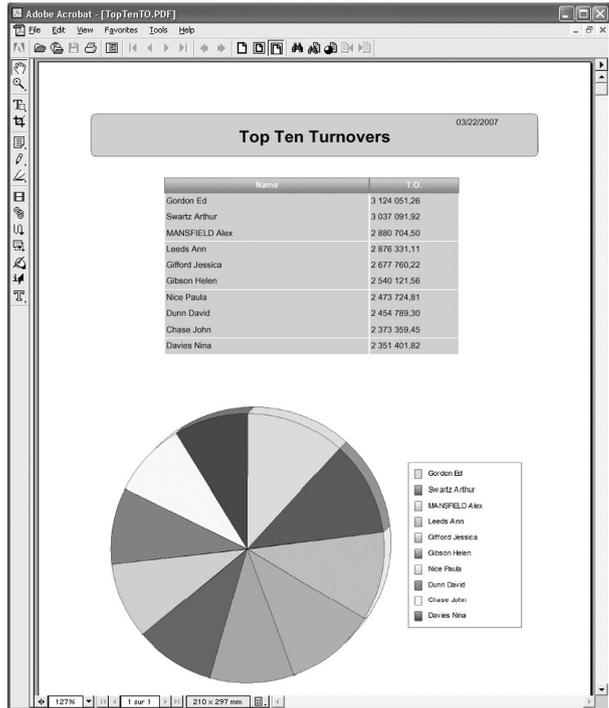
Some examples of reports:

Quantity	Ref.	Specification	Amount	Unit Price (BT)
1	000021NZZD	BRA WHITE 38B	297,69	297,69
2	009999ZBEM	KNICKERS PINK 20	28,32	14,16
2	345933CAL	KNICKERS MAXI DIM BLACK 18	4 026,00	2 013,00
2	009817KRS	KNICKERS WHITE 18	25,16	12,58
1	003248GNO	UNDERSHIRT DIM SMALL	57,04	57,04
1	341561CAL	UNDERWIRED BRA WHITE 38D	806,00	806,00
2	349824M-M	KNICKERS YELLOW 18	5 924,00	2 962,00
				1 324 006,84

Order form
(print preview in WinDev)



Table + Chart report
displayed in PDF format



Supplier Form

Supplier # 1

Company
Apex Corp.
2480 Washington St
13221 Syracuse NY

Contact
Jacobson Pete
Phone.: 315-741-5620
Fax: 01.91.80.06.42

Comment

Information form
of a supplier

The diagrams below present a simple definition of a report:

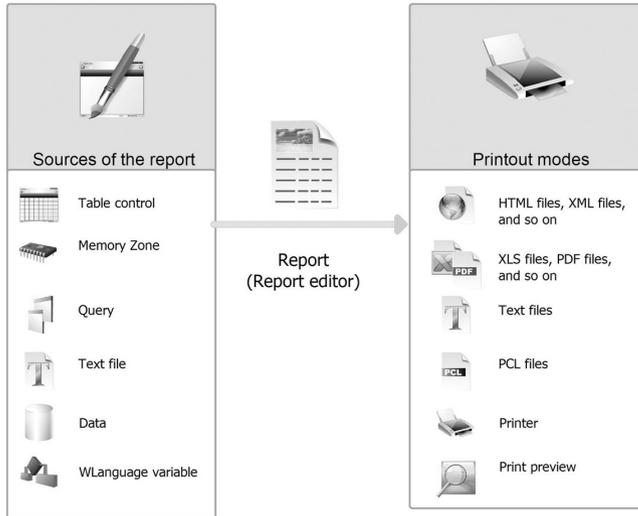
- the data to print comes from a data source (data file, HyperFileSQL view, query, memory zone, WLanguage variable or text file, etc.).
- the report groups, sorts and organizes the data.



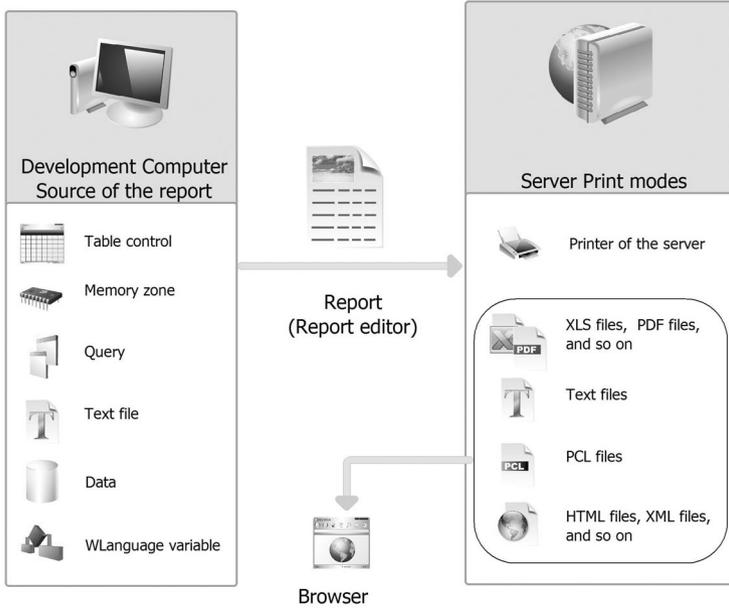


- the execution of the report can be directed to:
 - in **WinDev**: the screen, a PDF file, a HTML file, a RTF file, a printer or a fax, etc.
 - in **WebDev**: an HTML file, a RTF file, a PDF file, a printer or a fax, etc. If the report destination is a file, this file can be sent to the browser.
 - in **WinDev Mobile**: a PCL file or a printer.

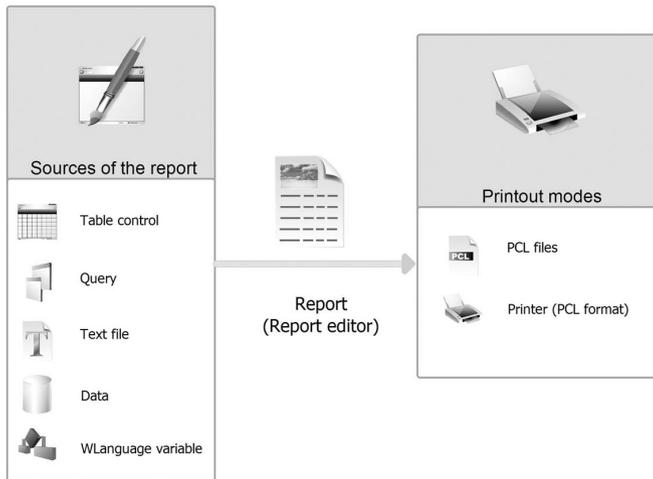
In WinDev:



In WebDev:



In WinDev Mobile:





1.2 The report editor

1.2.1 Quick overview

WinDev, WebDev and WinDev Mobile enable you to create reports with the report editor. The reports can be created with the data coming from:

- a file found in an analysis or in any data file,
- the result of a query,
- a view on a HyperFileSQL file,
- a memory zone,
- a Table control found in a window,
- a variable (variable global to the project or variable used in the current report),
- a text file,
- and so on.

The report editor also enables you 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 also have the ability to perform specific processes in WLanguage (the programming language of WinDev, WebDev and WinDev Mobile).

The report editor is easy to use and it proposes several useful features:

- *Choosing the destination of the report execution:*
You have the ability to perform a printout in a PDF, HTML, XML, XLS or RTF file.
In WebDev, this file is saved in a directory accessible in write mode on the server. This file can then be displayed on the browser.
In WinDev, the report can be previewed and printed from the print preview.
In WinDev Mobile, the report can be saved in PCL format and printed on a PCL printer.
- *Choosing the items to print:*
You have the ability to choose the items that will be printed (no need to print all of them).
- *Ability to print the data coming from several data files:*
The report can contain items coming from several data files. To simplify the data selection, 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 the 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. See “The sort”, page 38, and “The breaks”, page 39, for more details.
- *Printing labels:*
You have the ability to create and print custom labels.



- **Custom calculations:**
Custom calculations can be added to your report (VAT, etc.).
- **Totals and counters:**
Totals and counters can be added to the controls (calculated controls or controls linked to items). These totals and counters can be general, intermediate, by break, etc.
- **Images, borders, background colors:**
Charts, images, borders and background colors can be added to your report.
- **Choosing the fonts:**
The controls found in your report can have different fonts, different colors, etc.
- **Configuring the page headers and footers:**
Various information (date, page number, total number of pages, and so on) can be added to 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 instance).
- **Nested reports:**
Nested reports can be created by calling a report from the main report.
- **Choosing the number of columns:**
You can print a report's data in several columns.

1.2.2 Differences between the standard query editor and Reports & Queries

Two versions of the report editor are available:

- **the standard report editor.**
This version is supplied with WinDev, WebDev and WinDev Mobile. It can only be used when developing a WinDev/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 the report editor are not available in this version.

The icon  indicates the differences between the user version of the report editor and the standard report editor.

Standard report editor

The report editor enables you to create reports and to run them from the editor or from an application developed in WinDev, WinDev Mobile or WebDev.

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

However, no dongle is required for 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, etc.).

To find out the list of examples for using reports, display the guide ("? .. See the guide") and enter 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 the report editor (Reports & Queries - WinDev only)

Reports & Queries enables you to create and run reports directly 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 for instance).

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

No dongle is required to use Reports & Queries.

Some operations are required to create new queries or to modify existing queries with Reports & Queries. See "Distributing Reports & Queries with your WinDev application", page 231, for more details.

Documentation

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

Printed documentation

The printed documentation (the guide you are reading) is designed 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" guide.

Online help

The online help is intended to help you handle the editor.

To access the online help: select "? .. Summary", "? .. Index of the help" or press [F1].

Setup

Installing the standard report editor

The standard report editor is automatically installed along with WinDev, WebDev or WinDev Mobile.

Installing Reports & Queries

To install Reports & Queries, follow the instructions given when installing the WinDev application.

Minimum configuration for Reports & Queries

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

- Pentium with 128 MB of RAM (256 MB or more recommended).
- with 100 MB of disk space.
- in Windows 2000, 2003, XP or Vista.

Register yourself

The registration card supplied with your product is very important. It enables you to benefit from the free services offered by PC Soft to its customers: free Technical Support, information about new releases, product upgrades, etc.

Fill and return your registration card today.



1.3 Main interface of the report editor

To start the report editor, create a new report ("File .. New .. Report") or open an existing report ("File .. Open").



1 Toolbars and drop-down menu

Gives quick access to the main features of the report editor. You can:

- customize the toolbar ("Display .. Toolbars .. Customize the toolbars").
- display/hide the toolbar ("Display .. Toolbars .. Pop up bar").
- move the toolbar with the mouse.

2 Alignment rulers

Used to align the controls in the report. To display the rulers, press [CTRL]+[R].

3 Report being edited

Scrollbars allow you to scroll the large reports. You also have the ability to zoom in the report to better view the report's elements ([CTRL]+mouse wheel).

4 Vertical panes

Displays:

- the shortcuts to the main features available in the report editor ("Commands" pane),
- the elements (files, items, reports, controls, and so on) available in the current project ("Project Explorer" pane).
- the description of the report elements via the modifier.



**Important**

In Reports & Queries, only the "Project Explorer" and the "Modifier" panes are available.

5 Code window

Displays the processes associated with the reports, with its blocks and with its controls.

6 Description window

Enables you to view and/or modify the characteristics:

- of the selected controls.
- of the current report.

7 Opened documents

Displays a button for each open document in the report editor. To display the requested element in the foreground, click the corresponding "button".

8 Horizontal panes

Displays:

- the code elements (list of classes, list of local or global procedures, "Code" pane),
- the compilation errors (errors found in the WLanguage code) or GUI errors, "Compilation errors" pane),
- the search tool ("Find-Replace" pane),
- the debugger when testing an element ("Debugger" pane),
- etc.

1.4 Drop-down menus of the report editor

The tables below describe the actions of the menu options ("Insert", "Report", "Control", "Code" and "Display") specific to the report editor.

Insert menu

Option	Action
New control	Inserts the specified control into the report.
New break	Inserts a break.
New block	Inserts a new block into the current report.
New table	Inserts a table into the current report (only if no table is found in the report).
New column	Inserts a column into the table found in the report.
New internal report	Inserts an Internal Report control into the current report.
Date	Inserts a preset Date control.
Time	Inserts a preset Time control.



Page number	Inserts a preset control corresponding to the number of the current page.
Number of Pages	Inserts a preset control corresponding to the total number of pages in the report.
Page number / Number of pages	Inserts a preset control indicating the number of the current page in relation to the total number of pages in the report.
A template (inheritance)	Associates a report template with the current report.

Report menu

Option	Action
Description	Displays the description window of the report
Page layout	Displays the window for selecting the page format
Format of labels	Displays the description window for the format of labels (for a label report only)
Edit the query	Displays the base query of the report (for a report based on an embedded query)
Edit the sorts	Displays the description window of breaks and sorts
Edit the selection conditions	Displays the description window of selection conditions
Edit the breaks	Displays the description window of breaks and sorts
Edit the groups	Displays the description window for the groups of controls
List of templates used	Displays the list of templates used by the current report
List of overloaded controls	Displays the list of overloaded controls in the current report
Update the associated reports	Updates the reports associated with the current template
Reverse modeling of templates	Creates a flexible modeling from the template used on the current report.
Impact of template modifications	List of reports affected by the template's update.
Apply a skin template	Used to select the skin template that will be applied to the current report
Controls and blocks of the report	Returns the list of controls and blocks found in the report

Control menu

Option	Action
Description	Displays the description window of the selected controls
Edit the text	Used to modify the text of the selected control from the editor
Choose a style	Used to choose a style for the selected control.
Table	Used to manage the table found in the report
Groups of controls	Displays the description window for the groups of controls
Alignment	Used to manage the alignment options for the selected controls





Z-order	Used to manage the overlapping effects of controls
Bold	Enables (or not) the bold attribute for the text displayed in the selected controls
Italic	Enables (or not) the italic attribute for the text displayed in the selected controls
Underline	Enables (or not) the underline attribute for the text displayed in the selected controls
Border	Used to draw a border around the selected controls
Align to right	Aligns the text to the right in the selected controls
Align to left	Aligns the text to the left in the selected controls
Center	Centers the text in the selected controls
Justify	Justifies the text in the selected controls
Adapt the size	Adapts the size of the selected element to its content
Subscription	Displays the characteristics of the subscription for the selected control

Code menu

Option	Action
Run the report test	Used to run the test of the current report
End the test	Used to stop the current test
Report code	Displays the processes associated with the report in a code window
Code of the selection	Displays the processes associated with the selected control in a code window
Local procedures	Displays the local procedures of the report in the "Code" pane
All the report codes	Displays all the processes associated with the report, with its blocks and with its controls
Compile the report	Used to compile the report

Display menu

Option	Action
Show the report in 3D	Used to view the report in 3 dimensions
Language displayed	Enables you to select the language that will be used when the report is displayed in the editor
Zoom	Performs a zoom in the report editor according to the specified zoom factor.
Toolbars	Configures the toolbars displayed and the general interface.
Ruler	Enables or disables the display of alignment rulers
Grid enabled	Enables the snap-on grid
Grid visible	Makes the snap-on grid visible



Options	Configures the display of the report currently created in the editor
Description of the selection	Displays the description window of the selected controls





1.5 The main toolbars of the report editor

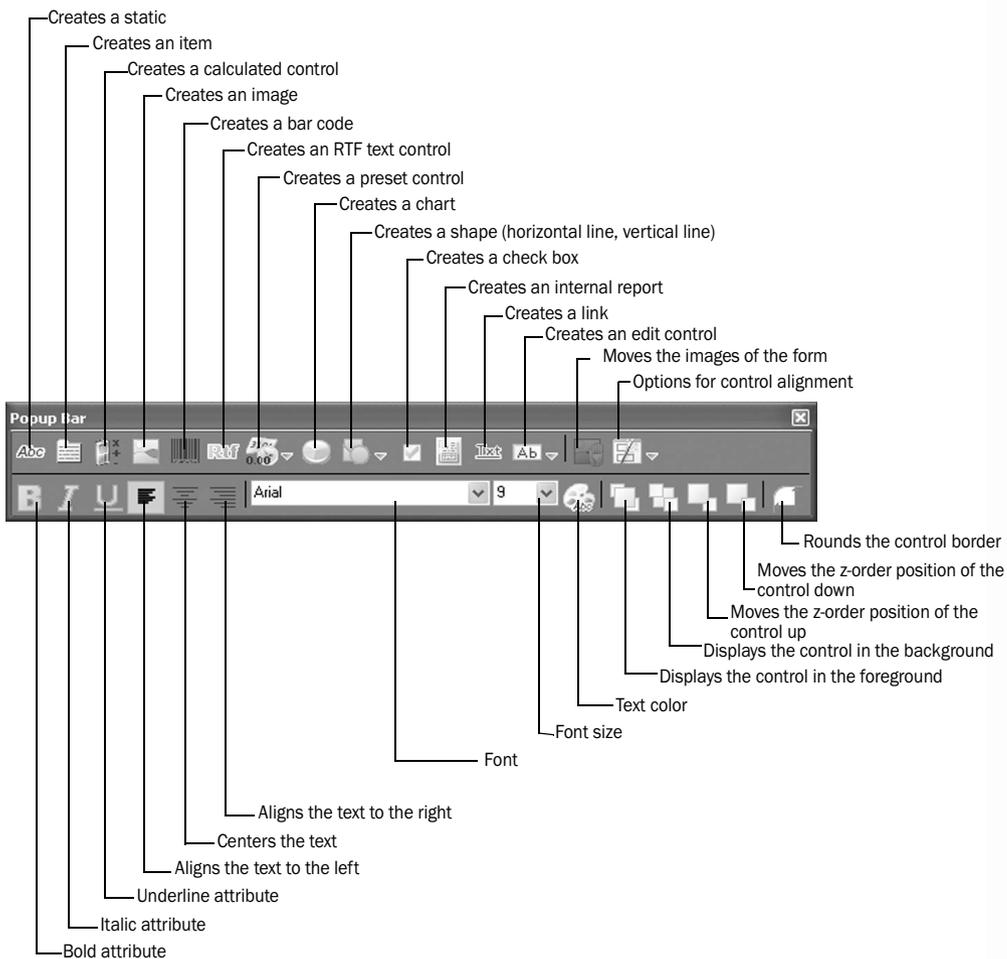
Several toolbars are displayed in the report editor:

- the toolbar of controls.
- the toolbar for aligning controls.
- the toolbar for the border of controls.

To display or hide these toolbars, select "Display .. Toolbars..".

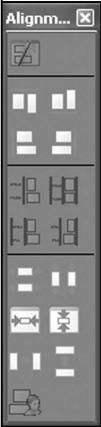
1.5.1 Toolbar of controls

This toolbar is used to create the different types of controls in a report and to configure the alignment of these controls.



1.5.2 Toolbar for aligning controls

This toolbar is used to easily define the position of the controls in a report.



	Interface checker	
Aligns the top borders		Aligns the bottom borders
Aligns the left borders		Aligns the right borders
Aligns the right inside borders		Justifies the inside and outside borders
Justifies the inside and outside borders to the left		Justifies the inside and outside borders to the right
	Adjusts the width	Adjusts the height
Centers horizontally		Centers vertically
Adjusts the horizontal spacing		Adjusts the vertical spacing
Supports the custom alignment		

1.5.3 Toolbar for the border of controls

This toolbar is used to draw borders around the controls in a report.



No border		
		Frames the control
Left border		
		Top border
Right border		
		Bottom border

1.5.4 Keyboard shortcuts

The main shortcuts that can be used in the report editor are as follows:

Action	Shortcut
Display the description of the control or cell	Alt + Return
Display the list of controls found in the current report	F12
Display the list of classes in the "Project Explorer" pane	Shift + F8
Display the list of global procedures in the "Project Explorer" pane	CTRL + F8
Display the list of local procedures in the "Project Explorer" pane	F8
Display the full name of the blocks / Display the short name of the blocks	F10
Display/Hide the border of the controls	F7
Display/Hide the name of the controls	F6
Display/Hide the margins of the report	F11





Display/Hide the rulers	CTRL + R
Help	F1
Undo the last action	CTRL + Z
Stop the test mode (when running the project test or the report test)	Shift + F5
Paste the content of the clipboard	CTRL + V or Shift + Ins
Copy the style of the selected element	CTRL + Shift + C
Copy the selected element	CTRL + C or Shift + Ins
Cut the selected element	CTRL + X or Alt + Del
Move the selected control or column to the right/left	right/left arrow
Display the code associated with the selected element in the "Context-sensitive view of code" pane	CTRL + Shift + F2
Edit the code of the selection	F2
Save the report	CTRL + S
Save as	CTRL + Shift + S
Save all	Alt + S
Close all the opened elements	CTRL + Shift + W
Print the documentation regarding the current report	CTRL + P
Hide the horizontal panes	CTRL + Q
Hide all (vertical and horizontal panes)	CTRL + W
Assign the bold attribute to the selected control	CTRL + G
Assign the italic attribute to the selected control	CTRL + I
New (Create an element)	CTRL + N
Open (Open an element)	CTRL + O
Open a project	CTRL + Shift + O
Restore (Undo the last action)	CTRL + Y or CTRL + Shift + Z
Underline the selected control	CTRL + U
Delete the selection	Del
Trace the project	Alt + F9
Run the test of the project	CTRL + F9
Run the test of the current report	F9
Show the report in 3D	Alt + D
Zoom at 100 %	CTRL + Alt + 0
Zoom + / -	CTRL + mouse wheel

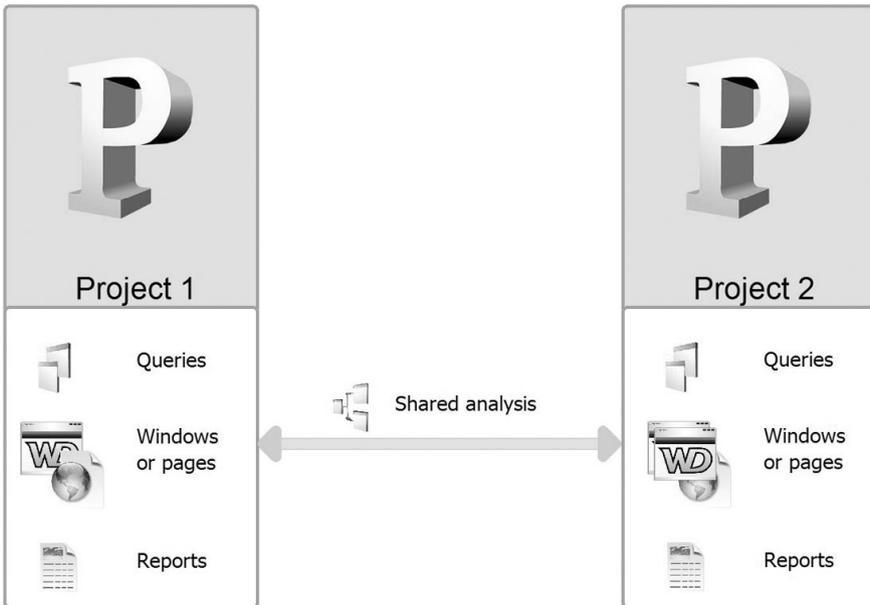
Caution: "+" means that the keys must be pressed simultaneously



2. HOW DO I CREATE A REPORT ?

2.1 Report and project

In most cases, a report is linked to a project. Indeed, to be able to check out 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 the project, select "File .. Open a project".

Notes:

- There is no need to open a project if the information to edit is not found in a data file (data found in a text file or in variables for instance).
- To create, modify or run a report, the analysis description must be installed on the computer or on the server (".WDD" file). This file is found in the runtime directory (deployment directory of the site in WebDev) or in an accessible directory.





2.2 Creating a report

2.2.1 Overview

To create a new report, select "File .. New .. Report". The wizard for report creation starts.

To create a report, you must:

1. Choose the report's type,
2. Select the data source for the report,
3. Describe the elements found in the report (controls, sort and selection conditions, etc.),
4. Generate and save the report corresponding to the selected options.

These steps are performed in the wizard for report creation. A report corresponding to the selected options is automatically created at the end of the 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.

The report can be run:

- from the report editor,
- from a WinDev or WinDev Mobile application,
- from a WebDev site.

2.2.2 The different types of reports

Several types of reports can be created in the report editor. All of them are available in the user version of the report editor and in the standard report editor.

The following types of reports are available:

- Table report
- Form report
- Label report
- Custom mailshot report (not available in WinDev Mobile)
- Blank report
- Report based on form
- Multi-column report
- Table + Chart report (not available in WinDev Mobile)
- Crosstab report

• Table report:

Reading the data source of the report

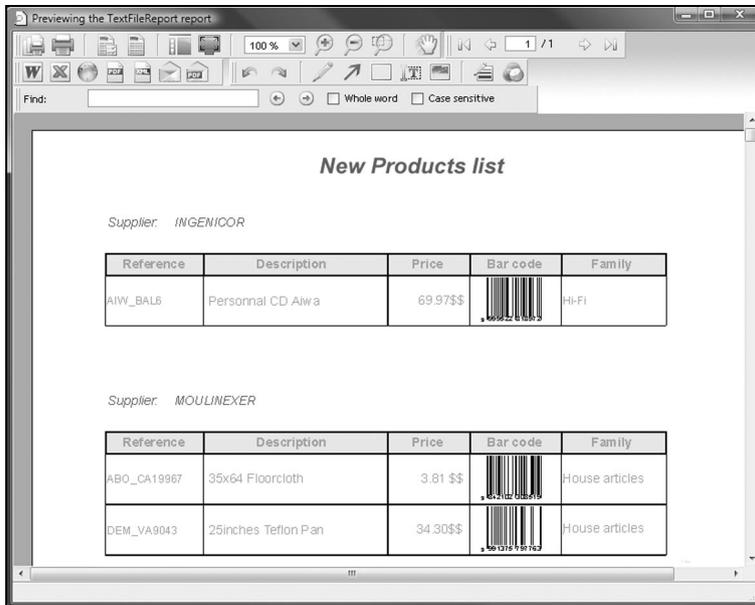
Each table row corresponds to a record read in the data source of the report (file, query, and so on).



Simple examples

- Summary table
- Invoice
- Detailed list
- Data grouped according to a criterion

See “Using tables in a report”, page 99, for more details.



• Form Report:

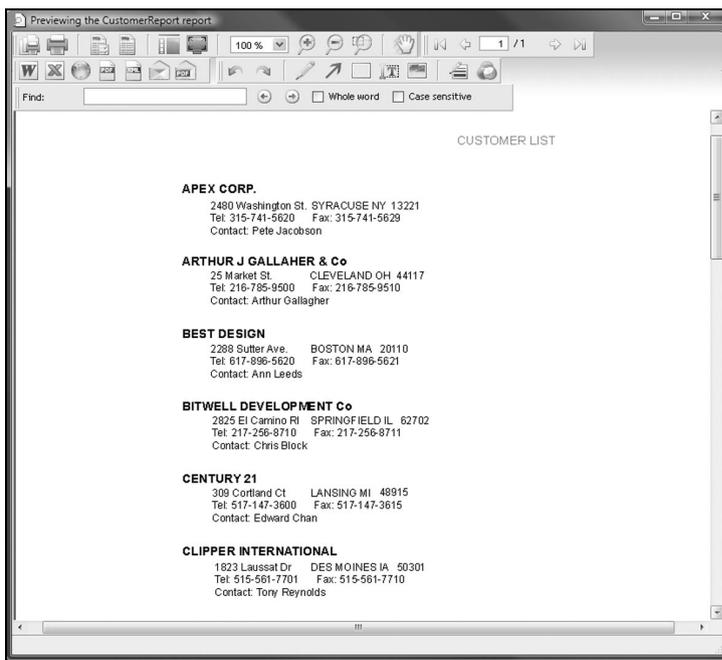
Reading the data source of the report

The data read in the data source is displayed in a form: each report page displays the data of several records.

Use examples

- Detailed list
- Data grouped according to a criterion





• Label report:

Reading the data source of the report

Each record read in the data source corresponds to a label.

Use examples

- Price labels
- Address labels

Information specific to the creation of labels

Specific information is required to create labels:

- the number of identical copies that will be printed for each label.
- the format of labels and/or the format of the label sheet. This format can correspond:

- a *custom format*:

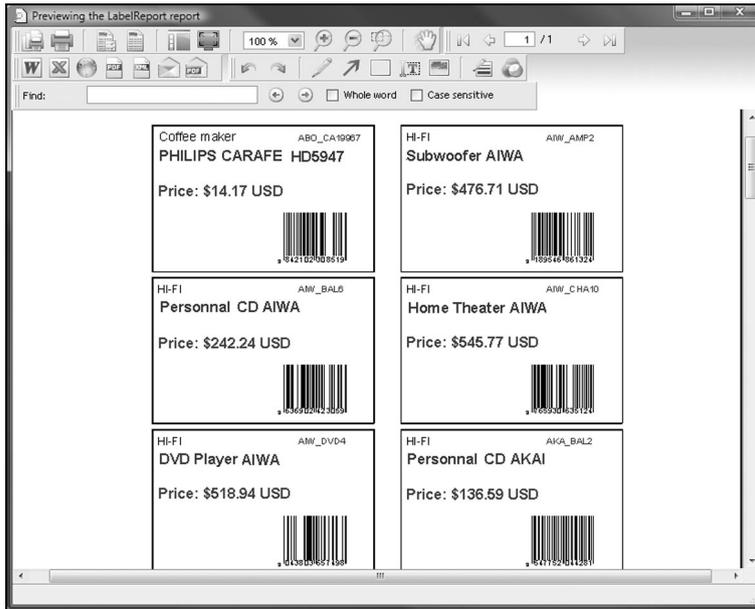
1. Select the "<Custom>" element in the "Manufacturer" combo box.
2. Specify the format of the label sheet on which the labels will be printed.
3. Specify the format of labels.
4. Specify the number of labels that will be printed across the page width.

- a *preset format*:

1. Select the brand of the labels used in the "Manufacturer" combo box.
2. Select the model of the labels used in the "Model" combo box.
3. The information specific to the labels used (caption, size of labels and number of labels printed across the page width) is displayed.

Note: Check "inches" to display the size of the labels in inches.





- Custom mailshot report (not available in WinDev Mobile):

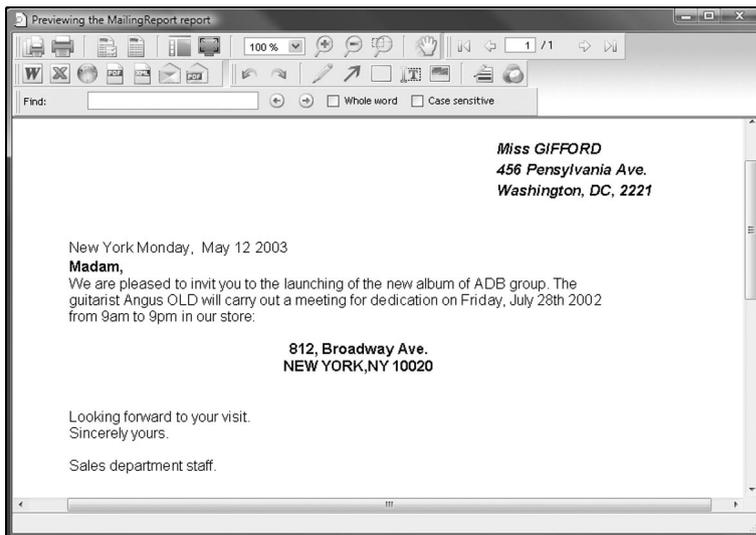
Reading the data source of the report

Each record read in the data source corresponds to a page.

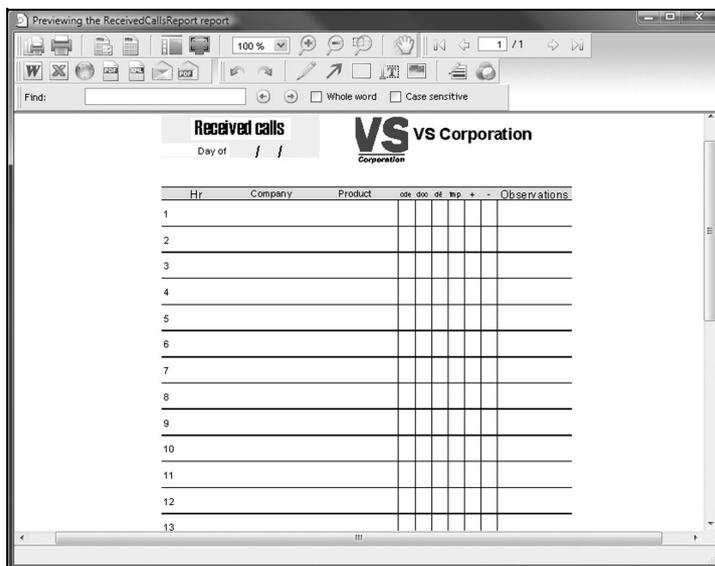
Use examples

- Mailshot
- Advertising letter
- Reminder
- Email





- Blank report (or free report):



Reading the data source of the report

Each page displays:

- a specific text.
- a standard form (filled manually afterwards).
- the data coming from one or more records.



Use examples

- Form for entering incoming calls
- Form for entering the absenteeism
- Form for entering the products received

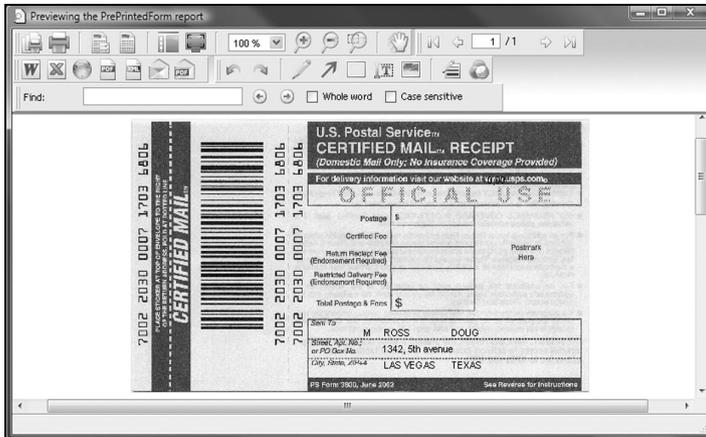
• Report based on form:

Reading the data source of the report

The data read in the data source is displayed in a form: each page displays the data of one or more records.

Use examples

- Postal forms
- Check
- Tax form



Information specific to the form

Specific information is required to create a report with form:

- the file containing the image of the form. The size of the report is automatically calculated according to the size of the form image.

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 form's image must be printed or not.

Note: If the form's image is printed in the report, it 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:
 - 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.
 - if the form contains the characteristics of several customers, several records must be printed per page.
In this case, the created report will contain 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 “Including a form in a report”, page 60, for more details.

• Table + Chart report (not available in WinDev Mobile):

Reading the data source of the report

The data read in the data source is displayed:

- in a table. Each table row corresponds to a record read.
- in a chart.

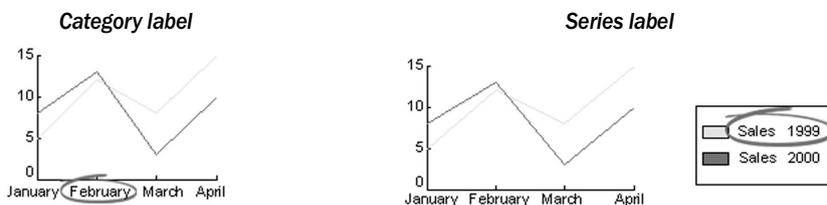
Use examples

- Table + Chart representing the expenses per category.
- Table + Chart representing the breakdown of gross revenue per product.

Information specific to the creation of the chart

Specific information is required to create a chart:

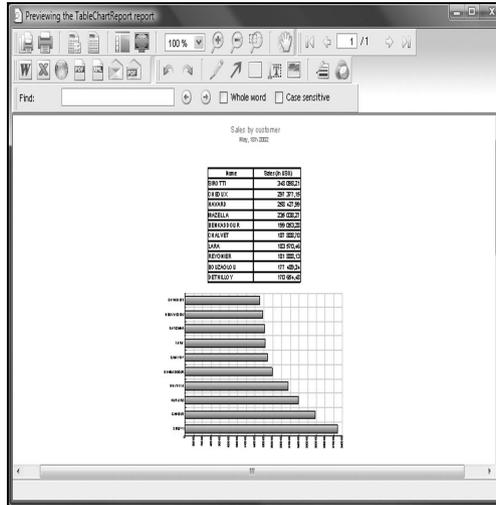
- the type of chart (stock, pie, line, etc.).
- the values displayed in the chart. The values come from the items associated with the report.
- the caption of the value. This caption comes from an item associated with the report. This caption corresponds to:
 - the label of a data category (or section).
 - the label of a data series.



- the frequency of the marks on the vertical/horizontal axis of the chart (if necessary).
- the style of the chart (relief, color gradients and gridlines).
- the titles of the axes if necessary.



Chart controls can also be created in an existing report. See “Chart control”, page 81, for more details.



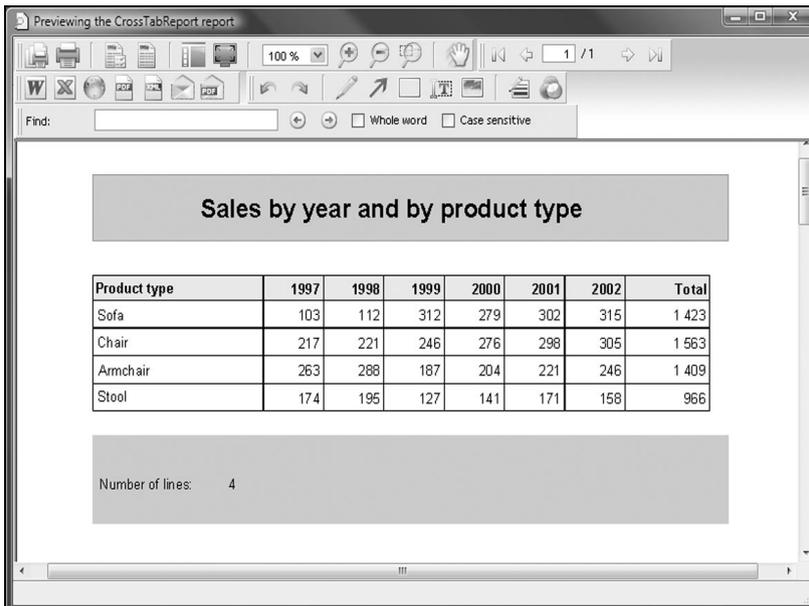
- **Crosstab report:**

Reading the data source of the report

The data read in the data source is displayed in a double-entry table.

Use example

- Table representing the gross revenue per product and per month (or per year).





2.2.3 Source of the report

The data source of the report defines the origin of the data printed in the report. Several data sources are available:

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



Important

The files and the file items that can be used in the reports created by Reports & Queries are defined by the developer in the data model editor of WinDev. For more details about the preparation needed for using a WinDev application with Reports & Queries, (See “Distributing Reports & Queries”, page 231).

- **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 the report was created.
- **Table control found in a window or in a page:** The data to print comes from a table control found in one of the application’s windows or in one of the site’s pages. The records are selected by the table and they are printed by the report.



Important

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

- **Existing variable:** the data to print comes from a WLanguage variable of the project or the report. This variable can be an array variable global to the project, an array variable global to a window or a page, an array variable global to a set of procedures.



Important

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

- **Other source:** The other data sources can be:
 - **Text file:** The data to print is read from a Text file.
 - **Memory zone** (not available in WinDev Mobile): the data to print corresponds to the records stored in a memory zone.



Important

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

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



Important

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



- **Programming the reading of the data source:** to program and select yourself the records to display in the report, choose a report based on a programmed data source. See 'Report based on a programmed data source', page 53, for more details.



Important

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

- **No data source:** The report will not be linked to any data source.

These data sources are presented in the "Data sources of a report", page 47.

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 the data source.

2.2.4 Breaks and sorts

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

- If the report is based on a data file, the search items of the 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 breaks can be defined: sorts must be done 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 the query is created. 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 can also use:

- WLanguage functions (*Left*, *Right*, and so on).
- 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 the project or a local procedure of the report. See "Sort procedure", page 142, for more details.

See the online help (keyword: "Sort, Report") for more details about sorting records.



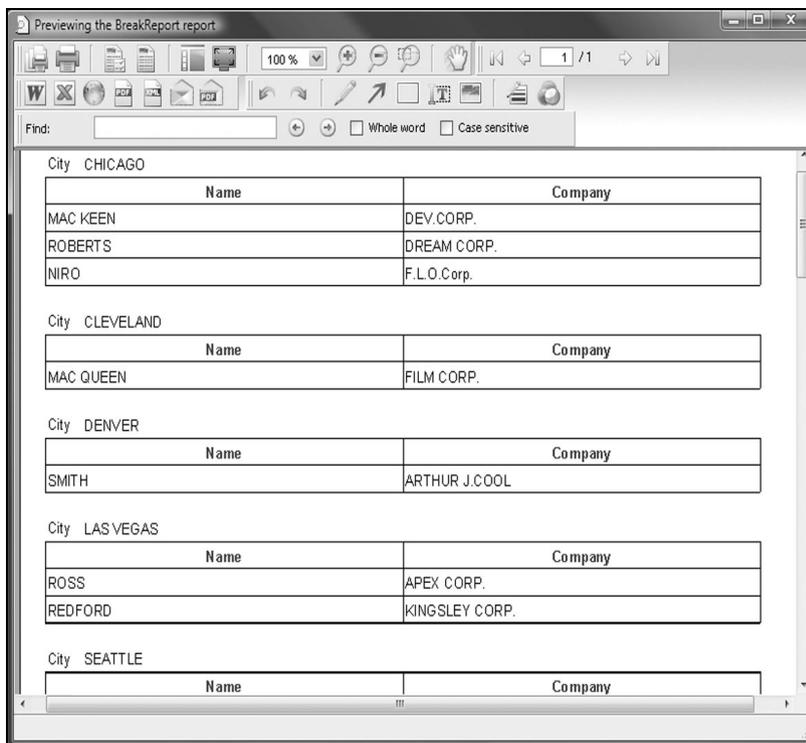


The breaks

The break is used to group the records corresponding to a sort criterion.

For example:

- to list customers by city, simply sort the customer by city then by name and perform a break on the city.
- to list the stock price per day, simply sort the stocks by date and create a break on the date.



Adding a break automatically generates a *Break Header* block and a *Break Footer* block. These blocks are used to:

- display the content of the break item (in the *Break Header* block).
- perform calculations on the records found in the break (in the *Break Footer* block).

See “Report blocks”, page 65, for more details.

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 the report.
- **programmed break**. The break is defined by programming in the “Break?” process of the *Break Header* block. See “Break process”, page 142, for more details about programming breaks.

See the online help (keyword: “Break, Report break”) for more details about breaks.



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 calculations that must be performed in a report are selected when the report is created. However, new calculations can be added to the created report via the calculated controls. See “Calculated control”, page 77, for more details.

2.2.6 Skin template

If the project is associated with a skin template, this skin template can be selected for your reports. Your reports and your application will have the same style. This skin template can be modified once the report has been created. See “Changing the skin template associated with the report”, page 63, for more details.

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

2.2.7 Printout format of the report

The format of a report is used to configure:

- the orientation of the report
- the paper format
- the printout margins
- the number of columns of the report. By default, a report has one column. If the report has more than one column, the report is said to be "multi-column".

For a "Label" report, you can also configure:

- the format of the label sheet
- the spacing between the labels

This information can be modified once the report is created. See “Describing and handling a report”, page 55, for more details.

2.3 Saving the report

To save a report, select "File .. Save". 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 the executable is created (WinDev/WinDev Mobile) or when creating the site's library (WebDev).

2.4 Running the report test

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

Note: the report test can only be run on the development computer. The data used for test correspond to the data found on the development computer.

The following processes are run during the test of a report:

- initialization process of the project.
- processes associated with the report.
- processes associated with the report blocks.
- processes associated with the report controls.



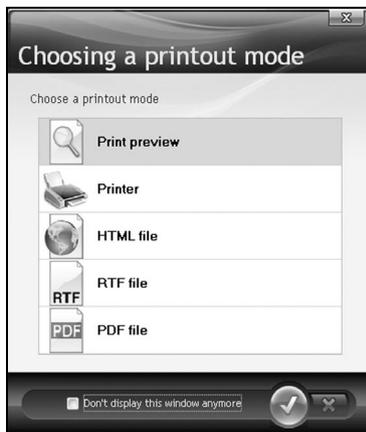


Note: The report test is available for the reports associated with no data source and for the reports based on a query, a data file or a text file. For the other types of reports, you must run the test of the window (or project) from which the report is printed.

2.4.1 Run the test

To run the test of the current report:

1. Click  or select "Code .. Run the report test" (F9). A screen allows you to choose the printout mode of the report:



The available options are:

- **Print preview:** used to preview the report.
- **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 a RTF file and displayed in a text editor (Word for instance).
- **PDF file:** the report is saved in a PDF file and displayed in a PDF reader program (Adobe Acrobat for instance).

See "Printing a report", page 43, for more details about these various modes.

2. Choose the printout mode and validate. If your report is not saved, WinDev, WinDev Mobile or WebDev automatically proposes to save it under the name given in the wizard for report creation.

3. Specify the additional information if necessary:

- for a **report with bounds**, a window allows you to enter the report bounds. See "Bounded report", page 48, for more details.
- for a **report based on a query with parameters**, a window allows you to enter the query parameters. See "Report based on a query with parameters", page 49, for more details.
- for a **report with parameters**, give a default value to the parameters in their declaration. See "Report with parameters", page 143, for more details.



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 via the taskbar (or press [Alt]+[Tab]) and confirm that the test must be stopped. The current editor at the time the test was started is displayed.
- **3rd method:** Go back to the editor via the taskbar (or press [Alt]+[Tab]) and click  found in the icon bar.

2.4.3 Directory of the data files used for test

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

1. Display the project description ("Project .. Project description").
2. In the "Files" tab, select the location of the test files:
 - **EXE directory of the project or current directory** (option selected by default): the data files will be located in the project's EXE subdirectory.
 - **Specific directory:** The data files will be found in the specified directory.
 - **Directory of the application's data:** this choice is recommended in Windows Vista. The "local" data files will be found in a directory accessible and modifiable by the application (without specific rights).

2.4.4 Managing the 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 files, it must be specified with **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 project's code

When running the test of a report, a WLanguage error occurs if a window is opened in the initialization code of the project or in the end code of the project. To avoid this error, **OnTestMode** must be used to configure the window opening. For example:

```
IF OnTestMode = False THEN Open(MyWindow)
```

However, you can use the functions for opening the system information boxes (**Info** or **Trace** for instance).





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 while running a report test in the editor	Printing a report from the preview window (WinDev only) *	Printing a report by programming (iPreview or iDestination and iPrintReport)
<i>Paper printout</i>	X	X	X
<i>Print preview</i>	X		X (WinDev only)
<i>HTML file (with style sheet)</i>	X	X	X
<i>HTML file (without style sheet)</i>			X
<i>RTF file</i>	X	X	X
<i>XML file</i>		X	X
PDF file	X	X	X
<i>Text file</i>			X
Sending a fax			X
PCL file			X
XLS file		X	X

* The destinations offered by print preview can be modified in the "Details" tab of the report description or by programming using **iParameter**.

See the online help for more details.

WinDev Mobile: All the options are available when running the test of a report in the editor. However, when printing a report by programming, the printouts can only be performed on a PCL printer or to a PCL file.

Note: By default, the print mode of the report is automatically asked when running the test of each report in a specific window. To avoid displaying this window when running the report test, uncheck "Ask for the print mode" in the edit options of the report editor ("Display .. Options .. Modify the options", "Printout" tab). In this case, the report being tested will be shown in the print preview window.

2.5.2 Print preview

The print preview is used to preview the layout of a report without printing it. The layout and the content of the report can be checked. You also have the ability to print the report.

Note: In WebDev and WinDev Mobile, the print preview is available in test mode only.

Several features are proposed by the print preview:

- Customizable zoom, and display of the report in full screen mode.
- Display the report pages as thumbnails.
- Display the total number of pages to print.
- Search in all the pages.
- Print the current page or all the pages.
- Direct printout.
- Choose the print destination via a button bar.

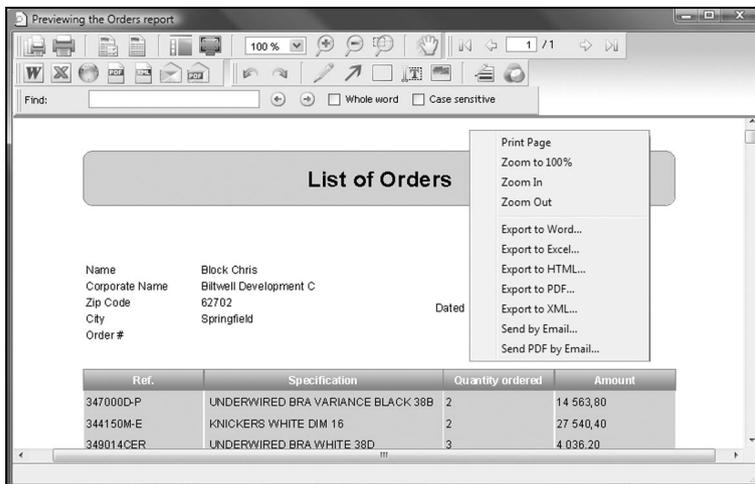


- Ability to annotate the displayed report. If the report is printed from the preview window, the annotations are also printed.

The button bar of the print preview

The button bar allows you to perform the following operations:

- Save and preview the report in RTF format (Rich Text Format).
- Save and preview the report in XLS format.
- Save and preview the report in HTML format.
- Save and preview the report in PDF format (Portable Document Format).
- Save and preview the report in XML format (eXtensible Markup Language).
- Preview the report in an email.
- Add a PDF file to an email.



A popup menu that includes most of the options found in the toolbar and in the button bar. See the online help for more details.

Notes:

- To preview a report, a printer must be installed on the current computer.
- Print preview performed with the WLanguage functions: If you print with the WLanguage functions, you can also benefit from the features of the print preview. The toolbar will be displayed and annotations can be made.
- To only display some of the buttons in the button bar of the print preview:
 - use ***iParameterPreview***. This function (used in the initialization code of the project) enables you to choose the buttons that will be displayed for all the application reports.
 - configure the buttons that will be displayed for each report in the report description ("Details" tab).
- When using the button bar, the print preview is automatically closed once the file is saved.



Important

If the current report can be modified by the end user, the following icons allow him to:

- modify the current report with: .
- create a new report with: .





The annotation bar

The annotation bar lets you perform the following operations:

- Draw using a brush (to circle some information for instance).
- Insert arrows.
- Insert frames (to frame an area for instance).
- Insert text (to add comments for instance).
- Insert images.

You can choose the thickness of the line as well as the color used, of course.

The search bar

The search bar lets you search for a word or a set of words in the report. The result of the search is highlighted in green.

Other features of the print preview (WinDev only)

- **Customizing the preview:** Your application uses a skin template? Why not use this skin template for the preview window. WinDev gives you the ability to customize the preview window. See the online help (keyword: "Customizing, Preview window") for more details.
- **Previewing a printout performed with the WLanguage functions:** The features of the print preview are also available when the printout is performed with the WLanguage functions.
- **Customizing the button bar:**
To display some of the buttons only:
 - use *iParameterPreview*. This function (used in the initialization code of the project) enables you to choose the buttons that will be displayed for all the application reports.
 - configure the buttons that will be displayed for each report in the report description ("Details" tab).
To configure the options of the different print modes, use:
 - *iParameterPDF* to configure the printing in a PDF file,
 - *iParameterExport* to configure the sending of emails,
 - *iParameterXLS* to configure printing in an XLS file.
- **Printing from the print preview:** To find out whether a printout was performed from the print preview, use *iDocumentPrinted*.
- **When using the button bar,** the print preview window is automatically closed after the document is printed in a file.







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 based on a data file”, page 47
- “Report based on a query”, page 48
- “Report based on a variable”, page 50
- “Report based on a HyperFileSQL Classic view”, page 50
- “Report based on a memory zone”, page 51
- “Report based on a Table control”, page 52
- “Report based on a text file”, page 53
- “Report based on a programmed data source”, page 54

3.2 Report based 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 (HyperFileSQL Classic or Client/Server, accessed via a Native Access, etc.). If the records come from several files, we recommend that you use a report based on a query (see next paragraph).

The creation of a report based on a data file is done through the report creation wizard (“File .. New .. Report”). This type of report can be created only if the current project is associated with an analysis.

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

- modify the data source of the report (“Data” tab of the report description).
- modify the search key (“Data” tab in the report description).
- define bounds (“Data” tab in the report description). These bounds are used to define the range of records that will be printed in the report
- add or delete Item controls linked to the data file associated with the report.

“Report based on a data file”, page 121, for more details.



Important

The data files and the file items that can be used in Reports & Queries are defined in the data model editor of WinDev. For more details about the preparation required for using a WinDev application with Reports & Queries, See “Distributing Reports & Queries with your WinDev application”, page 231.



3.2.2 Characteristics

Search key

A report based on a data file uses 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 based on a data file can be bounded on the search key used. For example, a report based on the "Customer" file can print all the customers or only the customers whose name starts with the letter "C" or "E" if the report is bounded.

See the online help (keyword: "Bound, Bounds of a report") for more details.

3.3 Report based on a query

3.3.1 Overview

A query enables you to easily select some records in one or more data files. A query can also be used to perform calculations on the selected data. You can, for instance, select all the orders placed by the customers living in Texas and calculate the total amount of orders per customer.

Using queries in a report enables you to easily create complex reports. The query preselects all the records. The report organizes the selected data and performs the calculations.

Caution: To optimize the printout time, the calculations (totals on items for instance) should be performed in the report (and not in the query).

Several modes are available to create a report based on a query:

- the query is created and saved in the query editor. The report is then based on the query file (".WDR" file). The query can be re-used in another report. The query can be modified at any time in the query editor. **The query is an independent query.**
- the query is created during the report creation. 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 reused in another report. **The query is an embedded query.**
- **the query is based on the data coming from a connection** (not available in WinDev Mobile). This type of report based on a query is kept for backward compatibility. See the online help for more details.
Note: This type of report based on a query is available even if no analysis is associated with the WinDev project.

Tip: In the query editor, the icon  can be used to automatically create a report based on the current Select query.

See "Report based on a query", page 126, for more details about printing a report based on a query.





3.3.2 The different types of queries

Report based on an independent query

The query must be created with the query editor before the report is created (See “How do I create a query ?”, page 170 for more details). 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 instance), a synchronization screen is displayed when the query is closed. Therefore, the modifications made to the query can be applied to all the elements of the project that uses this query (including reports). See “Synchronizing the query”, page 204, for more details.

Report based on an embedded query

The query is created during the report creation. This query will be modified from the report editor (“Report .. Description .. Data .. Edit the query”). 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 (i.e. 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 the report (or select “Report .. Edit the query”). The query is displayed in the query editor.
3. Select “File .. Export .. To an independent query”.
4. Save the new query.

Report based on a query with parameters

An independent query or an embedded query can accept parameters. In this case, one or several selection conditions expect a parameter when the query is run. This parameter can correspond to the value of an edit control found in an application window or in a page site.

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

- when running the test of the report based on a query with parameters, a window (or a page) allows you to enter the parameters of the query.
- at run time, the parameters of the query must be specified in *ilnitReportQuery* (before using *iPrintReport*).



Important

If the query is created in Reports & Queries, there is no need to use *ilnitReportQuery* to specify the parameters of the query. Indeed, when a report based on a query with parameters is run from Reports & Queries, a window comes up where the parameters of the query can be entered.

See “Printing a report based on a query with parameters”, page 127, for more details.

Note: See “Report with parameters”, page 143, for more details.



3.4 Report based on a variable

A report based on a variable lets you directly print data coming from a WLanguage variable. This variable can be:

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

Once the report based on a variable is created, you can:

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



Important

Reports & Queries cannot be used to create or customize reports based on variables.

See "Printing a report based on a variable", page 130, for more details about printing a report based on a WLanguage variable.

3.5 Report based on a HyperFileSQL Classic view

3.5.1 Overview

A report based on a HyperFileSQL view is used to print the records coming from a HyperFileSQL Classic or Client/Server view. **This type of report is kept for backward compatibility.**

Note: A HyperFileSQL view corresponds to a "memory image" of all or part of a data file. A view is stored in memory, which insulates it from the modifications made to the associated file. Once created (**HCreateView**), a view can be handled as any HyperFileSQL data file. See the online help (keyword: "HyperFileSQL view") for more details.



Important

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

Once the report based on a HyperFileSQL view is created, you can:

- modify the data source of the report ("Data" tab of the report description).
- add or delete Item controls linked to the HyperFileSQL view associated with the report.

See "Report based on a HyperFileSQL Classic view", page 132, for more details.

3.5.2 Characteristics

Base File

A report based on a HyperFileSQL view is linked to the base file of this HyperFileSQL view. The report is used to print the records coming from this base file.





Order for printing the records

The order for printing the records corresponds to the order of the records defined in the HyperFileSQL view (when the view was created or in *HSortView*).

3.6 Report based on a memory zone

3.6.1 Overview

A report based on a memory zone is used to print the records stored in a memory zone (created by *Mem-Create*). The memory zone must necessarily contains all the numbers of the file records to print.

You can for instance:

- store in a memory zone the numbers of the records selected by one or more processes.
- print the records only when the selection is performed.

To run the test of a report based on a memory zone, you must:

- create and initialize the memory zone.
- print the report.

Once the report based on a memory zone is created, you can:

- modify the data source of the report ("Data" tab of the report description).
- add or delete Item controls linked to the memory zone associated with the report.



Important

Reports & Queries cannot be used to create reports based on a memory zone. However, you will be able to customize the reports based on a memory zone created in the report editor (color, font, size, etc.).

See "Report based on a memory zone", page 134, for more details.

Notes:

- This type of report is not available in WinDev Mobile.
- *This type of report is kept for backward compatibility.*

3.6.2 Characteristics

Base File

A report based on a memory zone is linked to the base file of this memory zone. The report is used to print the records coming from this base file.

Order for printing the records

The order for printing the records corresponds to the order of the record numbers defined in the memory zone.



3.7 Report based on a Table control

3.7.1 Overview

A report based on a Table control is used to directly print the information displayed in a browsing table or in a memory table found in an application window or in a site page.



Important

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

A report based on a Table control is used to print:

- all the rows displayed in the table (the table data can be filtered using the magnifier).
- the rows selected by the user (multi-selection tables where the rows are selected via the [CTRL] key). This option is not available in WebDev.

A report based on a Table control can be created:

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

See the online help (keyword: "Table, Report based on a table control") for more details.

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

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

See "Report based on a table control", page 128, for more details.

3.7.2 Characteristics

Correspondence between records

If the report is based on a Table control:

- Each row of the Table control corresponds to a record.
- Each column of the Table control corresponds to an item control (even invisible columns).

For a table report, the order in which the columns are displayed in the report corresponds by default to the order of the columns in the window (or in the page) being edited.

In WinDev, if the user modifies the order of the columns in the window, the order in which the columns are displayed in the report is not modified dynamically.





LastName	City	Company
MAC QUEEN	CLEVELAND	FILM CORP.
ROSS	LAS VEGAS	APEX CORP.
SMITH	DENVER	ARTHUR J. COOL
MONTGOMERY	SEATTLE	BEST DESIGN
MAC KEEN	CHICAGO	DEV. CORP.

Previewing the RPT:tableCustomer report

Customer List

Name	City	Company
MAC QUEEN	CLEVELAND	FILM CORP.
ROSS	LAS VEGAS	APEX C CORP.
SMITH	DENVER	ARTHUR J. COOL
MONTGOMERY	SEATTLE	BEST DESIGN
MAC KEEN	CHICAGO	DEV. CORP.

Running the test of a report based on a Table control

The test of a report based on a table cannot be run directly from the report editor; indeed, the window or the page containing the table must be opened. Therefore, you must:

- program the printout of a report based on a table. See "Report based on a table control", page 128, for more details.
- run the test of the window/page (or project) used to start the printout.

3.8 Report based on a text file

3.8.1 Overview

In a report based on a text file, the information to print is read from a file in text format and directly assigned to the report controls.

This type of report is very useful when the information to print was imported into a file in text format (data imported from Excel for instance).

To create a report based on a text file, you can:

- use the wizard for report creation on a text file. The report editor automatically creates the report and reads the records. No programming is required.
- program the reading of the records. The wizard for report creation greatly simplifies the creation of the report. However, reading records from a text file requires WLanguage programming.

See the online help (keyword: "Text file, Report based on a text file") for more details.

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

- modify the data source of the report ("Data" tab of the report description).
- modify the method for reading the records found in the report (automatic or programmed). See the online help (keyword: "Text file, Report based on a text file") for more details.
- add or delete Item controls linked to the text file associated with the report.

See "Report based on a text file", page 136, for more details.

This list:

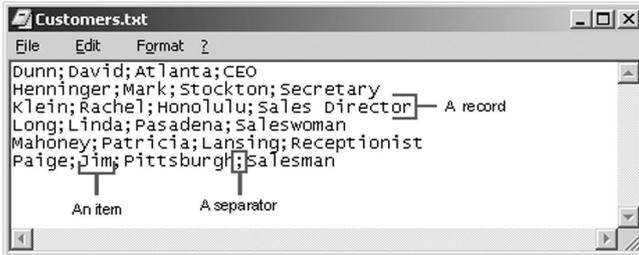
- contains **three items**: "Name", "Company" and "City".
- contains **as many records** as customers.



3.8.2 Characteristics

Correspondence between records

Each line found in the text file must correspond to a record. Each row contains various information, called items. The items are separated by a separator.



3.9 Report based on a programmed data source

In a report based on a programmed data source, the reading of the records that will be displayed in the report requires a full programming in WLanguage. The data source can have any type (HyperFileSQL file, query, text file, file from a third-party database, etc.).



Important

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

See the online help (keyword: "Data source of a report, Programmed data source") for more details.

When the report based on a programmed data source is created, you can:

- modify the data source of the report ("Data" tab of the report description).
- add or delete Item controls linked to the data source of the report.





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 was created, the description of this report can be modified at any time:

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

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

This chapter presents:

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

4.2 Viewing and modifying the report description

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

See the context-sensitive help of the description window for more details (, in the top right corner).

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

Note: the main options of the report are also accessible 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 can:

- modify the data source of the report ("Data" tab of the report description):
 - by changing the type of the 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 instance).
 - by changing the parameters of the selected data source (modify the query or the sort, implement bounds, and so on).
- configure the operating mode of the report if the data source is empty ("Details" tab of the 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, "Report .. Edit the query" is used to directly edit the query in the query editor. This query can then be modified.



4.2.2 Display options in the print preview

The configuration of the display options for print preview is done in the "Details" tab of the report's description. You can:

- 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.
- specify whether the report can be modified in "Reports and Queries" and under which conditions. See "Using Reports & Queries", page 236 for more details.
- specified whether the compiled code should be included. This option is only needed in very rare cases.
- specify whether spaces that complete items should be taken into account. This option only applies to auto-sizable controls associated with a text item filled with spaces.
- specify whether printing should be done page by page. In this case, the printout is sent to the print spooler after each page is printed.
- specify 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).

4.2.3 Managing the blocks

The report blocks can be managed in the report in edit or in the description window of the report.

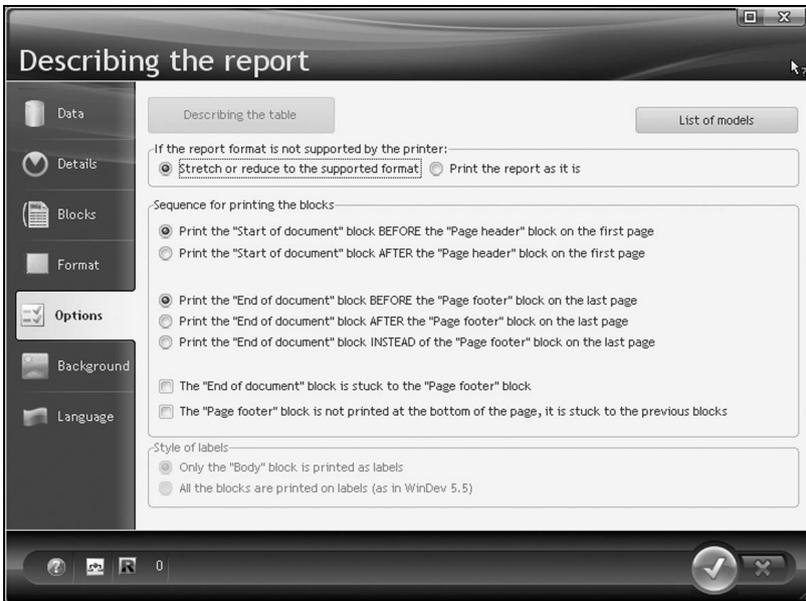
The blocks can be managed:

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

See "Report blocks", page 65, for more details about managing blocks.

4.2.4 Options

The "Options" tab of the description window of the report is as follows:





This tab is used to configure:

- the operating mode of the report if the page format is not supported by the printer.
- la position of the blocks when the report is printed (See "Rearranging the blocks", page 71, for more details).
- the style of the table if the report contains a table (See "Using tables in a report", page 99 for more details).
- the style of labels if the report is a label report.
- the list of templates used by the report.

4.2.5 Form and background image

Most types of reports can be created from a form or printed on top of a background image. The "Background" tab lets you define the image used as a form in the report. This image can be printed or not with the report.

Using a form is very useful for reports that need to 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 58, for mode details.

4.2.6 Language

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

The "Language" tab found in the project description lets you 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 entered in several languages:

- all the messages displayed while the report is printed. These messages are translated in the description windows.
- the caption of static controls or RTF controls: to translate these captions, modify the language displayed in the report ("Display .. Language displayed") and enter the translations in the report.
- 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 can:

- select "File .. Open".
- double-click a "WDE" file.
- double-click the name of a report in the "Project Explorer" pane.

To close the current report, select "File .. Close". If the report was modified, the report editor automatically proposes to save it.

To save the current report, select "File .. Save".

Note: Select "File .. Save as" if you want to rename and save the report.

Automatic retrieval

To avoid problems when a system lock occurs (power outage for instance), WinDev / WebDev / WinDev Mobile can automatically restore the elements opened in the editors ("Tools .. General .. General options of



WinDev/WebDev/WinDev Mobile .. General .. Backup for automatic restore").

When this system is enabled:

- an automatic backup of the open elements is regularly performed at specific time intervals (every 10 minutes by default).
- when a report is reopened after a system lock, an auto-restore window 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 unselected version is available in the window for version management ("File .. Versions") and it can be opened at any time. See "Version management", page 64, for more details.

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 instance, the addresses of the customers can be directly printed on registered mail forms or plane tickets, etc.

In **edit**, the form is displayed as background image of the report: the controls can be easily positioned on the pre-printed form.

For example:

The screenshot shows the Report Editor interface with a U.S. Postal Service Certified Mail Receipt form overlaid. The form includes the following fields and text:

- U.S. Postal Service™ CERTIFIED MAIL™ RECEIPT** (Domestic Mail Only; No Insurance Coverage Provided)
- For delivery information visit our website at www.usps.com.
- OFFICIAL USE**
- Postage \$
- Certified Fee
- Return Receipt Fee (Endorsement Required)
- Restricted Delivery Fee (Endorsement Required)
- Total Postage \$
- Sent To: M JOHN
- Street, Apt. No.:
- City, State, ZIP+4: LAS VEGAS
- PS Form 3800, June 2002
- See Reverse for Instructions

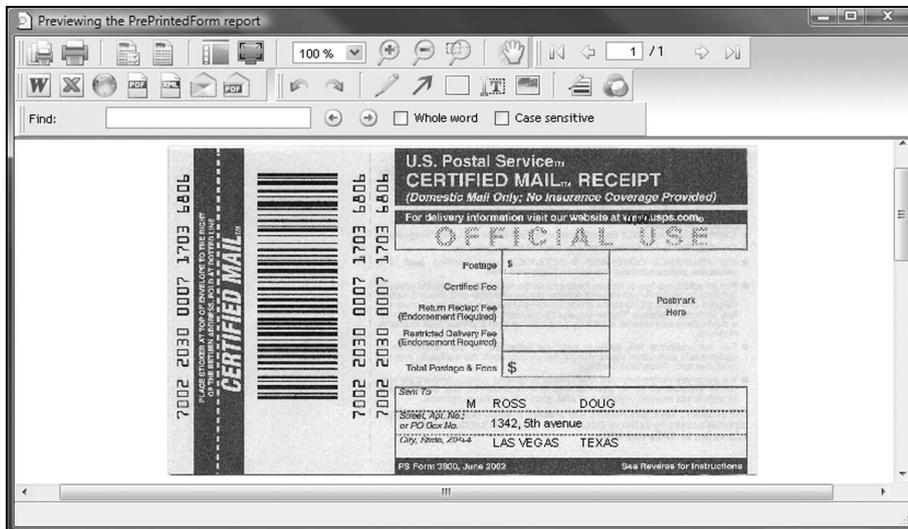
The report editor interface includes a toolbar at the top with various icons for editing and printing, and a status bar at the bottom showing "Reports" and "RPT_PrePrintedForm".





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:



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

- create a "Form" report. See the next paragraph for more details.
- include a form in an existing report.

4.4.1 Information specific to the form

Specific information is required to use a pre-printed form in a report:

- 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 595 x 842 pixels. For A4 format, the size of an image in 150 DPI must be 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 form's image must be printed or not.
Note: If the form's image is printed in the report, it 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:
 - if the forms 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.
 - if the form contains the characteristics of several customers, several records must be printed per page.
In this case, the created report will contain 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.



4.4.2 Including a form in a report

If needed, create 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 the A4 format, the size of an image in 75 DPI must be 595 x 842 pixels.

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

Note: The authorized image formats are: ".BMP", ".JPEG", ".GIF", ".PCD", ".PSD", ".TGA", ".TIF", ".PNG", ".ICO", ".EMF", ".WMF", ".CUR", ".OS2", ".PCX", ".ICW" or ".PDF".

To include a form in a report:

1. Display the "Background" tab of the report description ("Report .. Description").
2. Select the form's image 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 **Page-Number**. The report displayed in the report editor will contain a single page.
4. Specify whether the form's image 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 form's image, we recommend you check "Ignore the physical margins of the printer".
 - If the form's image must not be printed with the report (when printing on a pre-printed form for instance), "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 form's image is displayed in the background of the report being edited. 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 within each block according to the requested result.

To move the form image displayed in each block:

1. Click  found in the icon bar. The mouse cursor turns into a hand.
2. Click the block where the form's image must be moved.
3. To move the form's image of a block:
 - up or down: keep the mouse button down and move the mouse.
 - left or right: keep the mouse button and the [Shift] key down and move the mouse.
4. Click  to disable the mode for moving the image of the form.

Note: The mouse cursor must be moved within the block area otherwise it will not be taken into account.

4.4.4 Report based on a PDF file

If you have decided to display all the pages of a PDF file in the report, 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.

Note:

- This type of report can consume a large amount of memory resources (especially during the print preview).
- The use of a PDF file as 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 as report background.





4.5 Viewing the report in 3D

Your report displays controls that overlap? You want to find out the z-order position of your controls: which controls are displayed in the foreground ?

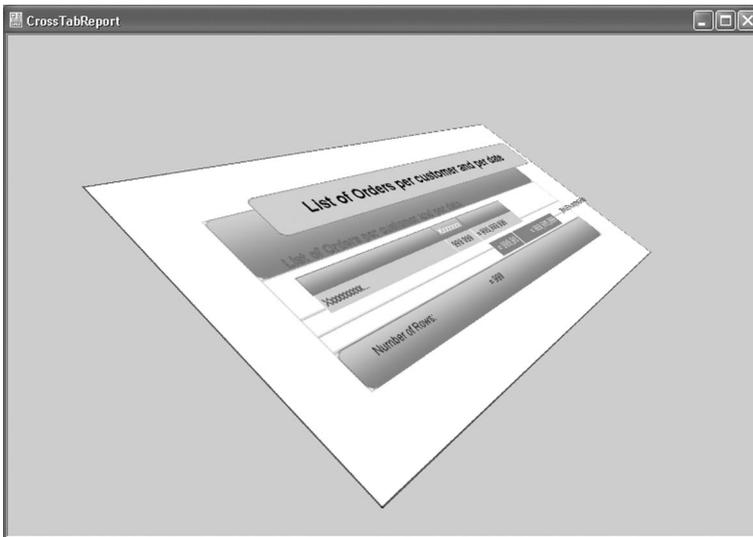
The two-dimensional display soon becomes inadequate for this type of operation.

The report editor gives you an all-round view of a report via the 3D display. The z-order position of the report controls is immediately visible. A control selected in this display mode remains selected when the mode changes back to two dimensions.

You have the ability to choose the best view angle via the mouse and via the keyboard keys.

Caution: a specific configuration is required to use this feature.

Example:



To display your reports in 3D:

1. Display the report in the editor.
2. Select "Display .. Show the report in 3D" (or press ALT + D).
3. The report is displayed in 3D.

Note: To display and animate a view in 3D of your report, press [ALT] + [SHIFT].

The mouse or the keyboard can be used to optimize the view of the report.

Moving with the mouse	
Rotation (left mouse button)	Click the report view and move the mouse while keeping the left button down.
Relocation (right mouse button)	Click the report view and move the mouse while keeping the right button down.



Moving with the keyboard	
Rotation	Use the arrow keys as well as the + and - keys on the numeric keypad. You can also use the numbers on the numeric keypad.
Relocation	Keep the SHIFT key down and use the arrow keys on the numeric keypad.
Zoom	Use the following key combination: - SHIFT + to zoom in - SHIFT - to zoom out.
Modifying the spacing between controls (modifications between the different display planes)	/ (on the numeric keypad) to reduce the spacing between the controls. * (on the numeric keypad) to increase the spacing between controls
Reinitializing the display	0 on the numeric keypad: full reinitialization 5 on the numeric keypad: reinitialization while keeping the spacing between controls.
Animating the view in 3 dimensions	[Scroll Lock]. [ESC] to cancel

Note: Press [ESC] to disable the 3D mode.

Possible actions when displaying a report in 3D

In 3D mode, only some actions can be performed on the report controls. You can for example:

- select a control.
- display the description of the control.
- display the source code of the control.

Conditions for displaying a report in 3D

To display your reports in 3D, the current computer must be equipped with:

- DirectX version 8.0.A (or later).
- A 3D accelerator card with up-to-date drivers.
- 16-bit or 32-bit display mode.





4.6 Changing the skin template associated with the report

When describing a report, a skin template can be applied to this report. A skin template lets you define the interface settings of the report: background image, size, control style, etc. A skin template can be applied to one or more reports (to harmonize the look of an application's reports).

To change the skin template or to apply a preset skin template to a report, select "Report .. Apply a skin template" and select the requested skin template.

Note: You also have the ability to create your own skin templates. See the online help (keyword: "Skin template, Creating a skin template for a report") for more details.



Important

Reports & Queries cannot be used to create skin templates of reports.

4.7 Change the report template associated with a report

When describing a report, a template can be applied to this report. A report template is used to define the common settings of a set of reports (presence of the same controls, code to run, etc.). A report template can be applied to one or more reports. The template's settings can be overloaded by the report using the template.

To change a template or apply a template to a report, select "Report .. List of templates used" and select the template you want.

Note: see the online help for more details about how to create report templates (keyword: "Template, report template").



Important

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



4.8 Managing the versions

The report editor is used to automatically manage the different versions of each report. The management of the report versions allows you to store the history of the main modifications made to a report. No need to save the same report under different names: the different versions are automatically managed by the report editor.

Indeed, the "WDE" file contains the description of the report as well as all the versions saved for this report. The size of the "WDE" file is proportional to the number of versions saved.

The report editor allows you to:

- save a new version of the current report,
- open a version of the current report,
- delete a version of the current report.

Saving a version

Saving a version is used to keep a backup of the report at a given time.

To save a version of the current report:

1. Select "File .. Versions". The window of current versions is displayed.
2. Click the "New" button. A window allows you to enter your own comments about the report version that will be saved.
3. Validate. The version is saved and the current report is automatically saved.

Note: The different versions of a report are listed in descending order (from the earliest one to the latest one), according to their date and time of backup.

Opening a version

Opening a version is used to create a new report corresponding to the selected version.

To open one of the versions of the current report:

1. Select "File .. Versions". The window of current versions is displayed.
2. Select the version to open.
3. Click the "Open" button. A new report is created in the report editor. This report corresponds to the selected version. The name of this report has the following format: <Report Name> <Date and Time of Version Creation>.

Tip: This new report can replace the initial report. To do so, close the initial report and save the new report under the same name as the initial report. **Caution:** all the versions of the initial report will be lost.

Deleting a version

Deleting the useless versions of a report reduces the size of the "WDE" file corresponding to the report.

To delete a version of the current report:

1. Select "File .. Versions". The window of current versions is displayed.
2. Select the version to delete.
3. Click the "Delete" button. The version is automatically deleted from the window for version management. This deletion will be effective the next time the report is saved.



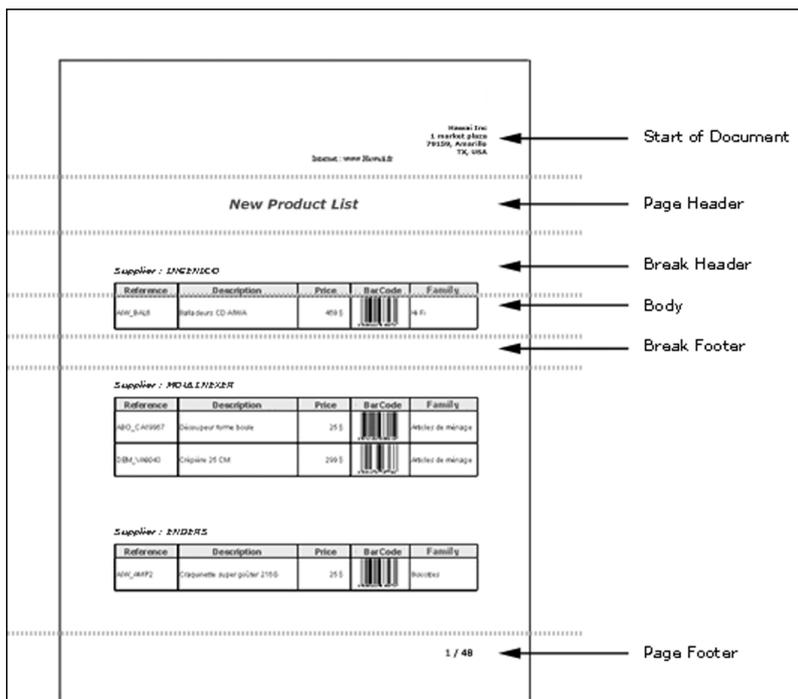


5. REPORT BLOCKS

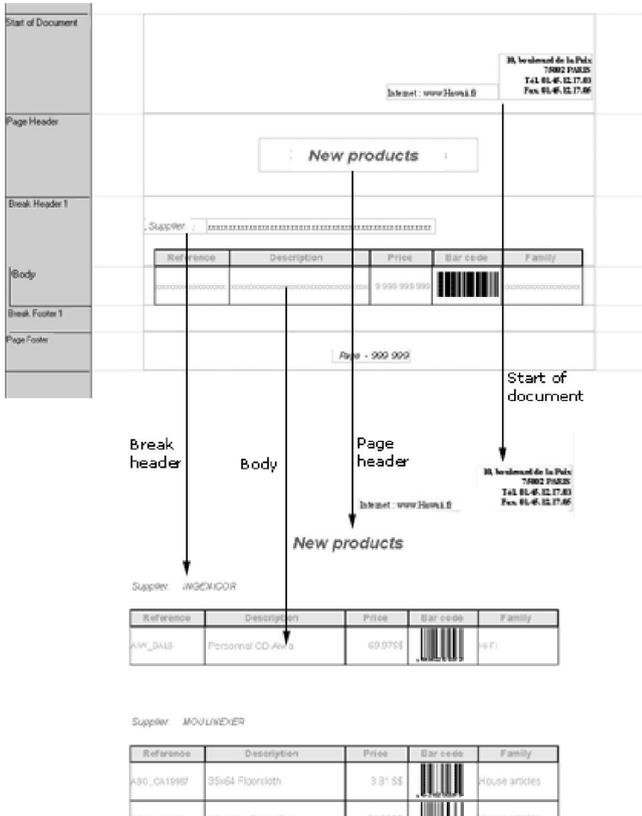
5.1 Overview

A printed report created by the report editor is made of several sections:

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



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



Report in edit

Previewing
the execution of
the report

5.2 The different types of blocks

A report is made of different types of blocks. Each block:

- can contain all types of controls.
- has specific parameters and printout 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 blocks 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
- *Break Header* block
- *Body* block
- *Body Complement* block
- *Break Footer* 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 the reports being edited. To display abbreviated captions, deselect "Display .. Options .. Show the full name of the blocks".

The different types of blocks and their characteristics are described later in this chapter.

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.

A window indicating the start of printout can be displayed in the process associated with this type of block.

5.2.2 Page Header block

The *Page Header* block is printed on each new page. By default, the *Page Header* block is printed just after the *Start of Document* block.

The *Page Header* contains information that must be printed at the top of each page (the title or a preset control indicating today's date for instance).

Caution: if the print destination is an HTML file, the *Page Header* block will be edited once only.

5.2.3 Break Header block

The *Beak Header* is optional. It is created when a break is requested on a sort argument. The *Break Footer* block necessarily exists if this block exists.

This block is run whenever the value of the sort argument changes.

Note: The *Break Header* block and the *Break Footer* block cannot be dissociated. However, one of these blocks can be made invisible (in the description window of the block or by programming with **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 to the end of page" checked in the "Details" tab of the block description window).

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



5.2.5 Body Complement block

The *Body Complement* block is linked to the *Body* block. This block is used to complete the printout down to the bottom of the page.

This block is printed whenever there is a free space between the *Body* block and the *Page Footer* block.

5.2.6 Break Footer block

The *Break Footer* block is optional. It is created when a break is requested on a sort argument. The *Break Header* block necessarily exists if this block exists.

This block is run whenever the value of the sort argument changes.

In most cases, this block contains counters or totals on the records matching the sort criterion.

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

5.2.7 Page Footer block

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 according to the total number of pages for instance).

Caution: If the print destination is an HTML file, the *Page Footer* block will be edited once only.

5.2.8 End of Document block

The *End of Document* block is printed once at the end of the report execution. In most cases, the *End of Document* block contains counters and/or totals (generated when creating the document or added thereafter).

5.2.9 Iteration block

The *Iteration* block is a free block, created manually in the report editor ("Insert .. New block"). In most cases, this block can be used as any other type of block. For example, an *Iteration* block can be used to replace a *Page Header* block to identify the even pages and the odd pages.

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

5.3 Characteristics of report blocks

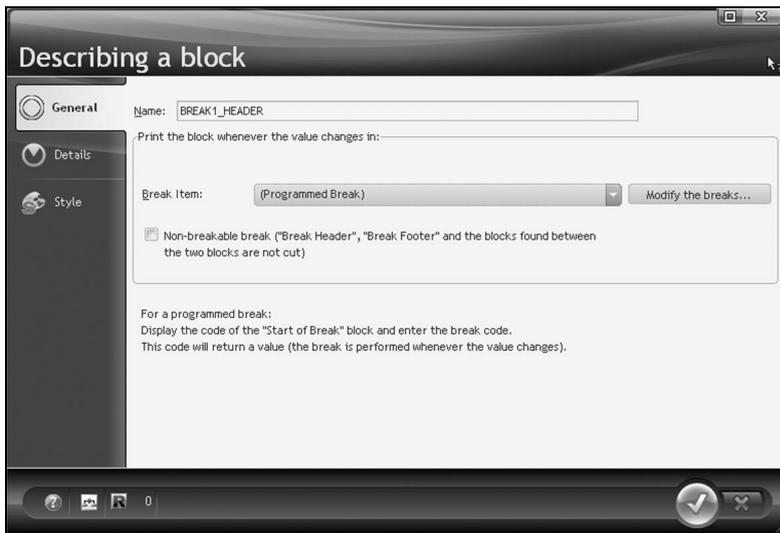
To display the description window of a block:

- double-click the caption of the block.
- select "Report .. Report controls and blocks" ([F12] key), select the block and click "Description".
- select "Block description" from the popup menu of the block (right-click the block caption).





For example, description window of a *Break Header* block:



Depending on the type of block, the description window is used to configure:

- the name of the block.
- the break item linked to the block (break blocks only).
- the parameters and the printout options of the block.
- the height of the block.
- the style of the block (background color, border, etc.).

Note: the characteristics of a block can also be viewed in the Modifier pane. To display this pane, select "Display .. Toolbars .. Modifier". Then, simply click the block's caption to view its description in the modifier.

5.4 Parameters and print options of the blocks

The parameters and the print options of a block are defined in the "Details" tab of its description (double-click the block caption).

The parameters and the print options of the report blocks are as follows:

- **Breakable:**
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.
- **Visible:**
The block can be displayed or not. If a block is invisible, the block and its controls will not be printed in the report.
This characteristic can be modified by programming with the *Visible* property.
- **Repeated to the end of the page:**
The block can be copied until it reaches the end of the page.



- **Page break after the block:**

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

Caution: If the print destination is an HTML file, the page breaks are not supported.

- **Completion to the end of the page:**

A block can be stretched until it reaches the page footer.

Note: This option is very useful if the block has a background image or a background color.

Caution: If the print destination is an HTML file, the page completion is not supported.

- **Extend the columns to the end of the page** (for blocks containing a table only):

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

Note: To extend the table columns until they reach to the end of the page, you can also add a *Body Complement* block. The *Body Complement* blocks are automatically extended to the end of the page.

Caution: If the print destination is an HTML file, the extension of columns to the end of the page is not supported.

- **Display the block in each column of the report (for the multi-column reports):**

If the report is a multi-column report, you have the ability to choose whether the block must be displayed on each column or on all the columns.

- **Keep room for the page footer when it is invisible** ('Page Footer' blocks only):

By default, the space taken by the Page Footer block will always be kept in the report pages (regardless whether the block is visible or not). With this option, the space taken 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.

5.5 Handling the blocks in the report editor

5.5.1 Create a block

To create a block in a report, use:

- "Insert .. New block".
- "Add a block" from the popup menu of the report (right-click the report currently created).
- the "Add a block" button in the "Blocks" tab of the report description.

Note: The *Break Header* and *Break Footer* 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*).

5.5.2 Delete a block

To delete a report block, use:

- "Delete the block" from the popup menu of the block to delete (right-click the block caption).
- the "Delete a block" button in the "Blocks" tab of the report description.
- the [Del] key (the block to delete being selected).

Note: The *Body* block cannot be deleted. This block can be made invisible (in the description window of the block or by programming with *Visible*).





5.5.3 Rearranging the blocks

The report editor gives you the ability to rearrange 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.

The order for printing the report blocks can be defined:

- in the "Options" tab of the description window of the report ("Report .. Description").
- in the "General" tab of the description window of each block ("Block Description" from the popup menu of the block).

See the online help (keyword: "Reports blocks, Print order") for more details.

5.6 Handling 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 the report blocks

The processes associated with the report blocks are as follows:

Processes	Runtime conditions
Initialization	<p>Run when the block is initialized. The order in which the processes are run when starting a report (including when running a test) is as follows:</p> <ul style="list-style-type: none"> • "Report opening" process. • "Initialization" process of the report blocks. • "Initialization" process of the 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 "Start of Print" process of the report. • perform block substitutions.
Pre-print	<p>Before printing a block, the following operations are performed:</p> <ul style="list-style-type: none"> • read the current record. • assign the values read in the block controls. • perform calculations on the values of the items. • run the "Pre-print" process of the block. • perform calculations on the block controls. <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 <i>Page Footer</i>, <i>Page Header</i> and <i>Body Complement</i> blocks if necessary).</p> <p>If the block is auto-sizable, the "Pre-print" process is run before knowing whether the block can be printed on the page or not.</p>



Pre-print (continues)	<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 <i>iPageEnd</i> in the <i>Break Header</i> block. <p>Caution: This process cannot be used to substitute the current block. However, it can request the substitution of another block (<i>iSubstBlock</i>).</p>
Post-print	<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 the controls that have just been printed. • substitute the blocks (taken into account for the next blocks). • print an Iteration block (<i>iPrintBlock</i>)
Break ?	<p>(<i>Break header</i> block only)</p> <p>Run before the "Pre-print" process of the <i>Break header</i> block. See "Break process", page 142, for more details.</p>

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

To view the processes specific to a block, use:

- "Code of the block" from the popup menu of the block (right-click the block caption).
- 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. Some WLanguage code can be used in these processes. See "Sequence of processes associated with a report", page 139, 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.
- by 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:

Property	Effect
Border	Enables you to find out and modify the characteristics of a block border.
BrushColor	Enables you to find out and modify the background color of a block
Height	Enables you to find out and modify the height of a block
Image	Enables you to find out and modify the background image of a block
InitialHeight	Enables you to find out the initial height of a block
Name	Enables you to find out the name of a block
State	Enables you to find out and modify the display status of a block
Visible	Enables you to find out whether a block is visible and to make a block visible/invisible
Width	Enables you to find out and modify the width of a block

See the online help for more details.







6. REPORT CONTROLS

6.1 The different types of controls

Several types of controls are available in the report editor:

- Static control
- Item
- Calculated control
- Image
- Bar code
- RTF control
- Preset control
- Chart
- Shape
- Check box
- Internal report
- URL link
- Edit 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 (keyword: "Report control (types of controls)") for more details.

6.1.1 Static control

The "Static" control is used to display a character string. This character string can contain:

- the content of an item (or variable) found in the data source of the report.
- 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 of the static control, the name of the item or control must be typed between [% and %].

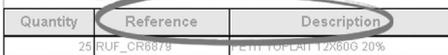
For example, the caption can correspond to: The [%SupplierName%] supplier sells the [%ProductName%] product at a price of [%PriceIOT%]. The SupplierName, ProductName and PriceIOT items will be replaced by their value during the execution of the report.

The character string can be entered over several lines. In this case, the control must be a multiline control and its size must be large enough.

Reminder: "Auto-Sizable" enables you to increase size of the control according to its content.

Examples:

Caption of a control → 

Caption of table columns → 

Quantity	Reference	Description
25	RUF_CR6879	PLHT TOP CART 12X60G 20%



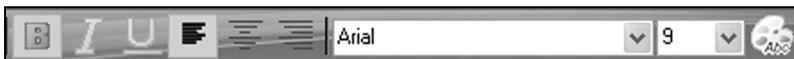
Modifying the caption

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 edit cursor.
3. Enter the new caption of the control.
4. Modify (if necessary) the formatting via the formatting toolbar:



Method 2:

1. Select the control.
2. Display the description window of the control ("Control .. Description").
3. Enter 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 a multilingual application or site, a different caption can be selected for each language supported by the report.

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

See "Multilingual feature in a report controls", page 87, for more details.

Changing a "Static" control into an "RTF" control

To display text in RTF format in a "Static" control, check "RTF format" in the "Details" tab of the control description ("Control .. Description").

The "Static" control becomes an "RTF" control. All the characteristics of the "Static" control are kept.

See "RTF control", page 79, for more details about the benefits of RTF controls.





6.1.2 Item control

The "Item" control is used to edit the content of a data file item or a variable. The item can be:

- an item found in the data source of the report (data file, query, variable, etc.).
- a data file item accessible by a link.
- a variable (global to the project or the report).

This control can also be assigned by programming.

Examples:

File used to list the characteristics of the customers

Name	Society	City
BROADHEAD	DIVAR 2/6	DALLAS
MONTGOMERY	SPLASH IND.	DENVER
BLAUWAERT	FILM & Co	LAS VEGAS
TREBBIA	TECHNI DEVELOPMENT	WASHINGTON

Order Form	
DIVAR 2/6	
M BROADHEAD	
XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
XXXXXXXXXXXXXXXXXXXXXXXXXXXX	

Control linked to a file item

To define the item linked to the "Item" control:

1. Display the "Link" tab in the description window of the control ("Control .. Description").
2. To link the control to an item or a variable that belongs to the data source of the report, select this item or variable directly in the "Linked item" list box.
3. To link the control to an item that does not belong to the data source of the report, select the "multi-file" option then select the item linked to the current control.

Special "Item" control: "Currency + Euro"

If the "Item" control is a "Currency + Euro" control, it can be used to manage:

- the *stored currency*: currency of the data found in the data source of the report. If the control is handled by programming, the currency used for the value will be the stored currency.
- the *displayed 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*: ability to link two "Currency + Euro" controls between themselves. 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 the report.
 - the tracker control: this control is used to display the value of the base control in the requested currency.

See the online help (keyword: "Currency + Euro") for more details.

6.1.3 Calculated control

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

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



Examples:

	Unit Price	Total
	2 20	37 34
	8 43	421 40
	5 07	152 03
	5 97	185 21
	4 25	106 14
Total		90 212 USD

Sum of the row

Sum of table

Specific "Calculated" control: "Currency + Euro"

If the "Calculated" control is "Currency + Euro" control, you can manage:

- the *stored currency*: currency of the data found in the data source of the report. If the control is handled by programming, the currency used for the value will be the stored currency.
- the *displayed 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*: ability to link two "Currency + Euro" controls between themselves. 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 the report.
 - the *tracker control*: this control is used to display the value of the base control in the requested currency.

See the online help (keyword: "Currency + Euro") for more details.

6.1.4 Image control

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

- be selected via the file picker (from the description window of the control).
- correspond to the content of an item or a variable of the data source (for example, if the data source is a HyperFileSQL file, the item can be an Image memo or a character string item containing the full path of the image).

The "Image" control is then linked to this item.

Example:

Bericourt	Emilyne	
Blanchard	Wissam	
Bokotza	Gizelle	





To define the item linked to the "Image" control:

1. Display the "Link" tab in the description window of the control ("Control .. Description").
2. To link the control to an item or a variable that belongs to the data source of the report, select this item directly from the "Linked item" list .
3. To link the control to an item that does not belong to the data source of the report, select the "Multi-file" option then select the item linked to the current 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 some text for instance.

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

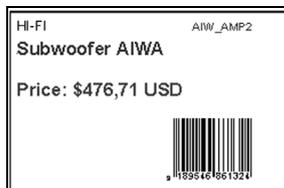
See "Multilingual feature in a report controls", page 87, for more details.

6.1.5 Bar Code control

The "Bar Code" control enables you 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 or a variable found in the data source of the report or it can be assigned by programming. The report editor proposes several types of bar codes that can be printed horizontally or vertically.

Example:



To define the item linked to the "Bar Code" control:

1. Display the "Link" tab in the description window of the control ("Control .. Description").
2. To link the control to an item or a variable that belongs to the data source of the report, select this item directly from the "Linked item" list.
3. To link the control to an item that does not belong to the data source of the report, select the "Multi-file" option then select the item linked to the current control.

6.1.6 RTF control

The "RTF" (Rich Text Format) control is used to display a formatted character string in your report. This character string is in RTF format. Specific tags are used to define the formatting of each character or character string. These tags are not displayed in the control.

Note: The text displayed in the control can:

- contain the value of an item or variable printed in the report.
To insert the value of a report item or control into the text of the static control, the name of the item or control must be typed between [% and %].

For example, the text of the control can correspond to: The [%SupplierName%] supplier sells the [%ProductName%] product at a price of [%PriceOT%]. The SupplierName, ProductName and PriceOT items will be replaced by their value during the execution of the report.

- come from a text item or variable found in the data source of the report.
If the control is linked to an item (or a variable), the name of the associated item is displayed in the "Linked item" list of the "Link" tab of the control description.



Example for using a "RTF" control:



Caption of a report that uses font of different colors.

Modifying the text of an "RTF" control

The text of an "RTF" control can be modified:

- by programming (*Caption*).
- in the report editor:
 1. Select the control (click the control for instance). The selection handles appear around the control.
 2. Click the control again. The mouse cursor turns into an edit cursor.
 3. Enter the new text of the control.
 4. The layout toolbar enables you to modify (if necessary) the formatting of the text:



Changing a "Static" control into an "RTF" control

To display text in RTF format in a "Static" control, check "RTF format" in the "Details" tab of the control description ("Control .. Description").

The "Static" control becomes an "RTF" control. All the characteristics of the "Static" control are kept.

Changing an "Item" control into an "RTF" control

To display text in RTF format in an "Item" control, check "RTF format" in the "Details" tab of the control description ("Control .. Description").

The "Item" control becomes an "RTF" control. All the characteristics of the "Item" control are kept.

Changing an "RTF" control into an "Item" control

To display the RTF tags or to link the control to an item other than a text item, uncheck "RTF format" in the "Details" tab of the control description ("Control .. Description").

The "RTF" control becomes an "Item" control. All the characteristics of the "RTF" control are kept.





6.1.7 Preset control

The "Preset" control is used to display standard information without 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:

*Miss GIFFORD
456 Pennsylvania Ave.
Washington, DC, 2221*

May 12 2003

Date of the day when the report is printed

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

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

6.1.8 Chart control

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

- Bar chart (grouped or stacked, vertical or horizontal),
- Line, area,
- Scatter, radar,
- Pie chart, donut, half-circle,
- Stock chart (Candlestick, MinMax or BarChart), etc.

Notes:

- This control is not available in WinDev Mobile.
- This control cannot be linked to a variable.

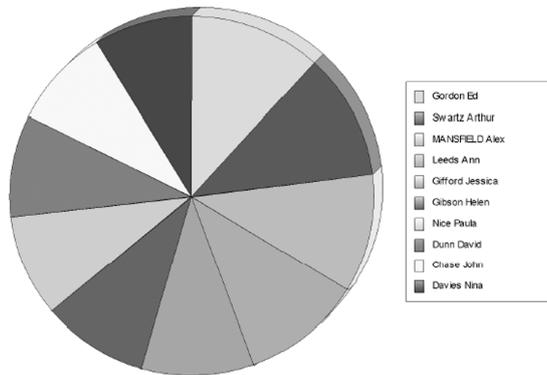


Example:

03/22/2007

Top Ten Turnovers

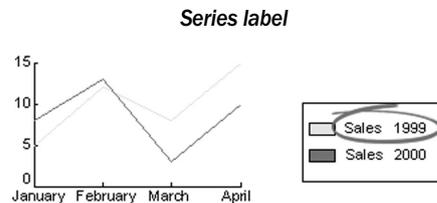
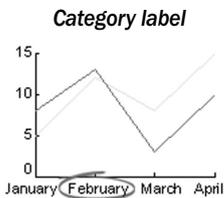
Name	T.O.
Gordon Ed	3 124 051,26
Swartz Arthur	3 037 091,92
MANSFIELD Alex	2 880 704,50
Leeds Ann	2 876 331,11
Gifford Jessica	2 677 760,22
Gibson Helen	2 540 121,56
Nice Paula	2 473 724,81
Dunn David	2 454 789,30
Chase John	2 373 359,45
Davies Nina	2 351 401,82



Information specific to the creation of the chart

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, etc.).
- the mode for filling the chart (source identical to the one of the report or fill by programming).
- the number of series.
- the source of the caption for the series. This caption comes from an item found in the data source associated with the report. This caption corresponds to:
 - the label of a data category (or section).
 - the label of a data series.





- For each series:
 - the color and the caption of the series.
 - the source of the series. The values of the series come from a control or from an item found in the data source associated with the report.
- the supply block.
- the frequency of the marks on the vertical/horizontal axis of the chart (if necessary).
- the style of the chart (relief, color gradients and gridlines).
- the titles of the axes if necessary.

You also have the ability to create a Chart report or a Table + Chart report. See "The different types of reports", page 29, for more details.

6.1.9 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 style, the thickness and the color of the line can be configured.

6.1.10 Check Box control

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

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

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

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

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

Example:



To define the item linked to the "Check Box" control:

1. Display the "Link" tab in the description window of the control ("Control .. Description").
2. To link the control to an item or a variable that belongs to the data source of the report, select this item directly from the "Linked item" list.
3. To link the control to an item that does not belong to the data source of the report, select the "Multi-file" option then select the item linked to the current control.

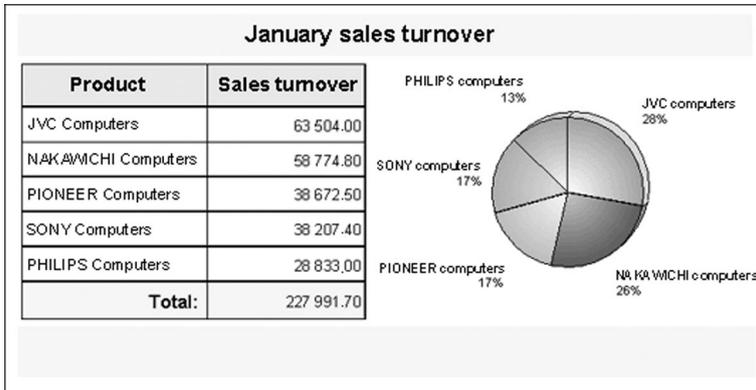


6.1.11 Internal Report control

The "Internal Report" controls can be 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 the online help (keyword: "Internal report") 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 use of *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 enable you to print several reports without programming:

- one below another,
- side by side.

See "Nested reports", page 113, or the online help (keyword: "Nested reports") for more details about nested reports.

6.1.12 URL Link control

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

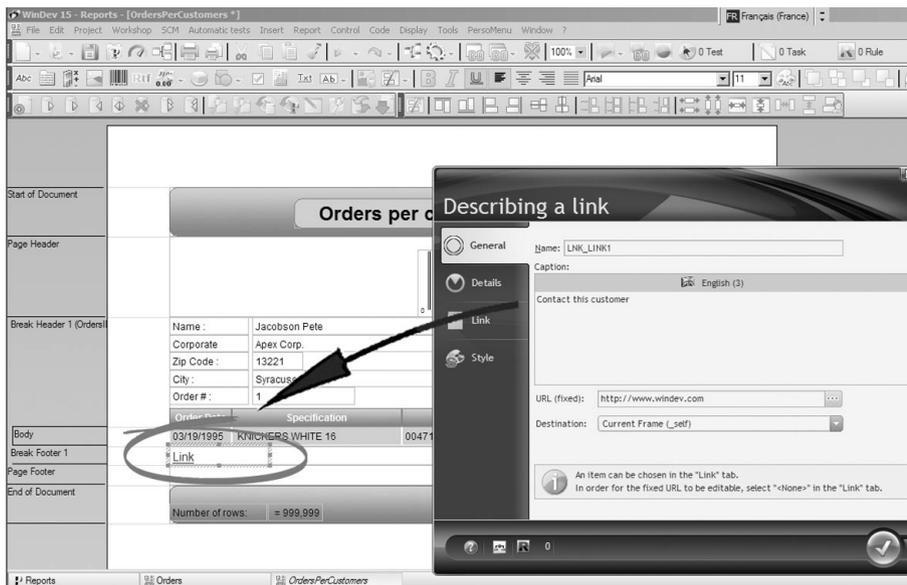
- a link to an email address,
- a link to an Internet site,
- etc.

The "URL Link" control is enabled only if the report is printed in an HTML file or displayed in the preview window. Otherwise, the "URL Link" control is visible but disabled.





Example:



To define the item or the variable linked to the "URL Link" control:

1. Display the "Link" tab in the description window of the control ("Control .. Description").
2. To link the control to an item or a variable that belongs to the data source of the report, select this item or variable directly in the "Linked item" list box.
3. To link the control to an item that does not belong to the data source of the report, select the "Multi-file" option then select the item linked to the current control.

Note: The URL links can be easily identified at runtime by .

6.1.13 Edit control

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

- to allow the user to enter some information to print,
- to allow the user to modify some information to print,
- etc.

The edit control is only active 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.

See "Editable reports", page 109, for more details about using edit controls.



6.2 Characteristics of report controls

The characteristics of a control can be viewed and/or modified in:

- the description window of the control.
- the modifier

The report editor also enables you to:

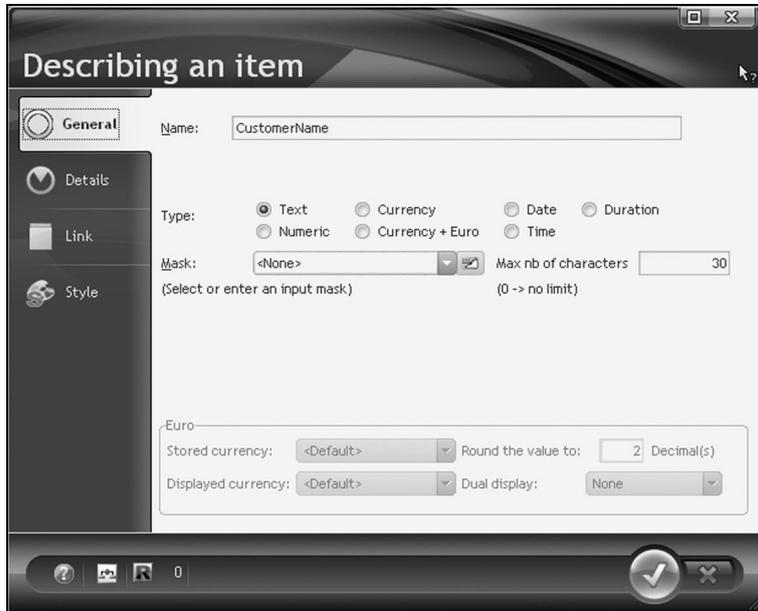
- modify the caption of a control,
- modify the name of a control.

6.2.1 Description window of a control

To display the description window of a control:

- double-click the control.
- select "Description" from the popup menu of the control (right mouse click).
- select the control and select "Control .. Description" ([ALT+Return] key).
- select the control and select "Display .. Selection description".
- select "Report .. Report blocks and controls" ([F12] key). Then, select the control and click the "Description" button.

For example, description window of an item control:



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 the selected controls into several tabs:

Tab	Characteristics
General	Used to specify the name of the control and the display characteristics of the control. For example: <ul style="list-style-type: none"> • the caption of the control (this caption can be multilingual). • the type of information displayed in the control (text, numeric, etc.). • the display mask used for the control (the mask can be multilingual). See the online help (keyword: "Mask, Multilingual mask") 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 for more details.
Details	Used to define: <ul style="list-style-type: none"> • the print parameters of the control (visible, breakable, etc.). • the size of the control. • the position of the control in the report and in the block to which the control belongs.
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.
Style	Used to define: <ul style="list-style-type: none"> • the style of the control (background color, border, etc.). • the font used in the control. Note: The font used in the control can also be defined via the icons found in the "Popup" toolbar.

Note: See the context-sensitive help of the description window for more details about each of the various tabs ( in the top right corner).

6.3 Multilingual feature in a report controls

In a multilingual application or site, the multilingual feature is managed by the report editor for the following elements:

- the static controls,
- 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 runtime language of the project is defined in the project description ("Project .. Project description", "Languages" tab).

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



6.4 Group of controls

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

In the report editor, all the controls of all groups are displayed: some controls may overlap. 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 healthcare form for instance, some controls may relate to men while others may relate to women. Grouping the related controls together enables you to make them visible or invisible according to the gender of the user.

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

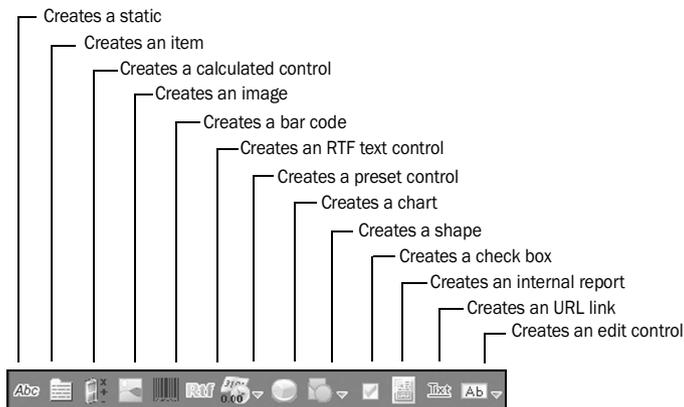
6.5 Handling the controls in the editor

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

6.5.1 Creating a control

To create a control in a report, you can:

- select "Insert .. New control" and select the type of control to insert.
- use the toolbar of the report editor.



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





6.5.2 Selecting one or more controls

The report editor enables you to select one or more report controls with the keyboard or with the mouse.

To select a control, click the control to select.

The selected control is displayed with handles.



The name of the selected control is displayed in the status bar. It can be modified directly.



To select several controls at once:

Method 1: with the [CTRL] key.

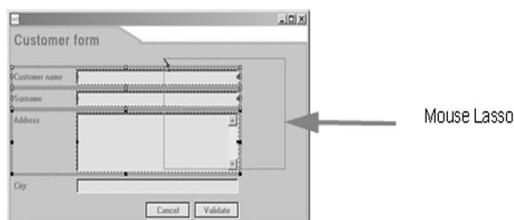
1. Press the [CTRL] key and keep this key down.
2. Select the controls with the mouse (click each control with the mouse).
3. Release the [CTRL] key. The selected controls are displayed with handles.

Notes:

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

Method 2: with the lasso.

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



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, keep the [Shift] key down during the selection steps.



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

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

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 source code of the copied control will be identical to the ones of the initial control.

To copy a control into the same report:

1. Select the control to copy.

2. Click the "Copy" icon () in the standard toolbar of the report editor.

Note: You can also select "Edit .. Copy" or "Copy" from the popup menu of the control.

3. Click the "Paste" icon () in the standard toolbar of the report editor.

Note: You can also select "Edit .. Paste" or "Paste" from the popup menu of the control.

4. Position the control with the mouse (the outline of the control appears in dotted lines and it helps you position the new control).

To copy a control from a report to another report:

Method 1:

1. In the first report, select the control to copy.

2. Click the "Copy" icon () in the standard toolbar of the report editor.

Note: You can also select "Edit .. Copy" or "Copy" from the popup menu of the control.

3. Display the destination report in the report editor.

4. Click the "Paste" icon () in the standard toolbar of the report editor.

Note: You can also select "Edit .. Paste" or "Paste" from the popup menu of the control.

5. Position the control with the mouse (the outline of the control appears in dotted lines and it helps you position the new control).

Method 2: (both reports must be displayed in the editor)

1. In the first report, select the control to copy.

2. Keep the left mouse button down and drag the control toward the destination report.

3. Release the mouse button once the control is positioned in the destination report (the outline of the control appears in dotted lines and it helps you position the pasted control).

6.5.4 Copying the style of a control

Copying the style of a control consists in copying the characteristics of a control style into another control. Only the style of the control 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 both controls are copied.

The style copy can be performed between two controls belonging to the same report or between two controls belonging to different reports.

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





To perform a style copy between two controls:

1. Select the control containing the style to copy.
2. Click  (or select "Edit .. Copy the style", [CTRL]+[Shift]+[C]).

The mouse cursor turns into a brush: 

3. Select the control where 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 a style to several controls, hold the [CTRL] key down when the style is copied to 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 enables you to align the moved controls (See "Real-time interface checker", page 92).

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: select the controls to delete and 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 and at run time.

You also have the ability to position a control in the home window of a report: the control is "off-report". This control will be visible in edit only. This control will not be printed.



An "off-report" control enables you, for instance, to store a report control without printing it. This enables you to use the content of the control in programming (like a variable).

To create an "off-report" control:

1. Select "Insert .. New control" and select the type of control to insert.
Note: you can also use the icons found in the report editor's toolbar.
2. Click the left part 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 your report:

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

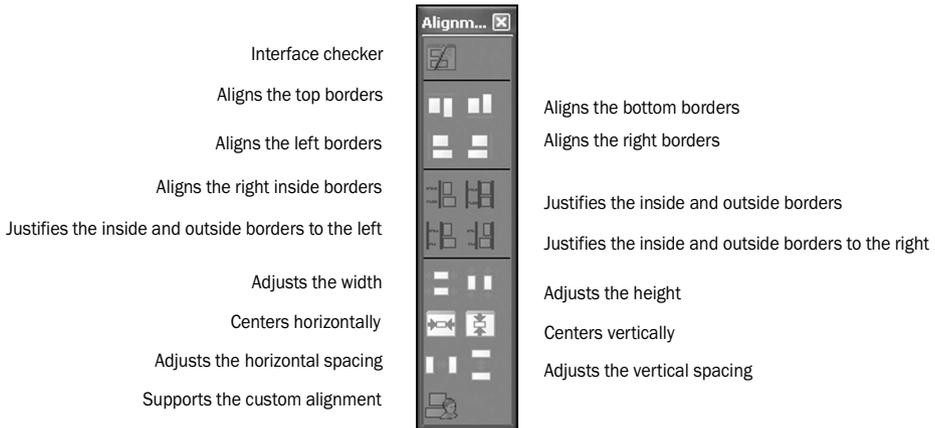
This paragraph presents the different methods for aligning the controls found in a report. See the online help (keyword: "Alignment") for more details.

6.6.1 Alignment options

The alignment options enable you to optimize the alignment of the controls in a report. These options are used to align several controls found in the report.

To align the controls, you can:

- select "Control .. Alignment".
- use the alignment toolbar ("Display .. Toolbars .. Alignment"):



6.6.2 Real-time interface checker

When a control is moved in a report, guides are automatically displayed by the real-time interface checker (also called automatic positioning). The objects snap to the guides, allowing you to align the control currently moved with the other controls found in the report.

The real-time interface checker is enabled by default.

To disable the real-time interface checker:

1. Select "Display .. Options .. Modify the options" (or press [F12]). The editing options of the report editor are displayed.
2. Display the tab named "Magnetism" and uncheck "Automatic positioning enabled".





6.6.3 Zoom

To perfectly align your controls, zoom in your report.

To specify a zoom coefficient:

- select "Display .. Zoom..." and choose the zoom factor.
- enter a zoom factor directly in the status bar.



To zoom in or out:

1. Press the [CTRL] key.
2. Modify the zoom factor with the mouse wheel while keeping the [CTRL] key down.

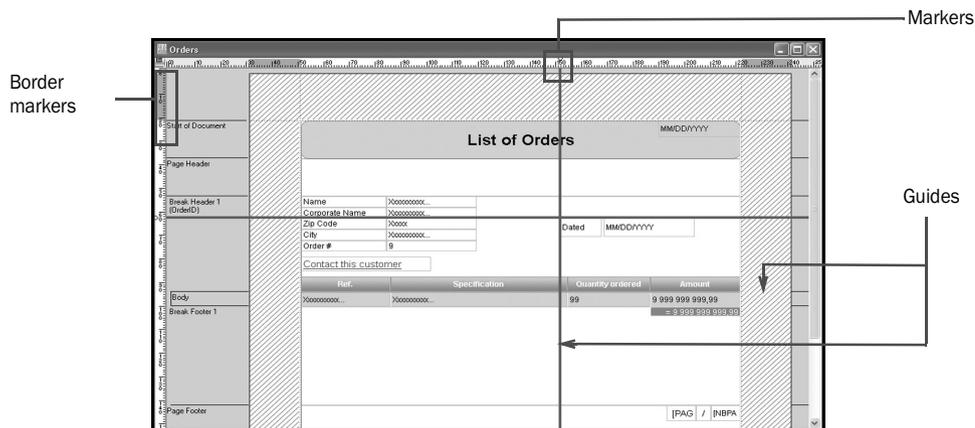
6.6.4 Rulers and markers

In the report editor, alignment rulers can be displayed when a report is created or modified.

These rulers have snap-on guides: any control that comes close to a guide is automatically "snapped" by it. This feature enables you to easily position, align or resize the controls found in a report.

To display the rulers, select "Display .. Rulers" ([CTRL]+[R]). Two types of guides can be used:

- the guides, used to align and organize the controls inside the reports.
- the border guides, used to define a margin of the same size on each side of the report.



Note: To view a report with or without guides, select "Guide visible" from the popup menu of the rulers. Whether they are visible or not, the guides are always enabled: any control moved toward a marker is automatically snapped by it.

The markers can be easily handled with the mouse in the report editor. When a marker is hovered by the mouse, the cursor turns into a double arrow.

To create a marker:

- **Method 1:**
Click one of the rulers found in the report. The marker is created on the nearest mark.
- **Method 2:**
Select "Add a marker" from the popup menu (right mouse click) of the ruler. The marker is created on the nearest mark.



To move a marker:

1. Click the marker found in the ruler. The cursor turns into a double arrow.
2. Move the cursor to the new marker position.

To delete a marker, select "Delete a marker" from the popup menu (right mouse click) of a marker.

To delete all the markers from a ruler, select "Delete all" from the popup menu of the ruler.

To copy the markers and the guides:

Select "Copy the guides" from the popup menu of the rulers.

Open the report where the guides and the markers must be copied.

Display (if necessary) the rulers in this report ("Display .. Rulers").

Select "Paste" from the popup menu of the rulers.

The border markers are used to define a "margin" in the current report. The border markers are located at the end of the rulers.

To move the left or right margin of the report, point to a border marker on the horizontal ruler. When the cursor turns into a double arrow, drag the margin.

To move the top or bottom margin of a report, point to a border marker on the vertical ruler. When the cursor turns into a double arrow, drag the margin.

Notes:

- The border markers cannot be deleted. If the borders are not required in a report, the border markers can be reduced to their minimum size.
- The opposite border markers are interdependent: when one border marker is moved, the opposite border marker moves accordingly.

6.6.5 Snap-on grid

The snap-on grid allows you to easily position and align the report controls.

The size of the grid is expressed in centimetres.

To display the grid in the report editor, select "Display .. Grid visible".

To use the magnetism effect of the grid, select "Display .. Grid enabled".

When the grid is enabled, it can be temporarily disabled by keeping the [Shift] key down while moving the elements.

Similarly, when the grid is disabled, it can be temporarily enabled with the [Shift] key.





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 conditions
Initialization	Run when the report is initialized. The order in which the processes are run when starting a report (including when running a test) is as follows: <ul style="list-style-type: none"> • "Opening" process of the report. • "Initialization" process of the report blocks. • "Initialization" process of the controls.
Pre-print	Run before printing the control.
Click	Run when clicking the control (in the preview window only). A star allows the user to easily identify the clickable controls
Entry in	Run when the input cursor enters in the control. 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 instance). 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, etc.). This process is available for the edit controls used in preview mode only.

Source code in WLanguage can be entered in these processes. See "Sequence of processes associated with a report", page 139, for more details.

To view and/or modify the processes of a control:

You can:

- select the control and press [F2].
- select the control and select "Code .. Code of selection".
- select the control and display the "Code" pane ("Display .. Toolbars .. Code").
- select "Code" from the popup menu of the control (right mouse click).
- click the "Code" icon in the control description.

6.7.2 Handling 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:

- by a WLanguage function,
- by a WLanguage property,
- by direct assignment.



For example:

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

6.7.3 Properties associated with the report controls

The following properties are associated with the report controls:

Property	Effect
Border	Enables you to find out and modify the characteristics of the border for a report control.
BrushColor	Enables you to manage the background color of a control
Caption	Enables you to find out and modify the text in a static control
Color	Enables you to manage the color of the text in a control
DisplayCurrency	Enables you to find out and modify the currency used when displaying a value in a "Currency + Euro" control
DisplayedItem	Enables you to find out and modify the item displayed by the report control
Empty	Enables you to find out whether the value of the item linked to a control (for the record currently printed) is filled or not
Font	Enables you to find out and modify the characteristics of the font used in a control
FontBold	Enables you to find out and modify the "Bold" attribute for the content of a control
FontCharset	Enables you to find out and modify the character set used by the font of a control
FontCondensed	Enables you to find out whether the characters of the text displayed in a control are condensed and to condense (or not) the characters of the text displayed in a control
FontExtended	Enables you to find out whether the characters of the text displayed in a control are extended and to extend (or not) the characters of the text displayed in a control
FontItalic	Enables you to find out and modify the "Italic" attribute for the content of a control
FontLarge	Enables you to find out whether the characters of the text displayed in a control are enlarged and to enlarge (or not) the characters of the text displayed in a control
FontName	Enables you to find out and modify the font used in a control
FontSize	Enables you to find out and modify the size of the font used in a control
FontStrikeOut	Enables you to find out and modify the "Strikeout" attribute for the content of a control
FontUnderlined	Enables you to find out and modify the "Underline" attribute for the content of a control





FutureHeight	Enables you to find out the future height of an auto-sizable control
Group	Enables you to find out whether the control belongs (or not) to a group of controls
Height	Enables you to find out and modify the height of a control (in centimetres)
HeightInPixel	Enables you to find out and modify the height of a control (in pixels)
HorizontalAlignment	Enables you to find out and modify the horizontal alignment of static controls, calculation controls, preset controls and item controls
Image	Enables you to find out and modify the background image of a control
InitialHeight	Enables you to find out the initial height of a control (when the report is opened)
InitialWidth	Enables you to find out the initial width of a control (when the report is opened)
InputMask	Enables you to find out and modify the input mask of a control
InternalReportName	Enables you to find out and specify the name of the report printed in an "Internal Report" control
MemoryCurrency	Enables you to find out and modify the currency used when handling a "Currency + Euro" control
Multiline	Enables you to find out whether a control is multiline and to modify this option
Name	Enables you to find out the name of a control
Size	Enables you to find out and modify the maximum number of characters found in a control
State	Enables you to find out and modify the status of a control
URL	Enables you to find out and modify the address associated with a URL control
Value	Enables you to find out and modify the value of a control
VerticalAlignment	Enables you to find out and modify the vertical alignment of static controls, calculation controls, preset controls and item controls
Visible	Enables you to find out whether a control is visible and to make a control visible/invisible
Width	Enables you to find out and modify the width of a control (in centimetres)
WidthInPixel	Enables you to find out and modify the width of a control (in pixels)
X	Enables you to find out and modify the x-coordinate of a control
XInitial	Enables you to find out the initial x-coordinate of a control (when the report is opened)
Y	Enables you to find out and modify the y-coordinate of a control
YInitial	Enables you to find out the initial y-coordinate of a control (when the report is opened)
ZOrder	Enables you 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 is made 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 internal reports (See "Internal Report control", page 84) or nested reports (See "Nested reports", page 113).

A table contains three sections:

- the **header row**, used to display the header of the table columns. This row is optional. If it is displayed, this row must be:
 - the first row of the table.
 - displayed in each new table.

In most cases, this row is displayed in a *Page Header* block or in a *Break Header* block.

- the **data rows**, used to display the information coming 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.

A single data row is displayed in the report editor. At run time, this data row is repeated as many times as necessary.

- the **total row**, used to calculate totals on the table columns.

In most cases, these rows are displayed in the *Break Footer* block or in the *Page Footer* block.

Note: The different total cells are not necessarily displayed on the same row.

For example:

Ref.	Specification	Quantity ordered	Amount	
347000D-P	UNDERWIRED BRA VARIANCE BLACK 38B	2	14 563,80	Header row
344150M-E	KNICKERS WHITE DIM 16	2	27 540,40	
349014CER	UNDERWIRED BRA WHITE 38D	3	4 036,20	Data rows
349636M-M	KNICKERS YELLOW 18	1	95 623,20	
001913UKC	UNDERPANTS BLUE XL	2	561,60	
057148XOY	KNICKERS WHITE 16	3	3 285,36	
349169M-E	KNICKERS SLOGGY S2	2	2 176,20	
345393PHO	BRA WHITE 38B	2	35 399,00	
			= 183 185,76	

Benefits of using tables

In edit and at run time, the 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.

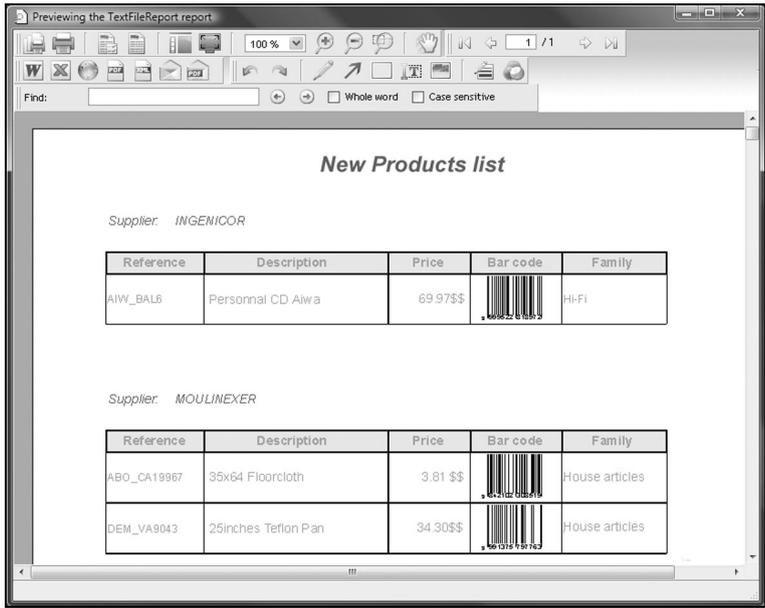
At run time, if a value cannot be entirely displayed in a cell of an auto-sizable table, all the cells found on the row will take the same height in order for the value to be entirely displayed.

The report editor proposes two types of tables:

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

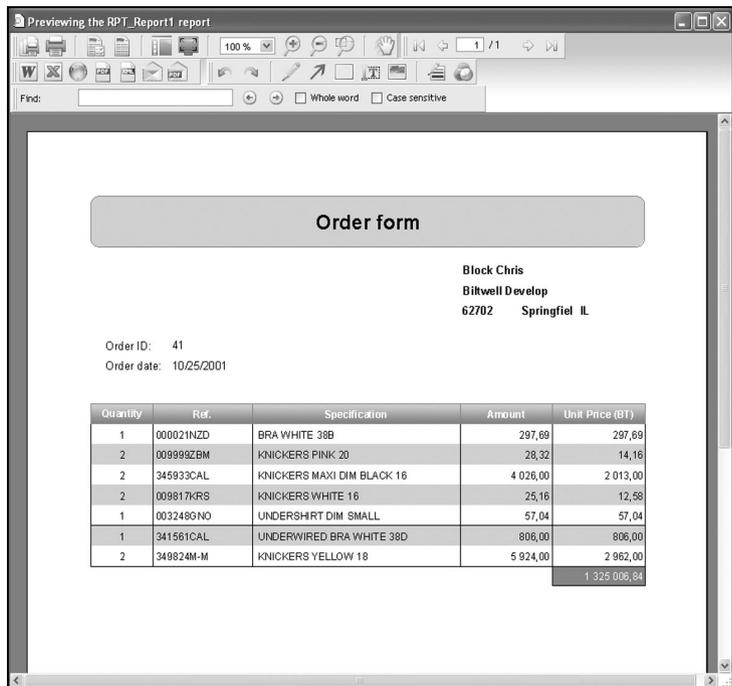


Some examples of reports with tables:



List of products (simple table)

Order form used to list the products ordered in a table (simple table)





Previewing the CrossTabReport report

100 %

Find: Whole word Case sensitive

Sales by year and by product type

Product type	1997	1998	1999	2000	2001	2002	Total
Sofa	103	112	312	279	302	315	1 423
Chair	217	221	246	276	298	305	1 563
Armchair	263	288	187	204	221	246	1 409
Stool	174	195	127	141	171	158	966

Number of lines: 4

Sales per year and per type of product (crosstab)

7.2 Creating a table in a report

The tables can be created in a report:

- when creating a Table report. See “Table report:”, page 29 for more details.
- when creating a Table + Chart report. See “Table + Chart report (not available in WinDev Mobile):”, page 35, for more details.
- when creating a Crosstab report. See “Crosstab report:”, page 36, for more details.
- by inserting a table into an existing report.

7.2.1 Creating a Table report

To create a Table report, select “File .. New .. Report”. The wizard for report creation starts.

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

- Selecting the type of report to create: Table.
- Selecting the data source of the report. See “Data sources of a report”, page 47, for more details.
- Selecting the sort items (if necessary). See “The sort”, page 38, for more details.
- Selecting the break items. See “The breaks”, page 39, for more details.
- Selecting the data that will be displayed in the report. This data will be displayed in the data row of the table.
- Choosing the calculations to perform in the report. These calculations will be displayed in the total row of the table.
- Selecting the format of the page.
- Selecting the skin template. See “Skin template”, page 40, for more details.



7.2.2 Creating a crosstab report

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

To create a Crosstab report, select "File .. New .. Report". The wizard for report creation starts.

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

- The item corresponding to the captions of the columns.
Note: if this item is a "Date" item, the wizard proposes to:
 - keep the day, the month or the year only 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 the day, the month or the year only 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 orders per year and per customer. The customer named "Smith" has placed 2 orders in 2005 and 3 in 2006.
 - 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 2005, 279 sofas were ordered.
 - an item.
For example: the score awarded by each referee to an ice skater. 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.

7.2.3 Inserting a table into a report

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

Note: Only a simple table can be inserted. A Crosstab table 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, select "Control .. Table .. New table". The wizard for table creation starts. Depending on the current data source, you can:

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

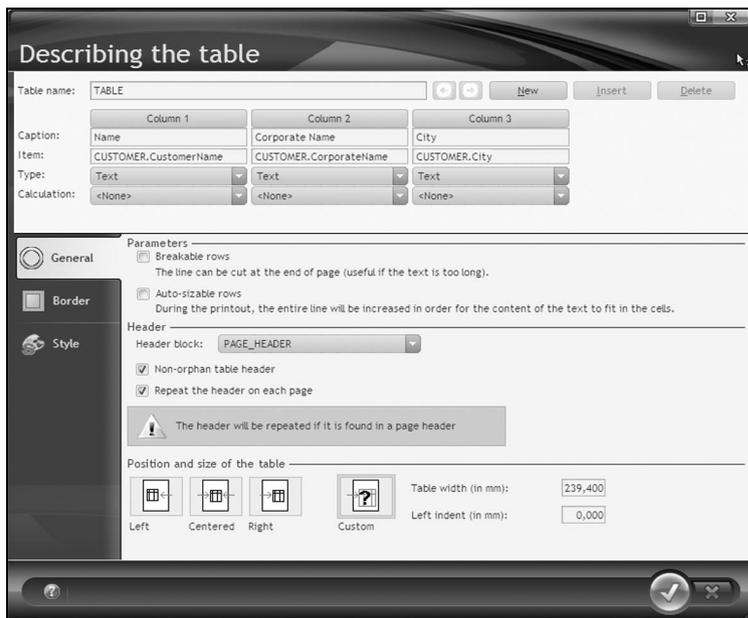
In all cases, the header row of the table is created in the header block and the data row is inserted into the *Body* block of the report.





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 totals). 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.

See the online help for more details ( in the top right corner and click the requested option).

Note: to directly display the characteristics of a column, select the column's cells (See "Selecting a table element", page 105, for more details) and select "Display .. Description".

7.3.1 Characteristics of the table

Three tabs are displayed in the bottom part if the table name 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 instance).
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 configure 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 table style. 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 part of the description window.

For each column the following is shown 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 instance. The type is automatically updated according to the selected item.
- 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.
- The **type of data** displayed in the column.

Two tabs are displayed in the bottom part if the name of the column is selected. These tabs are used to configure the column's settings.

"Column" tab: column settings

The "column" tab lets you configure the column's settings: type, mask, etc.

"Link" tab: Configure the column's link

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

7.4 Handling the cells of the table

The table cells can be handled individually or via the table's description. This paragraph presents how the table cells can be handled individually.

The main operations that can be performed are as follows:

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

Note: All the 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 cell.
- 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 table cell three times successively.

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 the 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 and select "Control .. Table .. Table Description".
- click a cell three times successively (all the table cells are selected) then double-click.

7.4.3 Adding a cell to a table

All the types of controls available in the report editor can be added to a table. These controls can:

- replace an existing cell in the table.
- be added to a column (a calculation added at the bottom of a column for instance).

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 the control. A wizard starts.
3. Select "Add to an existing column".
4. Select the column where the addition must be performed as well as the position of the control in the column (header, data or total).
5. Validate the addition of the cell.

7.4.4 Adding a column to a table

A column can be added to a table. This new column may (or may not) correspond to an existing control.

To add an existing control to a table as column:

1. Select the control.
2. You can:
 - "Drag and Drop" the control toward 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 the table" from the popup menu of the control. The wizard starts, allowing 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.
 - select "Insert .. New column".
2. Enter the caption of the header for the new column.
3. Select (if necessary) the item linked to the new column and validate the addition of the new column.



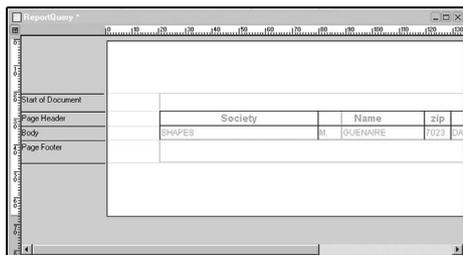
7.4.5 Resizing the table cells

The entire table is resized when a table cell is resized in the report editor. Two cases may occur:

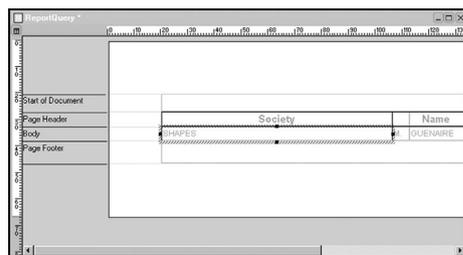
- **Modifying the width of a cell:**

All the cells found in the column will also be resized.

All the columns found after the column currently resized will be moved.



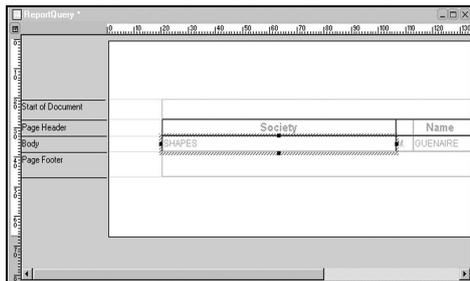
Before the resize operation



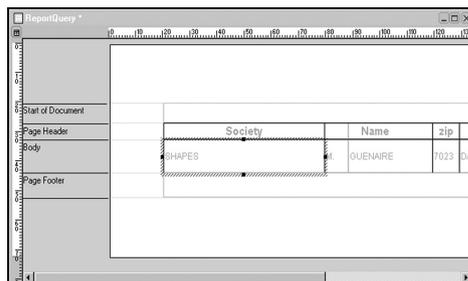
After the resize operation

- **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. Some 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 resize operation is performed.

Note: Border of the block and border of the cell

A cell can be stuck to a report block. In this case, the block may (or may not) be automatically resized when the cell is resized.

Indeed, the mouse cursor differs according to the object to resize:

- a small double black arrow indicates that the cell is currently resized,
- a large green double arrow indicates that the block is 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:

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

7.4.7 Removing or deleting a column from a table

A column found in a table can be:

- **removed from the table:** the column cells are no longer associated with the table. These cells can be used independently of the table.
- **deleted:** the column cells are no longer found in the table or in the report.

To remove a column from the table:

1. Select one of the cells in the column to remove.
2. Use one of the following options:
 - "Remove from table" from the popup menu of the control.
 - "Control .. Table .. Remove from table".

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

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 options:
 - the [Del] key on the keyboard.
 - "Delete" from the popup menu of the control.
 - "Control .. Table .. Delete column".
 - "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 enables you to create reports that include edit controls. These reports can be generated in PDF format.

This way you can distribute reports, in PDF format if necessary, that allow direct user input in the specified areas: administrative forms (tax return, postal forms, etc.).

Data can be just printed or stored into a database (when the data input is done from print preview). The report becomes a data input form then.

Edit controls can be used with all the types of reports. In most cases, edit controls are used on Form reports, which use a PDF as report background for instance (See “Using a form in a report”, page 58).

8.2 How do I proceed ?

8.2.1 Creating the report with edit controls

To create an editable report:

1. Create a report based on your needs (See “Creating a report”, page 29).
2. In the areas where data entry must be performed, create edit controls.
3. Save the report.
4. During printing, the user will be able to enter data directly in the edit controls. Input is available:
 - in the print preview.
 - in 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 the control (text, numeric, date, time),
- the associated mask: the masks offered 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 the caption and edit area.

Notes:

- Multiline edit control: if the users enter more lines than it is visible, the lines that are not visible will not be printed. During the input, the edit controls manage a vertical scrollbar.
- The edit control is not linked to an item from a file or a query, nor to a variable. To display information coming from a file, a query or a variable, the control must be assigned by programming.
- The border of the input area cannot be configured (thickness, color, etc.).
- You cannot use a currency type (the Euro symbol is not supported).



8.2.3 Processes associated with an edit control in a report

The processes associated with an edit control in a report are as follows:

Process	Runtime conditions
Initialization	Run when the report is opened.
Pre-print	Run before printing the control.
Entry	Run when the input cursor enters in the control.
Exit	Run when changing control (TAB key, change of control with the mouse, etc.).
Whenever modified	Run whenever the content of the control is modified (typing or erasing a character for instance).

8.3 Input in the preview window (WinDev only)

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

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

By default, users can move among the edit control by using the TAB key: it goes through the available controls from top to bottom and left to right.

Notes:

- The processes associated with the edit controls are run when the controls are handled in the print preview.
- Data from the database can be displayed by programming. To do so, link the report to a data file or a query, then assign the requested item to the edit control in the "Pre-print" process of the edit control.
- By programming, you can save data modified by an end user in the preview window. Let's look at 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 control" process.

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

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





8.4 Input in a PDF file

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

Notes:

- The edit controls enable information to be entered in the PDF file. This information will be used when printing. If the end user owns a complete version of Adobe Acrobat, the data entered can be saved in PDF format (Acrobat Reader does not allow to save data in PDF files).
- Data from the database can be displayed by programming. To do so link the report to a data file or a query, then assign the item requested to the edit control in the "Initialization" or "Pre-print" process of the edit control.

8.5 Input in a report based on a PDF form

Edit controls are well suited for "Form" reports. "Form" reports are based on an image file or a PDF file.

In the case of a PDF form containing several pages, you can create a report displaying the different pages of the PDF file ("Print all the PDF pages" option in the "Background" tab of the report description).

These pages are visible and can be modified in the report editor: you can position the edit controls on any page of the report.

See "Using a form in a report", page 58, for more details.







9. NESTED REPORTS

9.1 Overview

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 an order (number, date, customer, and so on).
- a report displaying the products of the current order.

Details of each order

Order # : 1

Order Date	Customer #	Amount
03/19/1995	2	7 928,16

Ref.	Description	Quantity	Amount
0006950PE	KNICKERS DIM WHITE 14	1	286,54
347452DIA	KNICKERS WHITE DIM 16	2	731,60
0054590OZ	UNDERWIRED BRA SILTEX WHITE	3	168,62
001629LHO	KNICKERS WHITE 16	2	170,50
347184M-M	UNDERWIRED BRA DIM WHITE 38	3	6 591,00
		11	7 928,16

Number of relevant products: 5

Order # : 2

Order Date	Customer #	Amount
04/27/1995	1 104	313 243,20

Ref.	Description	Quantity	Amount
344074-I	UNDERPANTS BLACK L	3	313 243,20
		3	313 243,20

Number of relevant products: 1

Note: The report editor also enables you to print secondary reports in a main report via the "Internal Report" controls. See "Internal Report control", page 84, for more details.

9.2 Creating a nested report

9.2.1 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. These reports can be oriented differently than the main report.



3. Printing the secondary reports in the main report with *iPrintReport*. This function must be used:

- in the "Pre-print" process of the block: the secondary report will be printed **before** this block.
- in the "Post-print" process of the block: the secondary report will be printed **after** this block.

If the nested reports have different orientations, you need to start the printing of the nested report in the "After printing page footer" code.

4. Configuring the page header and footer printing for the nested report (*iNestedHeaderFooter*). This function can be used:

- in the opening code of the main report: it will apply to all the nested reports that will be printed from the main report.
- directly before printing the nested report: it will only apply to the next nested report.

5. Printing the main report. See "Printing a nested report", page 115, for more details.

9.2.2 Example

The operations required to create nested reports are presented in the following example:



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

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

The "ReportProductOrdered" report is based on the "QueryProductOrdered" query. This report only prints the products selected by the "QueryProductOrdered" query.

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

1. Create a report named "OrderReport" ("File .. New .. Report"). This report corresponds to the main report.

2. Create a query named "QueryProductOrdered". See "How do I create a query?", page 170, for more details on how to create a query.

3. Create a report named "ReportProductOrdered" ("File .. New .. Report"). This report corresponds to the secondary report.

Note: The *Page Footer* block of this report will not be printed. See "Blocks in the secondary report", page 115, for more details about the blocks of the nested report.

4. Enter the following code in the "Post-print" process of the *Break Footer* block of "OrderReport":

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





9.3 Specific features of secondary reports

9.3.1 Blocks in the secondary report

When a secondary report is printed (with *iPrintReport*), all the blocks of the secondary report are printed except for:

- the *Page Header* block.
- the *Page Footer* block.

To print these blocks, 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 (of the secondary report).

9.3.2 Orientation of secondary reports

The secondary report and the main report can have different orientation (portrait/landscape). These orientations are applied in the print preview and in the direct printout.

In this version, the different orientations are not supported in the report export formats (PDF, RTF, etc.).

9.3.3 Page break in the secondary report

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

9.3.4 Secondary report based on a query with parameters

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

9.3.5 Secondary report based on a query that uses 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 used before *iPrintReport*.

9.4 Printing a nested report

To print a main report containing several secondary reports, you must:

1. Configure the destination of the main report:
 - with *iPreview* (print preview, print to an HTML file, and so on) in WinDev.
 - with *iDestination* (print to an HTML file, and so on) in WebDev or WinDev Mobile.
2. Specify the name of the 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 name of the report.
- *iListNestedReports* lets you get the list of the nested secondary reports in a main report.



Example:

The "OrderReport" report contains a main report and a secondary report. The "OrderReport" report is printed when the "Print" button is clicked. In this example, the click code of the "Print" button is as follows:

WinDev code:

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

WebDev code:

```
// Configures the destination of the printout
iDestination(iHTML, fWebDir() + "\Report.htm")
iPrintReport(OrderReport)
// Send the file to the browser
FileDisplay(fWebDir() + "\Report.htm")
```

WinDev Mobile code:

```
// Configures the destination of the printout
iDestination(iPCL, "\\MyNetwork\MyPrinter")
// Print the OrderReport report
iPrintReport(OrderReport)
```

9.5 Differences between the nested reports and the internal reports

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

- *Position of the secondary reports:*

In nested reports, the secondary reports are printed one after another. The reports can have different orientations. A report in portrait mode can be nested in a report in landscape mode.

In 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 can be reports with parameters. The parameters are passed when the nested report is printed (*iPrintReport*).

In an internal report, the secondary reports cannot be reports with parameters.

See "Report with parameters", page 143, for more details.

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

- internal reports. See "Internal Report control", page 84, for more details.
- reports based on queries. See "Report based on a query", page 48, for more details.





10. PRINTING A REPORT BY PROGRAMMING



Important

A report can be printed in Reports & Queries:

- by starting a print job ().
- by running the test of the report ( or "Code .. Run the report test").

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, the application's provider must have implemented this feature in the application.

The provider can for instance include a window allowing the end user to:

- select a report created in Reports & Queries.
- print this report.

10.1 Overview

Report creation (page 28) and report testing (page 40) were presented in previous chapters. This report can also be included in your applications or in your sites. The user or the Web user will have the ability to print the report directly from a button, a menu option, etc.

For a WebDev site, the server will prepare the report in the selected format (PDF, HTML, and so on). 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, etc.

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

10.1.1 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 (print preview, direct printout, generation of an HTML page, and so on) with *iPreview* or *iDestination*.
2. Start the printout with *iPrintReport*.

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

The following paragraphs present:

- printing a report based on a data file by programming.
- printing a report based on a query by programming.
- printing a report based on a HyperFileSQL view by programming.
- printing a report based on a memory zone by programming.
- printing a report based on a table by programming.
- printing a report based on a variable by programming.
- printing a report based on a text file by programming.
- printing a report based on a SQL query via a connection by programming.



10.1.2 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 printout, generation of an HTML page, generation of a PDF file, and so on) with ***iDestination***
2. Start the printout 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.
 - etc.

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

- Write rights must have been granted to you in 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 printouts would overwrite the printouts already performed.
Use the ***GetIdentifier*** or ***TimeSys*** functions to name your reports (for example: `FileName = "Customer-Report" + TimeSys() + ".HTML"`).

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

The following paragraphs present:

- printing a report based on a data file by programming.
- printing a report based on a query by programming.
- printing a report based on a HyperFileSQL view by programming.
- printing a report based on a memory zone by programming.
- printing a report based on a table by programming.
- printing a report based on a variable by programming.
- printing a report based on a text file by programming.
- printing a report based on a SQL query via a connection by programming.

10.1.3 Principle for printing in WinDev Mobile

The principle for printing a report in WinDev Mobile 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 printout parameters:
 - Printing on the printer selected by the user.
 - Printing on a printer selected during the development.
2. Printing the report.

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



**Notes:**

- The result of the printout differs according to the selected printer.
- In Windows Pocket 2002, the available memory is not sufficient to print reports.

Case 1: Printing on a printer selected by the user

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

To perform this type of printout, you must:

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

Example:

The CustomerReport report will be printed on the printer selected by the user in the configuration window.

This printer will be automatically used for all the other printouts performed by this application.

Corresponding code:

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

Case 2: Printing on a printer directly

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

- direct printout by defining the type of printer used.
- printing on one of the printers referenced on the Pocket PC.

Printing on a type of printer

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

1. Configure the print destination with *iDestination*. This enables you to specify whether the printout is performed in a PCL file, on a printer (infrared, WiFi, Bluetooth or network).
2. Use *iParameter* with "PrinterType" 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 CustomerReport report will be printed on the DeskJet 350 printer, connected by infrared to the Pocket PC.

Corresponding code:

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

Printing on a printer referenced on the Pocket PC

To print on a printer referenced on the Pocket PC (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 CustomerReport report will be printed on the "PrntInfra" printer configured on the Pocket PC.

This printer will be automatically used for all the other printouts performed by this application.

Corresponding code:

```
// Select the printer by programming
iConfigure("PrntInfra")
// Print the report based on the "CustomerTable" table
iPrintReport(CustomerReport)
```

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

The following paragraphs present:

- printing a report based on a data file by programming.
- printing a report based on a query by programming.
- printing a report based on a HyperFileSQL view by programming.
- printing a report based on a table by programming.
- printing a report based on a text file by programming.





10.2 Report based on a data file

Printing a report based on a data file uses the basic principle described above. However, specific information (bounds, parameters, and so on) can be taken into account by a report based on a data file. In this case, the report printout must be adapted. The following paragraphs present:

- printing a report based 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 based on a data file.

10.2.1 Printing a report based on a data file

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

1. Configure the destination of the printout:
 - In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
 - In WebDev: use *iDestination* (print to an HTML file, and so on).
 - In WinDev Mobile: use *iConfigure* or *iParameter*. These functions open a dialog box allowing the user to select the default printer.
2. Specify the name of the report to print in *iPrintReport*.

WinDev and WinDev Mobile example:



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

The "CustomerReport" report is printed when the "Customer" button is clicked.

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

WinDev code:

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

WinDev Mobile code:

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

WebDev example:



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

The "CustomerReport" report is displayed in a PDF file when the "Customer" link is clicked. Then, the PDF file is displayed in the browser.



In this example, the server click code of the "Customer" link is as follows:

```
// Generates a unique PDF file name
File is string = fWebDir()+"\"+"CustomerReport_" + ...
    DateSys()+TimeSys()+".pdf"
// Configures the destination of the printout
iDestination(iPDF, File)
// Prints the RptCustomer report
iPrintReport(CustomerReport)
// Sends the file to the browser
FileDisplay(File, "application/pdf")
// Deletes the file
fDelete(File)
```

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

10.2.2 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 the printout:

- In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
- In WebDev: use *iDestination* (print to an HTML file, and so on).
- In WinDev Mobile: use *iConfigure* or *iParameter*. These functions open a dialog box allowing the user to select the default printer.

2. Specify the following parameters in *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 "CustomerReport" 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 "CustomerReport" report is printed when the "CustomerD" button is clicked. In this example, the click code of the "CustomerD" button is as follows:

WinDev code:

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

WinDev Mobile code:

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





WebDev example:



The "CustomerReport" 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 "CustomerReport" report is printed in an HTML file and sent to the browser when the "CustomerD" button is clicked.

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

```
// Generates a unique file name
File is string = fWebDir()+"\"+"CustomerReport_" + ...
    DateSys()+TimeSys()+".pdf"
// Configures the destination of the printout
iDestination(iPDF, File)
// Prints the RptCustomer report
iPrintReport(CustomerReport,"D","D"+hMaxVal)
// Sends the file to the browser
FileDisplay(File, "application/pdf")
// Deletes the file
fDelete(File)
```

10.2.3 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 the printout:
 - In WinDev: use **iPreview** (print preview, print to an HTML file, and so on).
 - In WebDev: use **iDestination** (print to an HTML file, and so on).
 - In WinDev Mobile: use **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
2. Specify the following parameters in **iPrintReport**:
 - the name of the report to print.
 - the different parameters expected by the report (in the order specified in the report).

WinDev example:



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

- the text of the mailshot entered in the edit control named "MailshotText".
- the date for sending this mailshot entered in the edit control named "SendDate".

This report is printed when the "Print" button is clicked. The "ParamReport" report is based on the "Customer" file.

In this example:

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

```
// Open the preview window
iPreview()
// Print the report and pass parameters
iPrintReport(ParamReport, MailshotText, SendDate)
```



- the code of the report is as follows:

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

WinDev Mobile example:



The "ParamReport" corresponds to an invoice. This report expects in parameter the delivery date entered in the "DelDate" edit control.

This report is printed when the "Print" button is clicked. The "ParamReport" report is based on the "Customer" file.

In this example:

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

```
// Select the printer
iConfigure()
// Print the report and pass parameters
iPrintReport (ParamReport, DelDate)
```

- the code of the report is as follows:

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

WebDev example:



The "OrderReport" report is a report with parameters used to print the order of a customer identified by a customer number.

The "OrderReport" report is displayed in a PDF file. This PDF file is sent to the browser when the "Order" button is clicked.

In this example:

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

```
// Generates a unique file name
File is string = fWebDir()+ "\"+ "Invoice_" + ...
                DateSys()+TimeSys()+".pdf"
// Configures the destination of the printout
iDestination(iPDF, File)
// Generates the order form
iPrintReport (OrderReport, CUSTOMER.CUSTOMERID)
```

```
// Sends the file to the browser
FileDisplay(File, "application/pdf")
```

- the code of the report is as follows:

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





10.2.4 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 the printout:

- In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
- In WebDev: use *iDestination* (print to an HTML file, and so on).
- In WinDev Mobile: use *iConfigure* or *iParameter*. These functions open a dialog box allowing the user to select the default printer.

2. Specify the following parameters in *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 different parameters expected by the report (in the order specified in the report).

WinDev example:



The "ParamReport" report corresponds to a mailshot 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 text of the mailshot entered in the edit control named "MailshotText".
- the date for sending this mailshot entered in the edit control named "SendDate".

This report is printed when the "Print" button is clicked. The "ParamReport" report is based on the "Customer" file.

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

```
// Open the preview window
iPreview()
// Print the report while passing bounds
// and parameters
iPrintReport(ParamReport, "D","E", MailshotText, SendDate)
```

WinDev Mobile example:



The "ParamReport" 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 "DelDate" edit control.

This report is printed when the "Print" button is clicked. The "ParamReport" report is based on the "Customer" file.

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

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



WebDev example:

The "ParamReport" report is used to print the invoices of a specific customer for January 2008.

The lower bound corresponds to "01/01/2008" and the upper bound to "01/31/2008".

This report expects the customer number in parameter.

This report is displayed in a PDF file when the "Print" button is clicked.

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

```
// Generates a unique file name
File is string = fWebDir()+"\"+"ParamReport_" + ...
    DateSys()+TimeSys()+".pdf"
// Configures the destination of the printout
iDestination(iPDF, File)
// Prints the report while passing bounds and parameters
iPrintReport(ParamReport, "20080101", "20080131", CustNum)
// Sends the file to the browser
FileDisplay(File, "application/pdf")
```

10.3 Report based on a query

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

- Report based on a query without parameter.
- Report based 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 the report to print.
- **SourceName** used in the code of the report is used to find out and modify the name of the query associated with the report.

10.3.1 Printing a report based on a query without parameter

To print a report based on a query without parameter, you must:

1. Configure the destination of the printout:
 - In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
 - In WebDev: use *iDestination* (print to an HTML file, and so on).
 - In WinDev Mobile: use *iConfigure* or *iParameter*. These functions open a dialog box allowing the user to select the default printer.
2. Specify the name of the report to print in *iPrintReport*.

For example:

- To print the report named "QueryReport" in WinDev:

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





- To print the report named "QueryReport" in WinDev Mobile:

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

- To print the report named "QueryReport" in WebDev, in a file in HTML format:

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

10.3.2 Printing a report based on a query with parameters

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

1. Configure the destination of the printout:
 - In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
 - In WebDev: use *iDestination* (print to an HTML file, and so on).
 - In WinDev Mobile: use *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 the report to print in *iPrintReport*.

For example: The report named "QueryReport" 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 printout.

WinDev code:

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



WinDev Mobile code:

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

WebDev code:

```
// Generates a unique file name
File is string = fWebDir()+"\"+"QueryReport_" + ...
                DateSys()+TimeSys()+".htm"
// Configures the destination of the printout
iDestination(iHTML, File)
// Pass the parameters to the query associated with the
// report
iInitReportQuery(QueryReport, CompanyName, CompanyType)
// Prints the QueryReport report
iPrintReport(QueryReport)
// Extracts the name and extension of the generated file
FileName is string = fExtractPath(File, ...
                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 instance), the selection conditions corresponding to these parameters will be ignored.

10.4 Report based on a table control

A report based on a Table control can only 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 in the page containing the Table control.
2. Configure the destination of the printout:
 - In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
 - In WebDev: use *iDestination* (print to an HTML file, and so on).
 - In WinDev Mobile: use *iConfigure* or *iParameter*. These functions open a dialog box allowing the user to select the default printer.
3. Specify the name of the report to print in *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. See "Report with parameters", page 143, for more details.
- *SelectionOnly* is used to find and modify the records that will be printed in the report: all the records or the selected records only.





WinDev and WinDev Mobile example:



The information to print is found in the Table control named "CustomerTable" found in the "CustomerWin" window.

The report to print is named "CustomerReport" and it is printed when the "Print" button is clicked.

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

WinDev code:

```
// Open the preview window
iPreview()
// Print the report based on the "CustomerTable" table
iPrintReport(CustomerReport)
```

WinDev Mobile code:

```
// Select the printer
iConfigure()
// Print the report based on the "CustomerTable" table
iPrintReport(CustomerReport)
```

WebDev example:



The information to print is found in the Table control named "CustomerTable" found in the "CustomerPage" page.

The report to print is named "CustomerReport" and it is displayed in a PDF file when the "Print" button is clicked.

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

```
// Generates a unique file name
File is string = fWebDir()+"\"+"CustomerReport_" + ...
    DateSys()+TimeSys()+".pdf"
// Prints the report in a PDF file
iDestination(iPDF, File)
// Prints the report based on the "CustomerTable" table
iPrintReport(CustomerReport)
// Displays the report in PDF format
FileDisplay(File, "application/pdf")
// Deletes the file
fDelete(File)
```



10.5 Report based on a variable

10.5.1 Printing a report based on a variable

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

1. Initialize the value of the array variable used.
2. If needed, sort the array (with the **ArraySort** function).
3. Configure the destination of the printout:
 - In WinDev: use **iPreview** (print preview, print to an HTML file, and so on).
 - In WebDev: use **iDestination** (print to an HTML file, and so on).
 - In WinDev Mobile: use **iConfigure** or **iParameter**. These functions open a dialog box allowing the user to select the default printer.
4. Specify the name of the report to print with **iPrintReport**.

WinDev and WinDev Mobile example:



The "ReportggEvent" report is a report based on the ggEvent type variable named MyEvents. This report is used to print the list of events found in a Google calendar.

The "ReportggEvent" report is printed when clicking the "Event" button. In this example, the click code of the "Event" button is as follows:

WinDev code:

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

WinDev Mobile code:

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

WebDev example:



The "ReportggEvent" report is a report based on the ggEvent type variable named MyEvents. This report is used to print the list of events found in a Google calendar.

The "ReportggEvent" report is displayed in a PDF file when the "Event" link is clicked. Then, the PDF file is displayed in the browser.

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

```
// Generates a unique file name
File is string = fWebDir()+"\"+ ...
                "ReportggEvent_" + DateSys()+TimeSys()+".pdf"
// Prints the report in a PDF file
iDestination(iPDF, File)
// Prints the ReportggEvent report
iPrintReport(ReportggEvent)
// Displays the report in PDF format
FileDisplay(File, "application/pdf")
```





```
// Deletes the file
fDelete(File)
```

Notes:

- **SourceName** used in the code of the report is used to find out and modify the name of the variable associated with the report.
- The variable must have been created and initialized before you start printing the report.

10.5.2 Printing a report with parameters based on a variable

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

1. Initialize the value of the array variable used.
2. If needed, sort the array (with **ArraySort**).
3. Configure the destination of the report printout:
 - In WinDev, with **iPreview** (print preview, printout in an HTML file, etc.).
 - In WebDev, with **iDestination** (print to an HTML file, etc.).
 - 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 in **iPrintReport**:
 - the name of the report to print.
 - the different parameters expected by the report (in the order specified in the report).

WinDev and WinDev Mobile example:

A window lists the calendars and their events. The "ReportgglEvent" report is printed when clicking the "Print the calendar events" button. This report expects as parameter the calendar for which the events must be listed.

WinDev code:

```
// Open the preview window
iPreview()
// Print the ReportgglEvent report
// with parameter passing
iPrintReport(ReportgglEvent, Calendar[1])
```

WinDev Mobile code:

```
// Select the printer
iConfigure()
// Print the ReportgglEvent report
// with parameter passing
iPrintReport(ReportgglEvent, Calendar[1])
```

WebDev example:

The "ReportgglEvent" report is displayed in a PDF file when the "Print the calendar events" link is clicked. Then, the PDF file is displayed in the browser. This report expects as parameter the calendar for which the events must be listed.



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

```
// Generates a unique file name
File is string = fWebDir()+"\"+ ...
    "ReportgglEvent" + DateSys()+TimeSys()+".pdf"
// Prints the report in a PDF file
iDestination(iPDF, File)
// Prints the ReportgglEvent report
iPrintReport(ReportgglEvent, MyCalendar)
// Displays the report in PDF format
FileDisplay(File, "application/pdf")
// Deletes the file
fDelete(File)
```

Notes:

- The variable must have been created and initialized before you start printing the report.
- If additional parameters must be passed to the report, these parameters must be specified in *iPrintReport*. See "Report with parameters", page 143, for more details.
- The *SourceName* property in the code of the report is used to find out and modify the variable associated with the report.

10.6 Report based on a HyperFileSQL Classic view

To print a report based on HyperFileSQL Classic view:

1. Create the HyperFileSQL view with *HCreateView* (or *HMergeView*).
2. Configure the destination of the printout:
 - In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
 - In WebDev: use *iDestination* (print to an HTML file, and so on).
 - In WinDev Mobile: use *iConfigure* or *iParameter*. These functions open a dialog box allowing the user to select the default printer.
3. Specify the following parameters in *iPrintReport*:
 - the name of the report to print.
 - the name of the HyperFileSQL Classic view used.

Notes:

- The HyperFileSQL view must be created and initialized before the report is printed. Only a HyperFileSQL Classic view or HyperFileSQL Client/Server view can be used.
- If additional parameters must be passed to the report, these parameters must be specified in *iPrintReport*, after the name of the HyperFileSQL view used. See "Report with parameters", page 143, for more details.
- The *ViewName* property used in the report's code lets you find out and change the HyperFileSQL view associated with the report.
- The *SourceName* property in the code of the report is used to find out and modify the name of the data file from the HyperFileSQL view associated with the report.





WinDev example:



The user selects the reference of an order in a list. The "Print" button is used to:

- create the HyperFileSQL view named "OrderView" containing the selected order lines (**HCreateView**).
- open the preview window (**iPreview**).
- print the "ViewReport" report (**iPrintReport**).

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

```
// Create the view
IF NOT HCreateView("OrderView", ORDERLINE, "*", ...
  "Reference","", "", hViewExclusive + hViewBreakable) THEN
  Error(HErrorInfo())
ELSE
  // Open the preview window
  iPreview()
  // Print the ViewReport report
  iPrintReport(ViewReport,"OrderView")
END
```

WinDev Mobile example:



The user selects the reference of an order in a list. The "Print" button is used to:

- create the HyperFileSQL view named "OrderView" containing the selected order lines (**HCreateView**).
- print the "ViewReport" report (**iPrintReport**).

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

```
// Create the view
IF NOT HCreateView("OrderView", ORDERLINE, "*", ...
  "Reference","", "", hViewExclusive + hViewBreakable) THEN
  Error(HErrorInfo())
ELSE
  // Select the printer
  iConfigure()
  // Print the ViewReport report
  iPrintReport(ViewReport,"OrderView")
END
```

WebDev example:



The Web user selects the reference of an order in a list. The "Print" button is used to:

- create the HyperFileSQL view named "OrderView" containing the selected order lines (**HCreateView**).
- view the "ViewReport" report in a PDF file (**iPrintReport**).

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



```

// Create the view
IF NOT HCreateView("OrderView", ORDERLINE, "*", ...
    "Reference","", "", hViewExclusive + hViewBreakable) THEN
    Error(HErrorInfo())
ELSE
    // Generate a unique file name
    File is string = fWebDir()+"\"+"ViewReport_" + ...
        DateSys()+TimeSys()+".pdf"
    // Configure the destination of the printout
    iDestination(iPDF, File)
    // Print the ViewReport report
    iPrintReport(ViewReport, "OrderView")
    // Send the file to the browser
    FileDisplay(File, "application/pdf")
END

```

10.7 Report based on a memory zone

To print a report based on a memory zone, you must:

1. Create the memory zone with **MemCreate**.
2. Add records to the memory zone with **MemAdd** and sort these records (if necessary) with **MemSort**.
3. Configure the destination of the printout:
 - In WinDev: use **iPreview** (print preview, print to an HTML file, and so on).
 - In WebDev: use **iDestination** (print to an HTML file, and so on).
4. Specify the following parameters in **iPrintReport**:
 - the name of the report to print.
 - the name of the memory zone used.
5. Delete the memory zone with **MemDeleteAll** at the end of report printout.

Notes:

- To optimize the resources used on a Pocket PC, printing a report based on a memory zone is not available in WinDev Mobile.
- If additional parameters must be passed to the report, these parameters must be specified in **iPrintReport**, after the name of the memory zone used. See "Report with parameters", page 143, for more details.
- **MemZoneName** used in the code of the report is used to find out and modify the name of the memory zone associated with the report.
- **SourceName** used in the code of the report is used to find out and modify the name of the base file for the memory zone associated with the report.



**WinDev example:**

The user selects the customers to whom a promotional offer must be sent. The name of the selected customers is found in the list box named "Recipient". Once all the customers have been selected, the user clicks the "Print" button to print the "MemZoneReport" report.

When a click is performed on the "Print" button:

- the memory zone named "MemZone" is created.
- the record number of each selected customer is retrieved and added to "MemZone".
- the "ReportMemZone" report is printed.
- the memory zone named "MemZone" is deleted.

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

```
nSubscript is int
// Create the memory zone
MemCreate("MemZone")
// Browse the Recipient list and
// initialize the memory zone
FOR nSubscript = 1 TO Recipient..Occurrence
  // Find the current record
  HReadSeek(Customer, Name, Recipient[nSubscript])
  IF HFound(Customer) THEN
    // Add elements to the memory zone
    MemAdd("MemZone", "", HRecNum())
  END
END
// Open the preview window
iPreview()
// Print the report
iPrintReport(MemZoneReport, "MemZone")
// Delete the memory zone
MemDeleteAll("MemZone")
```

WebDev example:

The Web user selects the customer to whom a promotional offer must be sent. The name of the selected customers is found in the list box named "Recipient".

Once all the customers have been selected, the Web user clicks the "Print" button to print the "MemZoneReport" report.

When a click is performed on the "Print" button:

- the memory zone named "MemZone" is created.
- the record number of each selected customer is retrieved and added to "MemZone".
- the "ReportMemZone" report is displayed in a PDF file.
- the memory zone named "MemZone" is deleted.



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

```
nSubscript is int
// Create the memory zone
MemCreate("MemZone")
// Browse the Recipient list and
// initialize the memory zone
FOR nSubscript = 1 TO Recipient..Occurrence
// Find the current record
HReadSeek(Customer,Name,Recipient[nSubscript])
IF HFound(Customer) THEN
// Add elements to the memory zone
MemAdd("MemZone", "", HRecNum())
END
END
// Generates a unique file name
File is string = fWebDir()+"\"+"MemZoneReport_" + ...
DateSys()+TimeSys()+".pdf"
// Configures the destination of the printout
iDestination(iPDF, File)
// Prints the report
iPrintReport(MemZoneReport, "MemZone")
// Delete the memory zone
MemDeleteAll("MemZone")
// Sends the file to the browser
FileDisplay(File, "application/pdf")
```

10.8 Report based on a text file

Printing a report based on a text file is identical regardless of the method used to read the records (automatic or programmed). All you have to do is:

1. Configure the destination of the printout:
 - In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
 - In WebDev: use *iDestination* (print to an HTML file, and so on).
 - In WinDev Mobile: use *iConfigure* or *iParameter*. These functions open a dialog box allowing the user to select the default printer.
2. Specify the name of the report to print in *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. See "Report with parameters", page 143, for more details.
- **SourceName** used in the code of the report is used to find out and modify the name of the text file associated with the report (when the records are automatically read).

WinDev and WinDev Mobile example:



The "CustomerReport" report is based on the "Customer.TXT" file. This text file contains the characteristics of the customers.

The "CustomerReport" report is printed when the "Customer" button is clicked.





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

WinDev code:

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

WinDev Mobile code:

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

WebDev example:



The "CustomerReport" report is based on the "Customer.TXT" file. This text file contains the characteristics of the customers. This file is accessible from the server.

The "CustomerReport" report is displayed in a PDF file when the "Customer" button is clicked.

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

```
// Generates a unique file name
File is string = fWebDir()+"\"+"CustomerReport_" + ...
    DateSys()+TimeSys()+".pdf"
// Prints the report in a PDF file
iDestination(iPDF, File)
// Prints the RptCustomer report
iPrintReport (CustomerReport)
// Displays the report in PDF format
FileDisplay(File, "application/pdf")
// Deletes the file
fDelete(File)
```

10.9 Printout functions

The following functions can be used to easily print reports:

Functions	Effect
iInitReportQuery	Initializes the query linked to the report (for a query with parameters)
iNestedHeaderFooter	Prints (or not) the page headers and footers of the nested report
iPreview / iDestination	Configures the destination of the printout
iPrintReport	Prints a report created in the report editor

See the online help for more details.

Note: These functions are only required to print reports. Other print functions can be used to perform specific processes. These functions are presented in "WLanguage functions", page 148.







11. PROGRAMMING A REPORT

11.1 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.

WLanguage functions and properties can be used in these processes. Some functions and properties specific to reports are presented in this chapter.

See the online help (keyword: "Processes of a report") for more details.

11.2 Sequence of processes associated with a report

Several operations are performed when a report is printed. These operations are performed in the following order:

1.	Run the "Opening" process of the report
2.	Run the "Initialization" process of the blocks and the "Initialization" process of the controls
3.	Initialize the data source
4.	For the reports based on a file, a view or a memory zone: <ul style="list-style-type: none"> • Run the "Filter" process of the report • Run the procedures for sorting the report
5.	Initialize the margins and define the Portrait/Landscape mode
6.	Resize the report (if necessary) according to the specified format
7.	Print the <i>Start of Document</i> block and run its processes *
8.	Print the <i>Page Header</i> block and run its processes for each page of the report *
9.	Print loop: <ul style="list-style-type: none"> • Read the record • Run 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. • Perform calculations on the values of the items • Run 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 break and print one or more <i>Break Header</i> and <i>Break Footer</i> blocks. • Print the <i>Body</i> block <p>Note: When printing a <i>Break Header</i>, <i>Break Footer</i> or <i>Body</i> block, the following operations are performed:</p> <ul style="list-style-type: none"> • Run the "Pre-print" process of the block and the "Pre-print" process of the block controls • Perform calculations on the block controls • Resizing the auto-sizable controls and blocks • Print the block and run the "Post-print" process of the block



10	Print the <i>End of Document</i> block and run its processes *
11.	Print the <i>Page Footer</i> block and run its processes for each page of the report *
12.	Run the "Closing" process of the report

* The order for printing the *Start of Document*, *Page Header*, *Page Footer* and *End of Document* blocks can be modified in the "Options" tab of the report description ("Report .. Description").

Reminder: If the print destination is an HTML file or an XML file, the *Page Header* and *Page Footer* blocks are generated once only.

For more details about:

- the processes associated with the blocks, See "Processes of the report blocks", page 71.
- the processes associated with the blocks, See "Processes of the report controls", page 95.
- the different types of blocks, See "The different types of blocks", page 66.
- the sort procedures, See "Sort procedure", page 142.

11.3 Processes associated with the report

The processes associated with the reports are as follows (order of appearance in the code editor):

- Opening
- Reading the data
- Filter
- Closing

These processes are run when the report is printed.

11.3.1 "Opening" process

The "Opening" process is run when starting the report printout.

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 the data files if necessary.
- configure the print destination with *iPreview* or *iDestination*.
- pass parameters to the query linked to the report (*ilnitReportQuery*).
- retrieve the parameters passed to the report.
- change the data source of the report.

Password of the source file of the report

If the source file of the report is password-protected, this password must be specified before the file is opened (with *HPass* for instance). If the file has not been opened before the report is printed, 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 if:

- the user changes printer.
- the user uses one of the toolbar buttons (print to PDF, HTML, email, and so on).





In this case, the data printed in the report may differ from the data displayed in the preview.

The variables used by the report should be initialized in the "Report opening" process rather than in the code of the button that starts the printout.

11.3.2 "Reading data" process

The "Reading data" process is run when browsing the records to print (in the print loop), once the record is read.

Use examples

This process is used to:

- perform block substitutions (as the next record to print is known).
- select some records.
- perform specific calculations.

Note: this code returns True if there are still some records to print and False if there are no more records to print.

Reports based on a text file

For the reports based on a text file (with programmed read operation), the "Reading data" process is used to read the records found in the text file in order to print them.

11.3.3 "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 displayed in the report.

Use examples

This process is used to:

- select some records.
- perform advanced calculations (sub-total, and so on) for the reports based on files, views or memory zones.

Using the "Filter" process

If this process contains code lines, these code lines must return a value:

- Cancel to stop reading the records (and to stop printing).
- True to specify that the record corresponds to the filter and that it must be printed.
- False to specify that the record does not correspond to the filter and that it must not be printed.

11.3.4 "Closing" process

This process is run once the report printout is over, just before closing the report. This process should not contain any print functions.

Use examples

This process is used to:

- find out the status of the current printout (*iReportPrintingStatus*).
- open a dialog box.



11.4 Break process

If your report contains one or more breaks, the break process is associated with the *Break Header* block.

This process enables you to use a programmed break. Two types of programming can be implemented for the breaks:

- the code of the break returns True or False (True to perform the break, False not to perform the break).
- the code of the break returns the value of the break. If the value changes, the break is automatically performed by the report (see the example below).

The value of the break can correspond to the value of an item, a report control, a variable, etc.

Note: If the value of the break corresponds to the value of a report control, the returned value will be false if this control is found in the *Body* block of the report. Indeed, the break process is run before the controls found in the *Body* block are filled. In this case, you must return the value of the item linked to the control.

```
-- Code of the break
nSupplier is int = 0
// Retrieve the identifier of the supplier
nSupplier = ExtractString(gfsLine, 1, TAB)
// Find the name of the supplier
HReadSeekFirst(SUPPLIER, SupplierNum, nSupplier)
IF HFound() THEN
    // Initialize the control
    SupplierName = SUPPLIER.Company
END
// Return the identifier of the supplier
RESULT nSupplier
```

Important: The break code is run **only** if the corresponding break is a programmed break (not linked to a sort criterion).

See the online help (keyword: "Break in a report") for more details.

11.5 Sort procedure

Also, a sort procedure can be implemented in a reports 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. Select "Report .. Edit the sorts". The list of sort criteria is displayed.
2. Click "New sort criterion".
3. Check "Sort on WLanguage code".
4. Enter the name and value of the parameters for the sort procedure (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
```

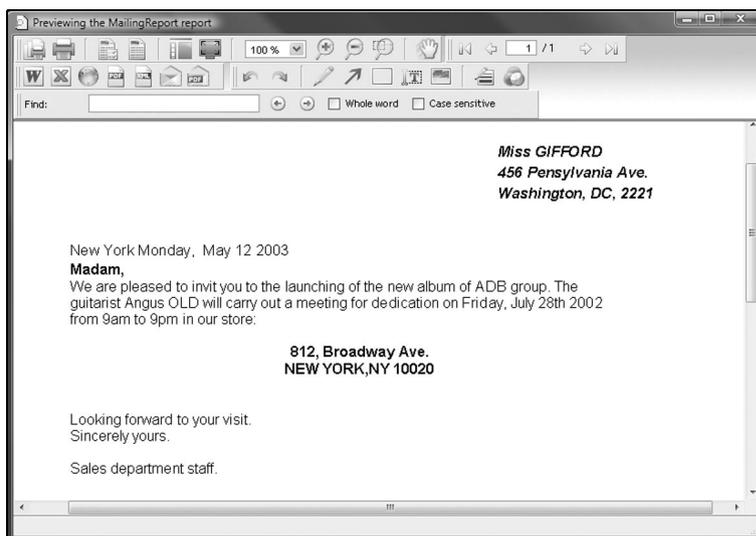




11.6 Report with parameters

When printing a report, some parameters can be passed to this report. The report is then 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 this report is run. This text corresponds to a parameter expected by the report.



11.6.1 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 parameters passed to a procedure may be optional. When declaring the procedure, the optional parameters must be described in last position (to the right), with a default value preceded by "=". See the online help (keyword: "Parameters, Parameters or a procedure") for more details.

- Using these parameters in the report controls.
3. Modifying (if necessary) the application or the site in order to specify the different parameters.
 4. Printing the report with its parameters.



Example:

The program required to create a report with parameters is illustrated by the following example:



The "ParamReport" report corresponds to a mailshot sent to the customers. This report expects in parameter the text of the mailshot entered in the "MailshotText" edit control.

1. Create the report named "ParamReport" ("File .. New .. Report .. Mailshot"). This report is linked to the "Customer" file of your analysis.
2. Customize this report.
3. Enter the following code in the "Opening" process of "ParamReport":

```
// Retrieve the parameter passed to the report and
// pass a default parameter for the test of the report
PROCEDURE ParamReport(sText = "Default Value")
```

4. In the *Body* block of this report, add the static control named "MailshotText" used to display the text of the mailshot.
5. Enter the following code in the "Pre-print" process of the "MailshotText" control:

```
// Retrieve the text of the mailshot
MailshotText = sText
```

6. In an application window or in a site page, add:
 - the edit control named "MailshotText" allowing the user to enter the text of the mailshot.
 - the "Print" button used to print the report.

11.6.2 Printing a report with parameters

To print a report with parameters, you must:

1. Configure the destination of the printout:
 - In WinDev: use *iPreview* (print preview, print to an HTML file, and so on).
 - In WebDev: use *iDestination* (print to an HTML file, and so on).
 - In WinDev Mobile: use *iConfigure* or *iParameter*. These functions open a dialog box allowing the user to select the default printer.
2. Specify the following parameters in *iPrintReport*:
 - the name of the report to print.
 - the parameters.

Notes:

- If the report is based on a HyperFileSQL view, the name of this HyperFileSQL view must be specified in *iPrintReport*, before the parameters. See "Report based on a HyperFileSQL Classic view", page 50, 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 “Report based on a memory zone”, page 51, 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 “Bounded report”, page 48 for more details.

Example:



The "ParamReport" report corresponds to a mailshot sent to the customers. This report expects in parameter the text of the mailshot entered in the "MailshotText" edit control.

The "ParamReport" report is printed when the "Print" button is clicked.

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

WinDev code:

```
// Open the preview window
iPreview()
// Print the report and pass parameters
iPrintReport(RptParam, MailshotText)
```

WinDev Mobile code

```
//Select the printer
iConfigure()
// Print the report and pass parameters
iPrintReport(ParamReport, CopyCaption)
```

WebDev code

```
// Configure the destination of the printout
iDestination(iHTML,"ReportPM_" + GetIdentifier() + ".htm")
// Print the report with parameter passing
iPrintReport(RptParam, MailshotText)
// Send the file to the browser
FileDisplay("ReportPM_" + GetIdentifier() + ".htm")
```

11.6.3 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, the following code must be entered in the "Opening" process of "ParamReport":

```
// Retrieve the parameter passed to the report and
// pass a default parameter for the test of the report
PROCEDURE ParamReport(sText = "Default Value")
```



11.7 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:

- by a WLanguage function,
- by a WLanguage property,
- by direct assignment.

For example:

```
// Print the "ParamReport" report
iPrintReport (RptParam)
// Modify the number of copies for the "Customer" report
Customer..NbCopy = 10
```

11.8 Keywords associated with reports

The following keywords can be used in the reports:

Keyword	Effect
MySource	Enables you to handle the current data source
MyReport	Enables you to handle the current report

See the online help for more details.

11.9 Properties associated with the reports

The following properties are associated with the reports:

Properties	Effect
AllModifiableWithReportsAndQueries	Enables you to find out whether a report can be modified in Reports & Queries
BottomMargin	Enables you to find out and modify the bottom margin of a report
BrowsedItem	Enables you to find out and modify the item used to automatically browse a report based on file
HSpace	Enables you to find out and modify the horizontal spacing between the labels in a "Label" report
HTML	Enables you to find out whether the report is printed in HTML mode
InterfaceModifiableWithReportsAndQueries	Enables you to find out whether the interface of a report can be modified in Reports & Queries
InternalReportName	Enables you to find out and modify the report printed in an Internal Report control
LabelHeight	Enables you to find out and modify the height of the labels in a "Label" report





LabelNumberToStart	Enables you to find out and modify the number of the label from which the printout will start
LabelWidth	Enables you to find out and modify the width of the labels in a "Label" report
LeftMargin	Enables you to find out and modify the left margin of a report
MaxValue	Enables you to retrieve and modify the upper bound of a report based on a file (the upper bound corresponds to the maximum value allowed for the search key in the source file of the report)
MemZoneName	Enables you to find out the name of the memory zone associated with a report
MinValue	Enables you to retrieve and modify the lower bound of a report based on a file (the lower bound corresponds to the minimum value allowed for the search key in the source file of the report)
MsgSortSel	Enables you to find out and modify the message displayed when sorting and selecting the records found in the report
Name	Enables you to find out the name of a report
NbCopy	Enables you to find out and modify the number of copies to print
NbLabel	Enables you to find out and modify the number of labels printed across the width of a page
NbLabelCopy	Enables you to find out and modify the number of label copies
NbPrintedRec	Enables you to find out the number of records already printed
NbRecRead	Enables you to find out the number of records read
ODBCConnectionName	Enables you to find out and modify the name of the connection used to handle a database accessed via an ODBC driver
ODBCDatabaseName	Enables you to find out and modify the name of the database used by the query run on a database accessed via an ODBC driver
ODBCPassword	Enables you to find out and modify the password associated with the user allowed to access a database via an ODBC driver
ODBCSQLCode	Enables you find out and modify the SQL code of the query run on a database accessed via an ODBC driver
ODBCUserName	Enables you to find out and modify the password of the user allowed to access a database via an ODBC driver
PageByPagePrint	Enables you to print a report page by page
PageEndStick	Enables you to manage the position of the Page Footer block
RightMargin	Enables you to find out and modify the right margin of a report
SelectionOnly	Enables you to find out and modify the mode for printing the lines selected in a report based on a table
SourceName	Enables you to find out the name of the data source of the report and to change the data source of the report
State	Enables you to find out and modify the display status of a block in a report



TopMargin	Enables you to find out and modify the top margin of a report
TotalNbRec	Enables you to find out and modify the number of records to print
ViewName	Enables you to find out the name of the HyperFileSQL view associated with a report
VSpace	Enables you to find out and modify the vertical spacing between the labels in a "Label" report

See the online help for more details.

11.10 WLanguage functions

Several WLanguage functions can be used in the processes of a report (created in the report editor). However, some of them are not recommended, such as the print functions specific to WLanguage printout or the functions for accessing the records used by the report.

11.10.1 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 execution of the report may be incorrect.

Print functions: *iEllipse*, *iEndPrinting*, *ilmageHeight*, *iPrint*, *iPrintBarCode*, *iPrintImage*, *iPrintWord*, *iPrintZone*, *iPrintZoneRTF*, *ilmageWidth*, *iLink*, *iNewLine*, *iXPos*, *iYPos*, *iSkipLine*, *iLine*, *iHLine* and *iVLine*.

Functions for accessing a record: All the HyperFileSQL Classic functions of WLanguage that modify the record loaded in memory or the current record found in the base file of the report should not be used in the processes of reports, blocks and controls. Indeed, these functions may modify the browse operation performed in the base file of the report.

These functions are: *HAdd*, *HCreation*, *HCreationIfNotFound*, *HLast*, *HWrite*, *HReadxxx*, *HModify*, *HPrevious*, *HSetPosition*, *HFirst*, *HCross*, *HSeek*, *HBackward*, *HNext*, *HDelete*, etc.



Important

The functions for modifying records (*HAdd*, *HModify*, *HDelete*, and so on) cannot be used in Reports & Queries.

11.10.2 Print functions of WLanguage that can be used in a report

The following printout functions can be used in a report:

Functions	Effects
iBorder	Prints a border at the specified coordinates
iCloseReport	Stops the current printout immediately
iConfigure	Configures the printer





iConfigureReport	Configures the printer by opening the configuration window. This window is initialized with the parameters specified when developing the report (orientation of the report, margins, and so on)
iCreateFont	Creates a new printout font
iDidotFont	Selects the unit that will be used for the height of the fonts (created by iCreateFont): DIDOT point or PICA point
iDocumentCanceled	Enables you to find out whether the current printout was canceled by the user
iDocumentPrinted	Enables you to find out whether the user has requested to print the document from the print preview
iEllipse	Prints an ellipse inside a rectangle
iEndReport	Forces the printout of a report to stop
iEscape	Sends an ESCAPE command to a printer
iFont	Selects the default font
iForceComplement	Forces a <i>Body Complement</i> block to print
iImageHeight	Calculates the height of the image to print (in millimeters)
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 (query based on a specific connection)
iListNestedReports	Returns the list of nested reports currently printed (reports run by iPrintReport in one of the processes of the report currently printed)
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 the nested report
iPageEnd	Forces the move to the next page while a report is printed
iPageHeight	Calculates the height of the printable page (in millimeters) while taking the margins (top or bottom) into account
iPageHeightLeft	Calculates the available height (in millimeters) left on the current page while taking the margins (top or bottom) into account
iPageNum	Returns or initializes the number of the page currently printed
iPageWidth	Calculates the width (in millimeters) of the printable page while taking the margins (left or right) into account
iParameter	Configures the printout



iParameterExport	Enables you to configure the exports performed from the print preview. This function is mainly used to configure the parameters for sending emails from the print preview.
iParameterPDF	Defines the protection options for the generated PDF file
iParameterPreview	Configures the display of each button found in the print preview
iParameterReport	Configures the printer by opening the configuration window. This window is initialized with the parameters specified when developing the report (orientation of the report, margins, and so on)
iParameterXLS	Defines the options for the XLS file generated during the printout.
iPicaToMM	Converts the coordinates of a point (expressed in system coordinates) into millimeters
iPreview / iDestination	Configures the destination of the report printout
iPrintBarCode	Prints a bar code in a rectangle
iPrintBlock	Forces a report block to print
iPrintBodyComplement	Forces a <i>Body Complement</i> block to print by specifying the height of the complement block
iPrintReport	Prints a report created in the report editor (nested report)
iRAZ	Reinitializes the printout parameters
iRoundBorder	Prints a rounded border at the specified coordinates
iSelectFont	Creates a new printout font from the window for selecting the Windows fonts
iSkipPage	Forces a page skip: the corresponding code is sent to the printer spooler
iStatusReportPrintout	Returns the status of the current printout
iSubstBlock	Substitutes a report block for another one while the report is printed
iTextHeight	Calculates the height of the font for the text to print (in millimeters)
iTextWidth	Calculates the width (in millimeters) of the text to print according to the specified font
iTransparentMagenta	Specifies whether the Magenta color will be considered as being transparent or not in the next printouts
iWindowCancel	Configures the display of the window used to cancel the current printout
iZoneHeight	Calculates the necessary height of a fixed-width area in order for the entire text to be printed in this area (in millimeters)

See the online help for more details.





INDEX

A

Add	
Cell to a table	105
Column to a table	105
Align	
Alignment options	92
Interface checker	92
Rulers and markers	93
Snap-on grid	94
Zoom	93
Align controls	92
Alignment markers	93
Alignment rulers	20, 93
Associated properties	
Report blocks	73
Report controls	96
Reports	146
Automatic retrieval	57

B

Bar	
Tools	20, 25
Bar code	79
Blank (type of report)	33
Blank report	33
Block	65
Associated processes	71
Associated properties	73
Body	67
Body Complement	68
Break Footer	68
Break Header	67
Characteristics	68
Create a block	70
Delete a block	70
Description	68
End of Document	68
Iteration	68
Page Footer	68
Page Header	67
Parameters	69
Position	57, 71
Print options	69

Programming	71
Rearrange	71
Setting	56
Start of Document	67
Types of blocks	66
Body	67
Body Complement	68
Bound	48
Bounded report	48
Break	38
Break process	142
Programmed	142
Break footer	68
Break header	67

C

Calculated	77
Calculations in a report	40
Cell	104
Add	105
Height	106
Resize	106
Selection	105
Width	106
Characteristics	
Report blocks	68
Report controls	86
Table	103
Chart	
Type of control	81
Chart + Table (type of report)	35
Check box	83
Circle	83
Closing a report	57
Code window	21
Column	
Add	105
Delete	107
Move	107
Remove	107
Select	105
Control	75
Align	92
Associated processes	95
Associated properties	96



Bar code	79	Query	48
Calculated	77	Table	52
Characteristics	86	Text file	53
Chart	81	WLanguage variable	50
Check box	83	Delete	
Copy	90	Block	70
Copy the style	90	Column	107
Create	88	Control	91
Delete	91	Version of a report	64
Description	86	Description	
Group of controls	88	Report	55
Image	78	Report blocks	68
Internal report	84	Report controls	86
Item	77	Table	103
Move	91	Description window	21
Multilingual	87	Displayed currency	77, 78
Off-report control	91	Drop-down menu	20
Preset control	81	Dual display	78
RTF control	79		
Select	89	E	
Shape	83		
Static	75	Embedded query	48
Types of controls	75	End of Document	68
URL link	84	End of print (process)	141
Copy			
Control	90	F	
Style of a control	90		
Create		Filter (process)	141
Block	70	Form	58
Controls	88	Include in a report	60
Report	29	Type of report	34
Report with parameters	143	Form (type of report)	30
Table	101	Form report	30
Crosstab report	36	Format of the page	40
Crosstab report (type of report)	36	Free (type of report)	33
Create	102		
Currency + Euro	77	G	
Custom mailshot (type of report)	32		
Custom mailshot report	32	Group of controls	88
		Guides	93
D			
		H	
Data file	47		
Print the report	121	History	64
Data source of the report		HTML	43
Description	55	HTML file	43
File	47	HyperFileSQL view	50
Memory zone	51	Print the report	132
Programmed data source	54		





I

Image	78
Independent query	48
Inserting a table	102
Interface	20, 92
Interface checker	92
Internal report	84
Item	77
Currency	77, 78
Dual display	77, 78
Iteration	68

K

Keyboard (shortcuts)	26
Keyboard shortcuts	26
Keywords	
Reports	146

L

Label (type of report)	31
Label report	31
Languages of a report	57
Line	83

M

Magnetism	92
Main window of the editor	20
Memory zone	51
Print the report	134
Move	
Column (table)	107
Control	91
Multilingual feature in report controls	87

N

Nested report	113
Nesting the reports	113
Number of Pages	81

O

Off-report control	91
Opened documents	21
Opening a report	57

Opening a version	64
Options for printing the blocks	69

P

Page Footer	68
Page header	67
Page number	81
Pane	20
Parameter	
Report	143
Report blocks	69
PDF	43
PDF file	43
Position of the blocks	57, 71
Pre-printed form	57
Preset	81
Print	
On a form	57
Report	117
Report based on a HyperFileSQL view	132
Report based on a memory zone	134
Report based on a query	126
Report based on a table control	128
Report based on a text file	136
Report with parameters	144
Print margins	40
Print preview	43
Printout	
HTML	43
Paper	43
PDF	43
Print Preview	43
RTF	43
Text File	43
XML	43
Process	
Break	142
End of print	141
Filter	141
Reading data	141
Start of print	140
Processes	
Report	140
Report blocks	71
Report controls	95
Programming	
Report	139
Report blocks	71



Q

Query	48
Embedded query	48
Independent query	48
Print the report	126
Query with parameters	49
Query with parameters	49

R

Reading data (process)	141
Rearranging the blocks	71
Rectangle	83
Registration card	19
Removing a column (table)	107
Report	13
3 Dimensions	61
Associated keywords	146
Associated properties	146
Available Types	29
Blocks	65
Bound	48
Calculations	40
Close	57
Creation	29
Data Source	47
Description	55
Form	57, 58
Format	40
Functions	148
Languages	57
Margins	40
Nested report	113
Print	43, 117, 121
Processes	140
Programming	139
Report controls	75
Restore	57
Save	40
Skin Template	40
Structure	14
Test	40
Using tables	99
Version	64
With parameters	143
Report based on a HyperFileSQL view	
Print	132
Report based on a memory zone	

Print	134
Report based on a query	
Print	126
Report based on a table control	
Print	128
Report based on a text file	
Print	136
Report based on a variable	
Print	130
Report based on file	
Print	121
Report based on form	34
Report data source	
HyperFileSQL view	50
Report Editor	
Configuration	19
Developer Version	18
Documentation	19
Drop-down Menus	21
Keyboard Shortcuts	26
Main Window	20
Minimum Configuration	19
Registration Card	19
Setup	19
Toolbar	25
User Version	18
Report editor	17
Report template	63
Report with parameters	143
Create	143
Print	144
Test	145
Reports and Queries	18
Resizing the table cells	106
Row	83
Rport	
Open	57
RTF	43, 79
RTF file	43
Running a report	121

S

Save	
Report	40
Version	64
Search key	48
Selecting one or more controls	89
Shape	83
Skin template	40, 63





Snap-on grid	94	Test	
Sort	38	Report	40
Source of the report	47	Report with parameters	145
Start of Document	67	Text file	43, 53
Start of print (process)	140	Print the report	136
Starting the report editor	20	Type	
Static	75	Type of blocks	66
Stored currency	77, 78	Type of controls	75
Structure of a report	14	Type of report	29
Style of the table	57		
System date	81	U	
System time	81	URL link	84
T		V	
Table	52, 99	Version management	64
Add a cell	105	Version of a report	64
Add a column	105	Delete	64
Characteristics	103	Open	64
Create	101	Save	64
Delete a column	107	Viewing the report in 3D	61
Description	103	W	
Inserting a table into a report	102	Wizard	
Move a column	107	Report creation	29
Remove a column	107	WLanguage functions	148
Resize the cells	106	WLanguage variable	50
Select an element	105	X	
Style	57	XML	43
Table cell	104	XML file	43
Table description	105	Z	
Table (type of report)	29	Zoom	93
Create	101		
Table + Chart report	35		
Table control			
Print the report	128		
Table report	29		
Table row			
Selection	105		





DEVELOP 10 TIMES FASTER

PART 2

Query editor



PCSOFT



QUERY EDITOR - SUMMARY

1. INTRODUCTION	157
1.1 What is a query ?	157
1.2 Query editor	159
1.2.1 Overview	159
1.2.2 Differences between the standard query editor and Reports & Queries	159
1.2.3 Main interface of the query editor.....	160
1.3 Drop-down menus of the query editor	162
1.4 Toolbar of the query editor	163
1.5 Keyboard shortcuts	164
1.6 Query and project	165
2. HOW DO I CREATE A QUERY ?	166
2.1 Overview	166
2.2 Select query	167
2.2.1 Overview	167
2.2.2 Creating a Select query	169
2.3 Insert query	169
2.3.1 Overview	169
2.3.2 Creating an Insert query.....	170
2.4 Update query	171
2.4.1 Overview	171
2.4.2 Creating an Update query	172
2.5 Delete query	174
2.5.1 Overview	174
2.5.2 Creating a Delete query.....	174
2.6 Union query	175
2.6.1 Overview	175
2.6.2 Creating a Union query via the description window	176
2.7 Query in SQL code	177
2.7.1 Creating a query in SQL code	177
2.7.2 Runtime modes of a query in SQL code	178
2.7.3 Available SQL commands	178
2.8 Creation mode of queries	182
3. DESCRIBING A QUERY	183
3.1 Overview	183
3.2 Description window	183
3.2.1 Displaying the description window of a query	183
3.2.2 Description window of a Select query	183
3.2.3 Description window of a Union query.....	188
3.2.4 Description window of a query (Insert, Update or Delete) or description window of a query in SQL code	189
3.3 Characteristics of a query	190
3.3.1 Items used in a Select query	190



3.3.2	Sorting the records selected by a Select query	192
3.3.3	Operations that can be performed in a Select query	193
3.3.4	Modifying the values displayed in the result of a Select query	194
3.3.5	Selection conditions	194
3.3.6	Sequence of selection conditions	197
3.3.7	Displaying the records corresponding to a given period	198
3.3.8	Limiting the number of records displayed in a Select query	198
3.3.9	Keeping (or not) duplicates	199
3.3.10	Joins between the files	199
3.3.11	Synchronizing the query	200
3.4	Opening, closing and saving a query	200
3.5	Managing the versions of a query	201
3.5.1	Saving a version	201
3.5.2	Opening a version	201
3.5.3	Deleting a version	202
3.6	Documentation of the query	202
3.7	Optimizing a query	203
4.	RUNNING THE TEST OF A QUERY AND USING IT	204
4.1	Test of the query	204
4.1.1	Exporting the query result to a text file	205
4.1.2	Exporting the query result to an Excel file	206
4.1.3	Exporting the query result to an XML file	206
4.1.4	Printing the result of the query	207
4.2	Using the result of a query	207
4.2.1	Query in a WinDev window or in a WebDev page	208
4.2.2	Query in a control of a WinDev window or WebDev page	208
4.2.3	Query in a report	209
4.3	Running the query and handling the result with the WLanguage functions	210
5.	ADVANCED QUERY	211
5.1	Overview	211
5.2	Sub-query	211
5.2.1	Sub-query as selection condition	211
5.2.2	Sub-query used as data source of a query	212
5.3	Query with parameters	212
5.3.1	Creating a query with parameters	212
5.3.2	Running the test of a query with parameters	212
5.3.3	Using a query with parameters	213
5.3.4	Two methods for passing parameters to a query	217





1. INTRODUCTION

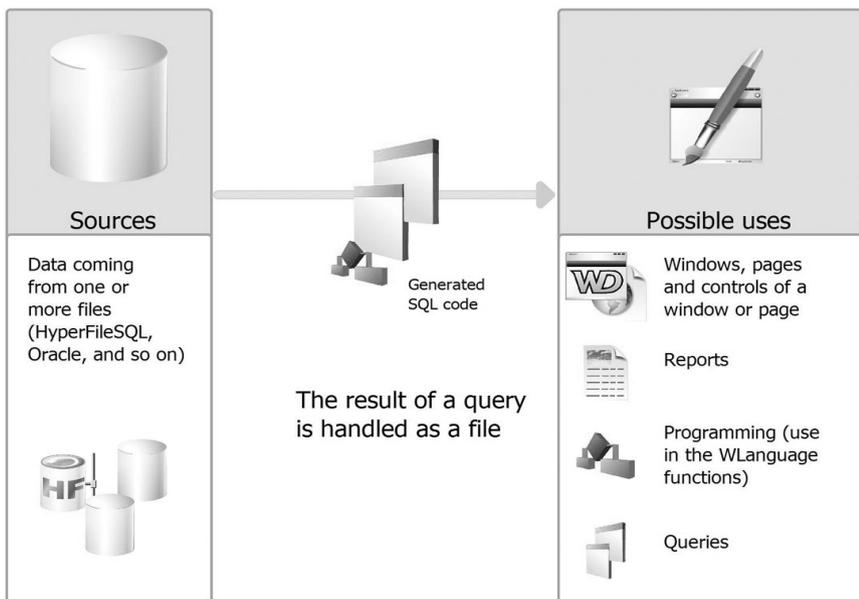
1.1 What is a query ?

A query is used to interrogate a database in order to view, insert, update or delete data.

The structure of the query defines the data used.

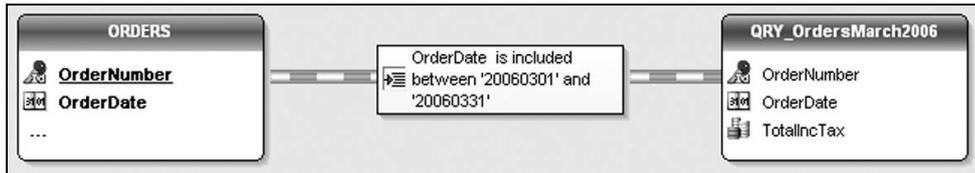
A query can interrogate one or more files.

The query editor is used to easily create queries, without programming.

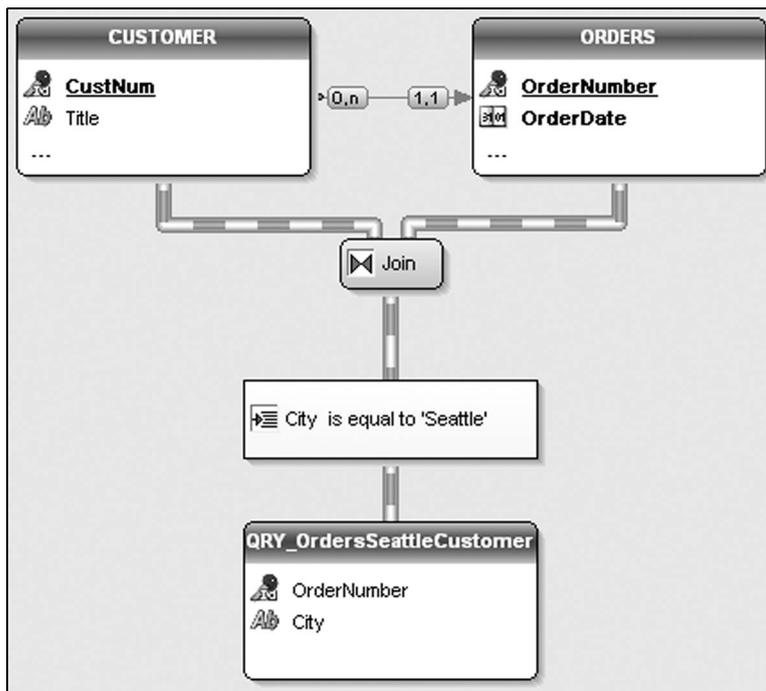


For example, in the *Sales Management* application, you want to find out:

- the orders placed in March 2006. You will have the ability to create a query to select the orders placed between 03/01/2006 and 03/31/2006 in the ORDERS file. The query named QRY_OrdersMarch2006 interrogates the ORDERS file.



- the numbers of the orders placed by the customers who live in Paris. You will have the ability to create a query that interrogates:
 - the CUSTOMER file to select the customers who live in Paris
 - the ORDERS file to display the order numbers
- The query named QRY_OrdersSeattleCustomer interrogates two files (CUSTOMER and ORDERS).





1.2 Query editor

1.2.1 Overview

The query editor is used to automatically create queries based on the data files. Creating a query is simple and intuitive. The query is automatically generated (in optimized SQL code) and the test of the query can be run immediately.

No programming hassle: windows, pages, tables, combo boxes, reports, etc. 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 be used when developing a WinDev application, a 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 the query editor are not available in this version.



The icon  indicates the differences between the user version of the query editor and the standard query editor.

Standard query editor

The standard query editor enables you to create queries and to run these queries from the editor or from an application developed in WinDev, WinDev Mobile or WebDev.

The dongle is required to use the query editor. It must be plugged into the parallel port or into the USB port of your computer when using the editor.

However, no dongle is required for the end users to run a query from a WinDev, WinDev Mobile or WebDev application.

Several examples are supplied with the standard 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 queries, display the guide ("? .. See the guide") and enter the "Query" keyword. The list of examples that use queries 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 the query editor (Reports & Queries - WinDev only)

Reports & Queries enables 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 existing queries with Reports & Queries. See "Distributing Reports & Queries", page 231, for more details.



Documentation

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

Printed documentation

The printed documentation (the guide you are reading) is 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 is supplied with the "Programming Guide", the "Tutorial" and the "Concepts".

Online help

The online help is intended to help you handle the editor.

To access the online help:

- select "? .. Summary" or "? .. Help index".
- press [F1].

Setup

Installing the standard query editor

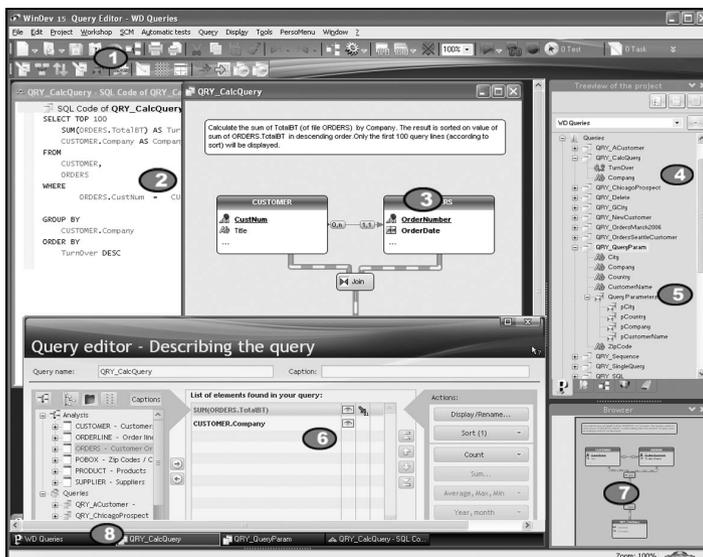
The developer version of the query editor is automatically installed along with WinDev.

Installing Reports & Queries

To install Reports & Queries, follow the instructions given when installing the WinDev application.

1.2.3 Main interface of the query editor

To display the query editor, all you have to do is create a new query ("File .. New .. Query") or open an existing query ("File .. Open").



1 Toolbar and drop-down menu

Gives quick access to the main features of the query editor. You can:

- customize the toolbar ("Display .. Toolbars .. Customize the toolbars").





- display/hide the toolbar ("Display .. Toolbars .. Popup bar").
- move the toolbar with the mouse.

2 SQL code of the current query

3 Graphic representation of the current query

Scrollbars are used to scroll the representation of the query if the size of this representation is quite important. You also have the ability to perform a zoom in the query editor according to a specified zoom factor:

- via "Display .. Zoom",
- with the mouse: [Ctrl] + mouse wheel.

4 Vertical panes

Displays:

- the shortcuts to the main features available in the current editor ("Command" pane),
- the frequently used features ("Command" pane),
- the elements (files, items, reports, controls, and so on) available in the current project ("Project" pane).



Important

In Reports & Queries, only the accessible elements are listed in the "Project" pane.

5 List of items and parameters for the selected query

If the query is run by programming (*HExecuteQuery*), the order in which the parameters are passed (query with parameters) corresponds to the order specified in this list. See "Query with parameters", page 216 for more details.



Important

The list of query items and parameters for the selected query is not available in Reports & Queries.

6 Description window of the current query ("Query .. Query description")

Enables you 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 enables you to easily move within this query.

8 Opened documents

Displays a button for each document (reports, queries, and so on) opened in the editor. To display the requested element in the foreground, click the corresponding "button".



1.3 Drop-down menus of the query editor

The tables below describe the actions of the menu options ("Query" and "Display") specific to the query editor.

Query option

Option	Action
Query description	Opens the description window of the query.
Edit the parameters	Displays the list of query parameters (queries with parameters only).
Edit the links/joins	Displays the description window of query joins (if the query is based on several linked files).
Run the query test	Runs the query in test mode and displays the result.
SQL code	Displays the SQL code corresponding to the query in the code editor.
Reverse engineering of the query	Generates the graphic display of the current SQL query
Consistency with the analysis	Displays the list of inconsistencies between the WinDev or WebDev analysis and the query (only if modifications have been made to the analysis and if the analysis and the project have not been synchronized).
Optimize the query	Defines the modifications that must be made to the analysis associated with the project to optimize the execution speed of the current query
Create a report based on this query	Generates a report based on the current query.
New Custom-Note	Creates a custom-note. A custom-note corresponds to a reminder associated with a project element.

Display option

Option	Action
Grid enabled	Enables the snap-on grid in the editor. This grid is used to position the query elements according to vertical and horizontal markers.
Grid visible	Displays the snap-on grid in the editor (displays horizontal and vertical markers).
Printout areas	Displays the printout area. The printout areas are delimited by dotted lines in the query editor. The graph of the query will be printed according to these areas.
Zoom	Performs a zoom in the query editor according to the specified zoom value.
Toolbars	Configures the toolbars and the general interface of WinDev or WebDev
Stream animation	Animates (or not) the streams of the query.

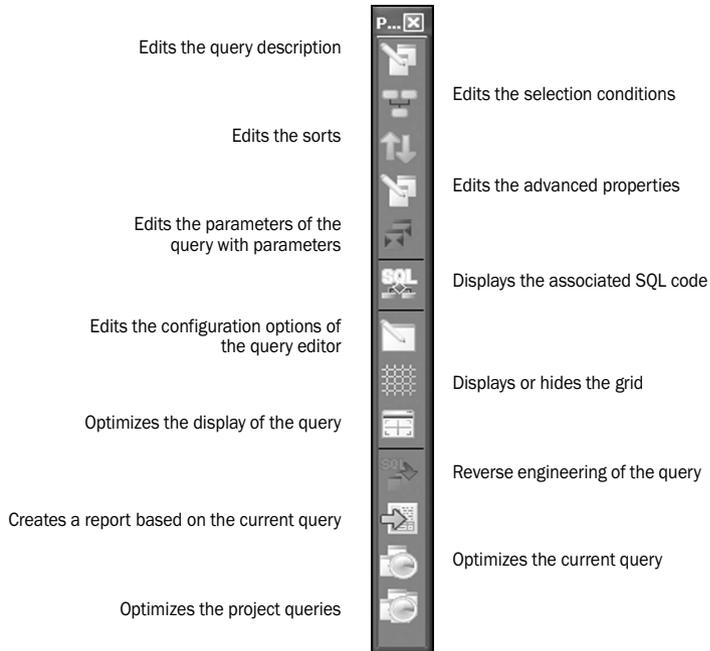
Note: the editing options of the query editor are accessible by selecting "Tools .. Options .. Options of the query editor".





1.4 Toolbar of the query editor

The toolbar of the query editor is as follows:



To enable or hide this toolbar, select "Display .. Toolbars".



1.5 Keyboard shortcuts

The main shortcuts that can be used in the query editor are as follows:

Display the list of classes in the "Code" pane	Shift + F8
Display the list of global procedures in the "Code" pane	Ctrl + F8
Display the SQL code in the "Code" pane	F2
Help	F1
Undo the last action	Ctrl + Z
Stop the test mode (when running the project test or the query test)	Shift + F5
Paste the content of the clipboard	Ctrl + V or Shift + Ins
Copy the selected element	Ctrl + C or Shift + Ins
Cut the selected element	Ctrl + X or Alt + Del
Edit the code associated with the selected query in the "Code" pane	Ctrl + Shift + F2
Save	Ctrl + S
Save as	Ctrl + Shift + S
Save all	Alt + S
Close all	Ctrl + Shift + W
Print the documentation regarding the current query	Ctrl + P
Hide all (vertical and horizontal panes)	Ctrl + W
New (Create an element)	Ctrl + N
Open (Open an element)	Ctrl + O
Open a project	Ctrl + Shift + O
Restore (Undo the last action)	Ctrl + Y or Ctrl + Shift + Z
Select all	Ctrl + A
Delete the selection	Del
Run the test of the current query	F9
Zoom + / -	Ctrl + mouse wheel

Note: + means that the keys must be pressed simultaneously.





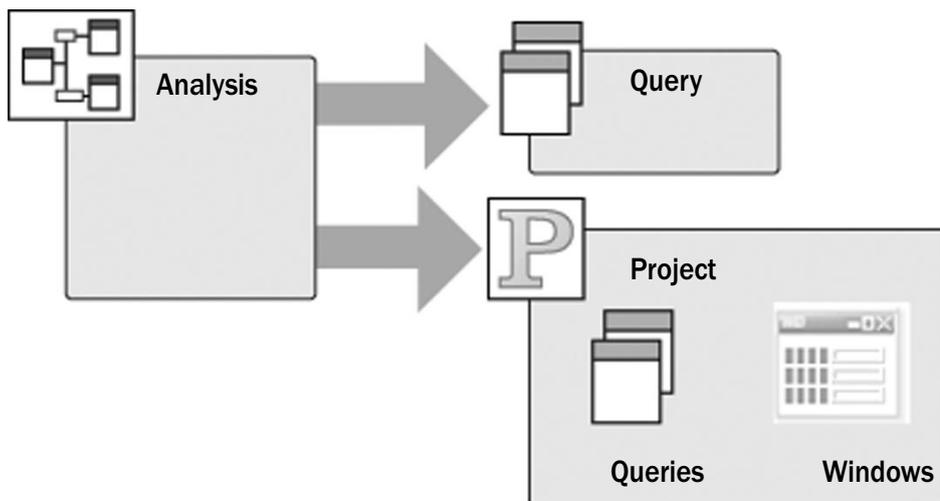
1.6 Query and project

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.

Also, a query must be linked to a WinDev or WebDev 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 or WebDev analysis contains the description of the files that will be used 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 open**, the query will be automatically linked to the project analysis.
- **if no project is opened**, the analysis used will have to be specified in the wizard for query creation.



2. HOW DO I CREATE A QUERY ?

2.1 Overview

To create a query, the query editor offers two methods:

- *create the query via a wizard or a description window* with automatic generation of the SQL code.
- *enter the SQL code.* See “Query in SQL code”, page 181, for more details.

The query editor enables you to create several types of queries:

- **Select query:** Used to view a selection of records coming from a database. See “Select query”, page 171 for more details.
- **Insert query:** Used to add records to a database file. See “Insert query”, page 173 for more details.



Important

Reports & Queries cannot be used to create Insert queries.

- **Update query:** Used to modify records in a database file. See “Update query”, page 175 for more details.



Important

Reports & Queries cannot be used to create Update queries.

- **Delete query:** Used to delete records from a database file. See “Delete query”, page 178 for more details.



Important

Reports & Queries cannot be used to create Delete queries.

- **Union query:** Used to combine selection queries. See “Union query”, page 179 for more details.

Note: All the types of queries can be based on items belonging to:

- to a file described in the analysis.
- a query that was previously created.

Insert, Update and Delete queries:

The ability to create Insert, Update and Delete queries is supported for compatibility with the SQL standard.

The use of native WLanguage commands (*HReadFirst*, *HReadNext*, *HAdd*, *HModify*, *HDelete*, and so on) is much more suitable for 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 performance and the simplicity of WinDev and WebDev are unmatched.



**Important**

Reminder: Reports & Queries cannot be used to create:

- Insert queries.
- Update queries.
- Delete queries.

Principle:

To create a query, you must:

- specify the method for creating the query (wizard/description window or SQL code).
- specify the type of the query (Insert, Update, etc.).
- specify (if necessary) the WinDev or WebDev analysis containing the description of the data files on which the query will be based.
- select the items to use.
- enter the characteristics of the query (selection conditions, sort, etc.).

The following paragraphs present the different types of queries and their creation mode.

2.2 Select query

2.2.1 Overview

A **Select query** (corresponding SQL command: SELECT) is used to select records in a database. This type of query is the most commonly used.

To perform this selection, the following information is required:

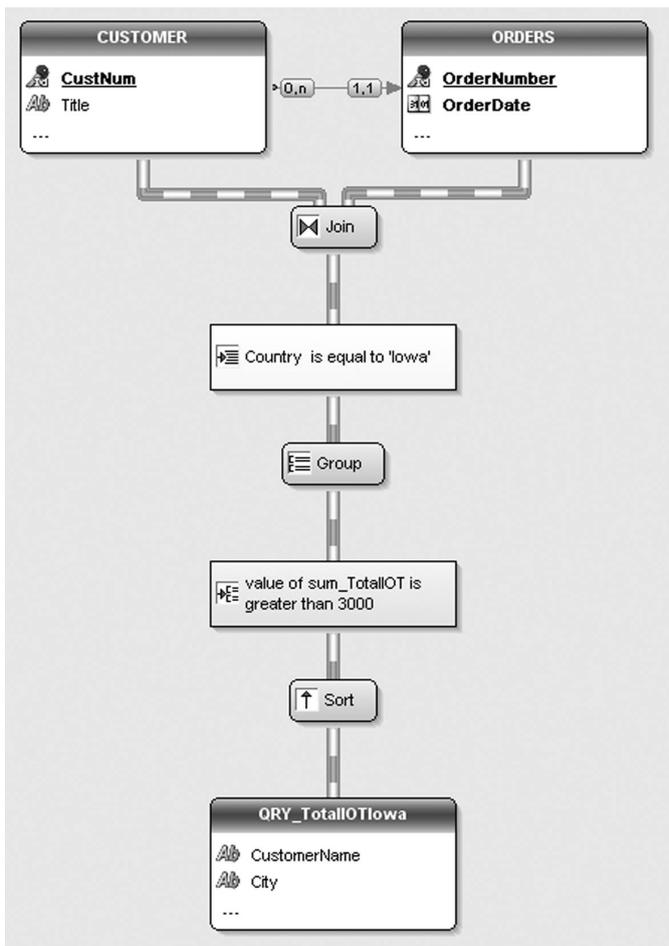
- the type of join between the files used (if several files are used),
- the selection conditions of the records (customers who live in Paris, products whose price is greater than 40 Euros, and so on),
- the grouping of the item values (grouping the products by type, the customers by age, etc.),
- the operations performed on the content of items (total amount of the orders for a customer, average sum spent per customer, and so on),
- the sort of the result items of the query (result sorted by zip code, price, and so on).

This information represents the characteristics of the query. See “Characteristics of a query”, page 194 for more details.

For example, you want to get the total of all the orders placed by people living in Iowa. Only the customers with a total amount of orders exceeding \$3,000 will be considered. The result will be sorted by total amount of orders in ascending order.



To perform this query, the result will be automatically grouped by customer name, by zip code and by city.



Data source of the query

Joins (see page 203 for more details)

Selection conditions (see page 198 for more details)

Sorting the records (see page 196 for more details)

Result items of the query

This query generates the following SQL code ("Query .. SQL code"):

```
SELECT CUSTOMER.CustomerLName AS CustomerName,
CUSTOMER.City AS City,
CUSTOMER.Country AS Country,
SUM(ORDERS.TotalIOT) AS sum_TotalIOT
FROM CUSTOMER, ORDERS
WHERE CUSTOMER.CustNum = ORDERS.CustNum
AND CUSTOMER.Country = 'Iowa'
GROUP BY CUSTOMER.Country, CUSTOMER.CustomerName, CUSTOMER.City
HAVING SUM(ORDERS.TotalIOT) > 3000
ORDER BY sum_TotalIOT ASC
```





2.2.2 Creating a Select query

To create a new Select query:

1. Select "File .. New .. Query". The wizard for query creation starts.
2. Specify that you want to create a Select query ("Select (SELECT)").
3. If your project is not linked to a WinDev or WebDev analysis, specify the analysis to which the query must be linked.
4. Validate the screen of the wizard. The description window of the query is displayed. This window is used to describe your query. This window is presented in details in "Description window of a Select query", page 187



Important

In Reports & Queries, the creation wizard asks all the necessary questions for creating a query.

5. Once the description of your query is validated, the graphic representation of the query is automatically displayed in the query editor.

Notes:

- You also have the ability to create a new Select query via a simplified wizard: Check "Use the simplified wizard" in the wizard for query creation.
Note: the simplified wizard does not let you manage "full-text" indexes.
- All the characteristics of a query can be modified ("Query .. Query description"). See "Characteristics of a query", page 194 for more details.
- To generate the SQL code corresponding to this query, select "Query .. SQL code". If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.
- You can also create a query from one or more 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 "Test of the query", page 208 for more details.

2.3 Insert query

2.3.1 Overview

An *Insert query* (corresponding SQL command: INSERT) is used to add records to a database file.



Important

Reports & Queries cannot be used to create Insert queries.

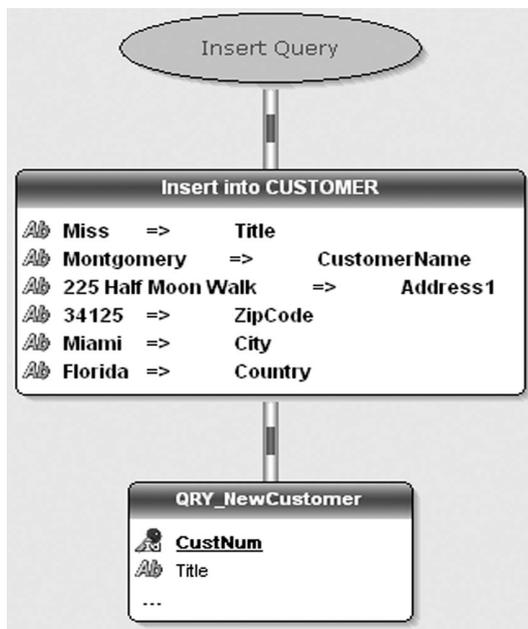
To perform this insertion, the following information is required:

- the file where the data will be inserted.
- the items where new values will be inserted.

This information represents the characteristics of the query. See "Characteristics of a query", page 194 for more details.



For example, creating a new customer in "Customer" file.



This query generates the following SQL code ("Query .. SQL code"):

```
INSERT INTO Customer
( Title, CustomerName, Address1, ZipCode, City, Country )
VALUES('Miss', 'Montgomery', '225 Half Moon Walk', '34125', 'Miami',
'Florida' )
```

2.3.2 Creating an Insert query

To create a new Insert query:

1. Select "File .. New .. Query". The wizard for query creation starts.
2. Specify that you want to create an Insert query ("Insert (INSERT)").
3. If your project is not linked to a WinDev or WebDev analysis, specify the analysis to which the query must be linked. Validate the screen of the wizard.
4. Select the data file where the data will be inserted. Go to the next screen.
5. Click the "Value/Parameters" column of the items for which new values will be inserted. 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 WinDev or WebDev analysis.
 - a specific value. In this case, specify this value.
 - a value found in a parameter. In this case, specify the name of the parameter.
 - a null value.
7. Validate your choice.





8. Repeat the operations 5 to 7 for each item for which a value must be inserted. Proceed to the next wizard screen.
 9. Enter the name of the query (name of the ".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. This caption will be used in the project documentation.
- Note:** A query name is automatically proposed from the caption.
11. Validate the description of the query. The graphic representation of the query is automatically displayed in the query editor.

Notes:

- To modify all the characteristics of a query, select "Query .. Query description". See "Characteristics of a query", page 194 for more details.
- To view the SQL code corresponding to this query, select "Query .. SQL code". If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.
- You can also create a query from one or more 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 "Test of the query", page 208 for more details.

2.4 Update query

2.4.1 Overview

An *Update query* (corresponding SQL command: UPDATE) is used to modify records in a database file.



Important

Reports & Queries cannot be used to create Update queries.

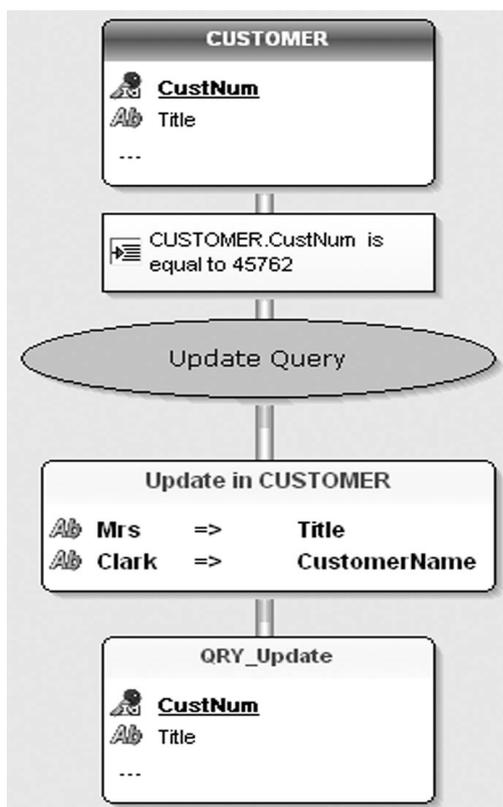
To perform this modification, the following information is required:

- the file where the data will be modified.
- the items where values will be modified.
- the selection conditions of the records to modify (customers who live in Paris, products whose price is greater than 40 Euros, and so on).

This information represents the characteristics of the query. See "Characteristics of a query", page 194 for more details.



For example, customer #45762 got married. Her title and her family name have been modified.



This query generates the following SQL code ("Query .. SQL code"):

```
UPDATE Customer
SET Title = 'Mrs', CustomerName = 'Clark'
WHERE Customer.CustomerNum = 45762
```

2.4.2 Creating an Update query

To create a new Update query:

1. Select "File .. New .. Query". The wizard for query creation starts.
2. Specify that you want to create an Update query ("Update (UPDATE)").
3. If your project is not linked to a WinDev or WebDev analysis, specify the analysis to which the query must be linked. Go to the next screen of the wizard.
4. Select the data file where the data will be modified. Go to the next screen of the wizard.
5. Click the "Value/Parameters" column for each item whose value must be modified. A window allows you to select the value or the parameter to modify.
6. Specify the value or the parameter containing the new value of the item. The modified values can correspond to:
 - the default value defined in the WinDev or WebDev analysis.





- a specific value. In this case, specify this value.
 - a value found in a parameter. In this case, specify the name of the parameter.
 - a null value.
7. Validate your choice.
 8. Repeat the operations 5 to 7 for each item for which a value must be modified.
 9. Define the selection conditions. See “Selection conditions”, page 198 for more details. Go to the next screen of the wizard.
 10. Specify (if necessary) the sequence of the selection conditions with the AND and OR conditional operators:
 - *Conditional operator AND*: the selected records will match both conditions. For example, the query selects the customers whose name starts with the letter "A" AND who live in Paris.
 - *Conditional operator OR*: the selected records will match one of the two conditions or both conditions. For example, the query selects the customers who live in Paris OR in London.
- Go to the next screen of the wizard.
11. Enter the name of the query (name of the ".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. This caption will be used in the project documentation.
- Note:** A query name is automatically proposed from the caption.
13. Validate the description of the query. The graphic representation of the query is automatically displayed in the query editor.

Notes:

- All the characteristics of a query can be modified ("Query .. Query description"). See “Characteristics of a query”, page 194 for more details.
- To generate the SQL code corresponding to this query, select "Query .. SQL code". If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.
- You can also create a query from one or more 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 “Test of the query”, page 208 for more details.



2.5 Delete query

2.5.1 Overview

A *Delete query* (corresponding SQL command: DELETE) is used to delete records from a database file.



Important

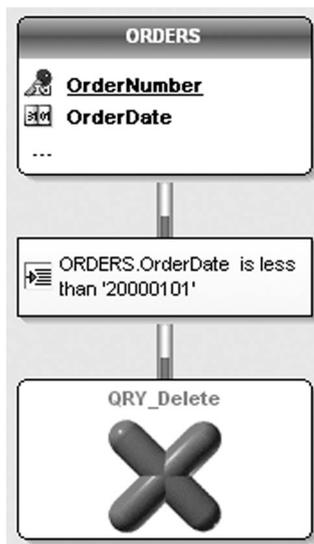
Reports & Queries cannot be used to create Delete queries.

To perform this deletion, the following information is required:

- the file where the data will be deleted.
- the selection conditions of the records to delete (customers who live in Paris, products whose price is greater than 40 Euros, and so on).

This information represents the characteristics of the query. See “Characteristics of a query”, page 194 for more details.

For example, deleting the orders placed before January 1st 2000.



This query generates the following SQL code ("Query .. SQL code"):

```
DELETE FROM ORDERS
WHERE ORDERS.OrderDate < '20000101'
```

2.5.2 Creating a Delete query

To create a new Delete query:

1. Select "File .. New .. Query". The wizard for query creation starts.
2. Specify that you want to create a delete query ("Delete (DELETE)").
3. If your project is not linked to a WinDev or WebDev analysis, specify the analysis to which the query





must be linked. Go to the next screen of the wizard.

4. Select the data file where the data will be deleted. Go to the next screen of the wizard.
5. Define the selection conditions. See “Selection conditions”, page 198 for more details. Go to the next screen of the wizard.
6. Specify (if necessary) the sequence of the selection conditions with the AND and OR conditional operators:
 - *Conditional operator AND*: the selected records will match both conditions. For example, the query selects the customers whose name starts with the letter "A" AND who live in Paris.
 - *Conditional operator OR*: the selected records will match one of the two conditions or both conditions. For example, the query selects the customers who live in Paris OR in London.

Go to the next screen of the wizard.

7. Enter the name of the query (name of the ".WDR" file corresponding to the query). This name will be used to identify the query in your programs.
8. Specify the caption of the query. This caption will be used in the project documentation.

Note: A query name is automatically proposed from the caption.

9. Validate the description of the query. The graphic representation of the query is automatically displayed in the query editor.

Notes:

- All the characteristics of a query can be modified ("Query .. Query description"). See “Characteristics of a query”, page 194 for more details.
- To view the SQL code corresponding to this query, select "Query .. SQL code". If modifications are performed in the description of the query, these modifications will be automatically carried over into the corresponding SQL code.
- You can also create a query from one or more 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 “Test of the query”, page 208 for more details.

2.6 Union query

2.6.1 Overview

The *Union queries* are used to combine several Select queries (SELECT statement). Each Select query must be based on 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 name starts with the letter 'A'. This query is based on the "Customer" file.
- selecting the prospects who live in Paris. This query is based on the "Prospect" file.

The "Customer" file and the "Prospect" file have the same structure in the analysis.

The records common to the two files ("Customer" and "Prospect") can:

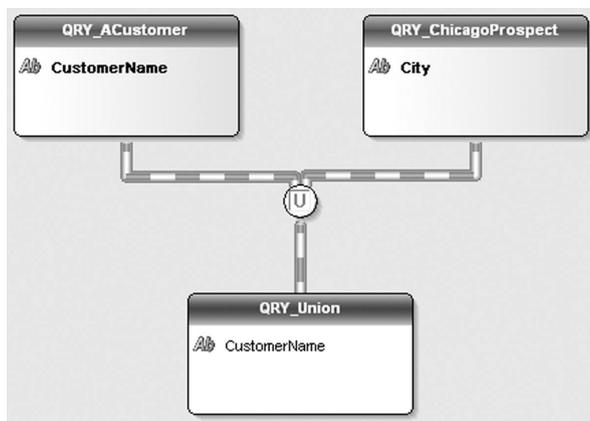
- appear only once in the result of the query (UNION operator).
- appear twice in the result of the query (UNION ALL operator).



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 result of the query.

Note: The items displayed in the result of the 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 Paris will only appear once in the result of the query.



This query generates the following SQL code ("Query .. SQL code"):

```
SELECT CUSTOMER.CustomerLName AS CustomerLName
FROM CUSTOMER
WHERE CUSTOMER.CustomerName LIKE 'A%'

UNION

SELECT CUSTOMER.CustomerLName AS CustomerLName
FROM CUSTOMER
WHERE CUSTOMER.City LIKE 'Paris'
```

In order for the people who are both customer and prospect, whose name starts with the letter 'A' and who live in Paris to appear several times in the result of the query, the SQL code must be:

```
SELECT CUSTOMER.CustomerLName AS CustomerLName
FROM CUSTOMER
WHERE CUSTOMER.CustomerName LIKE 'A%'

UNION ALL

SELECT CUSTOMER.CustomerLName AS CustomerLName
FROM CUSTOMER
WHERE CUSTOMER.City LIKE 'Paris'
```

2.6.2 Creating a Union query via the description window

To create a new Union query via the description window:

1. Select "File .. New .. Query". The wizard for query creation starts.
2. Specify that you want to create a Union query ("Union (UNION)").
3. If your project is not linked to a WinDev or WebDev analysis, specify the analysis to which the query must be linked.
4. Validate the screen of the wizard. The description window of the query is displayed. This window ena-





bles you to describe your query.

Note: This window is presented in "Description window of a Union query", page 192.

5. Enter the name and caption of the query.

6. Select the Select queries that will be used in the Union query (left box).

Note: The result items of the Union query will be the items of the first query run.

7. Specify whether duplicates must be kept or not ("Keep the duplicates/Delete the duplicates").

8. Sort the result of the query if necessary ("Sort" button).

9. Validate the description window of the query. The graphic representation of the query is automatically displayed in the query editor.

10. Select "File .. Save" to save the query.

Note: To view the SQL code corresponding to this query, select "Query .. SQL code". 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

The previous paragraphs explained how to create a query with the wizard. The query editor also enables you to create a query in SQL code directly (if you are familiar with this language).



Important

The following SQL commands are not allowed in Reports & Queries:

- INSERT
- UPDATE
- DELETE

2.7.1 Creating a query in SQL code

To create a query in SQL code:

1. Select "File .. New .. Query". The wizard for query creation starts.
2. Specify that you want to create a query by entering the SQL code directly ("Enter the SQL code").
3. If your project is not linked to a WinDev or WebDev analysis, specify the analysis to which the query must be linked. Validate the screen of the wizard.
4. Enter the name of the query (name of the ".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 the query. This caption will be used in the project documentation.

Note: The name of the query is automatically defined from the caption.

6. Validate the creation of the query. A code window is displayed.

7. Enter the SQL code of your query. You can insert into the query:

- comments by writing the "-" or "/" characters in front of them. Only comments about the selection conditions will be taken into account during the reverse engineering of the query; other comments will be deleted automatically. Comments about the selection conditions with parameters will not be proposed when running a GO of the query.
- WLanguage functions. The WLanguage functions must be prefixed by "WL.". The 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 ("File .. Save").



9. Click  to graphically represent the query (or select "Query .. Reverse engineering of the query").

Caution: this feature is only available if the SQL commands used in the query are recognized by the HyperFileSQL engine. See "Available SQL commands", page 182 for more details.

2.7.2 Runtime modes of a query in SQL code

The query editor enables you to easily run queries in SQL code. These queries can be:

- entered in the query editor. The query is run by *HExecuteQuery*.
- entered in *HExecuteSQLQuery* in the code editor.

Two runtime modes are available for 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, WinDev Mobile and WebDev. The SQL code of the query must use the commands listed below. In this case, the SQL code of the query is checked by the HyperFileSQL engine.
- **Running a 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 query is not checked by the HyperFileSQL engine. This option must be used if the query contains commands specific to a type of connection (Oracle, SQL Server, etc.). All the SQL commands supported by the accessed database can be used.

Note: The *hQueryWithoutCorrection* constant is also available when running the test of a query from the query editor:

- for a *Select query*: click the "Advanced" button in 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* query: display the "General" tab of the description window and check "Run with *hQueryWithoutCorrection* in test mode".

2.7.3 Available SQL commands

The "SQL commands" (statements, functions, clauses, and so on) that can be used in an SQL query checked by the HyperFileSQL engine are as follows:

ALL	Used with the SELECT statement: Selects all the records Used with the UNION operator: Selects the common records Used in the sub-queries. See "Sub-query", page 215 for more details.
ADD_MONTH	Adds months to the specified date
AS	Creates an alias for each item and for each file used
ASC	Used with ORDER BY : Defines the ascending order
ASCII	Returns the ASCII code of a character or of the first character of a string.
AVG	Calculates the average for a set of values
AND	Combines several selection conditions (both) See "Sequence of selection conditions", page 201 for more details
ANY	Used in the sub-queries. See "Sub-query", page 215 for more details
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
CASE	Evaluates a list of conditions and returns one of the possible results.
CEILING	Returns the rounded value of a number
CHAR_LENGTH	Returns the length (number of characters) of an expression
CHARACTER_LENGTH	Returns the length (number of characters) of an expression
COALESCE	Returns the first not-null expression among its arguments
CONCAT	Concatenates several strings together
COUNT	Returns the number of records found in a file or in a group of records
CREATE TABLE	Used to describe a data file and create it.
DELETE	Deletes records from a file
DESC	Used with the ORDER BY clause: Defines the descending order
DISTINCT	Used with the SELECT statement: Deletes the duplicated records (duplicates) from the query result
DROP TABLE	Enables you physically delete a data file
EXISTS	Used in the sub-queries. See “Sub-query”, page 215 for more details
FULL OUTER JOIN	Performs an outer join. See the online help for more details
FROM	Used with the SELECT statement: Defines the files used Used with the DELETE statement: Defines the files used
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 clause
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. See “Sub-query”, page 215 for more details
INNER JOIN	Performs an inner join. See the online help for more details
INSERT	Adds a record to a data file
INSERT with selection	Used to add the records returned by a SELECT query 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 the last day for the specified month
LEFT	Extracts the first characters from the content of a file item or character string
LEFT OUTER JOIN	Performs an outer join. See the online help for more details



LEN	Returns the size (number of characters) of the content of an item or character string
LENGTH	Returns the size (number of characters) of the content of an item or character string
LIKE	Selects the records for which the value of an item corresponds to a specified value (with wildcard characters)
LIMIT	Enables you to limit the number of records returned by the SELECT command.
LOWER	Converts the content of an item or character string into lowercase characters
LPAD	Returns a character string of a given length. To reach the requested length, the string is completed to the left: <ul style="list-style-type: none"> • by space characters. • by a given character or string.
LTRIM	Returns a character string: <ul style="list-style-type: none"> • without the space characters on the left. • without a list of characters.
MATCH ... AGAINST	Enables you to perform a query on a full-text index. Returns the relevance index of the processed record.
MAX	Returns the greatest value of an item for all the records selected in the file
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. See "Sub-query", page 215 for more details
NVL	Replaces the null values in a column by a substitution value
OCT	Returns the octal value of an integer (expressed in base 10)
OCTET_LENGTH	Returns the length (number of characters) of an expression
SUR	Used when creating a join. See the online help for more details
OR	Combines several selection conditions (one or the other) See "Sequence of selection conditions", page 201 for more details
ORDER BY	Sorts the selected records
PATINDEX	Returns the position of the first occurrence of a character string
POSITION	Returns the position of a character string in an expression





REPLACE	Returns a character string: <ul style="list-style-type: none"> • by replacing all the occurrences of a word found in character string by another word. • by replacing all the occurrences of a word found in a string.
RIGHT	Extracts the last characters from the content of a file item or character string
RIGHT OUTER JOIN	Performs an outer join. See the online help for more details
ROUND	Returns the date rounded to the specified format
RPAD	Returns a character string of a given length. To reach the requested length, the string is completed to the right: <ul style="list-style-type: none"> • by space characters. • by a given character or string.
RTRIM	Returns a character string: <ul style="list-style-type: none"> • without the space characters on the right. • without a list of characters
SELECT	Searches for records in one or more files
SET	Used with the UPDATE statement: Defines the names of the items used
SOME	Used in the sub-queries. See "Sub-query", page 215 for more details
SOUNDEX SOUNDEX_LIKE	Returns the phonetic value
SOUNDEX2 SOUNDEX2_LIKE	Returns the phonetic value (adapted to French)
SUBSTR	Extracts a sub-string from a given position
SUBSTRING	Extracts a sub-string from a given position
SUM	Returns the sum of all the values of an item for all the records selected in the file
SYSDATE	Returns the current date and time
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 the space characters on the left and on the right. - without a character string found at the beginning or at the end of the string. - without a character string found at the beginning of the string. - without a character string found at the end of the string.
TRUNC	Returns the date truncated to the specified format
UNION	Performs union queries
UNICODE	Returns the integer value defined by the Unicode standard of the first character in the specified expression.



UPDATE	Updates the file records
UPPER	Converts the content of an item or character string into uppercase characters
VALUES	Used with the <i>INSERT</i> statement: Defines the added values
WHERE	Used to specify the selection conditions of a query

See the online help for more details.

The code of the query can be directly entered if your database uses specific SQL keywords.

2.8 Creation mode of queries

The query editor proposes two methods for creating a query:

- Creating the query in the query editor directly ("File .. New .. Query"): This method was described in the above paragraphs. This query corresponds to an 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, table, looper), in programming, in another query, etc. **The query is said to be independent.**
- Creating the query from the wizard for creating the report or the control based on this query ("A query that I want to define now" in the wizard for report creation or for control creation). No specific file is created. **The query is said to be embedded.**
An embedded query cannot be used as data source for an element (report, and so on) other than the one where this query was created.

An embedded query can be converted into an independent query.

To switch from the embedded mode to the independent mode (i.e. 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 the report (or select "Report .. Edit the query"). The query is displayed in the query editor.
Caution: The report must have been saved at least once.
3. Select "File .. Export .. To an independent query".
4. Save the new query. The query is now an independent query.

To switch from the embedded mode to the independent mode (i.e. to extract a query from a control):

1. Display the window or the page in the editor.
2. Select the control.
3. Select "Embedded query .. Open the associated query" from the popup menu of the control. The query is displayed in the query editor.
Caution: The window of the page must have been saved at least once.
4. Select "File .. Export .. To an independent query".
5. Save the new query. The query is now an independent query.





3. DESCRIBING A QUERY

3.1 Overview

The steps for creating a query have been explained in the previous chapters. 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 result of the query,
- selection conditions,
- sort of records,
- etc.

Only the type of the query cannot be modified. Indeed, too many specific parameters are taken into account. To change the type of the query, 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 the 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 (see page 187).
- description window of an Insert query, Update query, Delete query or query in SQL code (see page 193).
- description window of a Union query (see page 192).

3.2.1 Displaying the description window of a query

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.
- select "Query .. Query description".

3.2.2 Description window of a Select query

The description window of a Select query is made of several sections. This description window is used to:

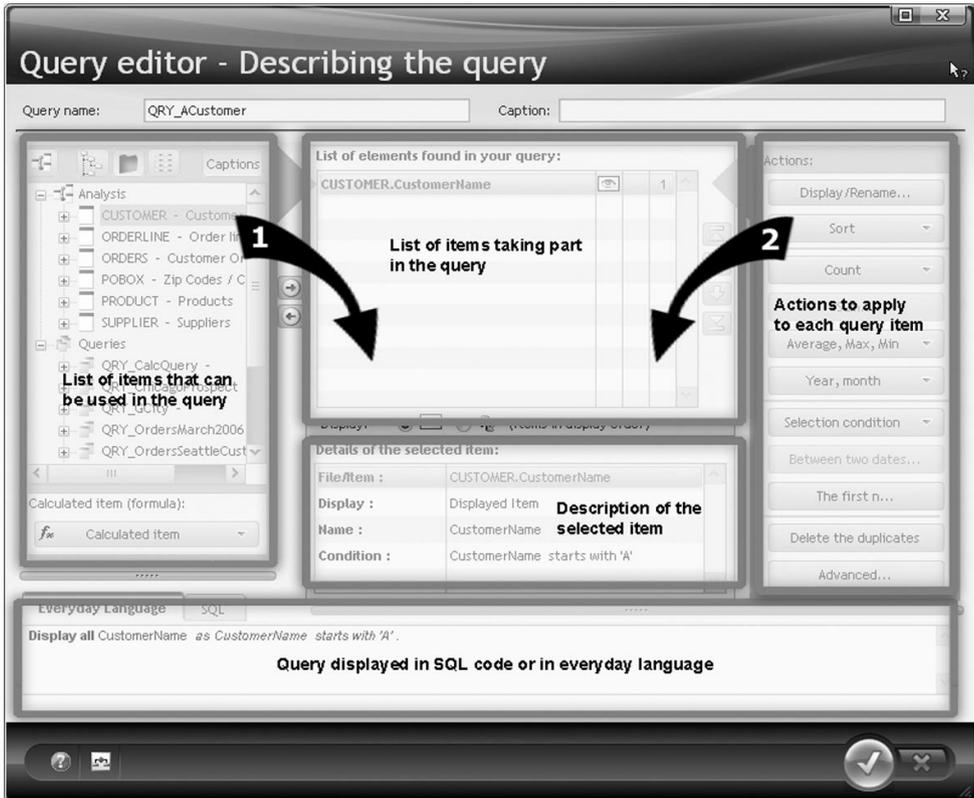
- modify the items used in the query,
- manage the selection conditions,
- view the query in SQL code or in natural language,
- manage the "duplicate" records, etc.

This description window is displayed:

- when creating a Select query ("File .. New .. Query").



- from a Select query displayed in the query editor ("Query .. Query description").



Note: The floating help of this window (green and transparent) is automatically displayed. The floating help is displayed only when the cursor is immobile. To remove this help:

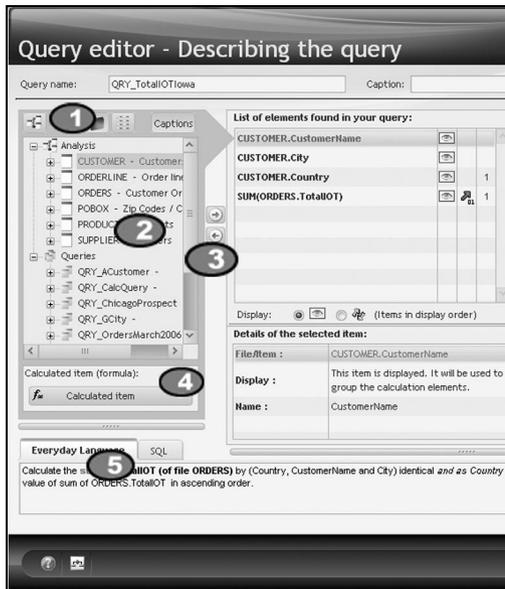
- for the current session: press [Alt]+[F1] together.
- completely: display the configuration options of the query editor ("Tools .. Options .. Options of the query editor") and select "Don't display the floating help".





This description window is made of three sections:

- **Left section:** This section contains the following elements:



1 Displays the analysis associated with the current project.

2 List of items that can be used in the query.

These items belong to:

- a file of the analysis associated with the current project.
- a query of the current project.

Note: To select several items, keep the [Ctrl] key down.



Important

The files and the items that can be used in Reports & Queries are defined in the data model editor. For more details about the preparation required for using a WinDev application with Reports & Queries, see “Distributing Reports & Queries”, page 231.

3 Used to add/delete the items selected to/from the query. The selected elements are displayed in the middle part.

4 Adds a new item to the result of the query. This item corresponds to a calculation on some items that can be used in the query.

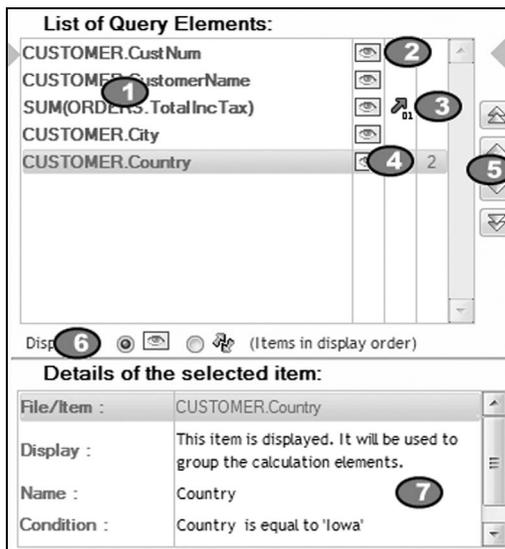
For example, the "PricelOT" item corresponds to the "PriceBT" item multiplied by the "VATRate" item.

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. See “Items used in a Select query”, page 194, for more details.
- 2 The values of this item will be displayed in the result of the query ().
- 3 Sort direction for the values of this item (ascending , 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 the items in the result of the query.
- 6 Modifies the display order in the list of query elements:
 - : corresponds to the display order of the elements in the result of the query.
 - : corresponds to the sequence of sorts in the result of the query.
- 7 Details of the options defined for the selected item.





- **Right section:** This section is used to define the different actions to apply to the query items. See “Characteristics of a query”, page 194, for more details about the various possible actions.



- 1 Item onto which the actions will be applied.
- 2 Used to:
 - display (or not) the values of the selected item in the query result.
 - rename the selected item in the query result (create an alias).
- 3 Sorts the values of the selected item in the query result.
- 4 Adds a new item to the result of the query. This item corresponds to:
 - the number of records found in a selection of records.
 - the number of records found in a data file.
- 5 Adds a new item to the result of the query. 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 average of a group of records.
 - the minimum value of a group of records.
 - the maximum value of a group of records.
- 7 Modifies the values displayed in the query result. You have the ability to display:
 - the year ou month from a date item.
 - the state from an item containing a zip code.
- 8 Defines the selection conditions. You can:
 - 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.
- 11 Deletes (or not) the "duplicate" records from the query result.
- 12 Used to:
 - manage the joins (add, modify, delete, reinitialize).
 - check (or not) the SQL syntax used (in test mode only).



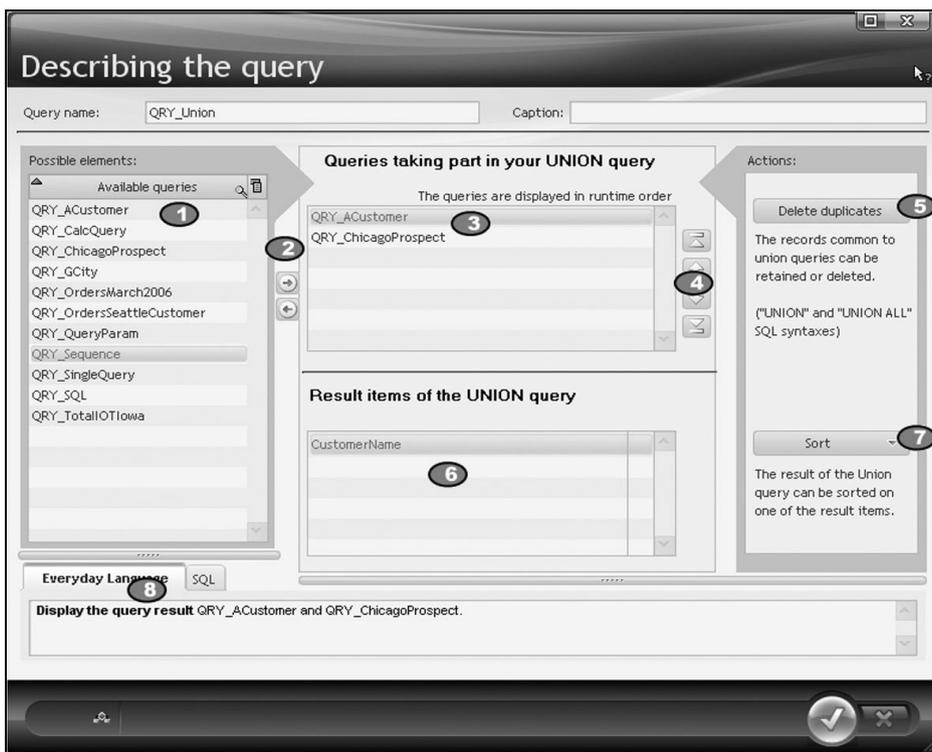
3.2.3 Description window of a Union query

The description window of a Union query is made of several sections. This description window is used to:

- change the queries used in the query.
- modify the order in which the queries are run.
Reminder: The items displayed in the result of the 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.

This description window is displayed:

- when creating a Union query ("File .. New .. Query").
- from a Union query displayed in the query editor ("Query .. Query description").



- 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/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.

Note: These items correspond to the items of 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 a query (Insert, Update or Delete) or description window of a query in SQL code



Important

Reports & Queries cannot be used to create:

- Insert queries.
- Update queries.
- Delete queries.

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 the caption of the query.
- specify whether the HyperFileSQL commands used must be checked or not by the Hyper File engine ("Run with *hQueryWithoutCorrection* in test mode").
- specify the connection used when running the test of the query.
- find out and/or modify the values to insert or to modify in the result of the query (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).

Note: See the context-sensitive help of the description window for more details about each of the various tabs ()



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 associated with the current application.**
These items correspond to the items of the data files described in the analysis. Calculations can be applied to these items.
You can use a full-text index item. The query will perform a full-text search.
- **items of a query in the current project.**
These items correspond to the items used in the queries included in the application.
- **count items.**
This type of item enables you to count the number of records found in the query, or the number of records found in a grouping of records.
- **calculation item.**
This type of item allows you to perform a calculation on the values of an item. See "Operations that can be performed in a Select query", page 197, for more details.
- **"calculated" items.**
This type of item allows you to add to the query result an item whose value corresponds to a calculation on items. For example, $\text{PriceBT} * \text{VATRate}$.

The items taking part in a query are chosen when the query is created. 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 query result are called *result items*.

To add an item to the current query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the left part of the window, double-click the item to add. This item is displayed in the middle part of the window.

To perform a full-text search:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the left part 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 used
 - the search value (value or parameter). If the value is described in the query, a wizard lets you enter all the needed conditions on the search value.
 - how to sort the result based on the relevance index of the search for each record.

Note: only records with a relevance index greater than 0 are kept in the query result.



**To add a count item to the current query:**

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the right part of the window, click "Count".
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 customers who have a cell phone ("CellPhone" item filled).
 - the number of values for another item in the query result: select "On another item" and select the requested item.
 - the number of records 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 how you count, you can specify whether you want to count the duplicates or not (Count(Distinct) in SQL).

To add a "calculated" item to the current query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the left part of the window, click 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 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 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.

To modify a "calculated item" in the current query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the calculated item to modify.
3. In the left part of the window, click the "Calculated item" button and select "Modify calculated item". The description window of a calculated item is displayed.
4. Modify the item's formula and validate. The item is automatically modified.

To rename a result item in the current query (alias creation):

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the item to rename.
3. In the right part of the window, click "Display/Rename".
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. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the item to display/hide.
3. In the right part of the window, click "Display/Rename".
4. Answer "No" to "Display this item?".

Note: To display/hide an item in the query result, 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. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the item to delete and click  (or press [Del]). The item is automatically deleted from the query.

Notes:

- Deleting an item also deletes the existing selection conditions for this item.
- To delete an item, you can also 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 query result clearer, the records selected by the query can be sorted according to one or more items. Sorting the records means classifying them in an ascending order or in descending order.

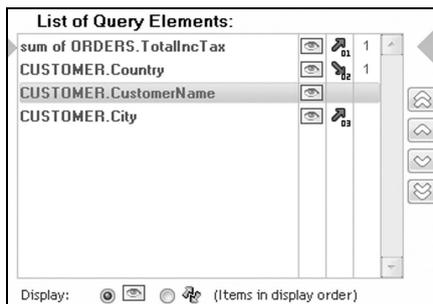
For example, the customers can be sorted by name, from A to Z.

When several sort items are defined, the sorts are nested. To find out the sequence of the different items, click  in the middle section.

Note: The digits found beside the arrows can also be used to find out the sequence of the different sort items.

For example, in the window below, the result of the query is sorted:

- by the total amount of orders,
- then by the city of the customers,
- finally by the name of the customers.



The arrows found to the right of the list are used to modify the sequence of the sort items.

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 on an item.
- modify the sort order for an item.

These operations are described in the following paragraphs.

To add a sort on an item to the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the requested item.
3. In the right section of the window, click "Sort" and select "Sort on the selected item".
4. Specify the sort order and validate.

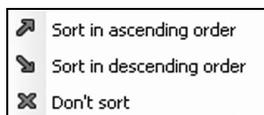
To delete a sort on an item from the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle section of the window, select the item for which the sort must be deleted.
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. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle section of the window, select the item for which the sort must be modified.
3. In the right section of the window, click "Sort" and select "Modify the sort order".

Tip: To add, modify or delete a sort, click the sort column of 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 enables you to perform operations on the values of an item. The possible operations are:

- the sum of the item values.
- the average of the item values.
- the calculation of the minimum value for all the values of the item.
- the calculation of the maximum value for all the values of the item.

These operations are performed:

- on all the records selected by the query.
- on subsets of records (automatically grouped by the query).

For example, you can find out:

- the total amount of the orders placed by the customers living in Paris,
- the order with the smallest amount,
- the average sum spent by the customer named Darwin,
- the total amount of orders per country and per year,
- etc.



The operations are defined when creating the query. However, you have the ability to add, modify or delete the operations performed in a query. These operations are described in the following paragraphs.

Caution: If the query result is displayed in a report, the operations to perform on the value of the items must be defined when describing the report. See "How do I create a report ?", page 28 for more details.

To add an operation into the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the item on which an operation must be performed.
3. To calculate:
 - the sum of the item values, click the "Sum" button.
 - the average of the item values, click the "Average, Max, Min" button and select "Average".
 - the maximum value for all the item values, click "Average, Max, Min" and select "Maximum".
 - the minimum value for all the item values, click "Average, Max, Min" and select "Minimum".
4. Validate. The operation is automatically added to the query.

To delete an operation from the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the calculation item to delete.
3. Press [Del] (or select "Delete the item" from the popup menu of the item).

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.
- the state from an item containing a zip code.

To modify the values displayed in the result of the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed
2. In the middle part of the window, select a date item or an item containing a zip code.
3. In the right section of the window, click "Year, state" and select:
 - "Month" to only display the month in the result of the query.
 - "Year" to only display the year in the result of the query.
 - "State" to only display the state in the result of the query.
4. Validate.

3.3.5 Selection conditions

The selection conditions of the records are applied to the items used in the queries. These conditions are used to select the records matching one or more specific criteria.

For example:

- customers whose name starts with "C",
- items whose price exceeds 500 Euros,
- orders placed between "01/01/2000" and "06/01/2000",
- etc.

Several selection conditions can be defined for an item.





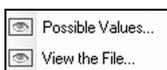
The selection conditions are defined when the query is created. However, you have the ability to add, modify or delete the selection conditions used in a query. These operations are described in the following paragraphs.

How to add a selection condition in a query ?

To add a selection condition to the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the item for which a selection condition must be added.
3. In the right section of the window, click "Selection condition" and select "New condition". The window for creating a selection condition is displayed.
4. Choose the selection criteria (equal to, greater than, and so on) of the condition. This condition can be applied to a given value, a parameter, a sub-query, etc.

Note: You can select a specific 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 ("Project .. Project description", "Live Data" tab). See the online help (keyword: "Live Data") for more details.

5. Indicate (if necessary) a comment on the condition. This comment will be displayed in the SQL code of the query (preceded by "//").
6. Validate. The selection condition is automatically added to the query.

To add a selection condition to the current Update or Delete query:

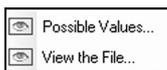
1. Select "Query .. Query description". The description window of the query is displayed.
2. Display the "Query" tab.
3. Select the item for which a selection condition must be added and click the corresponding "Condition" column. The wizard for creating a selection condition is displayed.
4. Select (if necessary) "I want to add a condition" (for an item on which at least one condition is performed).
5. Choose the selection criteria (equal to, greater than, and so on) of the condition. This condition can be applied to a given value, a parameter, a sub-query, etc.
6. Indicate (if necessary) a comment on the condition. This comment will be displayed in the SQL code of the query (preceded by "//").
7. Validate. The selection condition is automatically added.

How do I modify a selection condition in a query ?

To modify a selection condition in the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle part of the window, select the item for which a selection condition must be modified.
3. In the right part of the window, click "Selection condition" and select "Modify the condition". The window of selection conditions is displayed.
4. Select (if necessary) the condition to modify (for an item on which several conditions are performed).
5. Modify the selection condition.

Note: You can select a specific 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 ("Project .. Project description .. Live Data"). See the online help (keyword: "Live Data") for more details.

6. Indicate (if necessary) a comment on the condition. This comment will be displayed in the SQL code of the query (preceded by "//").
7. Validate. The selection condition is automatically modified.

To modify a selection condition in the current Update or Delete query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. Display the "Query" tab.
3. Select the item for which a selection condition must be modified and click the corresponding "Condition" column. The wizard for selection conditions is displayed.
4. Select "I want to modify an existing condition".
5. Select (if necessary) the condition to modify (for an item on which several conditions are performed).
6. Modify the selection condition.
7. Indicate (if necessary) a comment on the condition. This comment will be displayed in the SQL code of the query (preceded by "//").
8. Validate. The selection condition is automatically modified.

How do I delete a selection condition from a query ?

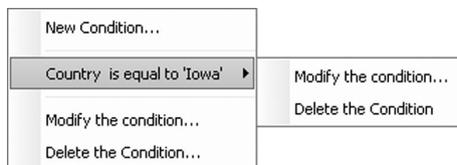
To delete a selection condition from the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the middle section of the window, select the item for which a selection condition must be deleted.
3. In the right section of the window, click "Selection condition" and select "Delete a condition".
4. Select (if necessary) the condition to delete (for an item on which several conditions are performed).
5. Validate the deletion. The selection condition is automatically deleted from the query.

To delete a selection condition from the current Update or Delete query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. Display the "Query" tab.
3. Select the item for which a selection condition must be deleted and click the corresponding "Condition" column. The wizard for selection conditions is displayed.
4. Select "I want to delete an existing condition".
5. Select (if necessary) the condition to delete (for an item on which several conditions are performed).
6. Validate. The selection condition is automatically deleted.

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:

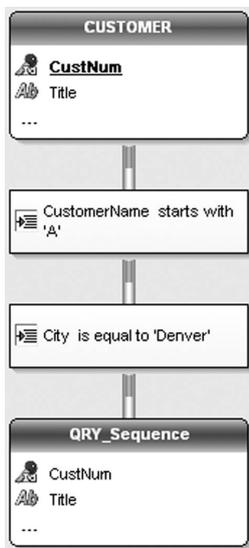




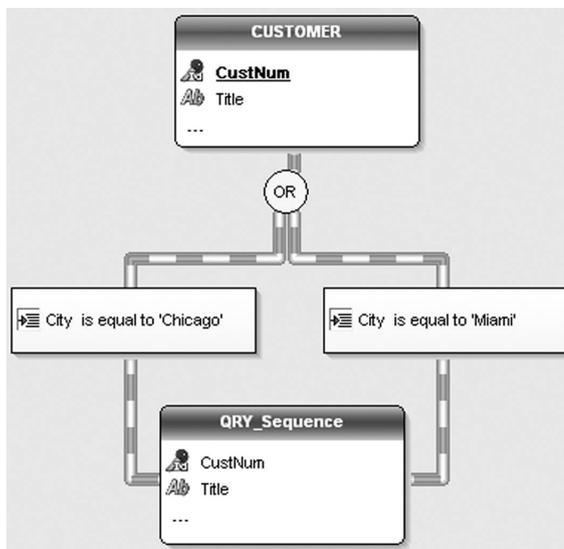
3.3.6 Sequence of selection conditions

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 operator AND:** the selected records will match both conditions. For example, select the customers whose name starts with the letter "A" **AND** who live in Paris.
- **Conditional operator OR:** the selected records will match one of the two conditions or both conditions. For example, select the customers live in London **OR** in Paris.



Conditional operator AND



Conditional operator OR

Two methods can be used to modify the sequence of selection conditions in the current query:

1st method: Via the description window of the query

To modify the sequence of selection conditions in the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the right part of the window, click "Selection condition" and select "Existing conditions".
3. Modify the sequence of the 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:

1. Select "Query .. Query description". The description window of the query is displayed.
2. Display the "And/Or" tab.
3. Modify the sequence of the 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:

1. In the graphic representation of the query, select the selection condition whose sequence must be modified.
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 taken into account.



3.3.7 Displaying the records corresponding to a given period

When a "Date" item is found in a Select 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,
- etc.

To display the records corresponding to a given period in the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed
2. In the middle section of the window, select a "Date" item.
3. In the right section of the window, click "Between two dates".
4. Specify the start date and the end date, then validate.

Note: For the "DateTime" items, when the start date and the end date correspond to values, the query editor proposes preset periods (current month, previous week, floating year, and so on). To use one of these preset periods, click "Preset period" and select the requested period.

3.3.8 Limiting the number of records displayed in a Select query

The query editor enables you to select:

- the first/last n records from the result of a query,
- n records from record x.



For example, you want to find out:

- the ten best customers: the customers who brought in the highest revenues,
- the 15 least selling products,
- etc.

To limit the number of records found in the result of the current Select query:

1. Select "Query .. Query description". The description window of the query is displayed.
2. In the right part of the window, click "The first n".
3. Specify the number of first or last records to select or the number of records to display from a given record.





3.3.9 Keeping (or not) duplicates

The query editor enables you to keep (or not) the duplicated records (duplicates) in the result of a Select query (corresponding SQL keyword: DISTINCT).

The duplicates are kept by default.

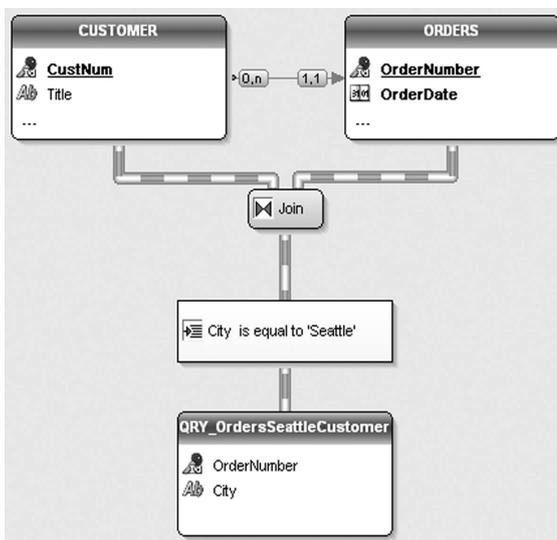
To delete the duplicates, click "Delete the duplicates" in the description window of the query.

To keep the duplicates, click "Keep the duplicates" in the description window of the query.

3.3.10 Joins between the files

When several files are used in a Select query, the query editor automatically detects 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 order 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 is automatically defined by the query editor.



Note: The links between the files are displayed in the graphic representation of the query only if the 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 to the current Select query:

1. Select "Query .. Edit the links/joins". The description window of the joins is displayed.
2. Click the "Add" button. A window allowing you to define a new join is displayed.
3. Specify the characteristics of the join to add.
4. Validate. The join is automatically added to the query.

This enables you to create multiple joins between two files.



To delete a join from the current Select query:

1. Select "Query .. Edit the links/joins". The description window of the joins is displayed.
2. Select the join to delete.
3. Click the "Delete" button. The join is automatically deleted. The joins between the files are optimized in order for the query to operate properly.

Reinitializing the joins of a Select query

Reinitializing the joins is used to cancel all the modifications made to the joins since the query was created. Some optimized joins are automatically re-created by the query editor.

To reinitialize joins of the current Select query:

1. Select "Query .. Edit the links/joins". The description window of the joins is displayed.
2. Click the "Reinitialize" button. The joins that were manually created are deleted.

3.3.11 Synchronizing the query

Synchronizing the query is used to update the WinDev or WebDev elements (reports, controls, and so on) based on the query according to the modifications made to the query.

Only the following modifications are saved and automatically synchronized when the query is closed:

- renaming the query.
- creating one or more result items.
- renaming one or more result items.
- deleting one or more result items.

Reminder: The result items correspond to the items whose value is displayed in the query result.

Whenever the query is saved, a group of information is created if modifications have been made to the query. A group of information contains the modifications made since the last backup of the query.

When the query is closed, the modifications made to the query are automatically applied to the project elements (reports, tables, and so on) that use the query: the query is synchronized.

To view the history of the modifications made to the current query, display the "Synchronize" tab of the configuration options of the query editor ("Display .. Configuration of queries").

The list of these modifications is stored in the ".WDR" file of the current query.

3.4 Opening, closing and saving a query

To open a query, select "File .. Open" and select one or more ".WDR" files. The query is immediately displayed in the query editor.

To close a query, select "File .. Close". If the query contains unsaved modifications, you have the ability to save the query. To save a query, select "File .. Save" or "File .. Save as" to rename and save the query.

Automatic retrieval

To avoid problems when a system lock occurs (power outage for instance), the query editor proposes a system used to automatically retrieve the elements opened in the different editors ("Tools .. XXX options .. General .. Backup for automatic restore").

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 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 ("File .. Versions") and it can be opened at any time. See the next paragraph for more details.

3.5 Managing the versions of a query

The query editor enables you to manage the different versions of each query.

The management of the query versions is used to store the history of the main modifications made to a query. No need to save the same query under different names: the different versions are automatically managed by the query editor and included in the query file (".WDR" file).

The size of the ".WDR" file is proportional to the number of versions saved. Each version saved uses disk space.

The query editor allows you to:

- save a new version of the current query.
- open a version of the current query.
- delete a version of the current query.

These operations are described in the following paragraphs.

3.5.1 Saving a version

Saving a version is used to keep a backup of the query at a given time.

To save a version of the current query:

1. Select "File .. Versions". The window showing the versions of the current query is displayed.
2. Click the "New" button. A window allows you to enter your own comments about the version to create.
3. Validate. The version is saved and the current query is automatically saved.

Note: The different versions of a query are listed in descending order (from the most recent one) according to the date and time of backup.

3.5.2 Opening a version

Opening a version is used to create a new query corresponding to the selected version.

To open one of the versions of the current query:

1. Select "File .. Versions". The window showing the versions of the current query is displayed.
2. Select the version to open.
3. Click the "Open" button. A new query is created in the query editor. This query corresponds to the selected version. The name of this new query has the following format: <Query Name><Date and Time of Version Backup>.

Tip: This new query can replace the source query. To do so, close the source query and save the new query under the same name as the source query. Caution: All the versions of the source query will be lost.



3.5.3 Deleting a version

Deleting the useless versions of a query reduces the size of the ".WDR" file corresponding to the query.

To delete a version of the current query:

1. Select "File .. Versions". The window showing the versions of the current query is displayed.
2. Select the version to delete.
3. Click the "Delete" button. The selected version is deleted from the window for version management. This deletion will be effective the next time the query is saved.

3.6 Documentation of the query

The query editor automatically creates a program documentation for each query. This documentation contains:

- the graphic representation of the query,
- the general information of the query (logical name and physical name of the query, type of query, analysis associated with the query, etc.),
- the information specific to the results of the query (result items, parameters of the query, sort criteria, and so on),
- the information about the characteristics of the query (selection conditions, groups, etc.),
- the SQL code corresponding to the query.

The content of this documentation can be customized, it may contain:

- the full information about the query.
- a selection of information.

To edit the program documentation of the current query in the editor:

1. Select "File .. Print the documentation".
2. Specify the information found in this documentation.
3. Choose the type of edit for the documentation.

Notes: The printout of the query representation can be configured according to:

- printout areas.
- a specified layout ("File .. Documentation layout").

To specify the printout areas of the graphic representation:

1. Select "Display .. Printout areas". Some borders outlining the areas that can be printed in the documentation are displayed in the query editor.
2. Reduce the display of the graphic representation ("Display .. Zoom" or [Ctrl] + mouse wheel) in order to view the entire query.
3. With the mouse:
 - moves 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).





3.7 Optimizing a query

The query editor enables you to optimize the execution speed of one or more queries found in your project. The optimization of a query is based on the following principle: a wizard defines the composite keys to modify and/or to add into the analysis associated with the project.



Important

Reports & Queries does not enable you to optimize the execution speed of a query.

To **optimize the execution speed of the current query**, click  (or select "Query .. Optimize the query"). The window for query optimization is displayed and it proposes the different modifications that can be performed in the analysis.

To **optimize the execution speed of one or more queries**, click . The window for query optimization is displayed and it proposes the different modifications that can be performed in the analysis.

Caution: Adding a lot of composite keys to an analysis increases the size of the index files and slows down the access to the data files.

Note: This option is available only if the following conditions are fulfilled:

- Live Data is enabled ("Project .. Project description", "Live Data" tab). See the online help (keyword: "Live Data") for more details.
- the data files are found in the test directory of the project ("Project .. Project description .. Files"). See the online help (keyword: "Test, Project") for more details.

Optimization and composite keys: to optimize your queries, the editor offers to create composite keys. The order of the components in the composite key is not important. There is no need to create a new key that just reverses the order of the components found in an existing key.

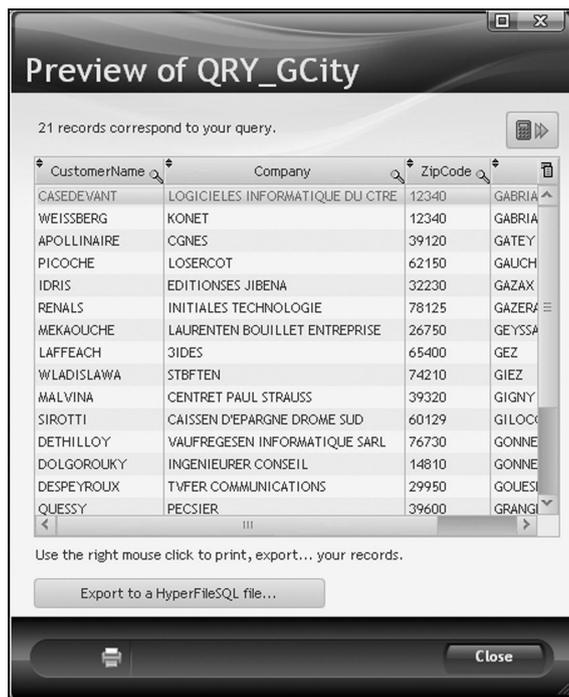


4. RUNNING THE TEST OF A QUERY AND USING IT

4.1 Test of the query

The query editor enables you to run the test of the query result. To do so, click  in the icon bar (or select "Query .. Run the query test").

When running the test of the query, the initialization code of the project is run.



Notes:

- Several tests of different queries can be run at the same time.
- The button  is used to automatically display the average, the total, the maximum value and the minimum value of the numeric result items.



Important

Reports & Queries cannot be used to create and/or test:

- Insert queries.
- Update queries.
- Delete queries.





The result of the test depends on the type of the 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. When running the test, this record is directly inserted into the relevant file.
- The result of an *Update query* corresponds to the records that will be modified. During the test, you can:
 - view the records that must be modified,
 - modify the records in the relevant file directly.
- The result of a *Delete query* corresponds to the records that will be deleted. During the test, you can:
 - view the records that must be deleted,
 - delete the records from the relevant file directly.

Result of the query

The result of the query test can be:

- exported to a text file (see page 209),
- exported to an XML file (see page 210),
- exported to an Excel file (see page 210),
- exported to a HyperFileSQL file,
- printed (see page 211).

A record found in the query result can also be copied to the clipboard ("Copy" from the popup menu).

Running the test of a query with parameters

When running the test of a query with parameters, a window used to specify the different parameters is displayed. See "Query with parameters", page 216 for more details.

Password

If one of the files involved in the query is password-protected, when running 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 the HyperFileSQL errors has not been 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 the project, this management of errors will be used.

Running the test of a query without checking the SQL code

For the queries that use files accessed via a Native Access or via an OLE DB provider, the HyperFileSQL engine may perform no query check.

See "Runtime modes of a query in SQL code", page 182, for more details.

4.1.1 Exporting the query result to a text file

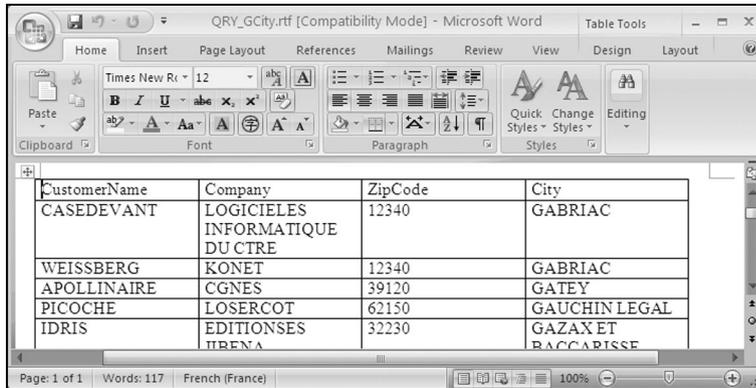
For the next operation, we consider that the result of the query test is displayed in the editor.

To export the query result to a text file:

1. Select "Export the table to Word" from the popup menu (right mouse click) of the query result. The window for selecting the text file is displayed.
2. Specify the name and location of the text file.



- Choose the type of separator used between the different items of each record ("Type"):
 - Separator: *Tabulation*: The items will be separated by tabulations.
 - Separator: *Semicolon*: The items will be separated by semicolons.
- Validate. The records found in the query result are automatically exported to the text file.



The screenshot shows a Microsoft Word window titled 'QRY_GCity.rtf [Compatibility Mode]'. The ribbon includes Home, Insert, Page Layout, References, Mailings, Review, View, Design, and Layout. The main content area displays a table with the following data:

CustomerName	Company	ZipCode	City
CASEDEVANT	LOGICIELES INFORMATIQUE DU CTRE	12340	GABRIAC
WEISSBERG	KONET	12340	GABRIAC
APOLLINAIRE	CGNES	39120	GATEY
PICOCHÉ	LOSERCOT	62150	GAUCHIN LEGAL
IDRIS	EDITIONSES JIBENA	32230	GAZAX ET BACCARISSE

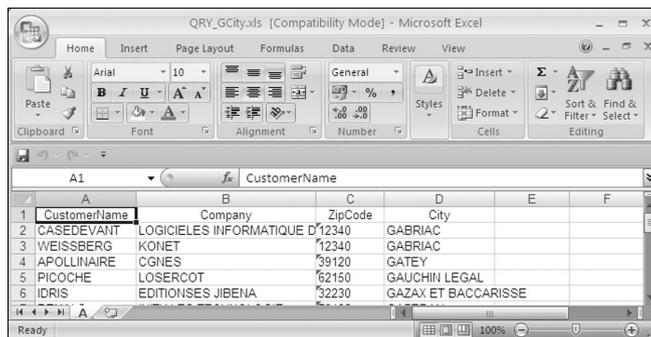
The status bar at the bottom indicates 'Page: 1 of 1', 'Words: 117', 'French (France)', and '100%' zoom.

4.1.2 Exporting the query result to an Excel file

For the next operation, we consider that the result of the query test is displayed in the editor.

To export the query result to an Excel file:

- Select "Export the table to Excel" from the popup menu (right mouse click) of the query result. The window for selecting the Excel file is displayed.
- Specify the name and location of the Excel file.
- Validate. The records found in the query result are automatically exported to the Excel file.



The screenshot shows a Microsoft Excel window titled 'QRY_GCity.xls [Compatibility Mode]'. The ribbon includes Home, Insert, Page Layout, Formulas, Data, Review, and View. The main content area displays the same table as in the previous screenshot, with the following data:

CustomerName	Company	ZipCode	City
CASEDEVANT	LOGICIELES INFORMATIQUE DU CTRE	12340	GABRIAC
WEISSBERG	KONET	12340	GABRIAC
APOLLINAIRE	CGNES	39120	GATEY
PICOCHÉ	LOSERCOT	62150	GAUCHIN LEGAL
IDRIS	EDITIONSES JIBENA	32230	GAZAX ET BACCARISSE

The status bar at the bottom indicates 'Ready' and '100%' zoom.

4.1.3 Exporting the query result to an XML file

For the next operation, we consider that the result of the query test is displayed in the editor.

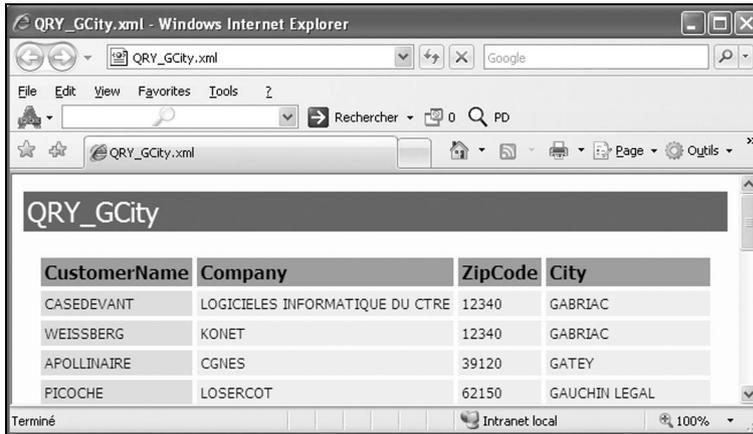
To export the query result to an XML file:

- 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.
- Specify the name and location of the XML file.
- Validate. The records found in the query result are automatically exported to the XML file. An XSL style





sheet is automatically generated.



4.1.4 Printing the result of the query

For the next operation, we consider that the result of the query test is displayed in the editor.

To print the result of the query:

1. Click the "Print" button (or select "Print" from the popup menu of the query result). The window for selecting the items to print is displayed.
2. Specify the items to print and validate. A print preview showing the query result is displayed.
3. Click to print the result of the query.

4.2 Using the result of a query

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, Update 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 214, for more details about using the query's result by programming.



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 by Reports & Queries.

To use a query in a window:

1. Select "File .. New .. Window".
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 is displayed.
4. In the creation wizard, select the query that must be associated with the window and the items that must 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. Select "File .. New .. Page".
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 must be associated with the page and the items that must 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 of a WinDev window or 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, and so on).

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 by Reports & Queries.

To display the result of the query in a list box, combo box, looper or table:

1. Open the window or the page where the list box, combo box, looper or table must be added.
2. Insert the control associated with the query ("Insert .. Control..."). 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 must be associated with the control and the items that must be displayed in the control.
5. Validate. The created control will display the information coming from the associated query.
6. Enter (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 result of the query in the current control.

```
HExecuteQuery(<Name of Associated Query>)
```

To use the result of the query in a control (other than a list box, combo box, looper or table):

1. Open the window or the page where a control associated with a query must be added.





2. Create the control associated with the query.
3. Select "Display .. Selection description". The description window of the control is displayed.
4. Display the "Link" tab in the description window of the control.
5. Select the item to display in "Linked item".
6. Validate. The created control will display the information coming from the associated query.
7. Enter (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 result of the query 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.
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 based on a query", page 48 for more details on how to use a query in a report.

Tip:  can be 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 result of a query is not necessarily displayed in a report, in a window, in a page, etc. This selection can also be used in programming.

Most of the HyperFileSQL functions (browse, search, read, etc.) can be used both on the data files and on the result of a query. To use these functions on a query, the query must be run beforehand (*HExecuteQuery*).

For example, the Customer34 query selects 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 is boolean
ResExecute = HExecuteQuery(Customer34)
// Error running the query ?
IF ResExecute = True THEN
    // Read the first record found in the result of the query
    HReadFirst(Customer34)
    WHILE HOut = False
        // Print an invoice for each record
        PrintCustomer(Customer34.CustomerLName, Customer34.CustomerFName, ...
            Customer34.CustomerAddress, Customer34.CustomerZIP, ...
            Customer34.CustomerCity)
        // Read the next record
        HReadNext(Customer34)
    END
ELSE
    // Display the error message
    Error("The query was not run.")
END
```

Password

If the query is linked to a password-protected analysis, this password must be specified when the analysis is opened (*HOpenAnalysis*) in the initialization process of the project.

If the query is based on one or more password-protected files, these passwords must be specified in *HPass* before the query is run (*HExecuteQuery*).





5. ADVANCED QUERY

5.1 Overview

The steps for creating "Simples" queries have been explained in the previous chapters. The query editor can also be used to create advanced queries. You have the ability to:

- use the query result in another query. This is the concept of sub-queries.
- use parameters in a query. The value of these parameters will be specified when the query is run. This 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 sales amount is greater than the average sales amount for orders. The average amount for orders is not known directly. This average amount will be returned by another query (called sub-query).

A sub-query can be used:

- as selection condition,
- as data source for the main query.

Limits:

- The sub-queries cannot be created in the query editor.
- The ORDER BY statement is not allowed in the sub-queries

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 [NOT] EXISTS (SELECT Item1[, Item2]
                   FROM File1 [, File2]
                   [WHERE ...])
```

```
SELECT Item1 [, Item2]
FROM File1 [, File2]
WHERE Item1 Operator (SELECT Item1[, Item2]
                     FROM File1 [, File2]
                     [WHERE ...])
```

In this syntax, <Operator> can correspond to:

- ALL, ANY, IN, NOT IN or SOME.
- a comparison operator (=, <, <=, >, >=, <> or !=).
- the combination of ALL, ANY or SOME and a comparison operator (=, <, <=, >, >=, <> or !=).



Example: The following SQL code is used to select the number of the orders whose total amount is greater than the average amount of all 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

The queries with parameters allow the user to easily select records according to his own requirements. All he has to do is specify the parameters that must be taken into account.

For example, a query with parameters can be used to select the orders placed by a specific customer. The user only has to select the name of the customer. The query will use this name in parameter.

To include a query with parameters in a WinDev application, in a WinDev Mobile application or in a WebDev site, you must:

- create a query with parameters in the query editor.
- modify the application or the site in order to run a query with parameters.
The query with parameters can be used with a window, a Table control (looper, list box or combo box), a report, etc.

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 do I create a query?", page 170, for more details).
2. When defining the selection conditions:
 - specify that the selection condition corresponds to a parameter.
 - give the name of this parameter.
See "Selection conditions", page 198 for more details.

5.3.2 Running the test of a query with parameters

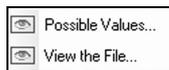
To run the test of a query with parameters:

1. Open the query with parameters in the query editor ("File .. Open").
2. Click  in the icon bar (or select "Query .. Run the query test").
3. Specify the value of the parameters that will be used to run the test of the query.





Note: You can select a specific 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 ("Project .. Project description .. Live Data"). See the online help (keyword: "Live Data") for more details.

4. Validate. The result of the query is displayed according to the specified values.

Note: If no parameter is selected (parameter not checked), the conditions that use the parameters will be ignored.

5.3.3 Using a query with parameters

A query with parameters can be used:

- in a window or in a page.
- in a table, list box, combo box or loop.
- in a report.
- in any case, the entered parameters can be managed in an edit control.

Using a query with parameters in a window or in a page

Principle

When creating a window (or a page), this window (or this page) can be based on a query with parameters. The parameters of the query can be entered in a previous window (or page) for instance.

To use a window (or a page) based on a query with parameters, the query must be run (*HExecuteQuery*):

- before opening the window (or the page).
- in the declaration code of the global variables of the window (or page).

The necessary parameters must be passed when the query is run. These parameters can be passed:

- directly in *HExecuteQuery*.
- by using the following notation <Query Name>.<Parameter Name>, then in *HExecuteQuery*.

See "Two methods for passing parameters to a query", page 221 for more details.

Example

To display the list of orders placed in year XXXX, the user performs the following operations:

- enter the requested year in the *EditParam* window.
- click the "Show the result" button. The *OrderList* window is displayed. This window is based on the *QryParamOrder* query. This query is used to select the orders placed on the date specified by the user.

The following code is used to:

- open the *EditParam* window,
- retrieve the value entered by the user,
- run the *QryParamCustomer* query.

1. Opening the window for entering parameters in the initialization process of the *OrderList* window:



```
-- Initialization process of the CustomerForm window
ParamQuery is string
// Open the EditParam window
// retrieve the input performed by the user
ParamQuery = Open(EditParam)
```

2. Running the query in the initialization process of the *OrderList* window:

```
// Run the query
HExecuteQuery(QryParamCustomer, hQueryDefault, ParamQuery)
// Similar code:
//QryParamOrder.Year = ParamQuery
//HExecuteQuery(QryParamOrder)
```

Using a query with parameters in a table, loop, list box or combo box

Overview

A table, a list box, a loop or a combo box can be used to display the content of a file or query.

A table, a list box, a loop or a combo box can be linked to a query:

- **independent.** This query already exists in the project: select this query among the queries of your project.
- **included in the table, list box or combo box:** this query will be created in the wizard for creating the Table control. This query will only be available for this control and it will be saved with the window that uses it. In this case, the window contains all the necessary elements.
- **with parameters:** this query (independent or embedded) requires parameters to run and to display the result in the table.

Note: this paragraph only presents the tables but the operating mode is similar for the list boxes, the combos boxes and the loopers.

Browsing tables based on independent queries

A browsing table based on an independent query (without parameter) displays the records of this query.

By default, the query is run and loaded in memory when it is first used. The result of a query used in a browsing table remains in memory when the associated window or page is closed. This query will not be re-run when it is re-used.

To re-run this query, you must use:

- **HCancelDeclaration:**
 - If **HCancelDeclaration** is not called, the next initialization of the table control (when the window is re-opened for instance) will not re-run the query. The data retrieved during the last query execution will be used. If the table is based on a query with parameters, see "Browsing table based on a query with parameters (embedded or not)".
 - If **HCancelDeclaration** is called, the next initialization of the table control (when the window is re-opened for instance) will re-run the query.
- **TableDisplay** (or **ListDisplay** for a list box): to force the query to re-run, use **TableDisplay** associated with the **taReExecuteQuery** constant.

Browsing tables based on an embedded query (without parameter)

A browsing table based on an embedded query (without parameter) displays the records of this query.

By default, the query is run and loaded in memory when the table is displayed. The result of an embedded query used in a browsing table is automatically cleared from memory when the associated window is closed. This query will be re-run when it is re-used.

To force the query to re-run, use **TableDisplay** (or **ListDisplay** for a list box) associated with the **taReExecute-**





Query constant.

Browsing table based on a query with parameters (embedded or not)

When a browsing table is linked to a query with parameters (embedded or not), some 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 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
```

In this case, the query is automatically run when the table is initialized.

Note: you can also use one of the following notations:

- MySource:

```
MySource.<Name of Parameter1> = xxx  
MySource.<Name of Parameter2> = xxx  
MySource.<Name of Parameter3> = xxx
```

- the indirection:

```
{ "<Query Name>.<Name of Parameter1>" } = xxx  
or  
{ "<Query Name>" }.<Name of Parameter1> = xxx
```

- **HExecuteQuery**.

One of these two methods can be used in the declaration code of the global variables of the window or in the initialization code of the table.

Special cases:

- Initializing a Table control or using **TableDisplay** without the *taReExecuteQuery* constant.
 - If no query parameter was modified since the last query execution, the query will not be re-run. The table will display the records corresponding to the last query execution.
 - Once the query parameters have been modified, **TableDisplay** (or **ListDisplay** for a list box) is used to re-run the query. The table displays the records corresponding to the specified parameters.
- Using **TableDisplay** with the *taReExecuteQuery* constant.

To force the query to re-run, use **TableDisplay** (or **ListDisplay** for a list box) associated with the *taReExecuteQuery* constant.

Reminder: the structure of the query parameters being reinitialized whenever the query is run, don't forget to redefine the requested parameters before you use this function.



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 the report is opened. The query can be run by *iInitReportQuery*. This function is used to run the query and to pass parameters to this query. This function can be called:

- in the initialization code of the report,
- before *iPrintReport*.

Example

The *CustomerForm* window is used to browse the characteristics of the customers. For each customer, the user can click the *PrintOrder* button to print the report named *ReportCustomerOrder*. This report is associated with the query named *QryCustomerOrder* that selects the orders placed by a given customer.

The following code is used to run the *QryCustomerOrder* query according to the current customer and to print the report:

```
-- Click process of the PrintOrder button
// Run the query according to the current customer
iInitReportQuery(ReportCustomerOrder, CustomerNum)
// Print the report
iPrintReport(ReportCustomerOrd)
```

Parameters coming from edit controls

In most cases, the parameters passed to a query with parameters are entered by the user in edit controls.

In order for your query to run even if no value was entered by the user, check "NULL if empty" for the edit controls ("Details" tab of the control description).

When this option is checked, if the edit control is empty, the value of the parameter passed to the query will correspond to the NULL constant. No error will occur when the query is run: the conditions that depend on this parameter will be ignored.

For example, the SQL code run is:

```
SELECT NameOfItems
FROM NameOfFiles
WHERE Item = {Param1}
```

In 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 NameOfItems FROM NameOfFiles
No value	Option unchecked	SELECT NameOfItems FROM NameOfFiles WHERE Item = ''
A value is entered	Option checked or unchecked	SELECT NameOfItems 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, in a page or in a control (list box, loop, combo box or table):

- pass the parameters in **HExecuteQuery** directly.
- use the following notation `<Query Name>.<Parameter Name>`, then **HExecuteQuery**.

Note: If the query with parameters is used in a report, the mode for passing parameters changes. 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 query parameters.

The syntax used is as follows:

```
<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 query parameters, you must specify the initialization mode of the query.

Notation `<Query Name>.<Parameter Name>`

To pass parameters to a query, all the query parameters can be specified to the HyperFileSQL engine before the query is run. This method enables you to specify the parameters in any order (useful for 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.
- No indirection operator and no dynamic compilation (**Compile**) can 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 that were previously specified are ignored.
- The value of the parameter can correspond to NULL.
- If one of the parameters is not specified, the corresponding condition is ignored

Do not specify 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.



For example: let's consider a query ("Customers_LastName_FirstName") whose SQL code is as follows:

```
SELECT * FROM CUSTOMER WHERE LNAME= {Param1} AND FNAME = {Param2}
```

- The 2 parameters are specified:
HExecuteQuery(Customer_LastName_FirstName, hQueryDefault, "Smith", "John")
will run the query:
SELECT * FROM CUSTOMER WHERE NAME = 'Smith' AND FIRSTNAME = 'John'
- Only the name is specified:
HExecuteQuery(Customer_LastName_FirstName, hQueryDefault, "Smith")
will run the query:
SELECT * FROM CUSTOMER WHERE NAME = 'Smith'
- Only the first name is specified. However, the name must be specified and it must correspond to Null.
sName is Variant = Null
// use the mandatory variant for the Null variable (not filled)
HExecuteQuery(Customer_LastName_FirstName, hQueryDefault, sLastName, "John")
Or
HExecuteQuery(Customer_LastName_FirstName, hQueryDefault, Null, "John")
will run the query
SELECT * FROM CUSTOMER WHERE NAME = 'John'





INDEX

A

Add	
Item	190
Join	199
Operation	194
Selection condition	195
Sort on an item	193
Automatic retrieval	200

B

Bar	
Icons	163
Tools	160, 163

C

Calculation	193
Characteristics of a query	190
Closing a query	200
Combining queries	175
Combo box	
Use a query with parameters	214
Count	190
Creation	
Query	166

D

DELETE	174
Delete	
Item	192
Join	200
Operation	194
Records	174
Selection condition	196
Sort on an item	193
Version of a query	202
Delete query	174
Creation	174
Description window	189
Selection condition	194
Test	204
Description	

Description window	183
Display	
First n records	198
Item	192
Last n records	198
Documentation	202
Duplicates	199

E

Embedded query	182
Excel	206
Exporting the query	205

F

File	
Excel	206
Text	205
XLS	206
XML	206
Full-text	190
Full-text search	190

G

Given period	198
--------------	-----

H

HExecuteQuery	210
HExecuteSQLQuery	210
hQueryWithoutCorrection	178

I

Independent query	182
INSERT	169
Insert query	169
Creation	170
Description window	189
Test	204
Inserting records	169
Interface	
Query Editor	160
Item	190



Count	190	Parameters	212
Operation	193	Pass parameters	217
Selection condition	194	Password	205, 210
Sort	192	Print	
		Documentation	202
		Result	207
		Printout area	202
		Project	165
		Q	
		Query	
		Advanced query	211
		Automatic restore	200
		Available types	166
		Characteristics	190
		Close	200
		Create	166
		Creation mode	182
		Definition	157
		Delete	174
		Delete query	174
		Description	183
		Documentation	202
		Duplicate	199
		Embedded	182
		Embedded query	182
		Execution	210
		Export	205
		Export the result	205
		First n records	198
		Independent	182
		Independent query	182
		Insert	169
		Insert query	169
		Item	190
		Join	199
		Last n records	198
		Open	200
		Operation	210
		Optimize the execution speed	203
		Overview	157
		Print	207
		Printout area	202
		Rename	200
		Reverse engineering	178
		Save	200
		Select	167
		Select query	167
		Selection condition	194
Count	190		
Operation	193		
Selection condition	194		
Sort	192		
		J	
Join	199		
Add	199		
Delete	200		
Reinitialize	200		
		K	
Keyboard (shortcuts)	164		
Keyboard shortcuts	164		
		L	
List box			
Use a query with parameters	214		
Looper			
Use a query with parameters	214		
		M	
Menu	160		
Option	162		
Menu options	162		
Mode of query	182		
Modify			
Selection condition	195		
Sequence of conditions	197		
Values displayed	194		
		O	
Open			
Query	200		
Version of a query	201		
Operation	193		
Add	194		
Delete	194		
Optimizing the execution speed	203		
		P	
Page			
Using a query with parameters	213		
Pane	161		



T		Using the result of a query	207
Table		V	
Use a query with parameters	214	Version	201
Test	204	Delete	202
Tools	163	Open	201
Type of queries	166	Save	201
U		W	
UNION	175	Window	
Union query	175	Main window of the editor	160
Creation	176	Use a query in a window	208
Description window	188	Window control	
UPDATE	171	Use a query in a control	208
Update		WLanguage	210
Records	171	X	
Update query	171	XML	206
Creation	172		
Description window	189		
Selection condition	194		
Test	204		



DEVELOP 10 TIMES FASTER

PART 3

Reports & Queries



PCSOFT



REPORTS AND QUERIES - SUMMARY

1. DISTRIBUTING REPORTS AND QUERIES	231
1.1 Overview	231
1.2 Distributing Reports and Queries with your WinDev application	231
1.3 Step 1: Configuring the project	231
1.4 Step 2: Configuring the analysis	232
1.4.1 Visible files and items	232
1.4.2 Translating the names of files and items	232
1.5 Step 3: Configuring the reports	233
1.6 Step 3b: Configuring the queries	233
1.7 Step 4: Creating the executable and distributing the application	234
1.8 Step 5: Installing the WinDev application on the end-user computers	235
2. USING REPORTS AND QUERIES	236
2.1 Starting Reports and Queries	236
2.2 Principle for using reports	236
2.2.1 Modifying a report	236
2.2.2 Using a modified report	236
2.2.3 Using a new report	237
2.3 Principle for using queries	237
2.3.1 Modifying a query	237
2.3.2 Using a modified query	237
2.3.3 Using a new query	238
3. LIMITS COMPARED TO THE STANDARD REPORT EDITOR OF WINDEV	239







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 (also called Report Editor User Version).

Reports & Queries replaces "WDEtat User Version". This module is now free and it can be distributed along with your WinDev applications. See the license for more details on distribution and conditions.

1.2 Distributing Reports & Queries with your WinDev application

To distribute Reports & Queries with your own WinDev application, you must:

1. Specify in the project that the application allows Reports & Queries to be run.
2. Define in the analysis the files and the items that can be handled by Reports & Queries.
3. Define the reports and the queries that can be modified in Reports & Queries.
4. Create the executable and the setup program of the WinDev application.
5. Install the application on the end-user computers.

Note: To reduce the size of the setup program, "Reports & Queries" can be downloaded directly from your application:

- via the automatic menu (automatic help menu that can be included in a WinDev application),
- via the print preview of a report,
- when using *RunReportsAndQueries*.

Translate the names of the files and items

You are distributing "Reports & Queries" in different countries (it is available in French and in English). Different languages are used: Spanish, Romanian, Chinese, Swedish, etc.

If the end user wants to create reports or queries, he must handle the names of files and items, the names used in the analysis by the developer of the application. In most cases, these names are in English, which is not suitable for the Chinese or Romanian user.

You now have the ability to specify a translation for the names of files and items in the analysis.

1.3 Step 1: Configuring the project

To use "Reports & Queries" from your application, the project associated with your application must be configured:

1. Display the description window of your project ("Project .. Project description").
2. Select the "Charter" tab and check "Allow Reports & Queries to be started".
3. Validate.

Notes:

- By default, the WinDev projects allow Reports & Queries to be run.
- The initialization code of the project is run when a report test or a query test is run from Reports & Queries. In this case, *OnTestMode* returns False.



1.4 Step 2: Configuring the analysis

1.4.1 Visible files and items

By default, all the files and their items can be used in Reports & Queries. The end users 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 instance), the WinDev analysis must be modified: simply make the files and/or items you want "invisible" in Reports & Queries.

Note: The reports and the queries created from WinDev and based on invisible files and/or items can be run and/or modified in Reports & Queries.

To define an invisible file in Reports & Queries:

1. Select the requested file in the data model editor of WinDev.
2. Display the description window of the file items ("Structure of files .. Description").
3. In the "Info" tab, clear the "Visible by the end user (Reports & Queries)" option.
4. Validate.

Note: All the 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 file containing the requested item.
2. Display the description window of the file ("Structure of files .. Items").
3. Select the requested item.
4. In the "General" tab, clear the "Visible by the end user (Reports & Queries)" option.
5. Validate.

Note: *VisibleEndUser* can also be used to find out and modify the visibility of an item in Reports and Queries

Caution: To take into account the modifications made to the WinDev analysis, the WinDev analysis must be generated ("Analysis .. Generation").

1.4.2 Translating the names of files and items

By default, when the files and 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 (not in the language of the user). You have the ability to propose a translation for the following elements in the WinDev analysis:

- File name,
- Caption of the file,
- Item name,
- Caption of the item.

Note: If your project must 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 file in Reports & Queries:

1. Select the requested file in the data model editor of WinDev.
2. Display the description window of the file items ("Structure of files .. Description").
3. In the "Reports & Queries" tab, specify the file name and its caption in the different languages supported by the analysis.

Note: The name used must not correspond to a file name found in the analysis.

4. Validate.



**To define the multilingual parameters of a file item in Reports and Queries:**

1. In the data model editor of WinDev, select the file containing the requested item.
2. Display the description window of the file ("Structure of files .. Items").
3. Select the requested item.
4. In the "Reports & Queries" tab, specify the item name and its caption in the different languages supported by the analysis.
5. Validate.

Caution: To take into account the modifications made to the WinDev analysis, the WinDev analysis must be generated ("Analysis .. Generation").

1.5 Step 3: Configuring the 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 the report ("Report .. Description").
3. In the "Details" tab, check "Report modifiable in Reports & Queries".
4. Check:
 - "Modifiable interface" so that only the interface of the report can be modified in Reports & Queries.
 - "All modifiable (interface and code)" so that the entire report can be modified in Reports & Queries.
5. Validate and save the report.

To make several reports of your application modifiable in Reports & Queries:

1. Select "Project .. Performance and improvements .. Wizard for improving the project". The wizard for improving the project starts.
2. Display the wizard screen named "Reports & Queries".
3. Select "Yes - Allow "Reports & Queries" to be run". Go to the next screen.
4. For each affected report, check:
 - "Modifiable interface" so that only the interface of the report can be modified in Reports & Queries.
 - "Total control of reports (modifiable interface and code)" so that the entire report can be modified in Reports & Queries.
5. Validate.

Notes:

- The `InterfaceModifiableWithReportsAndQueries` property is used to find out whether the interface of a report can be modified in Reports & Queries.
- The `AllModifiableWithReportsAndQueries` property is used to find out whether a report can be modified in Reports & Queries.

1.6 Step 3b: Configuring the queries

To make a query of your application modifiable in Reports & Queries:

1. Open the query in the query editor.
2. Display the description window of the query ("Query .. Query description").
3. Click the "Advanced" button.
4. In the "Reports & Queries" tab, check "Visible in Reports & Queries".
5. Validate and save the query.



To make several queries of your application modifiable in Reports & Queries:

1. Select "Project .. Performance and improvements .. Wizard for improving the project". The wizard for improving the project starts.
2. Display the wizard screen named "Reports modifiable by "Reports & Queries"".
3. Select "Yes - Allow "Reports & Queries" to be run". Go to the next screen.
4. For each affected query, check "Query visible" in order for the query to be visible in Reports & Queries.
5. Validate.

1.7 Step 4: Creating the executable and distributing the application

The method for creating and distributing a WinDev application that uses 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, and so on), create the corresponding executable ("Workshop .. Generate the Windows executable").

Create the setup program ("Workshop .. Create the setup procedure"). In the wizard for creating the setup program, in the "Additional modules" plane:

1. Check "Include the setup of Reports & Queries". The setup package of Reports & Queries is automatically sought. This setup package is found in the "Install\Reports & Queries" sub-directory of the WinDev setup directory.

Caution: Including Reports & Queries significantly increases the size of your setup program (about 120 MB).

The setup package of Reports & Queries can also be distributed separately. If Reports & Queries is not included in the setup, it can be downloaded directly from the application (via the automatic menu, from the print preview of a report or by the *RunReportsAndQueries* function. By default, the software is downloaded on an FTP site of PC SOFT. You have the ability to customize the download address of the software (site of a company for instance).

2. Specify (if necessary) the directories corresponding to the reports and queries modified or created by the end user:
 - the directory of the custom reports and queries corresponds to the directory of the reports and queries visible only by the user who created them.
 - the directory of the shared reports and queries corresponds to the directory of the reports and queries visible by all the users of the application.
3. Allow (or not) these directories to be modified when installing the application.

Note: *fReportsAndQueriesDir* is used to find out the directory of the custom and/or shared reports and queries.

Customizing the download address

If Report & Queries is not included in the setup program, its download will be offered automatically when using the automatic menu, when viewing a report or when starting the program.

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 must be found in the same directory as the executable of the application.
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 (WX15/latest for example)
```

Notes:

- If the download package is found at the root, write "." in the Dir entry.
- The name of the file to download is "WD150PACKER.exe".

1.8 Step 5: Installing the WinDev application on the end-user computers

The installation of an application that uses Reports & Queries is the same as the installation of 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 is not included in the setup program, its download will be proposed automatically when using the automatic menu, when viewing a report or when starting the program.



2. USING REPORTS & QUERIES

2.1 Starting Reports & 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 to the main window of your application ("Windows .. Main menu .. 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.

2.2 Principle for using the 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 to the directory of "Custom Reports & Queries" (defined when creating the setup procedure of the 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 code of the 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 users of the application, the end user must select "File .. Share 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:

- in the directory of "Custom reports and queries" first.
- then in the directory of "Shared reports and queries".
- and finally in the library of the WinDev application.

Note: An indication found in the title of the 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 must physically delete the file of the modified report (".WDE" file) from the directory of "Custom reports and queries" and/or from the directory of "Shared reports and queries".





2.2.3 Using a new report

The end users will be able to print their new reports directly from Reports & Queries:

- by starting a printout ().
- by running the test of the report ( or "Code .. Run 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 in your application.

You can for instance include a window that allows the end user to:

- select a report created in Reports & Queries.
- print this report.

2.3 Principle for using the queries

2.3.1 Modifying a query

When modifying a query, the corresponding file (".WDR" 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 the WinDev application).

This is the query that will be modified by the end user.

2.3.2 Using a modified query

Two types of queries are available:

- the queries used as data source for one or more reports.
- the other queries.

Query used as data source for a report

If the query is used as data source for a report, the modified query will be automatically included in the report.

Other query

The end users will be able to run their modified queries:

- directly from Reports & Queries by clicking  or by selecting "Query .. Run the query test".
- 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 instance include a window that allows the end user to:

- select a query created in Reports & Queries.
- run the query and display its result in a table (with *BuildBrowsingTable* for instance).



2.3.3 Using a new query

The end users will be able to run their new queries:

- directly from Reports & Queries by clicking  or by selecting "Query .. Run the query test".
- 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 instance include a window that allows the end user to:

- select a query created in Reports & Queries.
- run the query and display its result in a table (with *BuildBrowsingTable* for instance).





3.LIMITS COMPARED TO THE STANDARD REPORT EDITOR OF WINDEV

Reports & Queries cannot be used to:

- create reports based on a HyperFileSQL view.
- create reports based on a memory zone.
- create reports based on a table control.
- create reports based on a variable.
- create reports based on a programmed data source.
- create skin templates of reports.
- create Insert queries.
- create Update queries.
- create Delete queries.
- use functions for modifying records (*HAdd*, *HModify*, *HDelete*, and so on).

However, Reports & Queries allows you to customize all the types of reports created in the standard report editor of WinDev (color, size, font, and so on).



