

DEVELOP 10 TIMES FASTER



SUMMARY

Introduction

Preliminary points	15
Overview of tutorial.....	15
Legend of symbols used in this guide	16
How to access the online help?	16
If you are familiar with WEBDEV 22.....	18
What is WEBDEV used for?	19

PART 1 - DISCOVERING WEBDEV

Lesson 1.1. Discover WEBDEV

Overview	24
Starting WEBDEV.....	24
Development environment	25
The editor.....	25
The menu bar (ribbon) in details.....	26
The environment colors	27
WLanguage	28

Lesson 1.2. The Web development and WEBDEV

Principle of Browser/Server	30
How does it work?	30
And in WEBDEV?	30
Practical example.....	31
Intranet/Extranet/Internet	35
Principles	35
And in WEBDEV? (Classic/AWP, PHP, Static).....	35
Practical example.....	36
The method for creating a site.....	38

Lesson 1.3. My first pages

Overview	40
Opening the project.....	40
Creating a "form" page.....	41
Creating the page	41
Creating controls	42
Page improvements	46
Modifying the style of "Login" and "Password" controls.....	48
Page test.....	50
Saving the data in a data file	51
Creating the data file.....	51
Saving data	53
Page test.....	55

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

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

This documentation is not contractually binding. PC SOFT reserves the right to modify or delete any topic dealt with in this document.

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.
© PC SOFT 2018: This publication may not be reproduced in part or in full without the express consent of PC SOFT.

Viewing the data typed.....	55
Adding input checks.....	56
Mandatory input.....	56
Checking the password.....	57
Simple search in a database.....	58
Creating the page.....	58
Creating controls.....	58
Page test.....	61
Conclusion.....	62

PART 2 - MY FIRST DATABASE

Lesson 2.1. Overview

Overview of project developed in this part.....	66
---	----

Lesson 2.2. WEBDEV and the databases

Overview.....	68
HFSQL.....	69
The different modes for accessing the databases.....	69

Lesson 2.3. Project and analysis

Overview.....	72
Creating the project.....	72
Creating the analysis.....	74
Creating the description of data files.....	75
Creating a data file: using a preset file.....	76
Creating a data file: creating the file and items.....	77
Importing a CSV file.....	82
Direct import of existing data files.....	85
Creating links.....	86
Generating the analysis.....	92

PART 3 - INTRANET SITE WITH DATA

Lesson 3.1. Overview

Overview of site created in this part.....	98
--	----

Lesson 3.2. Principle for displaying a dynamic WEBDEV site

Principle for displaying a dynamic WEBDEV site.....	102
The WEBDEV session.....	102
The page contexts.....	102
Programming.....	102

Lesson 3.3. Pages for addition and modification

Overview.....	105
Creating a dynamic page used to list the products.....	105
Creating a page that is using a template.....	105
Creating controls.....	108
Page test.....	113
Modifying products via a dynamic "Product form" page.....	114
Creating the page.....	114
What should this page do?.....	114
Creating edit controls.....	115
Displaying data in the page.....	118
Creating buttons.....	118
Managing the product image.....	121
Displaying the form from the list of products.....	124
Adding a new product via the "Product form" page.....	127
Checking the product addition.....	129
Viewing the records.....	131

PART 4 - INTERNET SITE WITH DATA

Lesson 4.1. Overview

Overview of site created in this part.....	136
--	-----

Lesson 4.2. Principle for displaying an AWP site

Principle for displaying a WEBDEV AWP site.....	140
What is an AWP page?.....	140
Operating mode of AWP sites.....	140
How to share information (values) between AWP pages?.....	142

Lesson 4.3. Creating AWP pages

Overview.....	144
Creating an AWP page used to list new products.....	144
Creating the page.....	144
Creating the list of new products.....	146
Viewing the details of a new product.....	152
Creating the page.....	152
What should this page do?.....	153
Creating controls.....	153
Displaying data in the page.....	157
Displaying the form from the list of products.....	158
Managing styles.....	159
Importing a CSS style sheet.....	160
Applying the CSS styles.....	161
Linking the Internet site and the Intranet site.....	163

Creating a page used to display the general sales conditions.....	165
Creating the page.....	165
Creating the control.....	165
Conclusion.....	167

PART 5 - LET'S CONTINUE THE DEVELOPMENT

Lesson 5.1. Multicriteria search

Overview.....	172
Creating the query used to find orders.....	173
Creating the query.....	173
Query test.....	177
Using parameters in the query.....	178
Test of query with parameters.....	182
Creating the page proposing the multicriteria search.....	183
Creating a page.....	183
Creating the controls used to configure the criteria and to display the result.....	183
Optimizing the page display.....	199

Lesson 5.2. Printing an invoice

Overview.....	201
Principles for printing on Internet.....	201
Direct print.....	201
Generating documents in HTML, PDF, XML,	202
What is a report made of?	202
Creating the "Invoice" report	203
Creating the query.....	203
Creating the report based on a query.....	206
Modifying the "Invoice" report.....	212
Displaying the printed report from a button.....	216
Implementing the print	216
Print test	217

Lesson 5.3. Sending an email

Overview.....	220
A popup page to send emails	221
Creating the popup page.....	221
Creating controls used to type the email characteristics	221
Sending the email	223
Page improvements.....	224
Closing the popup page	224
Opening the popup page.....	226

Lesson 5.4. Identifying the user: the user groupware

Overview.....	229
Integrating the user groupware.....	230
Configuring the user groupware	231
Including the customer section in the site	231
Site test.....	233
Creating users and groups.....	234
Defining rights	236
Site test.....	238
Disabling the management of user groupware	239

Lesson 5.5. Managing the multilingual feature

What is a multilingual site?	241
Choosing the project languages.....	241
Localizing the project elements	244
Localizing an image.....	245
Localizing the menu	246
Localizing controls.....	246
Localizing a programming message.....	247
The translation tools.....	248
Direct input of translations	248
Translation with WDMMSG and WDTRAD	249
Other elements to translate: the framework messages	249
Programming the change of language.....	250
Adding a menu option	250
Programming.....	251
Project test.....	252

Lesson 5.6. The referencing

Overview.....	254
Reminder : AWP mode	254
The available methods.....	254
Referencing wizard.....	255
Properties of pages	257
Frequency of update	257
Importance in the site	257
Site map path.....	257
Physical naming of pages.....	258

PART 6 - DEPLOYING A SITE**Lesson 6.1. Deploying a site**

Overview	262
Required configuration	262
Installing and configuring the "WEBDEV Application Server - 10 Connections"	263
Installing the "WEBDEV Application Server - 10 Connections"	263
Configuration via the Hosting Control Center	264
Creating a deployment account.....	267
The different deployment modes.....	268
A detailed example: deployment by FTP.....	269
Correspondence between the directories of the development computer and the deployment.....	274

Lesson 6.2. Managing a site

Overview	276
Local WEBDEV administrator	276
Remote WEBDEV administrator.....	278

PART 7 - SPECIFIC WEB FEATURES**Lesson 7.1. The styles**

Overview	284
Discovering styles	284
A control: several elements	284
Discovering the styles via an example.....	285
Selecting the CSS style of a control.....	286
Editing the CSS style of a control.....	286
The WEBDEV styles	288
Implementing styles: a practical example	290
Overloading a CSS style for a control.....	291
Modifying a CSS style for all the controls	291
Styles and rich text	292

Lesson 7.2. The cookies

What is a cookie?	294
What is a cookie made of?	294
Practical example.....	295
How to use the cookies?	295

Lesson 7.3. Secure transactions and payment

Securing the information and the pages via TLS/SSL.....	297
Overview	297
Implementing secure transactions via the TLS/SSL protocol.....	297
Transactions secured by TLS\SSL in a WEBDEV site	297

Secure payment.....	297
Overview	297
System for secure payment in a WEBDEV site.....	299

Lesson 7.4. Anchoring and zoning

Overview.....	302
Creating a page in zoning mode.....	302
Implementing the anchors	305

PART 8 - RESPONSIVE WEB DESIGN**Lesson 8.1. Responsive Web Design**

What is Responsive Web Design?	310
Benefits.....	310
The concepts in WEBDEV.....	310

Lesson 8.2. Responsive Web Design and WEBDEV

Overview	312
My first page in Responsive Web Design mode	312
Creating a page	312
Fluidity and Responsive Web Design	314
Resolution blades	314
How to adapt the controls to the different blades?	316

Lesson 8.3. Advanced concepts

Controls specific to Responsive Web Design	320
The anchors.....	321
Positioning anchors.....	322
Size anchors	323
Order of controls in Responsive Web Design mode.....	324

PART 9 - USEFUL FEATURES OF WEBDEV**Lesson 9.1. The internal components**

Overview	334
Step by step.....	334
Step 1: Creating an internal component.....	334
Step 2: Using the internal component.....	338

Lesson 9.2. Automatic management of errors

Overview 342

 Operating mode..... 342

 Implementation..... 342

 Types of affected errors 343

Automatic management of errors: a training example..... 343

Lesson 9.3. Import/Export

Importing elements..... 348

Exporting elements 349

Specific import operations 350

 Importing a WINDEV project 350

 Importing an HTML page..... 351

PART 10 - OPTIMIZING AND DEBUGGING A PROJECT

Lesson 10.1. Overview

Overview 356

 Opening the project..... 356

Lesson 10.2. Project audits

What is an audit? 358

Static audit..... 358

 Procedure not run..... 360

 Orphan element 361

 Cleaning the project 362

Dynamic audit..... 362

Lesson 10.3. Performance profiler

Overview 366

Starting the performance profiler 366

Studying the result..... 367

Lesson 10.4. Debugging a project

Overview 373

Using the debugger..... 373

PART 11 - MANAGING A HFSQL CLIENT/SERVER DATABASE

Lesson 11.1. Introduction

Overview 382

Why switch a site to HFSQL Client/Server mode? 383

Lesson 11.2. Implementation of a Client/Server database

Overview 385

Installing a local HFSQL server 385

Creating a site that is using a HFSQL Client/Server database 386

Adapting a site to use a HFSQL Client/Server database..... 386

 Overview 386

 Adapting the example 386

Features available in HFSQL Client/Server mode 390

Lesson 11.3. Managing a Client/Server database

Overview 392

Configuring the computers 392

The HFSQL Control Center 392

Creating a user account in the HFSQL Control Center..... 394

Saving the database..... 399

Conclusion..... 399

PART 12 - SOURCE CODE MANAGER

Lesson 12.1. SCM

Introduction..... 404

SCM (Source Code Manager)..... 404

 Principle of SCM 404

 Creating the SCM database..... 405

Lesson 12.2. Including a project in SCM

Including a project in SCM 407

 Adding the project into SCM 407

 Opening a project from SCM..... 411

 Configuring SCM..... 412

Lesson 12.3. Using a project in the SCM

Overview 414

Modifying a project parameter 414

Modifying a project page..... 415

 Modifying the checked-out element..... 416

 Checking the checked-out element back in..... 416

Synchronizing the project 418

Off-line mode (or mobile mode).....	419
SCM administrator	419
Disconnecting from SCM	420

APPENDICES - THE WLANGUAGE BASICS

Appendices 1. Vocabulary

Main terms used	424
-----------------------	-----

Appendices 2. The variables

What is a variable?	432
Declaring a variable	433
Assignment and use.....	433
The types of variables.....	434
The scope of variables.....	434
Global scope.....	434
Local scope.....	435
Summary scope diagram.....	435
Simple operations on the variables	436
Tips.....	437
Details of variable type: the String variables.....	437
The String type.....	437
Practical example.....	439
Details of another variable type: the arrays	442
Declaration	442
Filling an array and accessing the elements.....	442
Advanced arrays.....	443

Appendices 3. The conditional statements

Overview	445
The IF and SWITCH statements.....	445
The IF statement	445
The SWITCH statement	446
Practical example: Using the IF and SWITCH statements.....	447
Project used.....	447
Creating the page	448
Creating the page controls for the conditional IF statement.....	448
Conditional IF statement	449
Creating the page controls for the conditional SWITCH statement	450
SWITCH statement	452

Appendices 4. The loops

Overview	455
The FOR statement	455
The LOOP statement	455
The WHILE statement	456

Practical example: Using loops	457
Project used.....	457
Creating the page	457
Creating controls	457
Page test.....	461

Appendices 5. The procedures

Overview	463
Types of procedures	463
Creating and calling a procedure	463
Global procedure.....	463
Local procedure.....	463
Internal procedure.....	464
Calling a procedure	464
Parameters of a procedure.....	464
What is a parameter?.....	464
Mandatory or optional parameters?	466
Result of a procedure	466
Practical example: Using a procedure	467
Project used.....	467
Implementation.....	467
Creating and using a procedure	470
Conclusion.....	472

Appendices 6. Questions/Answers

Questions/Answers.....	474
How to view the element to which the current process belongs?	474
How to print the source code?	474
How do I perform a "find and/or replace"?.....	474
What is the meaning of the "+" and "-" signs in the code editor?	475
Is it possible to identify the person who wrote a code line?	475
Is it possible to find out the number of a code line?	475
Is there a method to easily display the syntax or the help about a function?	476
What are the useful shortcuts in the code editor?	476

Conclusion

INTRODUCTION

Preliminary points

Caution: This manual is a tutorial. We advise you to check the online help when using WEBDEV.
The purpose of the tutorial is to help you discover WEBDEV, become familiar with the editors and teach you the concepts of WEBDEV.

This manual does not cover all the WEBDEV features.

You should plan on spending a few hours to follow this course and to learn WEBDEV: this is a good investment!

If you try to develop a site before following this tutorial, you will lose time, and a lot more than the few hours you would have spent on this tutorial.

This tutorial was designed so you can approach it in two different ways:

- either you follow all the detailed exercises in each lesson (recommended method).
- or, if you are in a hurry and already experienced, you can read it without doing the exercises (all the exercises have screen shots). However, in order to quickly assimilate the main concepts, we recommend that you follow the tutorial step by step.

WEBDEV evolving all the time, the screen shots found in this tutorial may differ from the windows and pages displayed in your product.

Overview of tutorial

The tutorial was designed to progressively teach you how to use WEBDEV. By following this tutorial:

- you will discover the main concepts explained informally; these are the concepts you must learn and understand.
- you will also be asked to perform operations that illustrate the concepts just explained.

As you progress through the tutorial, if you want to take a closer look at a concept or if you want to get more details about a programming function, see the online help (accessible from the editors). The size of a lesson is not necessarily proportional to its relevance ...

Don't forget to also take a look at the examples supplied with WEBDEV: they are very instructive!

Legend of symbols used in this guide

-  This symbol indicates the duration of the lesson. Please note that the actual time may vary according to your level of experience.
-  An example is available to complement the lesson. The examples are available via the home window of WEBDEV.
-  This symbol introduces a "Tip": reading the associated text is strongly recommended.
-  This symbol introduces a "Warning": reading the associated text is essential.
-  This symbol introduces a "Note": reading the associated text is recommended.
-  This symbol introduces a feature specific to Internet: reading the associated text is strongly recommended.

Notes The tutorial may have evolved since this document was published. Don't hesitate to check the online version of tutorial, the PDF file can be accessed from the WEBDEV menu: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Tutorial (PDF)".

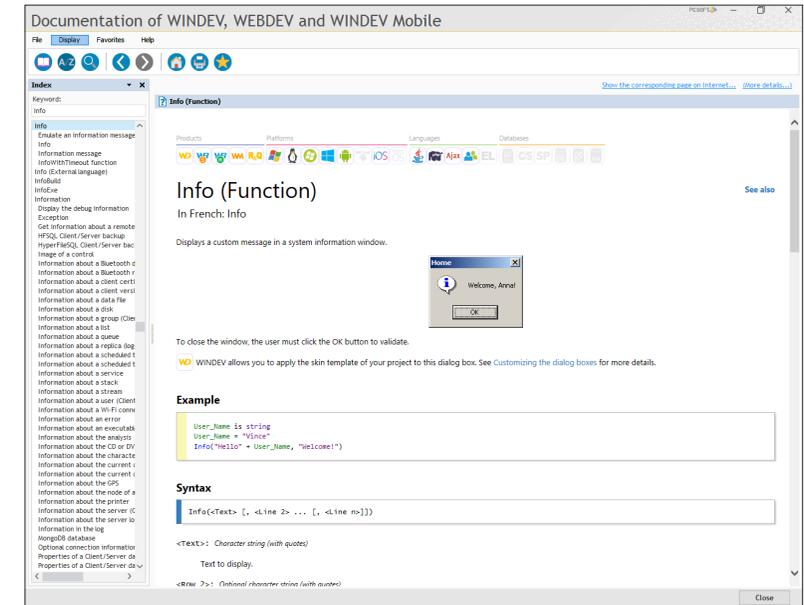
How to access the online help?

The online help of WEBDEV allows you to get detailed information about the 3500 WLanguage functions. It also contains the help about the editors and the controls, tips, ...

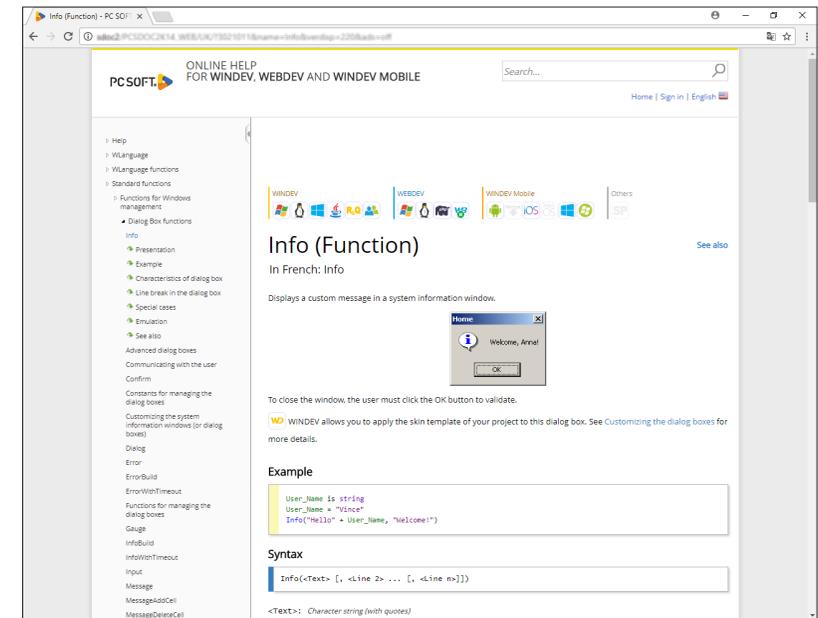
The online help is available at any time in WEBDEV:

- In the code editor, a specific help is available for each function via the [F1] key.
- Each dialog box displayed by WEBDEV proposes a button  allowing you to access the corresponding help page.
- The help menu of the editors ("Help" option available on the "Home" pane, in the "Online help" group of the WEBDEV menu) allows you to start the online help.

- ▶ The help can be displayed:
 - in a specific "help browser":



- in an Internet browser, if you have access to Internet:





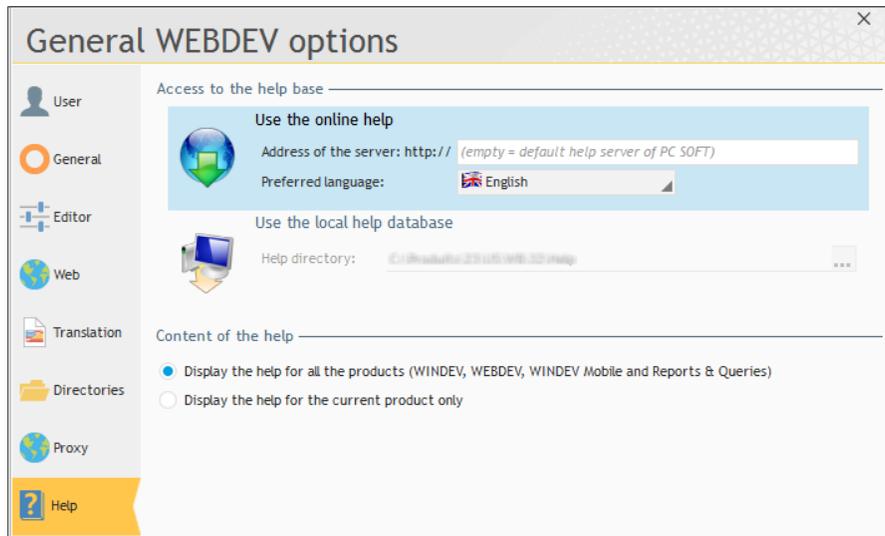
Notes

The online help of WINDEV, WEBDEV and WINDEV Mobile on Internet is available from any computer equipped with an Internet access, without the product being necessarily installed. This help is updated on a regular basis.

Each Web user can add comments about the documentation pages: personal notes, examples, links, ...

To start the Internet online help from the product:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
2. In the "Help" tab, select:
 - the access mode to the help database.



- the help content: help common to WINDEV, WEBDEV and WINDEV Mobile or help for the product currently used.

If you are familiar with WEBDEV 22...

If you are familiar with WEBDEV 22, following this tutorial will do no harm: it's a good opportunity to "review" the WEBDEV features!

What is WEBDEV used for?

WEBDEV is an IDE (Integrated Development Environment). It allows you to develop Internet and Intranet sites in many fields:

- E-commerce (sales, rentals, bookings, ...)
- Multimedia (description of companies, "showroom" sites, ...)
- Intranet (logins, secure accesses, ...)
- ...

WEBDEV is a full development environment that includes all the tools required for developing and maintaining Internet or Intranet sites.

Unlike other programming languages, there is no need to find and add modules to be able to design, check and install a site.

The WEBDEV 5GL (5th Generation Language), WLanguage, will surprise you by its simplicity: a few hours are all you need to get the hang of it, a week is usually all it takes to fully master its potential!

No more programming hassle, WLanguage is available in English and in French!

PART 1

Discovering
WEBDEV



LESSON 1.1. DISCOVER WEBDEV

This lesson will teach you the following concepts ...

- Starting WEBDEV



Estimated time: 5 mn

Overview

WEBDEV is an IDE (Integrated Development Environment) targeted for Internet/Intranet development: e-commerce, multimedia, ...

The developed sites can give access to information stored in the databases.

WEBDEV allows you to create:

- static Internet/Intranet sites. These sites manage data that does not change (corporate sites, sites available on a CD, ...).
- dynamic Internet/Intranet sites, that manage data. The WEBDEV sites access all the databases, relational or not, available on the market.

In this tutorial, you will learn how to create your sites (with or without database) and how to improve them by using the features proposed by WEBDEV.

Starting WEBDEV

- ▶ Start WEBDEV 23 (if not already done).
- ▶ If WEBDEV 23 was never started before, a welcome wizard is displayed:
 - If you worked with an earlier WEBDEV version, this wizard allows you to retrieve the existing configurations.
 - If you are a new user, this wizard allows you to configure your environment. This allows you to choose the screen configuration used and to configure the Control Centers. See the online help for more details.
- ▶ If WEBDEV 23 was already started, identify yourself if necessary. The development environment starts. The home window is displayed. This home window is used to:
 - create a project,
 - open an existing project,
 - open an example,
 - open one of the projects found in the tutorial.

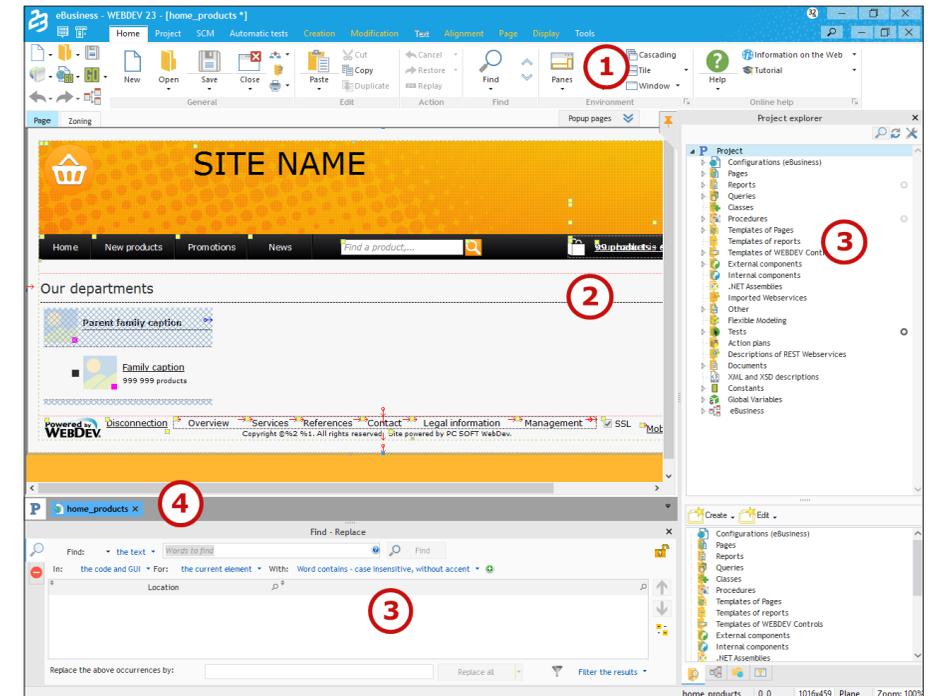
Let's take a look at the development environment of WEBDEV.

Development environment

The editor

The development environment of WEBDEV includes a specific interface and several editors allowing you to create the different elements of your applications.

For example, the page editor is used to create pages, the report editor is used to create reports, ... All the editors are using the same environment:



1. **Menu of editors**, displayed in the shape of a ribbon (we'll see how to use it in the next paragraph).
2. **Current editor** (page editor here). This space allows you to view the element currently created or modified in WYSIWYG (What You See Is What You Get).

3. **Panes**. The WEBDEV interface includes several horizontal and vertical panes allowing you to quickly access different types of information.

Some examples:

- The "Project explorer" pane (displayed on the right) is used to list all the project elements by category.
- The "Find - Replace" pane (displayed at the bottom) is used to quickly perform searches in your project.

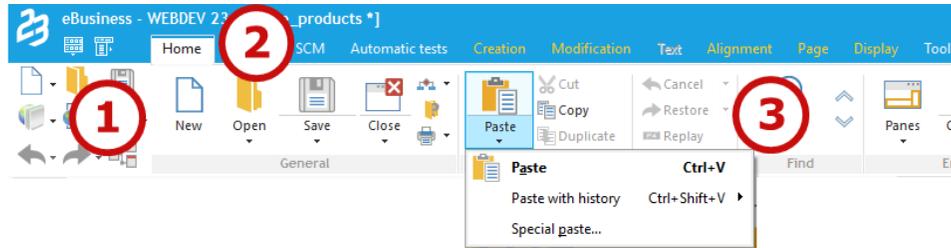
These panes can be hidden by pressing [CTRL] + [W] if necessary.

4. **Bar of opened documents**. This bar is used to quickly view all the opened elements. A simple click on the button corresponding to the element displays it in its own editor.

The menu bar (ribbon) in details

The menu bar of WEBDEV is presented in the shape of a ribbon. This ribbon includes panes in which the options of the editors are grouped.

We are going to take a closer look at the main elements of the ribbon, as well as how we will be using it in this tutorial.



The different ribbon elements

The ribbon includes three areas:

- the button area, on the left (1).
- the pane area, at the top (2).
- the option area (3).

Let's take a closer look at these areas.

The button area (1)

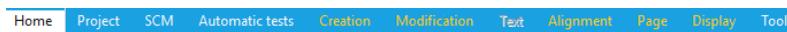


The button area groups the **quick access buttons**. These buttons are used to perform the most usual operations, common to all the editors: save, open, create, ...

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

- The product logo is used to display the "About" window, the custom menus and the drop-down menus found in the former interface of editors.
- The 2 other logos are used to restore the toolbars and the drop-down menus found in the former interface of editors.

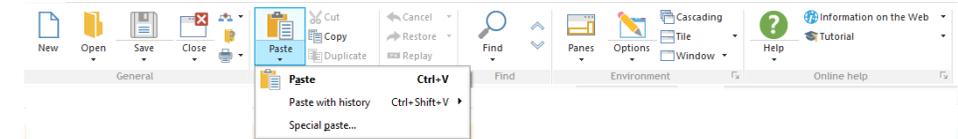
The pane area (2)



The different ribbon panes are used to access the options of the different editors for the current project. Several types of panes are available:

- the current pane: The pane tab appears in light gray and a blue line is displayed at the top of tab.
- the popup panes, specific to the current element: The pane name is displayed in orange.
- the available panes: The pane name is displayed in white.

The option area (3)



The options displayed in the ribbon differ according to the selected pane. Several types of options are available:

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

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

In this tutorial, to identify a menu option, we will be talking about panes and groups.

For example:

To display the help, on the "Home" pane, in the "Online help" group, click "Help".

The environment colors

The environment is using a light theme by default.

Several other themes are also available:

- Light theme, grey ribbon. In this mode, the menu bar is not colored anymore: it is grayed.
- Grey theme. In this mode, the environment and the different interface windows are displayed on a light gray background.
- Dark theme. In this mode, the environment and the different interface windows are displayed on a black or dark gray background.
 - ▶ To modify the theme used by the environment:
 1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
 2. In the "Editor" tab, in the "Themes" area, select the theme to use.
 3. Validate. The theme will be taken into account during the next start of WEBDEV.

Note: To improve the readability of this manual, the light theme will be used for the different images that illustrate the operations to perform.

WLanguage

WLanguage is the programming language common to WINDEV, WEBDEV and WINDEV Mobile.

If you are not familiar with this language, we advise you to follow the lessons presenting the programming concepts from page 431. These lessons present the different types of variables, the main WLanguage statements, the procedures, ...

LESSON 1.2. THE WEB DEVELOPMENT AND WEBDEV

This lesson will teach you the following concepts ...

- Principle of Browser/Server
- Intranet / Extranet / Internet



Estimated time: 30 mn

Principle of Browser/Server

How does it work?

An Internet or intranet site operates as follows :

- The client (the Web user) is using a browser to access the site.
- The browser sends a request to the server that is hosting the requested site. In this request, it indicates the page that must be displayed and different parameters allowing the server to build the corresponding page.
- The server receives this request, processes it and returns the corresponding "HTML" page. HTML (HyperText Markup Language) is the language used by all the browsers to display the Web pages.

Therefore, two types of processes are available :

- Processes executed at the browser's level, on the user's computer.
- Processes performed at server level.

The code run at browser level is called JavaScript code. Browsers only know how to run JavaScript code.

And in WEBDEV?

With WEBDEV, everything is developed:

- in WYSIWYG ("What You See Is What You Get") in the editor: your pages are visually identical in creation and at run time.
- in WLanguage for the programming side.

WEBDEV converts your page created in the editor into HTML page that can be read by the browsers.

The server code is executed in WLanguage.

The browser code is converted into JavaScript.

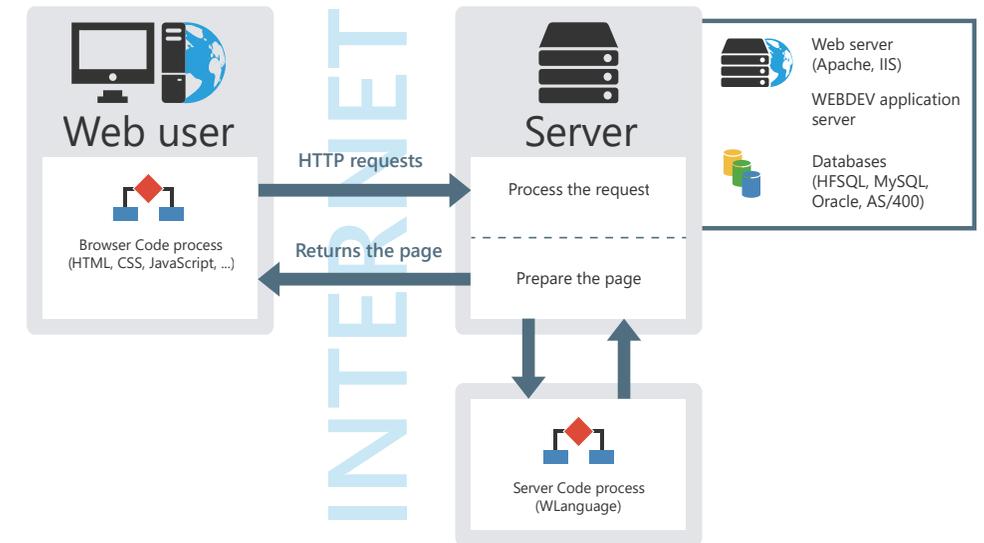
To create a site with WEBDEV, a single language is required: WLanguage.

However, two types of code are available: server code and browser code.

Why this distinction between server/browser? For performance reasons.

Indeed, between the browser and the server stands Internet, with its response time, latency, ...

Some simple operations can be performed directly on the browser, thus avoiding having to reach the server.



Practical example

- ▶ To better understand the difference between the server processes and the browser processes, a simple example was prepared for you:
 1. Start WEBDEV 23 (if not already done). Display the home window of WEBDEV if necessary: press [CTRL <].
 2. Open the "WEBDEV concepts" project. To do so, in the home window, click "Tutorial" and select the project named "WEBDEV concepts (Exercise)".



Tip

If the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "WEBDEV concepts (Exercise)". All the projects found in the tutorial are available from this menu.



Notes

If the UAC is enabled in Windows, a specific window may be displayed. This window indicates that the WD230Admin.exe program will be run on the current computer. Grant the authorization. This program corresponds to the local WEBDEV administrator that is used to run the test of sites developed with WEBDEV. Its features will be presented later in this tutorial.

► Open the "PAGE_Registration" page in the editor: double-click its name in the project explorer.



Notes

The project explorer is used to display in the environment the list of all elements found in the project. These elements are grouped by theme: Pages, Procedures, ...
 To display the project explorer:
 1. On the "Home" pane, in the "Environment" group, expand "Panels".
 2. In the list of panes that is displayed, select "Project explorer".



► This page contains edit controls and a "Register" button. The "Register" button must perform two things:

1. Check that all the controls have been filled.
2. Save the value of each control in the database.

► Let's see the code associated with the button:

1. Select the "Register" button.
2. Display the popup menu of the button (right mouse click) and select "Code".
3. The code editor is displayed with the different processes linked to the Button control. To see all the processes associated with the button, press the [Page Up] key on your keyboard.

```

Initialization of BTN_Subscribe (server)
If Error: by program When Exception: by program
...
Click (onclick) of BTN_Subscribe (browser)
// Check whether the "EDT_LastName" control is equal to an empty string (excluding spaces and punctuation)
IF EDT_LastName == "" THEN
// The control is empty, display an error message to the user
Error("Enter your name")
// Return in edit into the "EDT_LastName" control (without running the rest of code)
ReturnToCapture(EDT_LastName)
L END
// Check whether the "EDT_FirstName" control is equal to an empty string (excluding spaces and punctuation)
IF EDT_FirstName == "" THEN
// The control is empty, display an error message to the user
Error("Enter your first name")
// Return in edit into the "EDT_FirstName" control (without running the rest of code)
ReturnToCapture(EDT_FirstName)
L END
// Check whether the "EDT_Email_address" control is equal to an empty string (excluding spaces and punctuation)
IF EDT_Email_address == "" THEN
// The control is empty, display an error message to the user
Error("Enter your email address")
// Return in edit into the "EDT_Email_address" control (without running the rest of code)
ReturnToCapture(EDT_Email_address)
L END
// Check whether the "EDT_Password" control is equal to an empty string (excluding spaces and punctuation)
IF EDT_Password == "" THEN
// The control is empty, display an error message to the user
Error("Enter your password")
// Return in edit into the "EDT_Password" control (without running the rest of code)
ReturnToCapture(EDT_Password)
L END
// The browser code is ended, the page will send the values fo controls to the server and will
// ask it to run the server code of the button
Click of BTN_Subscribe (server) AJAX
// Reset the customer structure
    
```



Notes

The code editor enables you to write the code for your processes. The code editor presents the events associated with each control, which means the events on which a specific process can be run.
 Note: The processes are displayed in the order in which they will be run. For example, the processes associated with the Button control are:

- Initializing.
- Browser click.
- Server click.

Let's study the code displayed: the server code and the browser code are identified by different colors:

- The browser code (that will be run on the computer of Web user) is colored in green.
- The server code (that will be run on the server) is colored in yellow.

In our example, all the input checks are performed in browser code (green code). For example, the EDT_LastName control must not be empty. The corresponding code is as follows:

```
// Check whether the "EDT_LastName" control is equal to empty
// string (excluding spaces and punctuation)
IF EDT_LastName ~= "" THEN
    // The control is empty, display an error message to the user
    Error("Enter your name")
    // Return in edit into the "EDT_LastName" control (without
    //running the rest of code)
    ReturnToCapture(EDT_LastName)
END
```

This check is performed in browser code because there is no need to go back to the server to check whether the controls are filled.

This is used to avoid useless round trips and to reduce the wait for the Web user: the navigation is more fluid.

Once the browser code was run, the page sends the values typed to the server. Then, the server runs the server code. In the server code, you have the ability to process the information received.

In our example, the information received is added into the database via the following code:

```
// Reset the customer structure
HReset(Customer)
// Retrieve the values of controls in the customer structure
ScreenToFile()

// Add the customer into the database
HAdd(Customer)
```

This operation cannot be performed in the browser code because the database is common to all the users and therefore located on the server.

- ▶ Close the code window (use the cross in the top right corner).
- ▶ Close the page displayed in the editor (use the cross in the top right corner).

Intranet/Extranet/Internet

Principles

An **Intranet** or **Extranet** site is often considered as being a management application in Web mode, which means an application run in a browser.

This Web application can present:

- business features intended for specific users,
- processes that must be secured: not everyone should be able to access the application

The Web application can be accessed:

- from a company network only, in which case we talk of **Intranet site**.
- from Internet, in which case we talk of **Extranet site**.

In both cases, the Web application is secured via a management of logins, passwords, rights, ...

An **Internet site** is a site meant for "public" consumption (business users or regular users).

An Internet site must be easily found on the Web. Some examples: presentation site, e-commerce site, ...

To bring more Web users to your site, the site must be referenced by the search engines. In order for the search engines to reference each page properly, an additional constraint appears: the site pages must be accessible at any time. But this constraint is also a guarantee of simplicity for the Web user: he can easily copy/paste a link from a page and re-use this link whenever required.

And in WEBDEV? (Classic/AWP, PHP, Static)

In WEBDEV, to develop an **Intranet** or **Extranet** site, the "classic" mode is more suitable because it includes the following features: integrated security, automatic contexts.

Indeed, the classic mode includes automatic sessions. The session identifier is included in the URL. The address for the pages depends from this identifier which changes for each connection.

Drawback: The search engines cannot index this site.

In WEBDEV, to develop an **Internet** site, you can choose one of the following modes:

- The AWP mode (Active WEBDEV Page).
- The PHP generation engine.
- The static mode if your site contains preset pages only (no database).

Practical example

► First of all, let's see the operating mode of an Internet page:

1. In the "WEBDEV_Concepts" project (that was opened at the beginning of this lesson), open the "PAGE_Internet" page in the editor: double-click its name in the project explorer.

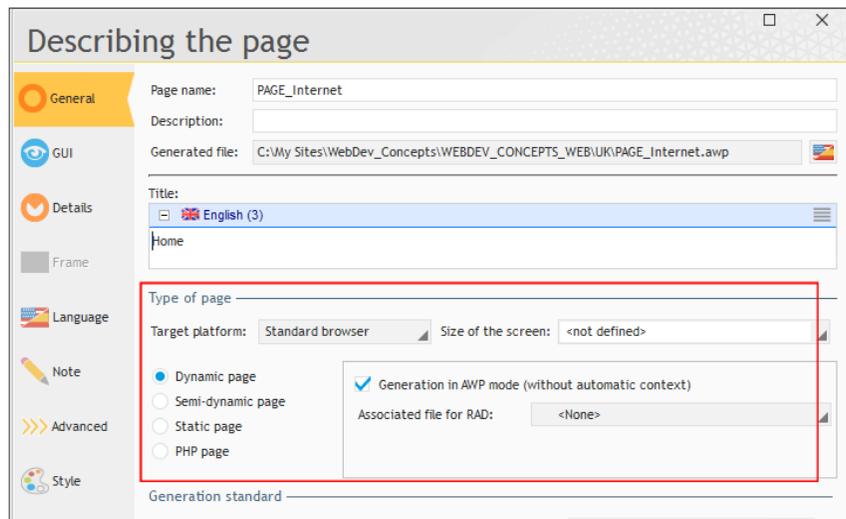


TIP

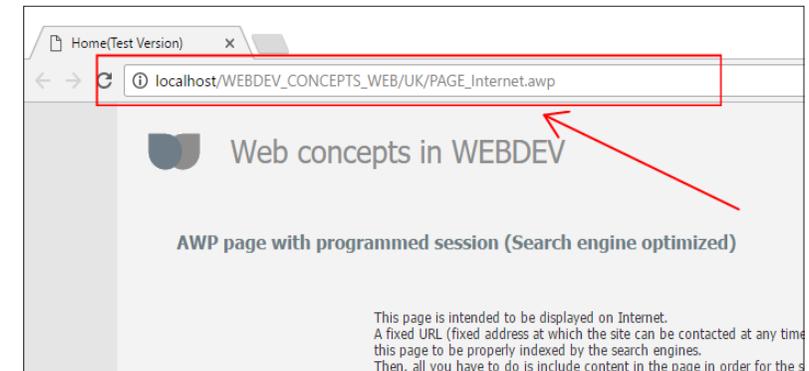
To quickly find a page in the current project, press [CTRL] + [E]. A window is displayed, allowing you to perform a search on all the pages containing the set of letters typed in the search area. Simply select the page you want and validate and the page opens up in the editor.

2. Let's check the page type:

- On the "Page" pane, in the "Description" group, click "Description" (you can also select "Description" from the popup menu of the page).
- In the "General" tab, the page is defined as a dynamic page with an AWP generation mode.



- Validate the description window of page.
3. Run the test of this page: click  among the quick access buttons.
 4. The page appears in your default browser.
 5. Study the address displayed in the browser: this page has a fixed address. If you copy this address and if you paste it in a new browser, the same page will be displayed!



6. Close the browser.

► Now, let's check the behavior of an Intranet page:

1. Open the "Page_Home_of_Intranet_site" page in the editor: double-click its name in the project explorer.
2. Let's check the page type:
 - On the "Page" pane, in the "Description" group, click "Description" (you can also select "Description" from the popup menu of the page).
 - In the "General" tab, the page is defined as a dynamic page without generation in AWP mode. This type of page is used in Intranet/Extranet mode.
3. Run the test of this page: click  among the quick access buttons.
4. The page comes up in the browser.
5. Study the address displayed in the browser. The URL has the following format: `http://localhost/WEBDEV_Concepts/CONNECT_MODETEST/<connection identifier>` where <connection identifier> is a string that changes at each connection. The address changes at each connection therefore the security is improved. However, this behavior is not compatible with a referenceable site.



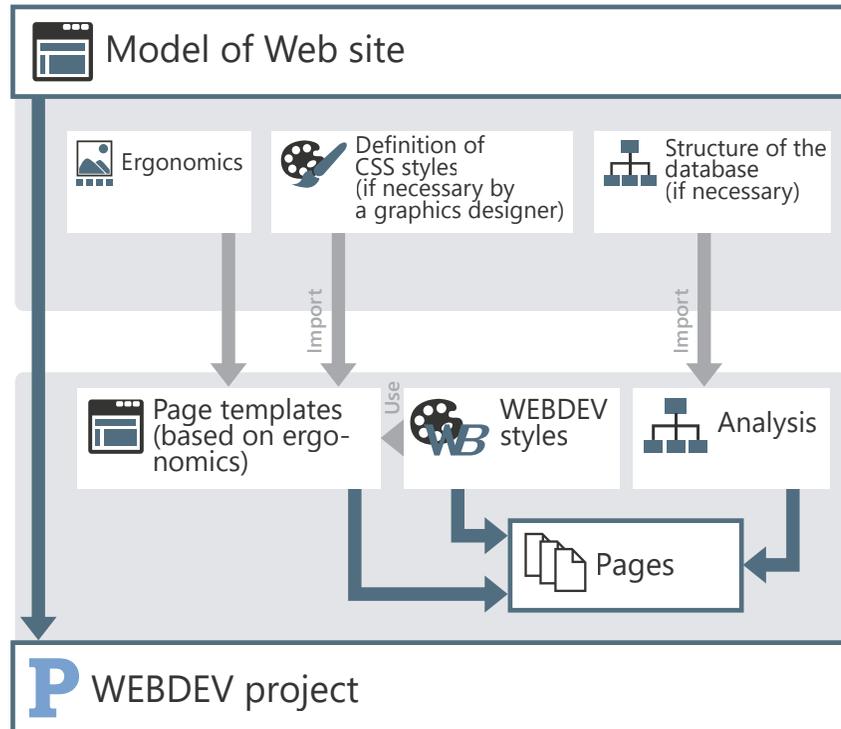
6. Close the browser.

The method for creating a site

Now that the main Web concepts have been presented, let's see how to develop a WEBDEV site. To create a site with WEBDEV, use the following method:

- Creating the site model:
 - Model of pages (user experience).
 - Defining the CSS styles (if necessary).
 - Designing the database structure.
- Development in WEBDEV:
 - Creating a project that groups all site elements.
 - Creating the page templates that will define the site style. These templates are based on the model of pages, the CSS styles and the WEBDEV styles.
 - Define the database (data model), that can re-use an existing structure.

DESIGN



WEBDEV development

LESSON 1.3. MY FIRST PAGES

This lesson will teach you the following concepts ...

- Creating a form page
- Saving data
- Checking the input
- Simple search in a database



Estimated time: 30 mn

Overview

To start developing with WEBDEV, we are going to create pages.



Notes

The pages are used to display or type information on the screen. The Web user can directly act on the pages via controls, buttons, ...

These examples will allow you to understand the operating mode of Internet/Intranet sites and will allow you to handle your first database.

Opening the project

- ▶ Start WEBDEV 23 (if not already done). Close (if necessary) the current project to display the home window.
- ▶ Open the "My_First_Pages" project.
To do so, in the home window, click "Tutorial" and select the "My first pages (Exercise)" project.



Important!

In this section, we will focus on the creation of simple pages. The "My_First_Pages" project is an empty project that was created beforehand. The creation of a project will be presented in another lesson.



Notes

A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My first pages (Answer)".

Creating a "form" page

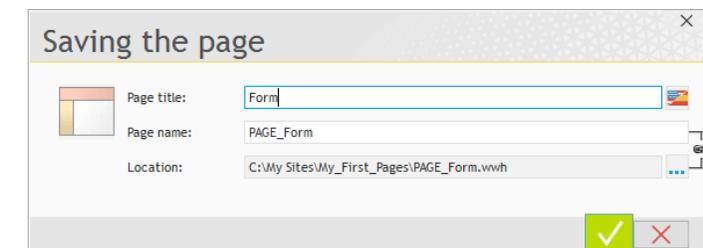
We are going to create a registration form. This form will contain several controls used to identify the person.

Creating the page

- ▶ To create the page:
 1. Click  among the quick access buttons.



2. The window for creating a new element is displayed. This window is used to create all the elements that can be associated with a project.
3. Click "Page" then "Page". The wizard for page creation starts.
4. Select "Blank page" then "Simple layout" and validate (green button at the bottom of window). The page is automatically created in the editor. The backup window of page is displayed. This window displays:
 - The page title. This title will be displayed in the title bar of browser.
 - The page name. This name corresponds to the logical page name. This name will be used to handle the page by programming.
 - The location. This location corresponds to the backup directory of file corresponding to the page.
5. Type the page title: "Form".



6. The page name (that will be used in programming) is automatically deduced from the page title. If this name does not suit you, you have the ability to modify it and to specify a title that differs from the page's name.
7. Validate (green button).



Notes

For each page created in the editor, WEBDEV generates several types of files :

- A "WWH" file that contains the page description for the page editor. This file is saved in the project directory.
- An "AWL" file that corresponds to the page description for the WEBDEV engine in the EXE sub-directory of project.
- An "HTM" file that contains the HTML code and the JavaScript code of the page that will be sent to the browser. This file is saved in the "\<Project name>_WEB\XX" sub-directory of project directory (one sub-directory per language, for example "FR" for French, "UK" for English, ...).

Creating controls

Let's create the different form controls. These controls will allow the user to specify the data required for registration.

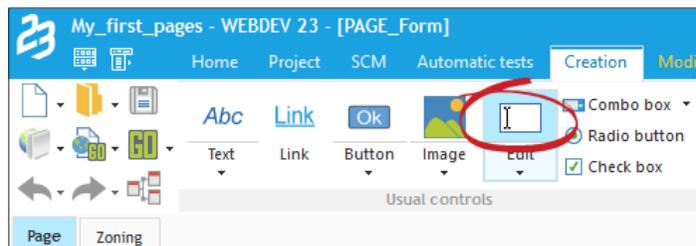
The form includes the following controls:

- LastName
- FirstName
- Address
- Email
- Login
- Password

All these controls are edit controls, where the user will type data.

► To create an edit control:

1. On the "Creation" pane, in the "Usual controls" group, click .



2. The control currently created follows the move of the mouse.
3. Click inside the blank page to create the control. The control is created and handles are displayed to indicate that the control is selected.



Tip

You must create a control? Display the "Creation" pane of WEBDEV: all the available controls are accessible in this pane.

► To modify the control caption:

1. Click the control (simple click): the caption becomes editable. If you prefer to use the keyboard, press the [SPACE] key or the [ENTER] key in order for the caption to become editable.



2. Type "Name" and press the [ENTER] key to validate. The caption is immediately modified. The control name is also modified: it appears in the tooltip when the control is hovered by the mouse cursor: EDT_Name.



Notes

Study the control name proposed by WEBDEV: this name starts with the letters "EDT_". This prefix is automatically added because the project is using a programming charter.

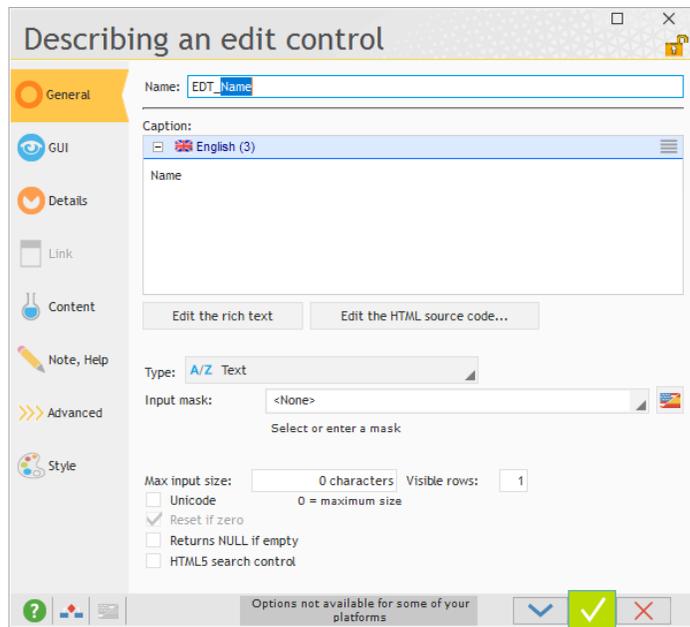
The programming charter is used to define a prefix for each type of object, allowing you to quickly identify the element :

- a page starts with "PAGE_" (we saw it when we saved the page).
- an edit control starts with "EDT_".
- a button starts with "BTN_", etc.

You have the ability to disable this charter if you don't want to use it : in the ribbon, on the "Project" pane, in the "Other actions" group, expand "Charter" and uncheck "Use the charter".

A simple modification was performed on the edit control: change its caption.

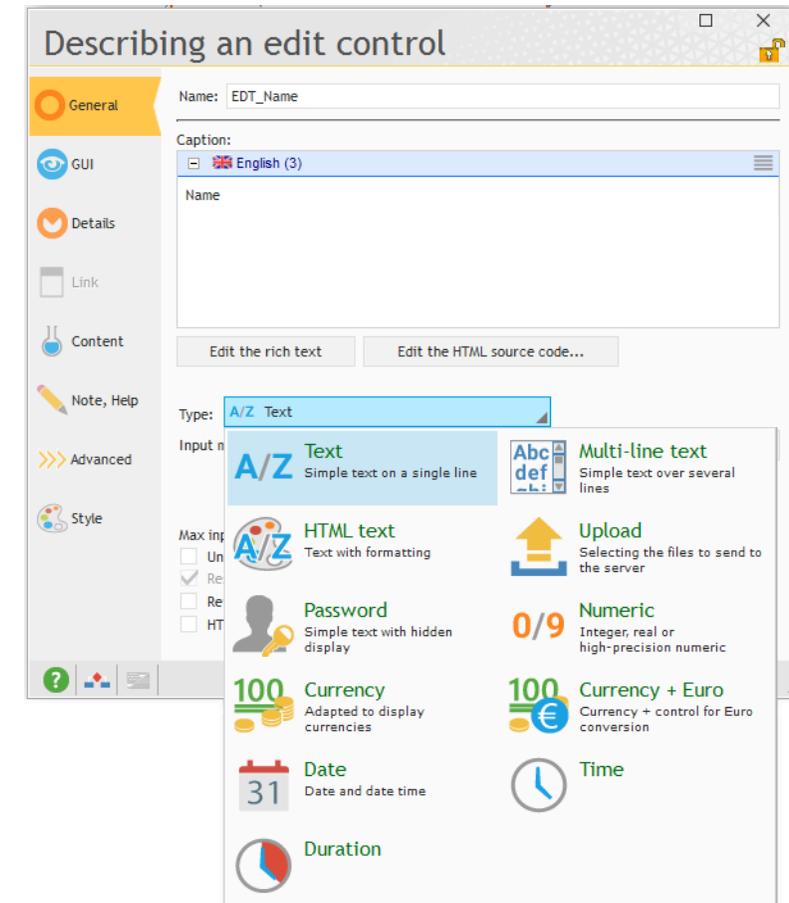
- ▶ To modify the detailed characteristics of control, all you have to do is double-click the control. A description window of control is displayed:



- ▶ We are going to modify the maximum input size: all you have to do is type the new size (30 in our case) in the "Max input size" control. Several other characteristics can be modified. To save the modifications, click the green validation button.
- ▶ Now, it's your turn. Use the same method to create the following edit controls, below the "Name" control:

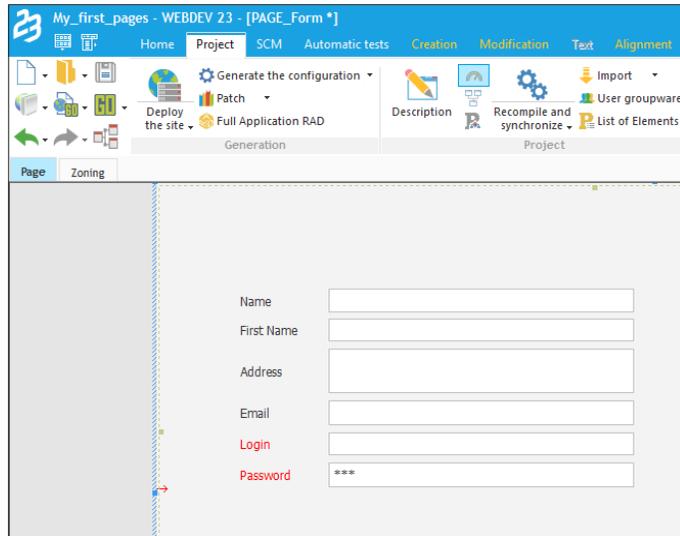
Caption	Type	Size
FirstName	Text	30
Address	Multi-line text	
Email	Text	255
Login	Text	10
Password	Password	10

You may have noticed that these controls do not have the same type. To modify the type of the edit control, display the description window of control.



The basic form is created. Let's improve it now.

Let's see the page that we want to get:



- The "Address" edit control is used to type several lines.
- The different edit controls are aligned.
- The "Login" and "Password" controls have a red caption.

Page improvements

We are going to perform some modifications in the interface of this page.

First of all, let's take a look at the "Address" control. This control must display several lines on the screen. Let's make it bigger.

Then we'll align the controls inside the page.

Finally, we will modify the style of the "Login" and "Password" controls.

- ▶ To enlarge the "Address" control:
 1. Click the "Address" control.
 2. Handles come up.
 3. With your mouse, grab the handle at the bottom of the edit area and increase the size of the area.
- ▶ This way you can increase the width of the various controls in the page.

- ▶ To align the controls in the page:

1. Select the "Name" control then all the edit controls found in the page (keep the [CTRL] key down while clicking the different controls).



Notes

Caution: the first selected control is essential. Indeed, the alignment options are based on:

- The size of first selected control: it will be used as reference for the size of other controls.
- The position of first selected control: all the selected controls will be aligned in relation to the first selected control.

2. Display the "Alignment" pane of WEBDEV menu. This pane contains all the alignment options available for the controls..



3. We want the left and right sides of controls to be aligned. Click "Justify (Ins and Out)".



Notes

If you don't know which alignment to choose, hover the different options proposed by the "Alignment" pane of WEBDEV. When an option is hovered by the mouse cursor, the controls selected in the page are automatically positioned according to the hovered option.

If the positioning suits you, all you have to do is click the option.

If the positioning does not suit you, all you have to do is click inside the page: the controls are moved back to their initial position.

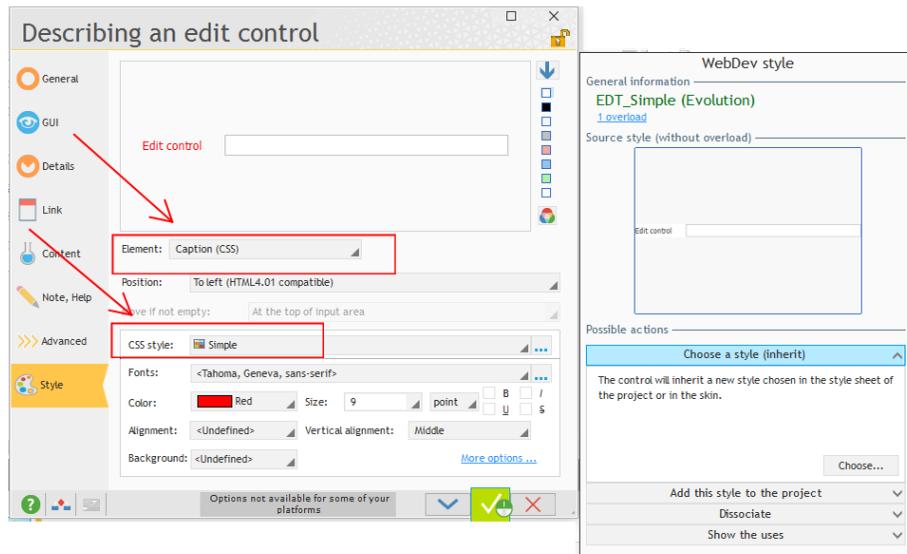
4. Save the page: click  among the quick access buttons (or press CTRL S).

Modifying the style of "Login" and "Password" controls

To display the caption of "Login" and "Password" controls, we are going to modify the default style associated with these controls.

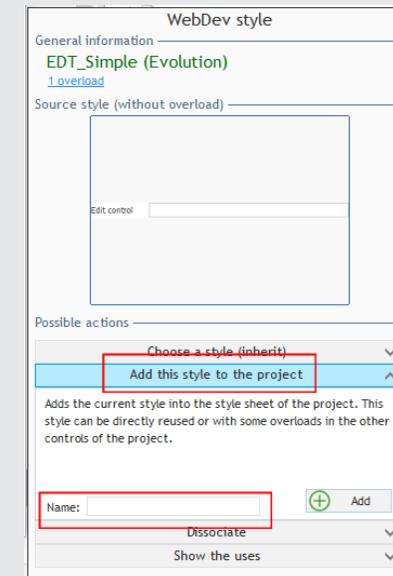
► To modify the default style:

1. Select the "Login" control.
2. Display the description window of control (double-click the control for example).
3. In the "Style" tab, select the "Caption (CSS)" element.
4. Expand the "Color" combo box and select the red color.



This style can be added to the style sheet of project in order to be re-used. You must:

1. Click the "Add this style to the project" button in the "WEBDEV style" pane displayed on the right of description window.
2. Give a name to the style.
3. Click the "Add" button.



To re-use the created style and to associate it with another control:

1. Select the control whose style must be modified.
2. Display the popup menu of control (right mouse click) and select "Choose the WEBDEV style".
3. In the window that is displayed, select the requested style and validate.

Note: Choosing an existing style will be performed in lesson 5 (page 188 - 189).



Notes

5. Validate the description window of "Login" control.

► To apply the same style to the "Password" control:

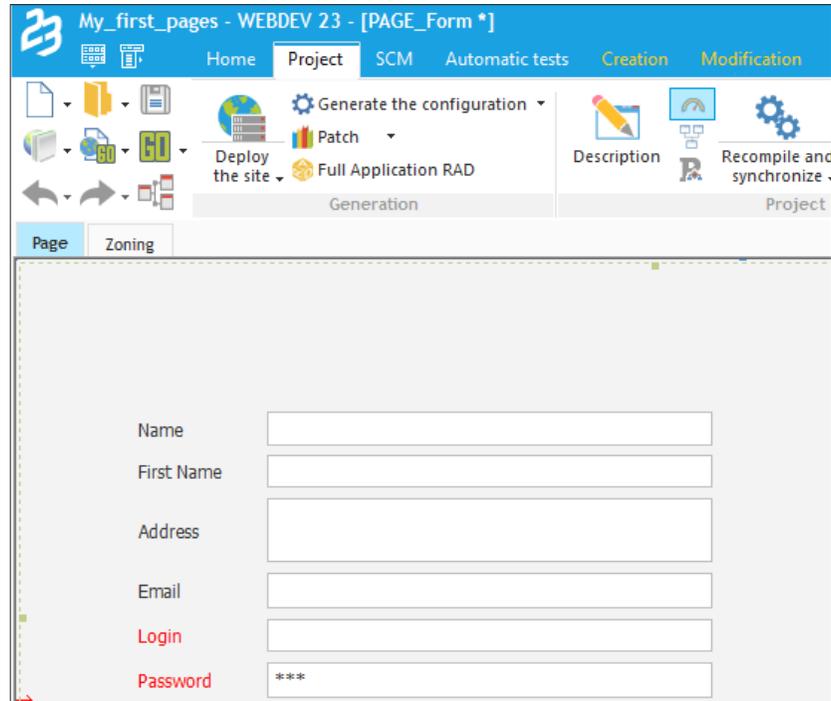
1. Select the "Password" control.
2. Press the [F4] key.



Notes

The [F4] key is used to apply the last modifications performed to the selected controls. In our case, the change of color is automatically applied to the "Password" control.

3. You get the following interface:



Page test

- ▶ Run the test of this page (click the icon  among the quick access buttons). You can enter data but this data is neither processed nor stored.
- ▶ Close the browser.

Saving the data in a data file

When creating a form, the first thing that we want to do is save the data typed. This data can be saved in a text file, in an XML file or even in a database.

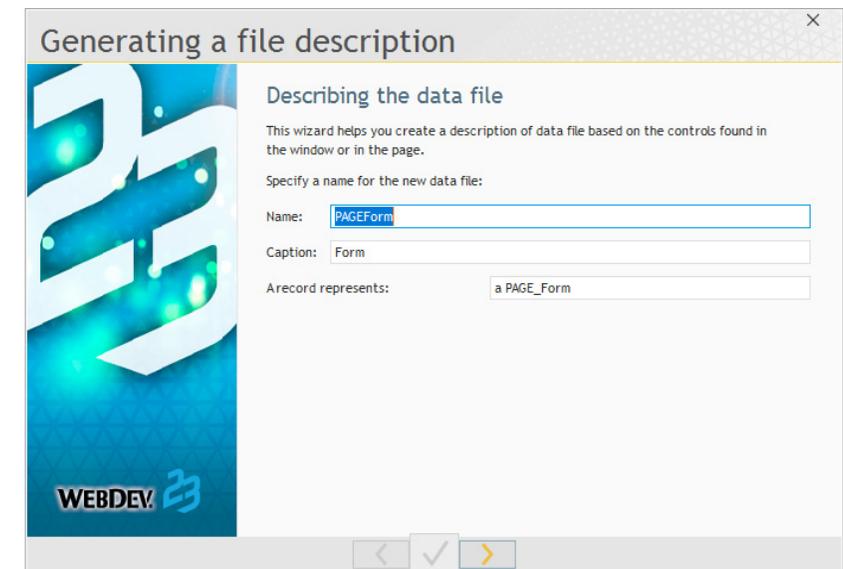
We have chosen the last option.

In WEBDEV, the database description is performed in a specific editor, the data model editor. This editor will be presented in the next section.

In this example, we are going to define the database linked to the created controls via a straightforward feature: the reverse engineering.

Creating the data file

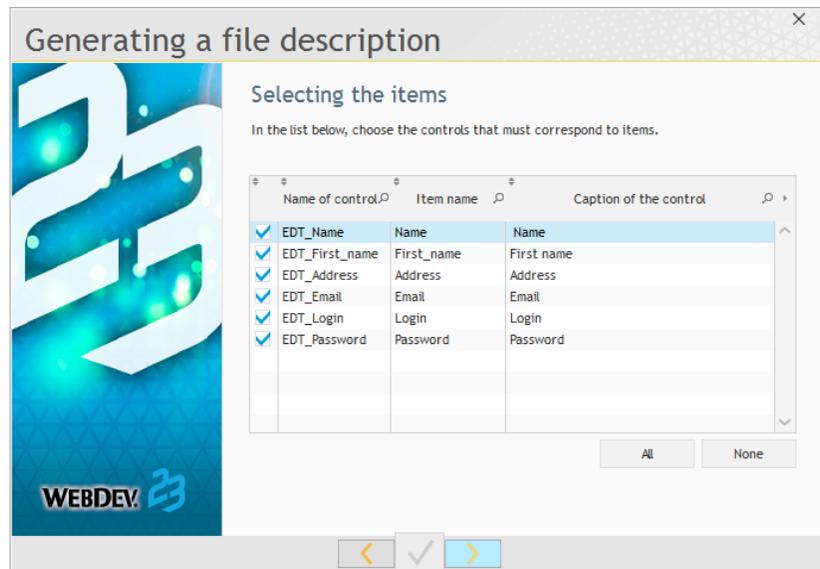
- ▶ To automatically create the data file:
 1. In the ribbon, on the "Page" pane, in the "Edit" group, expand "Other actions" and select "Generate a file description".
 2. A new editor appears: the data model editor. The data model editor contains the description of all data files that will be used by the project.
 3. The wizard for generating a data file starts.



4. The name of the data file is automatically filled with the name of the page. We are now going to modify some elements:

- Enter "Register" in the name.
- Enter "Registration" in the caption.
- Enter "a registration" in the last field.

- Go to the next step (yellow arrow at the bottom of wizard).
- Choose the controls to retrieve. In our case, all the page controls correspond to an item.

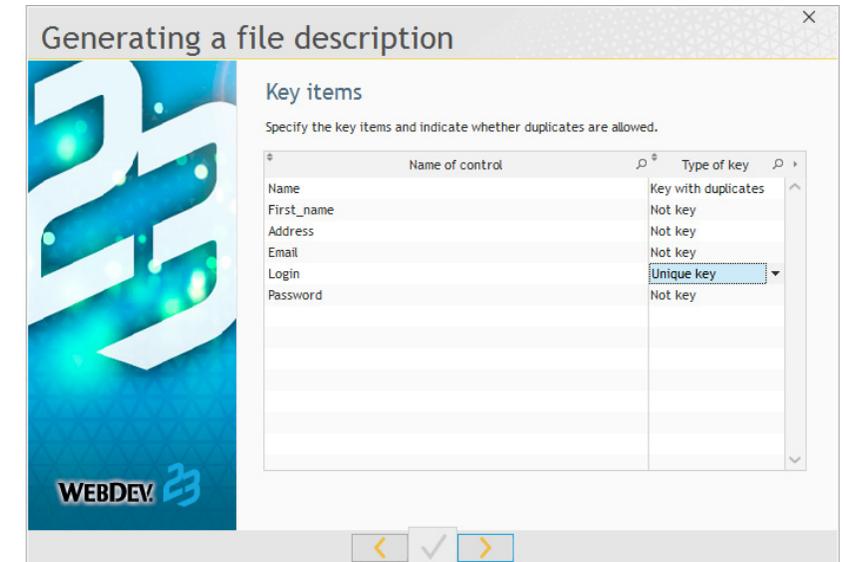


- Go to the next step.
- This step is used to define the keys (also called indexes).



Notes A key provides faster access when performing searches (key with duplicates) and/or it is used to make sure that the value of an item can be saved once only (unique key).

9. In this example, the "Name" item will be a key with duplicates and the "Login" item will be a unique key. The file will be able to contain two people with the same name but not with the same login.



- Go to the next step.
- Validate the summary. The file description appears in the data model editor.
- Save the analysis: click among the quick access buttons (or press CTRL S).
- Close the data model editor.
- A window is displayed, proposing to synchronize the project. It is used to check the differences between the created pages and the data defined in the analysis. This step is performed for each analysis modification, when going back to the project pages.
- Click "Yes".

Let's not spend more time on the data model editor. It will be presented in an another lesson.

Saving data

- Let's go back to the registration form in order to add the data typed by the Web user into the data file. The document bar is found at the bottom of editor. In this bar, a button is displayed for each document opened in the editor.



- "P" represents the project.
- "PAGE_Form" represents the "Form" page.

- ▶ A button is required to save the data from our form. This button will be used to validate the information typed in the page and to save data in the Registration file.
- ▶ To create a Button control:
 1. On the "Creation" pane, in the "Usual controls" group, click  .
 2. The control currently created follows the move of the mouse.
 3. Click inside the page to create the button (below the edit controls for example). The control is created.
 4. Press the space bar: the caption becomes editable.
 5. Type "Save".
 6. Press the [ENTER] key to validate.
- ▶ To type the code associated with the button:
 1. Display the popup menu of control and select "Code" (or press the [F2] key).
 2. The processes associated with the button are displayed. The yellow codes correspond to processes performed on the server and the green codes correspond to processes performed on the browser.
 3. We want to save data in the database. The access to the database can be performed in Server code only: therefore, the code must be typed in the Server click code (yellow).
 4. Type the following code:

```
PageToFile ()
HAdd(Registration)
Info("Record added.")
```

5. Let's study this code:
 - **PageToFile** is used to retrieve the content of edit controls found in the page in order to transfer it to the data file.
 - **HAdd** is used to write the data transferred into the data file.
 - **Info** is used to display a message.
6. Save (CTRL S) and close the code editor (cross at the top right).

Page test

- ▶ Run the test of this page:
 1. Click  among the quick access buttons.
 2. Type the following data:
 - LastName: Doe
 - FirstName: Paul
 - Address: Flower Rd, 75000 Paris
 - Email: paul.doe@atandt.com
 - Login: polo
 - Password: polo
 3. Click the "Save" button. The browser displays a message indicating that the record is added!
 4. Close the browser.

Viewing the data typed

WEBDEV proposes a tool used to view the content of data files while developing the site (when the viewing pages have not been created yet for example).

This tool is named WDMAP. It will be used to check whether the specified information was saved.

- ▶ To start WDMAP:
 1. In the ribbon, on the "Tools" pane, in the "Database" group, click "WDMAP".
 2. Select the "Registration" file. The file content is displayed.

WDMAP - Registration.fic					
File	Edit	Display	Tools	Parameters	?
Analysis Name	C:\My Sites\My_first_pages\My_First_Pages.wdd				
File name	Registration				
File path	C:\My Sites\My_first_pages\EXE\				
Rec #.	RegistrationID	FirstName	Name	Password	
1	1	John	Doe	john	

3. We can see the data we've entered.
4. Close WDMAP ("Close" button).

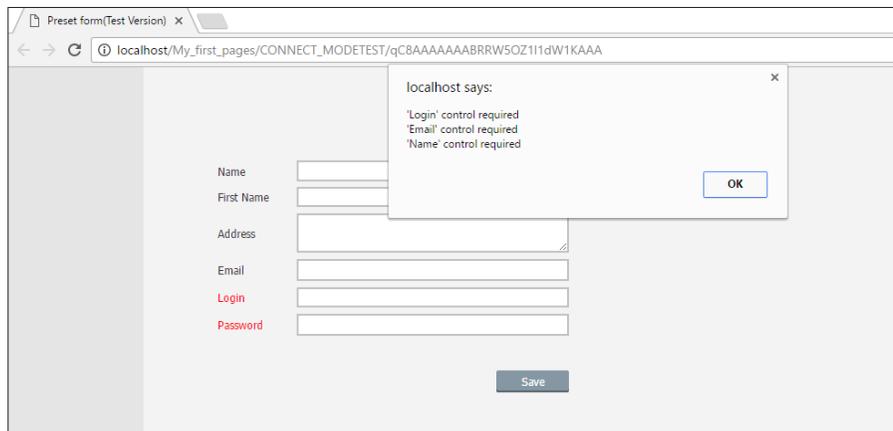
Adding input checks

We are now going to improve our form by adding input checks. We are going to:

- make mandatory the input of the name, email and login.
- have the password be entered twice to check it.

Mandatory input

- ▶ Go back (if necessary) to the "PAGE_Form" page by clicking its name in the bar of opened documents.
- ▶ To force the input in the "Name" control:
 1. Double-click the "Name" control: the description window of control is displayed.
 2. Display the "Details" pane.
 3. Check "Mandatory input". If this option is checked, WEBDEV will automatically check whether this edit control was filled.
 4. Validate the description window.
- ▶ To apply this modification to the "Email" control:
 1. Select the "Email" control (mouse click).
 2. Press the [F4] key: the last action performed on a control is re-run on the selected control.
- ▶ Similarly, apply this modification to the "Login" control.
- ▶ Run the test of this page:
 1. Click  among the quick access buttons.
 2. In the window that is displayed, click the "Save" button.
 3. An information box is automatically displayed to indicate that mandatory controls have not been filled.



4. Validate the information box.
5. Close the browser.

Checking the password

To check the password, we are going to create the edit control allowing the user to type his password twice. Then, we'll enter the code needed for the checking.

- ▶ Create a new edit control. This control caption is "Check" and is a "Password" type.
- ▶ The code for password check must be typed in the processes associated with the "Save" button.
 1. Click the "Save" button and press the [F2] key.
 2. This check consists in comparing the value typed in the "Password" control and the one typed in the "Check" control. The server is not required to perform this check: it can be performed locally in the browser.
 3. Therefore, we are going to type the following code in the browser "Click" process:

```
IF EDT_Password<>EDT_Check THEN
  Info("Error, the passwords are different.")
  EDT_Password = ""
  EDT_Check = ""
  ReturnToCapture (EDT_Password)
END
```

4. Let's study this code:
 - The **IF** statement is used to perform an action on a condition. In our case, the action is performed if the "Password" and "Check" controls are different ('<>' operator).
 - In case of difference, the edit controls are "cleared": an empty string is assigned to them.
 - **ReturnToCapture** is used to position the cursor in the specified control (the "Password" control here) without running the code that follows. In our case, if the passwords typed are different, the "Password" control takes focus and the process is stopped: the server code that is used to save the information in the database will not be run.

- ▶ Run the test of this page:

1. Click  among the quick access buttons.
2. Type the following information:
 - LastName: "Doe"
 - FirstName: "Peter"
 - Email: "Peter.Doe@gmail.com"
 - Login: "Peter"
 - Password: "Peter"
 - Check: "Louis"
3. Click the "Save" button.
4. An error message is automatically displayed to indicate that the passwords are different.
5. Validate this message.
6. Close the browser.

Simple search in a database

We have seen how to create an input form and how to save values in a database. Let's continue our first WEBDEV discovery by performing a search in the database.

We are going to create a login page where the login and the password will be typed. Then, we will check whether the information typed is correct.

Creating the page

- ▶ To create the page:
 1. Click  among the quick access buttons.
 2. The window for creating a new element is displayed: click "Page" then "Page". The wizard for page creation starts.
 3. Select "Blank page" then "Blank - Simple layout" and validate.
 4. The page is created and the backup window is automatically displayed.
 5. Type the page title: "Login".
 6. Validate.

Creating controls

The login page will contain the following controls:

- Two edit controls:
 - "Login" to type the login.
 - "Password" to type the password.
- Two buttons:
 - "Connect" to check the login and password
 - "Register" to open the registration page (that was already created).

These types of controls have already been created, let's see a reminder of the operations to perform:

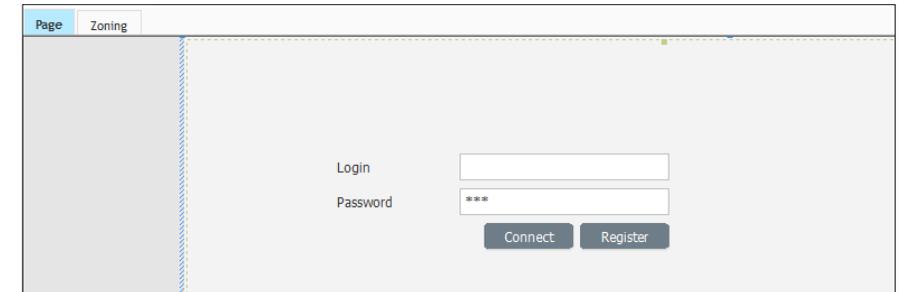
- ▶ To create an edit control:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click inside the blank page to create the control.
 3. Click the control (simple click): the caption becomes editable.
 4. Type the caption ("Login" and "Password") and press the [ENTER] key to validate. The caption is immediately modified.



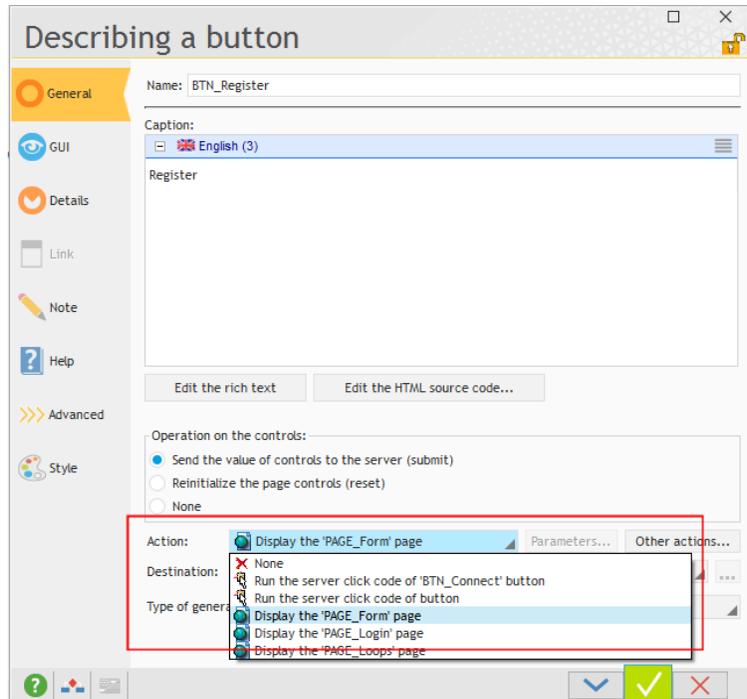
Notes

The type of "Password" control must be "Password". To modify the type of the edit control, display the description window of control.

- ▶ To create a Button control:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click inside the page to create the button (below the edit controls for example). The control is created.
 3. Press the space bar: the caption becomes editable.
 4. Type the caption ("Connect" and "Register") and press the [ENTER] key to validate. The caption is immediately modified.
- ▶ You get the following page:



- ▶ Now let's see the operating mode of buttons. First let's take a look at the "Register" button.
 1. Select the "Register" button.
 2. Bring up the description window of the button (double-click the control).
 3. In the "General" tab, in the "Action" list, we are going to specify the action to perform. In our case, we are going to open the registration page: select "Display the 'PAGE_Form' page".



4. Validate the description window.



Notes

No code is required: in the editor, the button is associated with the action to perform (opening a page in this case).

► For the "Connect" button, we must type the WLanguage code required to perform the search:

1. Select the "Connect" button and display the associated processes ([F2] key).
2. Type the following code in the "Click (server)" process:

```
HReadSeekFirst(Registration,Login,EDT_Login)
IF NOT HFound(Registration) THEN
    Error("Login not found.")
ELSE
    IF EDT_Password = Registration.Password THEN
        Info("You are connected.")
    ELSE
        Error("Wrong password.")
    END
END
END
```

3. Let's study this code:

- This code is run as server code because we are performing a search in the database found on the server.
- **HReadSeekFirst** is used to find a value in a data file. The search is performed on the entire name by default. To perform a "Starts with" search, all you have to do is add a parameter: the **hStartWith** constant.
- **HFound** is used to find out whether the search performed in the Registration file was successful or not. The "IF NOT HFound" code is used to define the action to perform if the login is not found ("Unknown login" displayed) as well as the action to perform if the login is found.
- If the login is found, the password typed in the EDT_Password control is compared to the one found in the data file (for the specified login). If the two passwords are the same, then the user is connected, otherwise an error message is displayed.



Notes

The code presented here is for educational purpose. In a real project:

- The password must not be stored in clear. We advise you to encrypt the password.
- We advise you to display the same error message in case of incorrect login or password (to avoid an attempt to hack the logins).

4. Save (CTRL S) and close the code editor (cross at the top right).

Page test

► Run the test of this page:

1. Click  among the quick access buttons.
2. Type the following data:
 - Login: polo
 - Password: polo1
3. Click the "Connect" button. An error message is displayed.
4. Validate the message and modify the password: "polo".
5. Click the "Connect" button. You are connected.
6. Close the browser.

Conclusion

By creating two pages only, we have discovered several WEBDEV features:

- the creation of controls: edit controls and buttons,
- the possibilities for aligning and organizing the controls in the page,
- the addition of records into data files,
- the sequence of pages,
- the search for record in the data files,
- the use of server code and browser code,
- ...

After this overview, we will go into more details in the next section, especially regarding the analysis and the definition of database characteristics before moving on how to build a full site.

DEVELOP 10 TIMES FASTER

PART 2

**My first
database**

LESSON 2.1. OVERVIEW

This lesson will teach you the following concepts ...

- Overview of project developed in this part



Estimated time: 5 mn

Overview of project developed in this part

In this part, we are going to create a project associated with a HFSQL database (database supplied with WEBDEV).

You will discover some key elements for developing a site that is using data files:

- Creating a WEBDEV project.
- Describing the data files used by the site.

In another section, we will concentrate on the development of elements (page, reports, ...) found in a site with data that will be developed from A to Z, from the interface creation (GUI) to the final distribution. You will see the main points for developing a site.

In this section, we are going to create a database used to manage orders.

The same type of database will be used in part 3 of this tutorial to develop a full site. The database used is HFSQL Classic, the free database supplied with WEBDEV. The HFSQL Client/Server database will be presented later in this tutorial.

LESSON 2.2. WEBDEV AND THE DATABASES

This lesson will teach you the following concepts ...

- Vocabulary used.
- The different modes for accessing the databases.



Estimated time : 5 mn

Overview

You may have to handle data when creating an application. To store the data, you must create a "database".

In WEBDEV, when creating a project that is using data, you must first create an "analysis".

An "analysis" contains the description of data files (or tables). The application data will be stored in the data files.

When running the application, these descriptions will be used to create the database and/or the data files. The information will be stored in this database or in these data files.



Notes

Several tools for performing maintenance operations on the HFSQL databases are supplied with WEBDEV. They can be accessed from the HFSQL Control Center.

WEBDEV supports most of the database formats (nearly all of them). The most common formats are:

- HFSQL, the database system supplied in standard with WEBDEV. The HFSQL database is available in Classic or Client/Server mode.
- AS/400, Access, Sybase, Informix, ...
- Oracle, SQL Server, MySQL, xBase, ...
- Any database accessible in SQL language in Windows.
- Text (ASCII files).

Several methods (also called "access modes") can be used to access the data:

- Native Connector (also called Native Access),
- OLE DB access,
- Direct ODBC access,
- ODBC access via OLE DB.

HFSQL

HFSQL is a database that is very powerful, very fast and very reliable.

HFSQL operates in Windows and Linux, on mobile devices (iOS, Android, Windows), on networks of any size and type, and it automatically manages hundreds of concurrent accesses.

HFSQL is available in Classic version and in Client/Server version.

HFSQL can be freely distributed with your WEBDEV sites.

HFSQL proposes all the database features, especially:

- the log process,
- the transactions,
- the replication,
- the triggers,
- the stored procedures,
- the clusters, ...

See the online help for more details about implementing these features.

In the different sections of this tutorial, we will be using a HFSQL Classic database then a HFSQL Client/Server database.

The different modes for accessing the databases

Native Connector (Native Access)

A Native Connector is using a database format directly and exclusively. This optimized type of access is developed specifically for each database format.

In WEBDEV, a Native Connector is available for the following databases:

- HFSQL Classic or Client/Server (standard)
- xBase (standard)
- Access (standard)
- XML (standard)
- SQLite (standard)
- Oracle (optional)
- AS/400 (optional)
- SQL Server (optional)
- Sybase (optional)
- Informix (optional)
- DB2 (optional)
- Progress (optional)
- MySQL (optional and free)
- PostgreSQL (optional and free)
- MariaDB (optional and free)

For any question regarding the Native Connectors, contact our sales department!

The WLanguage **SQL*** and **HRead*** functions can be used with this type of access. Therefore, the code is portable and independent of the database.

Direct ODBC access

An access via ODBC is using a multi-database access standard. The ODBC layer must be installed on your computer. In most cases, this layer is already installed in the recent Windows versions. This can be checked in the control panel of Windows by selecting "ODBC data source" (or "ODBC administrator" according to the Windows version used).

Caution: some databases may not be accessible via this method. If you want to use this type of access, check whether an ODBC driver exists and install this driver if necessary.

Only the WLanguage **SQL*** functions can be used with this type of access.

OLE DB access

An access via OLE DB is an access that is using a multi-database access standard. This type of access is based on MDAC (Microsoft Data Access Component).



Caution!

If you are using an OLE DB access, MDAC must necessarily be installed on the user computers (version 2.6 or later).

Some databases may not be accessible via this method. Check whether an OLE DB driver exists before you use this type of access.

The WLanguage **SQL*** and **HRead*** functions can be used with this type of access.

ODBC access via OLE DB

In summary, it is a "mix" of OLE DB and ODBC. This is the "heaviest" method and the least efficient one in term of performance. It should not be used on small databases.

The WLanguage **SQL*** and **HRead*** functions can be used with this type of access.

LESSON 2.3. PROJECT AND ANALYSIS

This lesson will teach you the following concepts ...

- Creating a project
- Creating an analysis



Estimated time: 40 mn

Overview

To create a site with a database, you must:

- Create the project linked to the site. This project will group all the site elements (pages, codes, queries, reports, ...).
- Create the analysis linked to the project. The analysis is used to describe all the data files handled by the site.

Creating the project

► To create the project:

1. Start WEBDEV (if not already done). Close the current project if necessary.
2. In the home window, click the "Create a project" button and select "Site". The wizard for project creation starts. The different wizard steps help you create your project. The information specified in this wizard can be modified later.

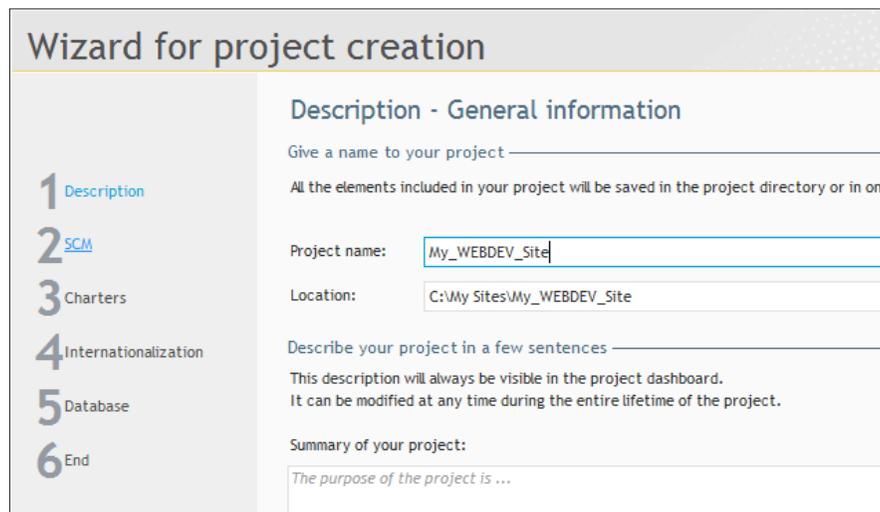


Notes

Tip: To create a project, you can also:

1. Click among the quick access buttons of WEBDEV menu.
2. The window for creating a new element is displayed: click "Project".

3. First of all, the wizard allows you to type the project name, its location and its description. In our case, this project will be named "My_WEBDEV_Site".



4. By default, WEBDEV proposes to create this project in the "\\My Sites\My_WEBDEV_Site" directory. You can keep this location or modify it via the [...] button.



Notes

When you develop for the Web, avoid using accented characters in the names of elements (projects, pages, ...).

For the project summary, type "The purpose of the project is to manage the products".

5. The different wizard sections are indicated in the left margin. You can easily go from a section to another one by clicking on it. The other steps of the "Description" part being not fundamental, click "Charters" directly.

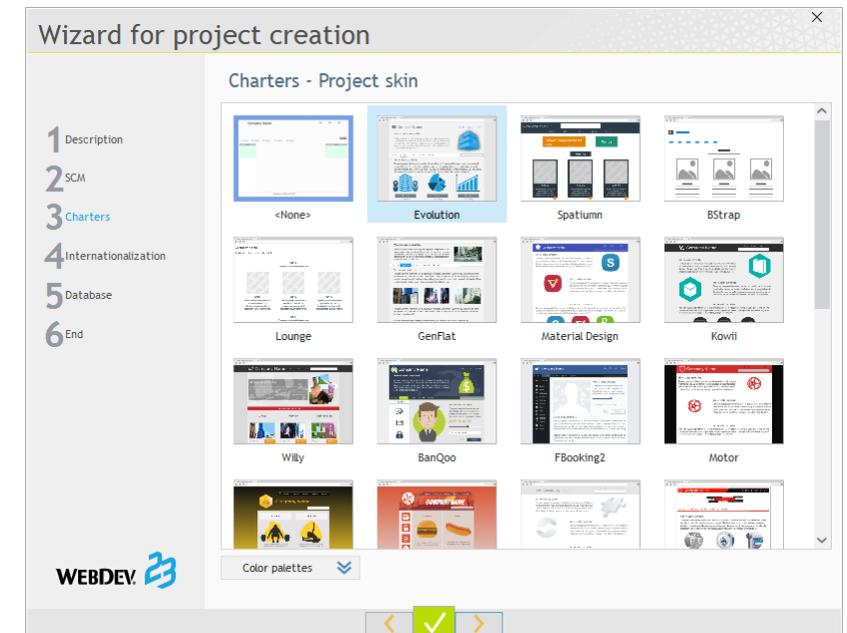
6. This section is used to define the different charters associated with the project. For the programming charter, don't modify the suggested options. Go to the next step via the "Next" arrow found at the bottom.

7. This step is used to define the style book (also called "Skin"). The style book is used to define the site style.



Notes

The skins are used to standardize the visual aspect of a site and they allow you to easily change style. The skin provides the images, the fonts, the texture, the shape of buttons and the styles available for the project.



Select "Evolution".

8. We are now going to specify the information regarding the database. Click "Database" directly.

9. Select "Yes, create a new database" and validate. The wizard for analysis creation starts.



Notes

To better understand the lessons found in this section and to optimize your training, we advise you to create the "My_WEBDEV_Site" project.

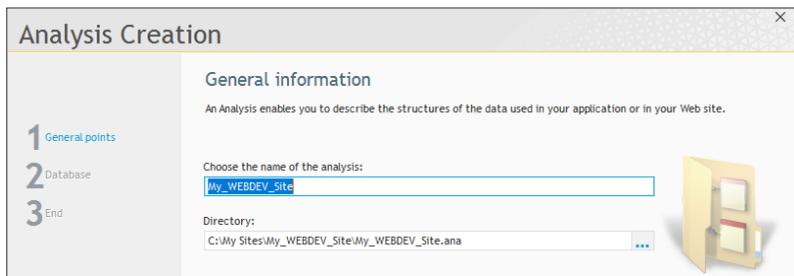
A corrected example can be access at any time to check the validity of operations performed.

To open this corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My WEBDEV site (Answer)".

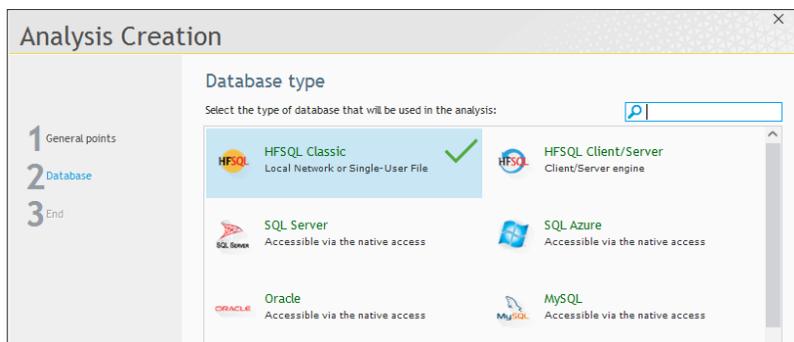
Creating the analysis

► The wizard steps for creating the analysis are as follows:

1. Specify the analysis name and directory. By default, the analysis name corresponds to the project name and the analysis directory is a ".ana" directory in the project directory. We will keep these default parameters. Go to the next wizard step.



2. You now have the ability to choose the types of databases used by the project. Select HFSQL Classic (the database proposed by default with WEBDEV).



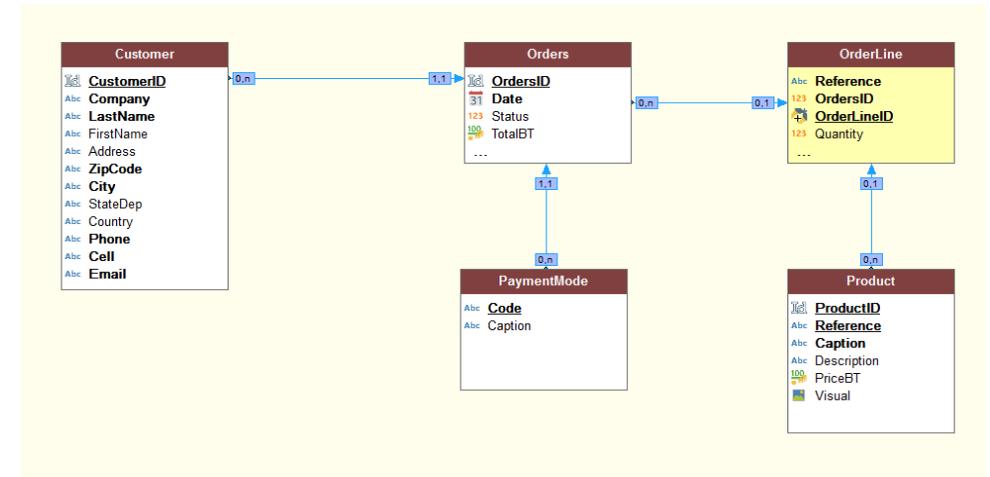
Go to the next wizard step.

3. Validate. The wizard for creating a data file is automatically started.

Creating the description of data files

Our application for managing customers and orders will be associated with the following analysis. This analysis includes five different data files (tables):

- Customer,
- Orders,
- PaymentMode,
- OrderLine,
- Product.

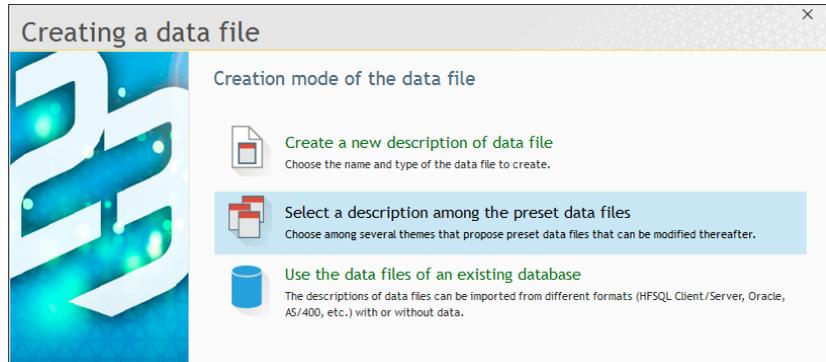


To create the data files of this analysis, we will be using the different methods proposed by WEBDEV.

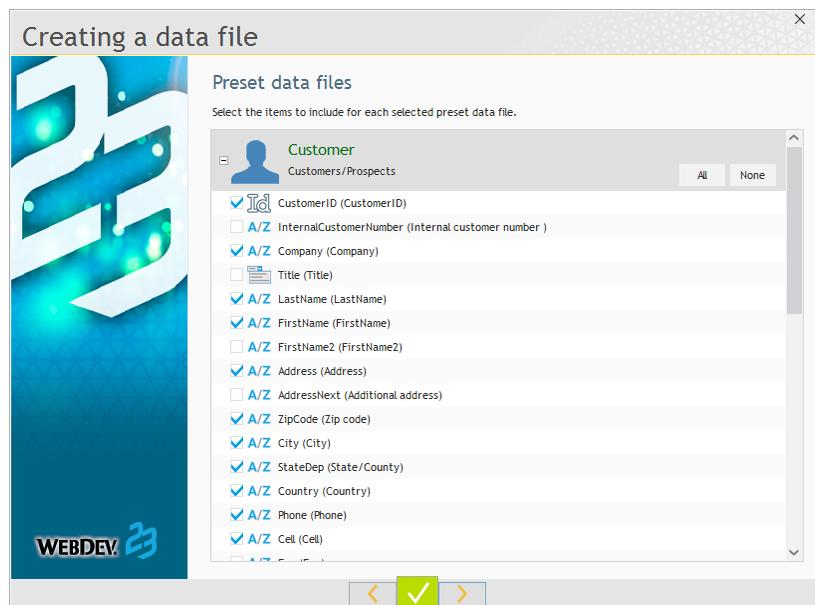
Creating a data file: using a preset file

► The wizard steps for creating a data file are as follows:

1. In the wizard, select "Select a description among the preset data files". Go to the next wizard step (arrow at the bottom of wizard).

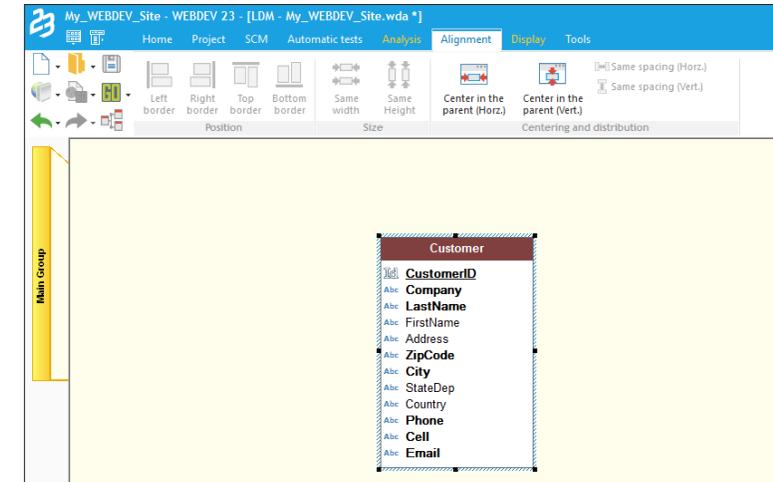


2. The list of preset data files is displayed. We are going to create the "Customer" file. In the list of data files, select "Customer". Go to the next step.
3. The wizard proposes the list of items that will be included in Customer file. This list is very impressive because it allows you to manage several types of Customer files.



4. Click "None" to deselect all the items. Then, check the following items: CustomerID, Company, LastName, FirstName, Address, ZipCode, City, StateDep, Country, Phone, Cell, Email.
5. Go to the next wizard step.

6. Validate the wizard. The "Customer" file is automatically created in the data model editor.



7. The window for creating a new element is displayed. We are now going to create the data file containing the orders.

Creating a data file: creating the file and items

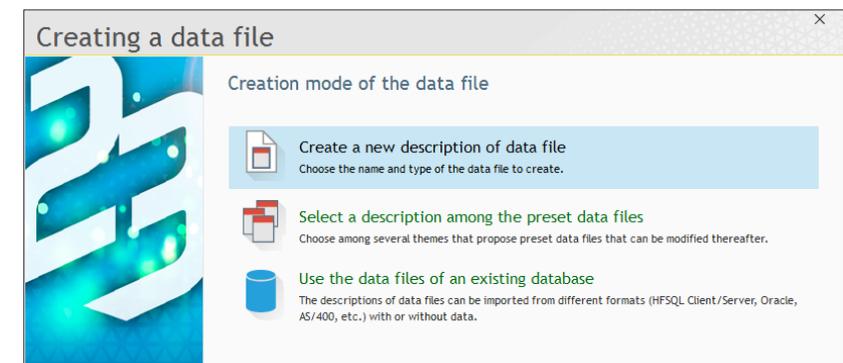
- To create a data file from the window for creating a new element:
1. Click "Data" then "Data file".



Notes

You also have the ability to create a data file from the data model editor: on the "Analysis" pane, in the "Creation" group, click "New file".

2. The wizard for creating a new data file starts.
3. In the wizard, select "Create a new description of data file". Go to the next wizard step.



4. We are going to create the "Orders" file. Type its name ("Orders") in the wizard. This name will be used:
- to handle the data file by programming. The variable associated with the file will be "Orders".
 - to build the name of associated physical data file ("Orders.fic" file).

The caption and description of elements represented by the file records are automatically displayed.



Notes

In the wizard, the control "A record represents" is used to get an understandable caption during the description of links between data files. A caption is automatically proposed from the file name. In our case, type "An order".

General parameters

Name of Data File

Name:

Caption:

A record represents:

Automatic identifier

Automatic identifier on 8 bytes (mandatory for the server replication)

Automatic identifier on 4 bytes

None

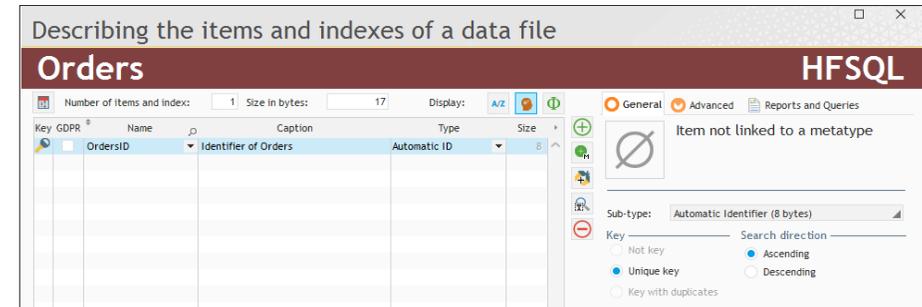
5. In the "Automatic identifier" area, keep "Automatic identifier on 8 bytes". If an automatic identifier is defined on the data file, it means that the data file includes a unique key, automatically managed by WEBDEV.



Notes

To create the identifier (an identifier is a unique key), you have the ability to create a numeric item whose type is "Automatic identifier". This identifier is automatically managed by WEBDEV. Whenever a record is added into the data file, WEBDEV automatically assigns a value to the file identifier. This value is unique.

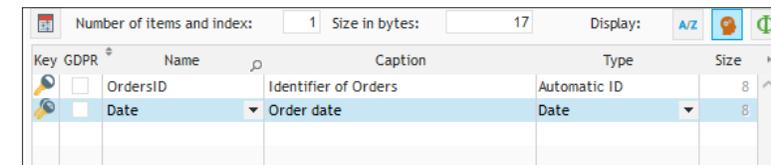
6. Go to the next step and select the type of database associated with the data file. We are going to work on HFSQL Classic data files. Go to the next step.
7. Click the green button to validate. The data file is automatically created in the analysis. The description window of items is opened.



We are going to create the items of "Orders" file. In the description window of data file, you will notice that an item was automatically created: "OrdersID". This item corresponds to the automatic file identifier. This item includes the letters "ID" and the file name.

We are going to create the other items of this data file.

- First, we are going to create the "Date" item. This item will contain the order date.
1. In the description window of items, double-click in the "Name" column of first empty row. This column automatically becomes editable. Type "Date".
 2. Click the "Caption" column. The item name is automatically displayed. We are going to modify the item caption by clicking on it: type "Order date". In the "Type" column, the "Text" type is automatically selected. Expand the list and select the "Date" type.



3. This item will be a key item (index) in our data file: the keys are used to accelerate the accesses to data and the sorts.
- For a database in SQL format, the engine is using the best indexes.
 - For a sequential browse of a data file, all you have to do is specify the browse index (which means the key).

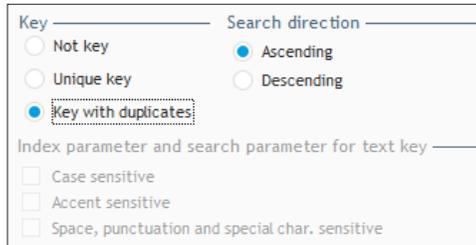


Notes

The concept of key is part of the item characteristics. When creating an item, you have the ability to specify whether it is:

- not key,
- unique key: the value of this key will be unique in the entire data file (which means in all the file records)
- key with duplicates: the value of this key can be found several times in the data file.

4. The key definition is performed as follows: re-select the line of "Date" item to enable the description controls found on the right of screen. Then, all you have to do is specify the type of key used. In our case, the date is a key with duplicates.



5. You must also define the search direction of the key. The search direction is used to define the default sort for this item. In our case, when a read operation is performed based on this key, the default sort order will be "ascending".

► We are now going to create the "Status" item that is used to find out the status of the order.

1. Position on a new table row. Type:

- the name: Status
- the caption: Order status
- the type: Radio button, List box, Combo box. In the window that is opened, you have the ability to select the type of control created by default for this item. It will be a radio button in this case. Validate the window.

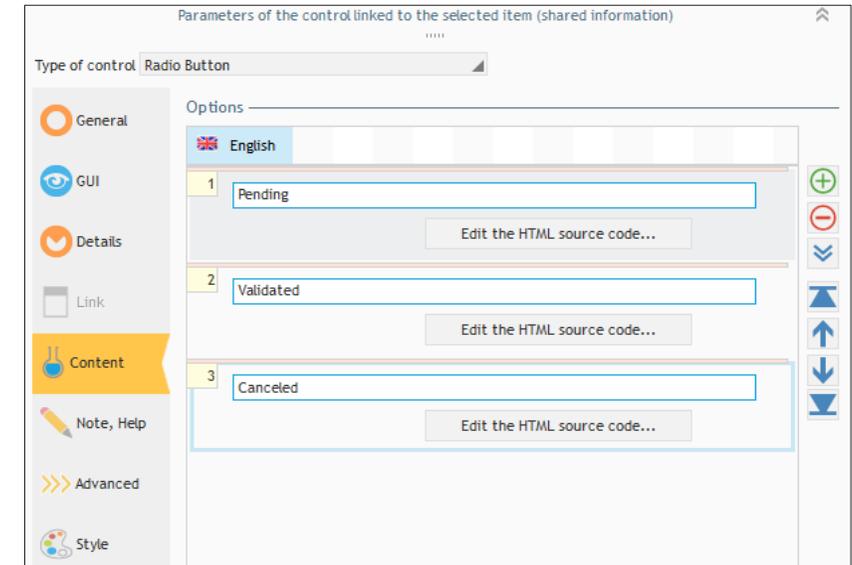
2. In the lower section of the screen, click the double arrow to display the parameters of the control linked to the selected item.



The information typed here will be automatically used when creating pages linked to the data file. You will find here the control type and the caption.

We are going to enter the different options corresponding to the order status in the "Content" tab:

- Click the "Content" tab.
- The option 1 corresponds to "Pending". Type "Pending" in the edit control found on the right of number 1.
- Type "Validated" in the edit control found on the right of number 2.
- Click the "+" button to add a new option in the radio button.
- Type "Canceled" instead of "Option 3".



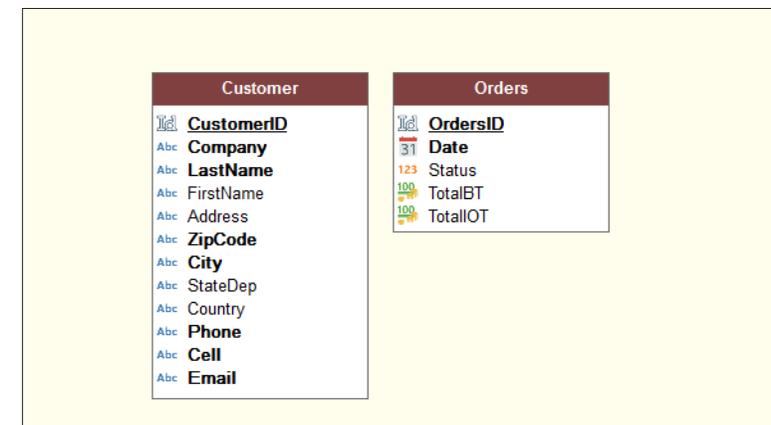
3. Click the double arrow again.

4. Similarly:

- Position on a new table row and create the "TotalBT" item. This item is a "Currency" item.
- Position on a new table row and create the "TotalIOT" item. This item is a "Currency" item.

5. That's it, the description of the "Orders" file is over. Validate the description window of items.

6. The "Orders" file appears in the data model editor.



Importing a CSV file

Another method will be used to create the "PaymentMode" file that contains the characteristics of payment: importing a CSV file.



Notes A CSV file is a text file that is using a specific format. This file contains data on each line. The data is separated by a separation character (usually a comma, a semi-colon or a tab).

From the CSV file containing the data, WEBDEV will create:

- the description of data file in the analysis,
- the HFSQL data file with the data found in the CSV file.

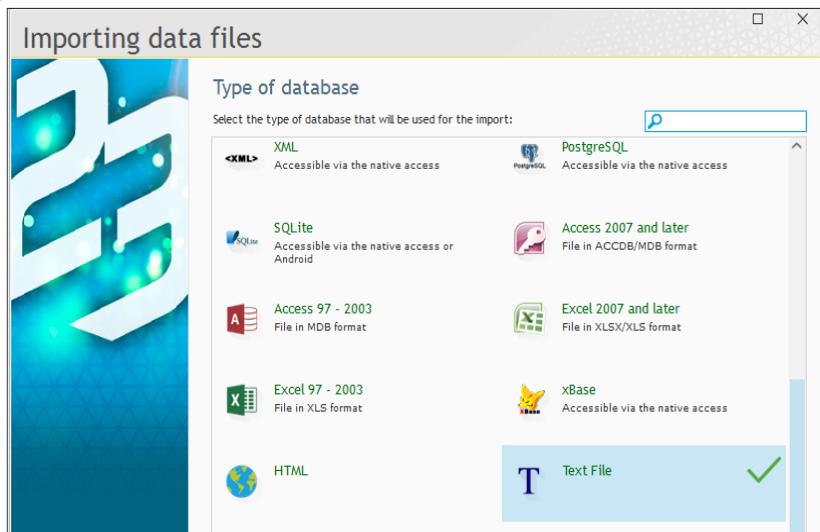
► To import a CSV file into the analysis:

1. On the "Analysis" pane, in the "Creation" group, expand "Import" and select "Import the descriptions of files/tables".

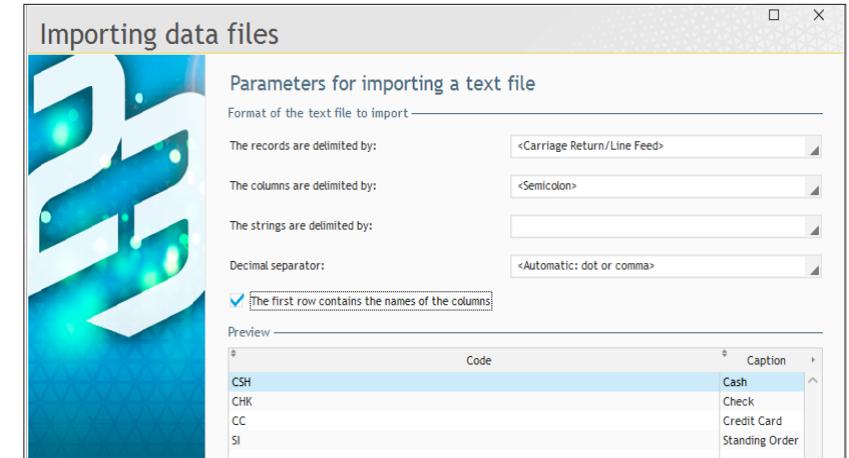


Tip To import a CSV file (or any other file) into the analysis, you also have the ability to Drag and Drop the CSV file (from the Windows explorer) to the data model editor. This will be presented in the next paragraph.

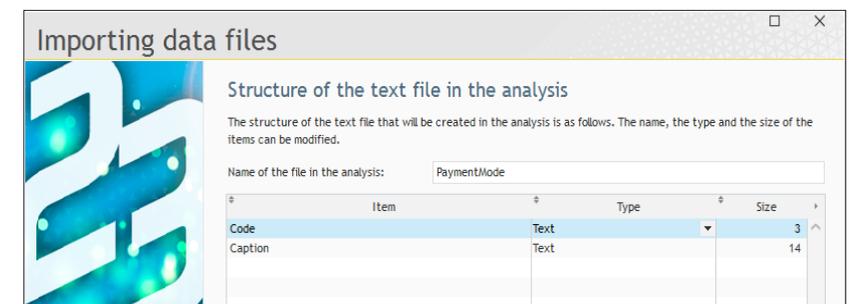
2. The wizard for importing a file starts.
3. Go to the next step.
4. Select the format of the file to import. In this case, select "Text file". Go to the next wizard step.



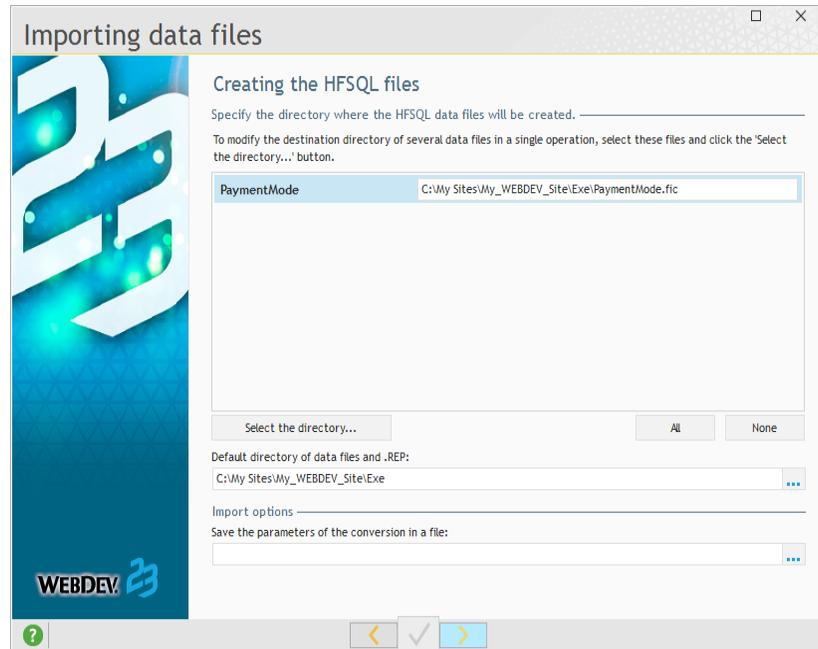
5. Specify the path of file to import: "\\Tutorial\Exercises\My_WEBDEV_Site\PaymentMode.csv" found in the setup directory of WEBDEV.
6. Go to the next wizard step.
7. Specify the following import parameters:
 - Records delimited by: "<Carriage return/Line feed>"
 - Columns delimited by: "<Semicolon>"
 - Strings delimited by: "<None>"
 - Decimal separator: "<Automatic: dot or comma>"



8. Don't forget to check "The first row contains the names of the columns".
9. Go to the next step.
10. The structure of data file that will be created is displayed. Keep the default options. Go to the next step.



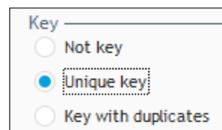
11. The content of CSV file will be automatically converted to HFSQL format. The wizard proposes to create the HFSQL file in the project directory. Keep the proposed options and go to the next step.



12. Validate the wizard. WEBDEV creates the data file.

► Let's study the description of imported data file:

1. Select the "PaymentMode" file and select "Description of data file" from the popup menu.
2. In the window that is displayed, modify the file caption: delete "(Imported)".
3. Click to display the description of items found in the data file.
4. This data file contains no automatic identifier and no unique key. We are going to switch the "Code" item to unique key:
 - Position the selection bar on the "Code" item if necessary.
 - In the right section of screen, click "Unique key".



5. We get the following data:

PaymentMode					
Number of items and index:		2	Size in bytes:		28
			Display:		A/Z
Key	GDPR	Name	Caption	Type	Size
<input type="checkbox"/>		Code	Code	Text	3
<input type="checkbox"/>		Caption	Caption	Text	14

6. Validate the description window of items then the description window of the file.

Direct import of existing data files

The last method for creating data files consists in importing the existing HFSQL data files. This method will be used to create the "Product" file.

► To import HFSQL data files:

1. In the Windows file explorer, open the following WEBDEV sub-directory: "\\Tutorial\Exercises\My_WEBDEV_Site".
2. Select the "Product.fic" file.
3. Drag and Drop the "Product" file to the data model editor of WEBDEV.
4. The import wizard starts. Validate the different screens. The data file appears in the data model editor.

All the necessary data files are now found in the data model editor.



Caution!

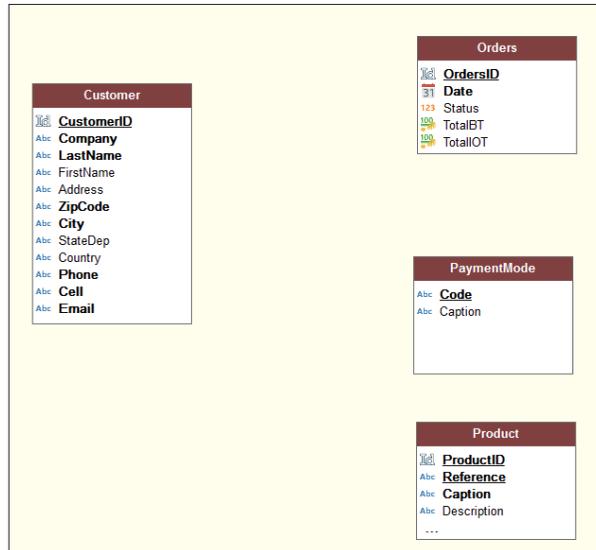
Only the description of the "Product" file was imported into the analysis of our project. The data found in the "Product" file have not been imported into the project.

To handle the data found in the file that was just imported, copy (in the file explorer) the "Product.fic", "Product.mmo" and "Product.ndx" files (found in the "\\Tutorial\Exercises\My_WEBDEV_Site") to the EXE sub-directory of the directory of your project.

Note: To open the file explorer on the directory of your project, on the "Home" pane, in the "General" group, click .

Creating links

All the file descriptions required by the application for product management have been created.



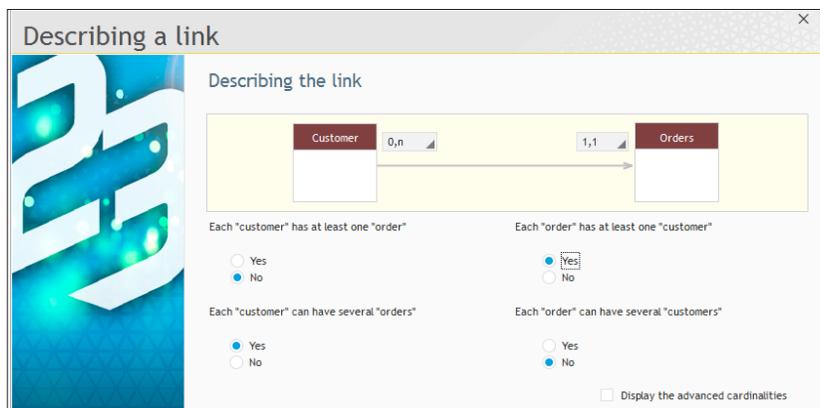
The image shows four data file descriptions:

- Customer:** CustomerID (Key), Company, LastName, FirstName, Address, ZipCode, City, StateDep, Country, Phone, Cell, Email.
- Orders:** OrdersID (Key), Date, Status, TotalET, TotalOT.
- PaymentMode:** Code, Caption.
- Product:** ProductID (Key), Reference, Caption, Description.

We are now going to create the links between the data files. A link is used to define the integrity constraints (cardinalities) between two data files.

► Let's create the link between "Customer" file and "Orders" file: a customer can have one or more orders, and each order is linked to a customer.

1. On the "Analysis" pane (found in the ribbon), in the "Creation" group, click "New link". The mouse cursor turns into a pen.
2. Click the "Customer" file then the "Orders" file.
3. The wizard for link creation starts.
4. Answer the questions asked by the wizard:

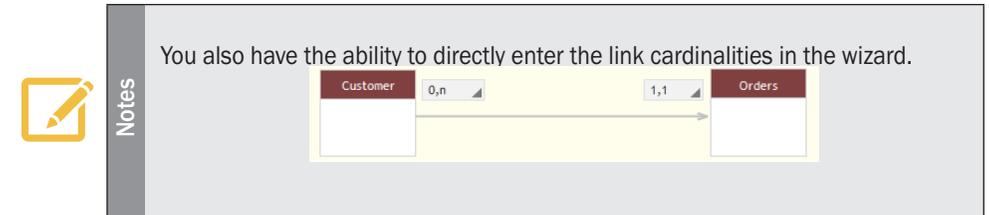


The screenshot shows the 'Describing a link' wizard. It displays a diagram of the link between 'Customer' (0,n) and 'Orders' (1,1). Below the diagram are four questions with radio button options:

- Each "customer" has at least one "order": Yes, No
- Each "order" has at least one "customer": Yes, No
- Each "customer" can have several "orders": Yes, No
- Each "order" can have several "customers": Yes, No

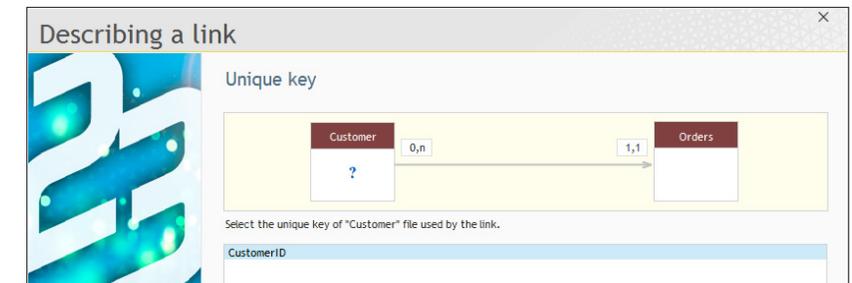
There is a checkbox for 'Display the advanced cardinalities' which is currently unchecked.

- Each Customer has at least one Order: No
- Each Customer can have several Orders: Yes
- Each Order has at least one Customer: Yes
- Each Order can have several Customers: No



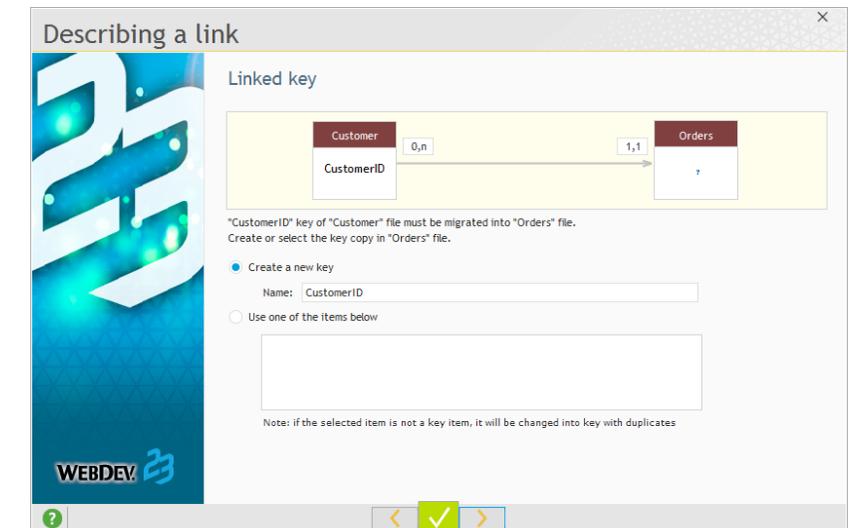
The 'Notes' box contains a diagram showing a link between 'Customer' (0,n) and 'Orders' (1,1). The text reads: "You also have the ability to directly enter the link cardinalities in the wizard."

5. Go to the next step. The wizard automatically proposes the key used by the link (CustomerID).



The screenshot shows the 'Describing a link' wizard at the 'Unique key' step. It displays the link diagram with a question mark in the 'Customer' box. Below the diagram, it says: "Select the unique key of 'Customer' file used by the link." A list box below contains 'CustomerID' selected.

6. Display the next wizard screen. The wizard offers to create the "IDClient" key in the Orders file to store the customer corresponding to the order.



The screenshot shows the 'Describing a link' wizard at the 'Linked key' step. It displays the link diagram with 'CustomerID' in the 'Customer' box and a question mark in the 'Orders' box. Below the diagram, it says: "'CustomerID' key of 'Customer' file must be migrated into 'Orders' file. Create or select the key copy in 'Orders' file." There are two radio button options: 'Create a new key' (selected) and 'Use one of the items below'. Under 'Create a new key', there is a text input field with 'CustomerID' entered. A note at the bottom states: "Note: if the selected item is not a key item, it will be changed into key with duplicates."

7. Accept this option by going to the next screen.

8. This screen is used to define the integrity rules that will be automatically applied. In our case, you have the ability to choose the requested behavior when deleting a customer as well as the requested behavior when modifying the customer identifier.
9. Validate the integrity rules by going to the next screen.
10. Click the green arrow. The link is automatically created in the data model editor.

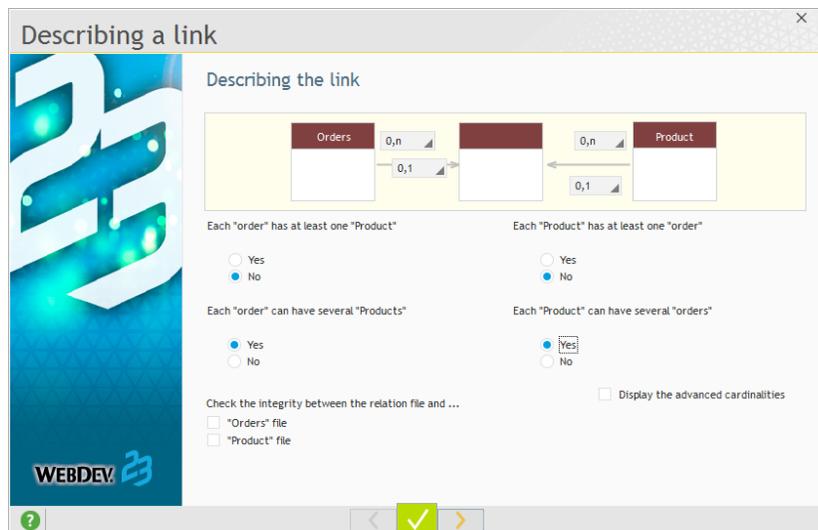
- ▶ Similarly, create a link between the "PaymentMode" file and the "Orders" file. These two files are linked as follows:
 - An order must have a payment mode.
 - A payment mode can be used in several orders.

In the wizard:

- The cardinalities are as follows: PaymentMode (0,n), Orders (1,1)
- The link key corresponds to the "Code" item.

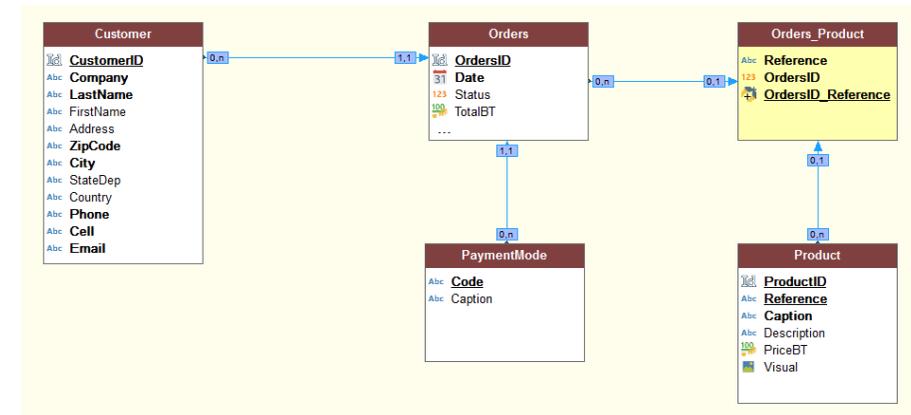
- ▶ We are now going to create a link between the "Orders" file and the "Product" file. This link will be used to create a link file, the file of order lines.

1. Similarly, create the link between files.
2. Answer the questions asked by the wizard:
 - Each Order has at least one Product: No
 - Each Order can have several Products: Yes
 - Each Product has at least one Order: No
 - Each Product can have several Orders: Yes



3. Go to the next step. The wizard proposes to create a relation file. Keep "Create the relation file automatically" and go to the next step.

4. The wizard proposes the unique key of Orders file: "OrdersID". Go to the next step.
5. Validate the creation of the key by going to the next step.
6. Keep the default options regarding the integrity rules and go to the next step.
7. The wizard proposes the unique key of Product file. Select "Reference". Go to the next step.
8. Validate the creation of the key by going to the next step.
9. Keep the default options regarding the integrity rules and go to the next step.
10. Click the green arrow. The relation file is automatically created in the data model editor.

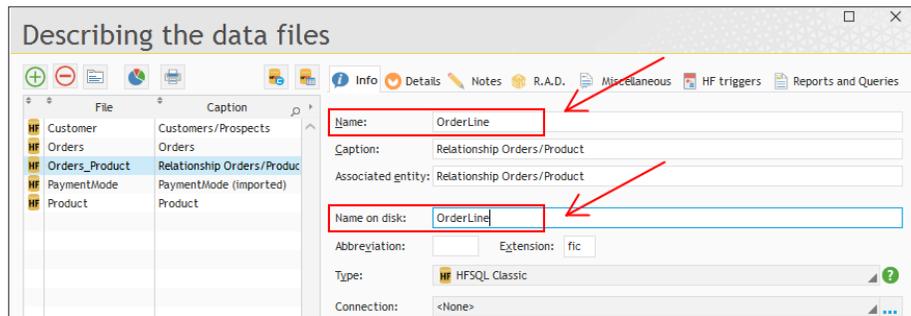


We are now going to modify the relation file that was created by WEBDEV. Indeed, this file will contain the order lines.

We are going to:

- Modify the name of the file.
- Modify the name of its items.
- Add items to find out the quantity of products ordered and the total of the order line.

- ▶ First, we are going to rename the file. A similar operation was already performed when the caption of imported file was modified. However, in this case, not only the caption will be modified: we are also going to rename the physical file linked to the file description.
 1. Select the "Orders_Product" file. Select "Description of data file" from the popup menu.
 2. In the window that is displayed, modify:
 - the file name: "OrderLine".
 - the name on disk: "OrderLine".



- ▶ We are now going to modify the items of this relation file.
 1. Click to display the description of items found in the data file.

OrderLine					
Key	GDPR	Name	Caption	Type	Size
		Reference	Reference	Text	50
		OrdersID	OrdersID	Numeric	8
		OrdersID_Reference	Identifier of Orders_Product	Composite Key	58

2. This file includes 3 items. Position the selection bar on the "OrdersID_Reference" item. This item is a composite key.

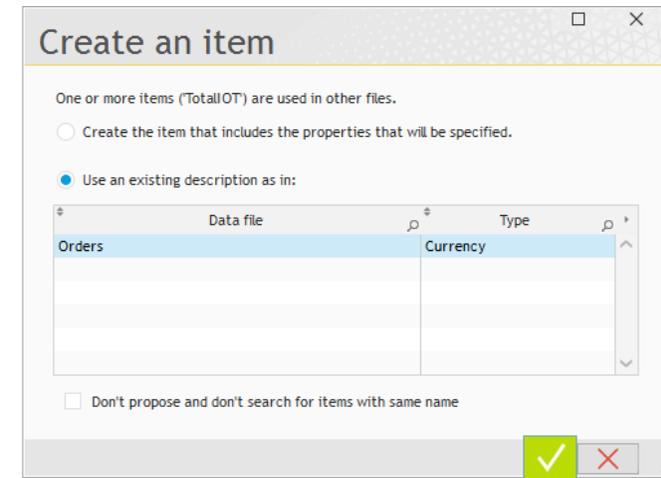


Notes

A composite key is a set of items that constitute an index. This type of key is used to browse the data file according to complex criteria or to perform specific searches on several items at the same time.

3. To rename this item:
 - Click the "Name" column.
 - Replace "OrdersID_Reference" by "OrderLineID".
 - Click the "Caption" column.
 - Replace the current caption by "Identifier of OrderLine".

- ▶ We are now going to add 3 new items into this relation file: Quantity, TotalIOT and TotalBT.
 1. Position on a new row and create the "Quantity" item. This item is a "Numeric" item.
 2. Position on a new table row and create the "TotalIOT" item. This item is a "Currency" item.
 3. A window is displayed, indicating that the item already exists in the analysis and proposing to re-use its characteristics:

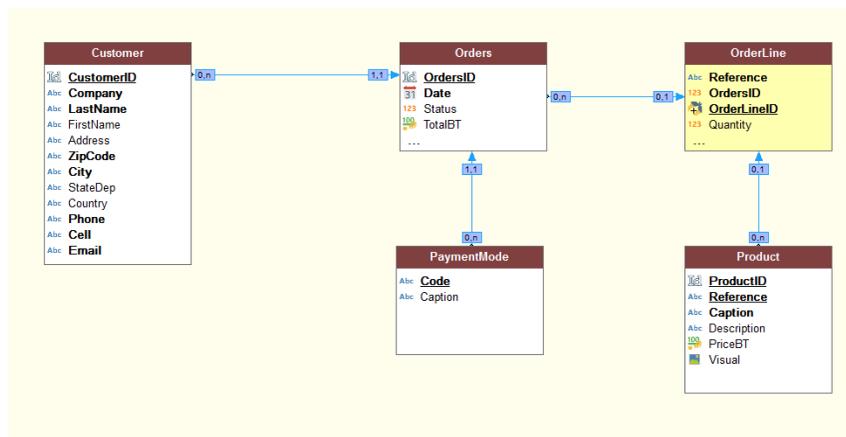


4. Keep the options selected by default and validate (green button).
5. Position on a new table row and create the "TotalBT" item. This item is a "Currency" item. Once again, use the existing description.
6. The description of items for the "OrderLine" file is as follows.

OrderLine					
Key	GDPR	Name	Caption	Type	Size
		Reference	Reference	Text	50
		OrdersID	OrdersID	Numeric	8
		Identifier_of_OrderLine	Identifier of Orders_Product	Composite Key	58
		Quantity	Quantity	Numeric	4
		TotalIOT	TotalIOT	Currency	10
		TotalBT	TotalBT	Currency	10

7. Validate the description of items (green button at the bottom of screen) and the file description.

► The analysis is as follows:



Generating the analysis

Generating the analysis consists in validating the modifications performed in the analysis (creation of data files, addition or deletion of items, ...) and to apply them to the entire project (pages, linked controls, reports, ...).

The generation is automatically offered when you close the data model editor after modifications have been performed.

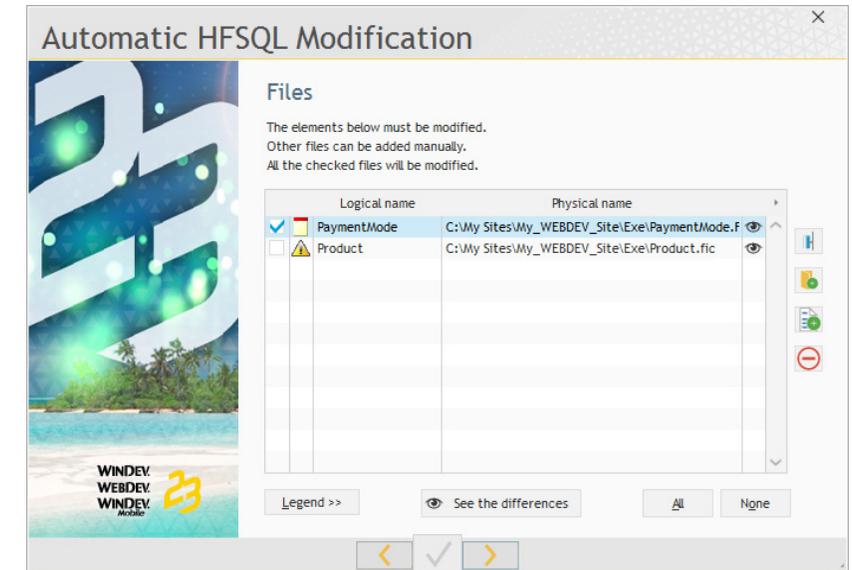
You also have the ability to generate the analysis manually. That's what we are going to do.

- To generate the analysis:
 1. In the data model editor, on the "Analysis" pane, in the "Analysis" group, click "Generation".
 2. The analysis generation is automatically started.

The descriptions of the data files found in the analysis have been modified.

- To update the data files of the application, WEBDEV proposes to synchronize the structure of deployed data (also called "modification procedure of data files"). This operation is used to update the physical data files (".fic" files) according to their description in the analysis.
- Click the "Start" button.

- The wizard for automatic modification starts. Validate the different screens until you reach the screen that lists the files to take into account:
 - The PaymentMode file must be updated. Keep this file selected.
 - The Product file was copied into the project directory. WEBDEV proposes to associate it with this analysis. Check the file. Go to the next step.



- The wizard proposes to save the existing data files, don't change anything and go to the next step.
- The wizard proposes to type the passwords for protecting the data files modified by the automatic modification. Keep the options proposed by default and go to the next step.
- The list of data files to modify is displayed. Validate the wizard.
- The data files are updated.

► Close the data model editor.

The main steps for creating an analysis have been presented.

When your analysis is described and generated, you can:

- Create a full site via RAD (Rapid Application Development). See the online help for more details.
- Create a full custom site. This method will be used to develop a full site based on a database in section 3 of this tutorial. This site will be based on an analysis corresponding to the one that was created in this section.

PART 3

**Intranet site
with data**



LESSON 3.1. OVERVIEW

This lesson will teach you the following concepts ...

- Overview of site created in this part



Estimated time: 5 mn

Overview of site created in this part

The parts 3, 4 and 5 of this tutorial will allow us to develop a site for order management that is using data files in HFSQL Classic format.

In part 3, we are going to develop the Intranet section of the site. This Intranet site will be used to view, enter and modify products. **This site is a dynamic WEBDEV site.**



Notes

Referencing

In a dynamic WEBDEV site, only the home page of the site can be referenced. The referencing of a WEBDEV site is presented in “Lesson 5.6. The referencing”, page 253.

Part 4 will allow you to develop the Internet section of the site. You will have the ability to list the new products and to view their details.

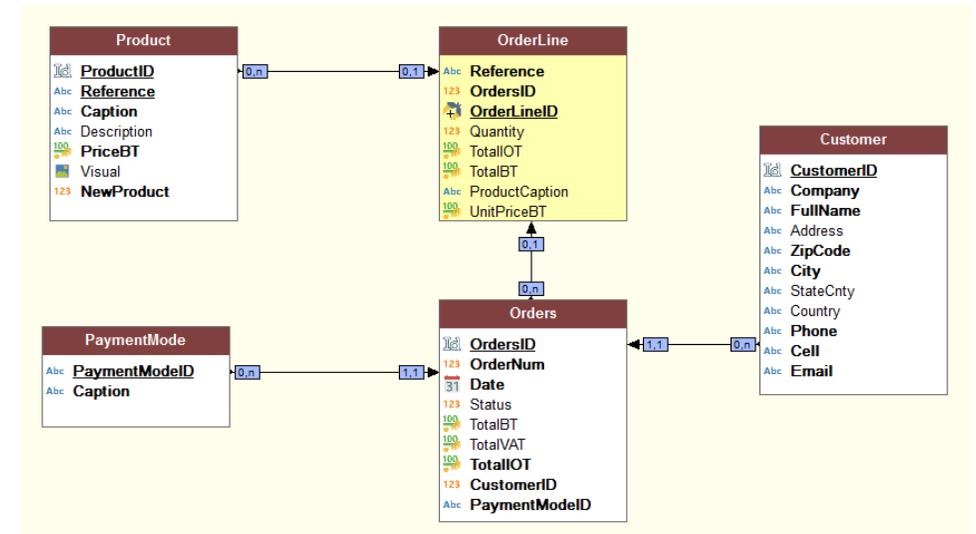
Part 5 will allow you to continue the site development, by proposing the following features:

- Printing reports,
- Managing use levels,
- Sending emails,
- Managing the multilingual feature,
- ...

The development of this site will also allow us to talk about site deployment in part 6. You will see the main points for developing a WEBDEV project.

The creation of the project and analysis was presented in the previous section. We won't go back to it. We are going to work with a project that already contains an analysis and pre-filled data files. The site's analysis is an improved version of the analysis that was created in the previous section.

Let's see the analysis used:



This analysis contains the description of 5 data files:

- Customer,
- Orders,
- OrderLine,
- Product,
- PaymentMode.

This analysis is straightforward and it is used to manage orders.

The site that we are going to develop being quite long (it is a full site that will allow us to discover the main features of WEBDEV), several intermediate projects are available:

- A project used to perform the operations from part 3.
 - A project used to start the operations from part 5. This project includes all the operations performed in parts 3 and 4.
 - A final project used to perform the operations in part 6. This project includes all the operations performed in parts 3, 4 and 5.
- The different projects that can be opened are listed at the beginning of each lesson as well as the operations that must be performed to open them.

LESSON 3.2. PRINCIPLE FOR DISPLAYING A DYNAMIC WEBDEV SITE

This lesson will teach you the following concepts ...

- Page contexts
- WEBDEV session



Estimated time: 5 mn

Principle for displaying a dynamic WEBDEV site

The WEBDEV session

When the Web user connects to a dynamic WEBDEV site, a session is automatically created on the server.

This session contains the page contexts corresponding to each page opened by the Web user.

The session will be ended when the Web user exits from the browser and when the session time-out is exceeded. This time-out is defined in the WEBDEV administrator ("Configuration" tab, "Disconnect the idle users since" option).

The page contexts

A page context is automatically created on the server whenever a page displayed in the browser. This page context contains all the elements that have been required to build the page viewed by the Web user:

- the global variables,
- the local variables,
- the server processes,
- the connections to the databases,
- the file contexts, ...

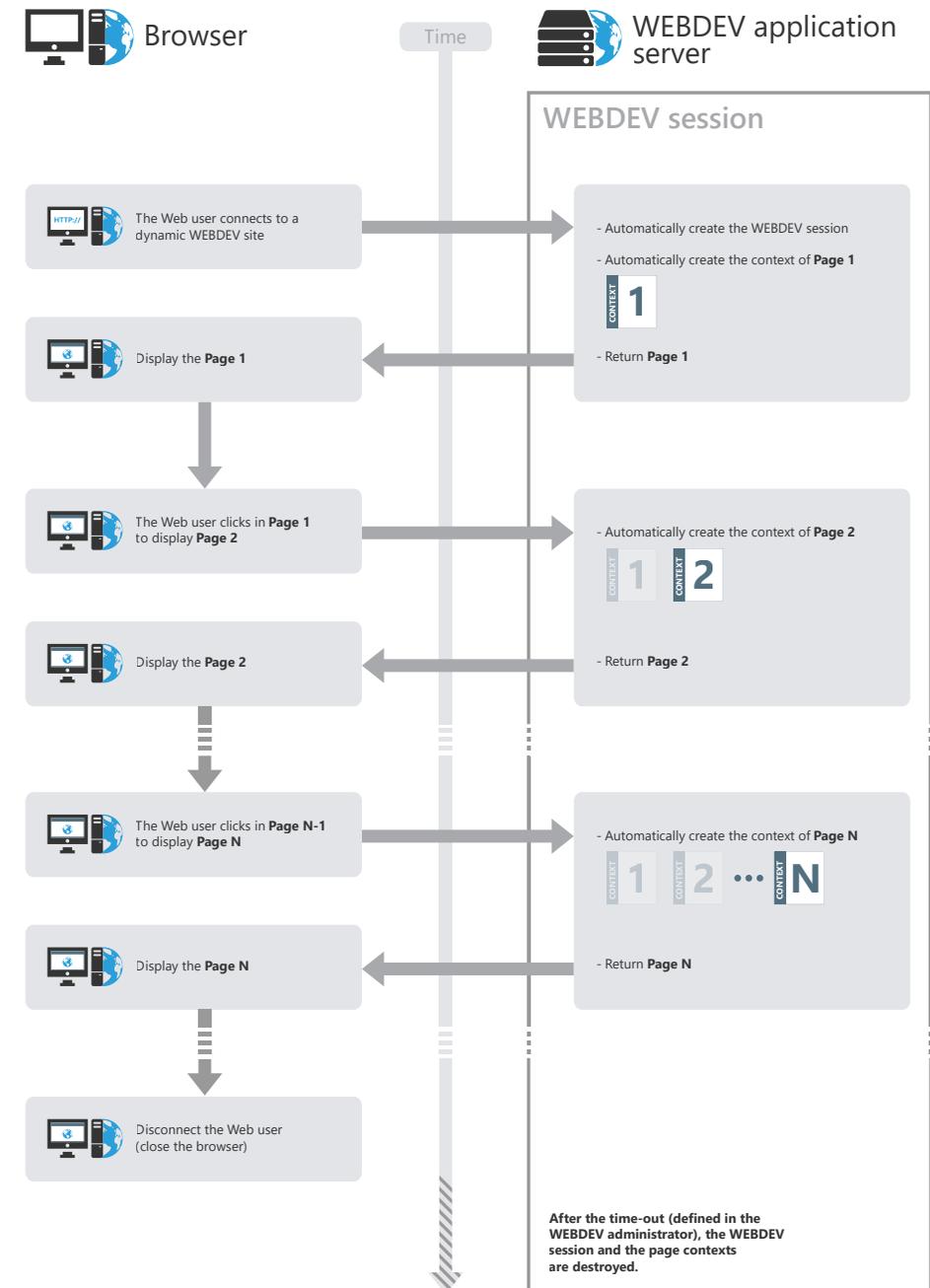
The page contexts remain in memory on the server until the end of WEBDEV session.

If the same page is called several times:

- If **PageDisplay** is called to display the page, the page context is destroyed and re-created.
- If **PageRefresh** is called to display the page, the same page context is re-used.

Programming

The WEBDEV session and the page contexts are automatically managed by default. There is nothing to program.



LESSON 3.3. PAGES FOR ADDITION AND MODIFICATION

This lesson will teach you the following concepts ...

- Creating a dynamic page used to list the products.
- Creating a dynamic page whose type is product form.
- Managing the addition and modification of a product.



Estimated time: 50 mn

Overview

We are going to create the different pages of the dynamic WEBDEV site used to list, add and modify products. These operations will allow you to discover:

- how to use WEBDEV for creating dynamic pages,
- how to access the database.

These operations will also allow you to use some very useful features of WEBDEV.

- ▶ Open the "Full WEBDEV Site" exercise:
 1. Close (if necessary) the current project to display the home window.
 2. In the home window, click "Tutorial" and select "Full WEBDEV Site (Exercise)".



Answer

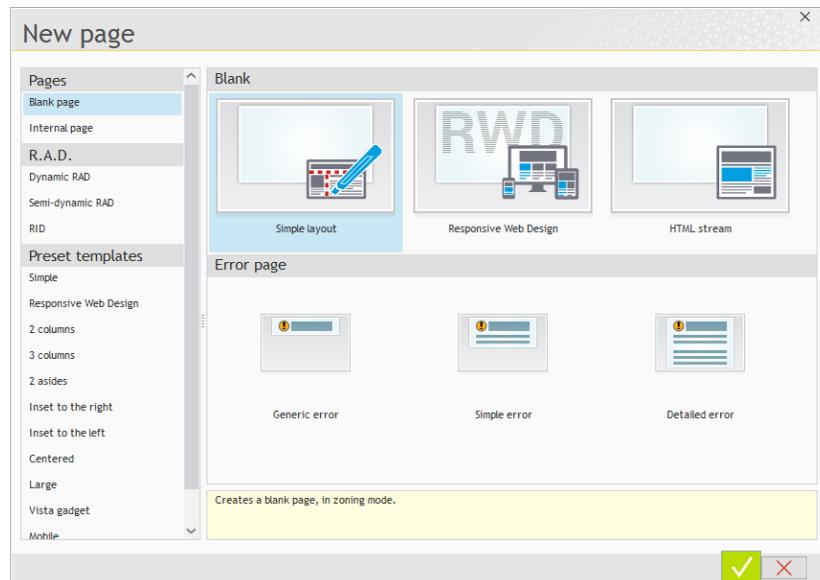
A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (With pages)".

Creating a dynamic page used to list the products

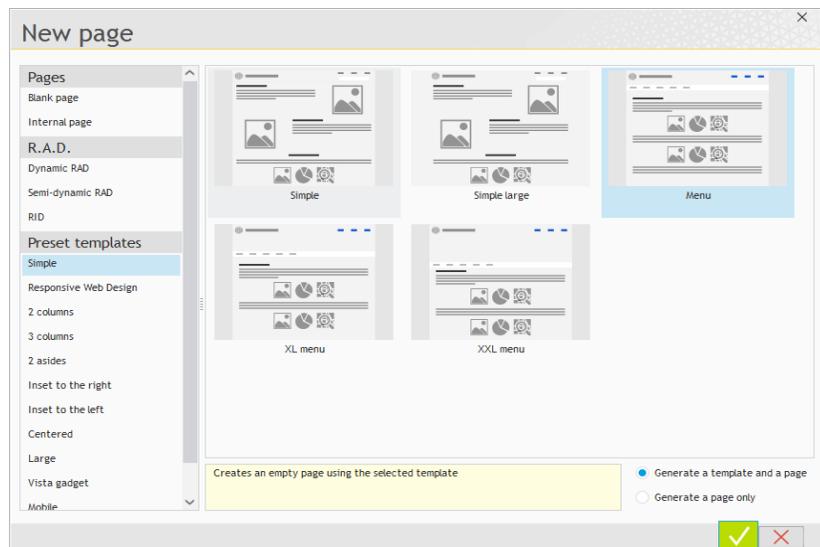
To create the page used to list the products, we are going to create a blank page then add all the controls: this will allow us to present all the steps required to create this type of page.

Creating a page that is using a template

- ▶ To create a blank page:
 1. Click  among the quick access buttons (or use the Ctrl N shortcut).
 2. The window for creating a new element is displayed: click "Page" then "Page".
 3. The wizard for creating a new page is displayed.



4. In the list of preset templates, select "Simple" then, in the list on the right, select the "Menu" template: it is a template with a header at the top, a menu and the page body below.



Notes

A preset template allows you to group part of the interface and processes at a single location. We recommend that you use templates to re-use and standardize the interface of your sites.

5. Make sure that "Generate a template and a page" is checked.
6. Validate the wizard (green button).
7. The new page is displayed in the editor.

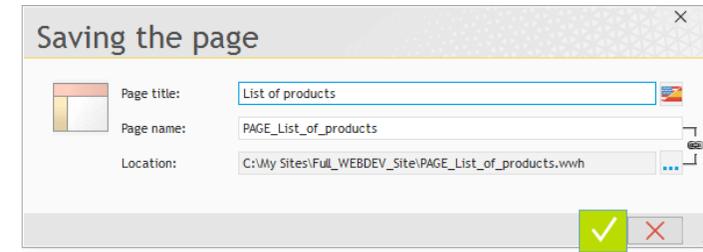


Notes

When validating the wizard for page creation, WEBDEV automatically creates:

- the page that will be displayed in the browser. This page is displayed in the editor directly and it can be modified.
- the associated page template, corresponding to the preset template that was chosen. To modify the page template, it must be edited. We'll see this operation later in this lesson.

8. The backup window of page is displayed. Type the page title: "List of products". The name ("PAGE_List_of_products") is automatically proposed.



9. Validate.
- By default, this page contains the different controls used to display the possible overviews. In our case, we are going to delete these controls:
 1. Select the controls found in the page (press CTRL A).



Notes

In the editor, by default, only the page controls are accessible. You cannot edit the controls and processes found in the template. By pressing the CTRL A keys, only the page controls are selected. The controls of the page template are not modified.

2. Delete the controls ([Delete] key).
3. Only the controls corresponding to the page template remain in the page.
4. Save the modifications performed in the page: click among the quick access buttons (or press CTRL S).

Creating controls

To create the list of products, we will be using a Looper control. This control will be linked to the "Product" data file.



Notes

What is a Looper control?

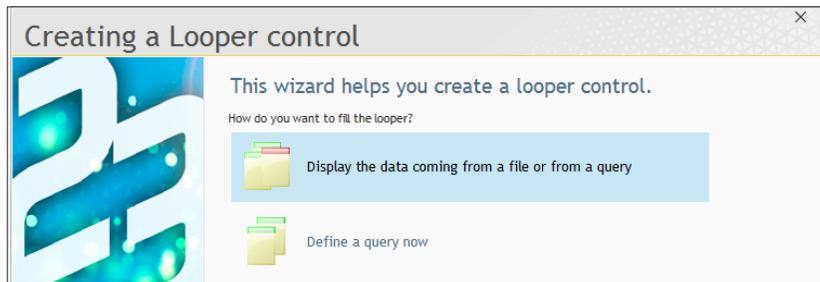
The best way for displaying elements in list format in a Web page is to use a Looper control. A Looper control includes an area (containing several controls) that is repeated a given number of times.

The data found in each repeated area can be different.

► To create the Looper control:

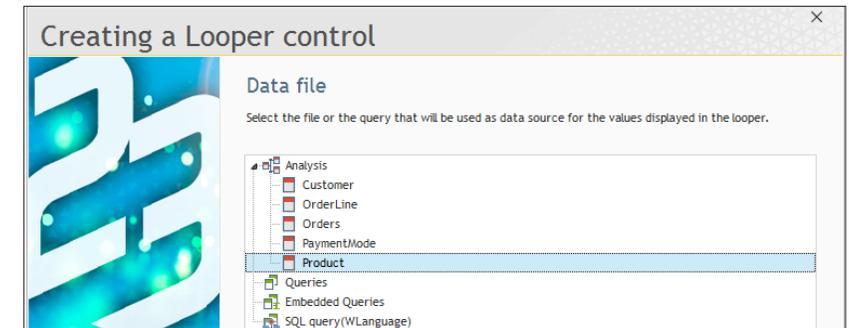
1. On the "Creation" pane, in the "Data" group, expand "Looper" and select "Looper".
2. Click in the top left corner of the page, just below the menu: the wizard for creating the Looper control starts.
3. The wizard wants to know how the Looper control will be filled:
 - by programming (this feature will be presented in a next lesson),
 - from the database, using the a data file or a query,
 - from a WLanguage variable.

In this case, the Looper control must display all the products: select "Display the data coming from a file or from a query".



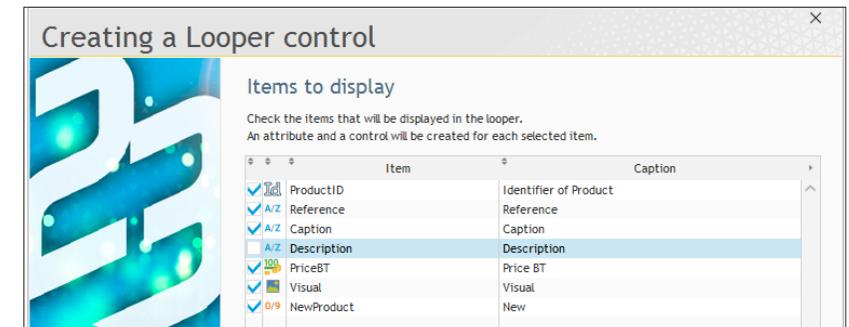
4. Go to the next step by clicking the arrow at the bottom of the screen.

5. The next wizard screen proposes the different data files and queries found in the current project. Expand "Analysis" if necessary and select the "Product" file.



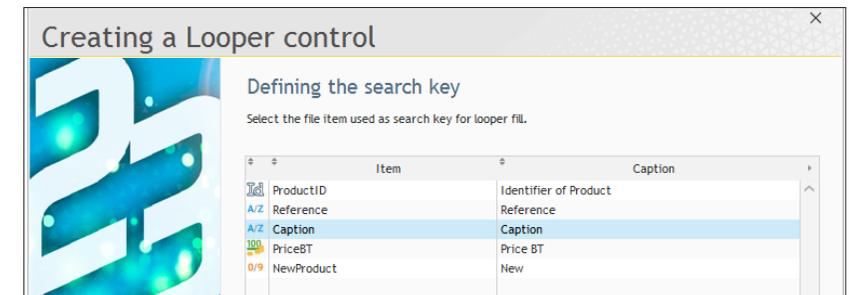
6. Go to the next step.

7. The wizard proposes the list of items found in the data file. By default, all the items are checked in order to be displayed in the Looper control. In our case, we are going to display all the items EXCEPT for the product description.



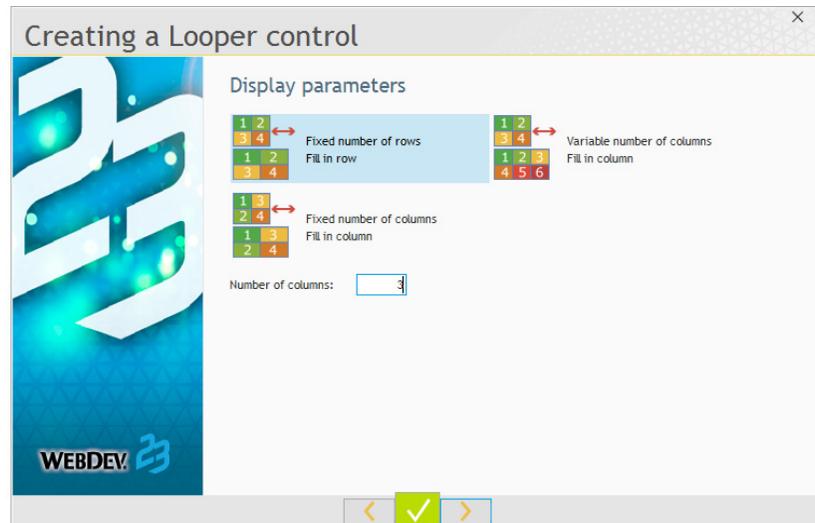
Uncheck "Description" and go to the next step.

8. Then, the wizard proposes to select the search key, which means the default sort for the data displayed in the Looper control. The items proposed in the wizard correspond to the items defined as key items in the analysis. The products will be sorted according to their caption.

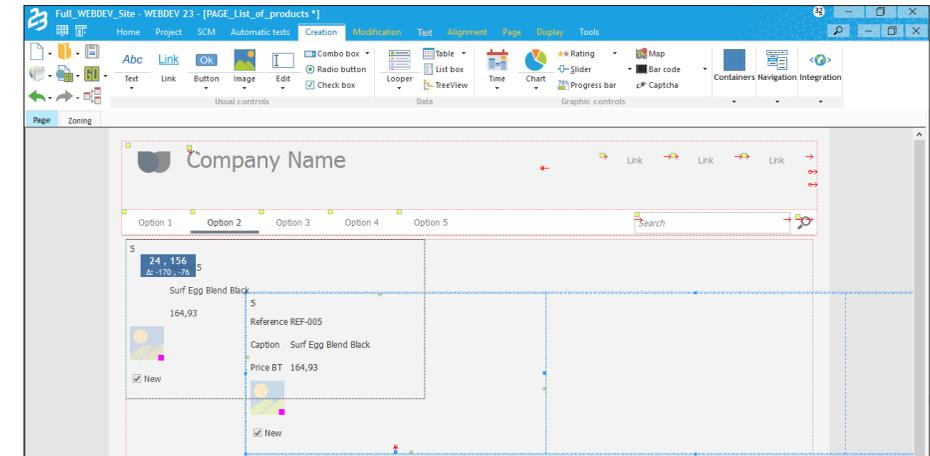


Select "Caption" and go to the next step.

9. We are now going to select additional parameters for creating the Looper control:
- The Looper control is in Classic mode: the entire data will be displayed when loading the page.
Select the "Classic" operating mode.
 - The Looper control is using an unlimited number of rows. Indeed, all the products must be accessible in the page directly. The page will be enlarged in order to display all the products.
Select "No limit" in the "Maximum number of rows per page" area.
10. Go to the next step. The Looper control will display the products on 3 columns.
- Keep the default option: "Fixed number of columns, Fill in row".
 - In the "Number of columns" area, replace 2 by 3.



11. Validate the wizard. The Looper control is automatically created in the page.
- We are going to reposition the control in the page:
 - Select the control.
 - Move the control with the mouse in order to position it in the top left corner of the page.
When the control reaches the limit of display area in the page template, green lines are displayed: these lines are used to position the control inside the display area.



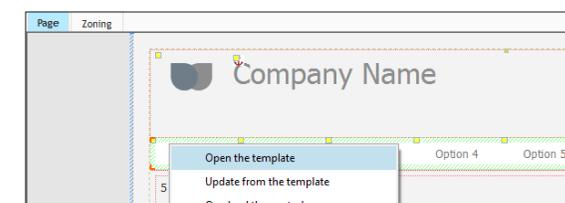
Then, reduce:

- the size of control rows (the size of columns) in order for the control to be entirely displayed in the page.
- the size of caption in order for the control to be positioned in the column.

Let's take a look at the control that was just created: the data is already displayed in the control, even in the editor.

This concept is called "Live data": the content of data files found on the development computer is used in the pages or reports handled in the editor. This feature is very useful to define the size of controls found in a page for example.

- We are going to hide the control containing the product identifier. Why hide it instead of deleting it? Because this identifier will be useful in the rest of our development.
 - Select the "STC_ProductID" control. This control corresponds to the identifier (control with a number at the top of looper).
 - Display the description window of control ("Description" from the popup menu).
 - In the "GUI" tab, uncheck the "Visible" option.
 - Validate the description window of control.
- We are now going to modify the menu of our page in order to display the option used to list the products. The menu is found in the page template. Therefore, we are going to modify the page template.
 - Display the popup menu of the menu (right mouse click) and select "Open the template".



- Select the menu.

3. Click "Option 1". A yellow border appears around the menu. This yellow border indicates that the menu is in "Edit" mode: it can be modified.



4. Display the description window of option (select "Option description" from the popup menu).
5. In the "General" tab:
 - Type the option caption: replace "Option 1" by "List of products".
 - In the "Action" combo box, select the "Display the PAGE_Product_List page" action.
 - Validate the description window of option.
6. Press the [ESC] key to exit from the edit mode.



Notes

We have modified the menu displayed in the template. You also have the ability to modify other template elements such as the logo or the "Company name" caption.

► To update the pages that use the template:

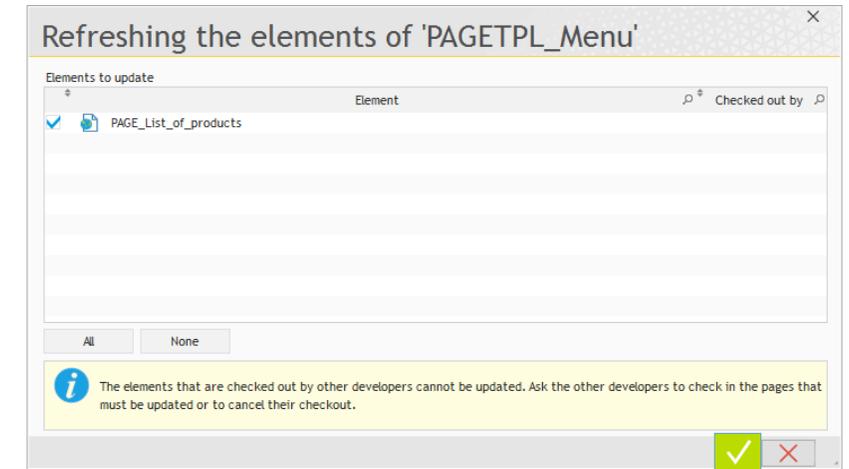
1. In the orange bar of template, click : this button lets you propagate the template's modifications to all the pages that use the model.



Notes

If you do not propagate the template modifications into your pages, a synchronization of templates will be proposed during the test of your page (and also before a deployment).

2. In our case, a single page is proposed.

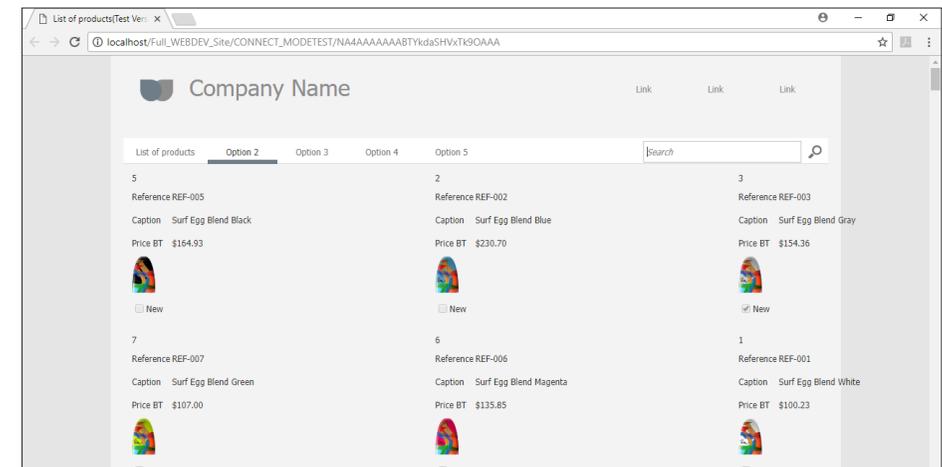


3. Validate the template update window.
4. Close the page template.

Our page is ready to be tested.

Page test

- We are going to run the test of the page that was just created.
 1. Click  among the quick access buttons.
 2. The page is automatically displayed in the browser.



3. You have the ability to scroll the products with the page scrollbar.
- Close the browser. The WEBDEV editor is redisplayed.

Modifying products via a dynamic "Product form" page

Now that the list of products is displayed, you may want to modify a product. Now, we are going to create a page for displaying the product details in order to modify it.

Creating the page

- ▶ To create a page used to display the product details:
 1. Create a new blank page.
 - Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Page" then "Page".
 - The wizard for page creation starts.
 - In the "Based on a project template" area, choose "PAGETPL_Menu" and validate the wizard.
 2. The backup window of page is displayed. Type the page title: "Product form". The name ("PAGE_Product_form") is automatically proposed.
 3. Validate.

What should this page do?

This page is intended to modify the characteristics of the product currently selected in the Looper control. This type of page is called a "Form" because it corresponds to a descriptive form of the requested element.

In our case, this page will be used to display the content of different items found in the "Product" data file.

First, we are going to indicate to the page the product that will be displayed. To do so, all we have to do is declare a parameter in the page.

- ▶ To declare a parameter in the page:
 1. Press the [F2] key. The code editor displays the different page processes.
 2. Type the following code in the "Global declarations" process:

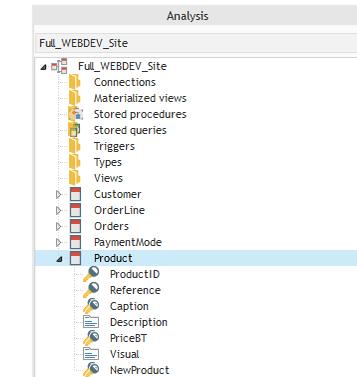
```
PROCEDURE MyPage (gnProductID is int on 8 bytes)
```

3. Let's study this code:
 - The PROCEDURE keyword in the "Global declarations" process is used to associate a procedure with the page opening.
 - The procedure is named "MyPage". At run time, this keyword will be automatically replaced by the name of current page.
 - The procedure expects the gnProductID variable (that is an 8-byte integer) in parameter. This variable corresponds to the product identifier that will be displayed in the page. The type of this variable is identical to the type of the corresponding ProductID item described in Product file.
4. Close the code editor.

Creating edit controls

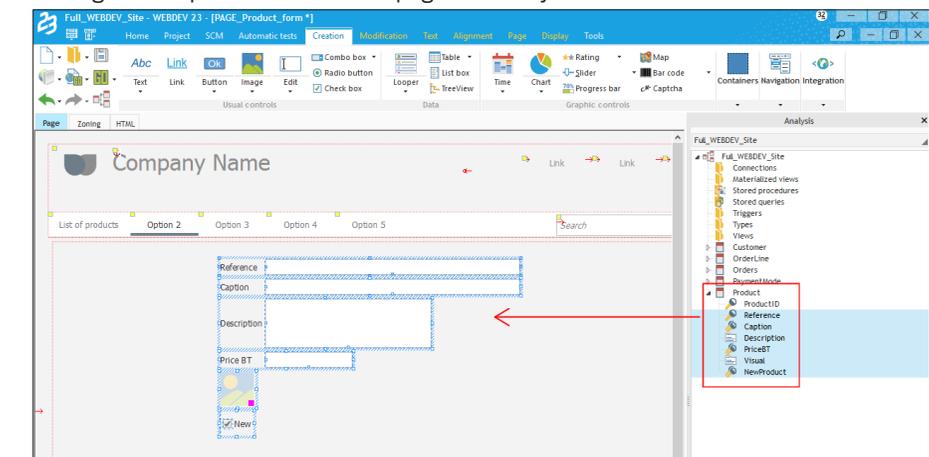
We are now going to create the edit controls used to display information about the selected product in the page.

- ▶ To create the controls in the page:
 1. Display the "Analysis" pane if necessary: in the ribbon, on the "Home" pane, in the "Environment" group, expand "Panels" and select "Analysis". The different data files described in the "Full_WEBDEV_Site" analysis are displayed in the pane.
 2. Click the arrow on the left of "Product" file: the items found in the data file are listed.

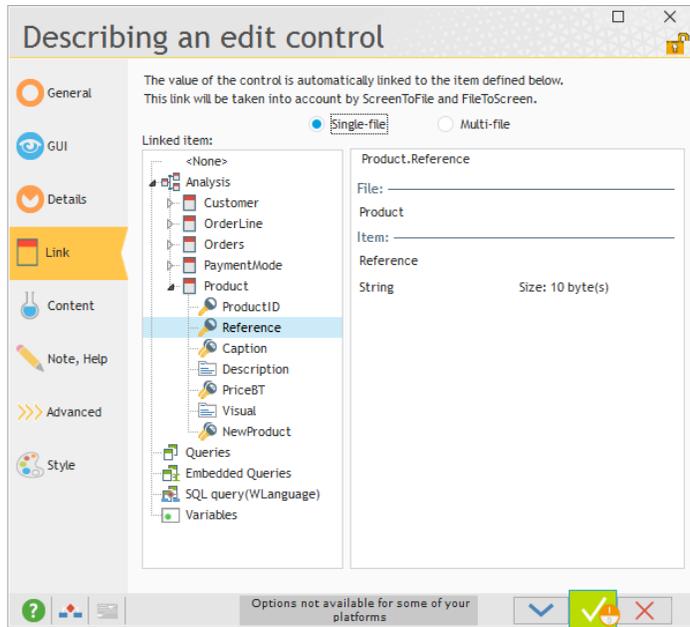


3. With the mouse, select all the items displayed in the pane (except for the "ProductID" item). You can for example use the lasso of the mouse or the multi-selection by keeping the [CTRL] key down.

4. Drag and Drop these items to the page that was just created.



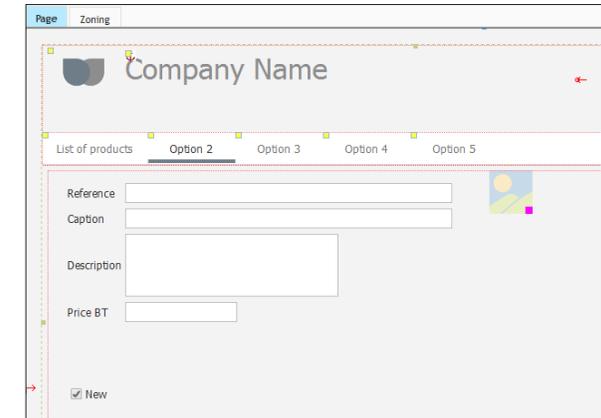
5. Different controls are automatically created in the page. These controls are linked to the corresponding item in the data file. To check this:
- Select the "Reference" control for example.
 - Display the popup menu (right mouse click) and select "Description".
 - Display the "Link" tab of description window:



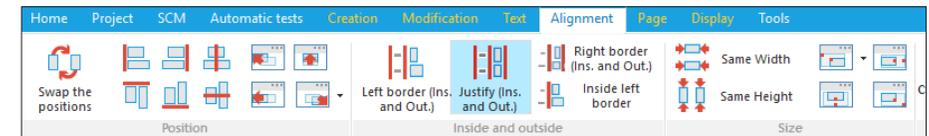
This tab allows you to see that the current edit control is linked to the "Reference" item found in the "Product" data file.

6. Close the description window.
7. Save the page (CTRL S).

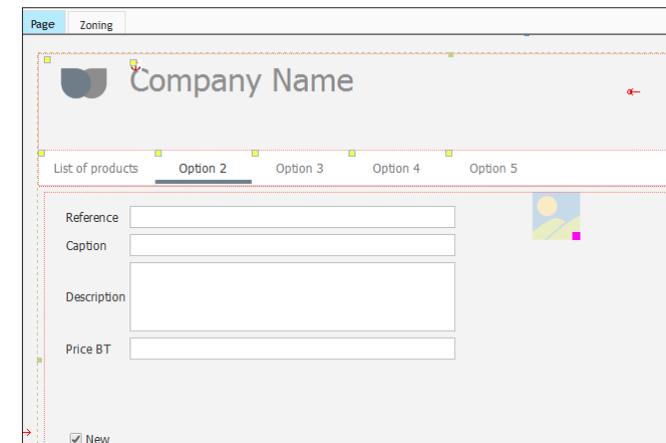
- Reposition the controls in the page in order to get the following interface:



- Enlarge the Image control used to visualize the image associated with the product.
- We are now going to align the edit controls in the page and to give them the same size:
 1. Select the "Description" control then all the edit controls found in the page (keep the [CTRL] key down while clicking the different controls). The first selected control will be used as reference for the size of other controls.
 2. Display the "Alignment" pane of WEBDEV menu. This pane contains all the alignment options available for the controls..



3. We want the outside and inside borders of controls to be aligned. Click "Justify (Ins. and Out.)".



4. Save the page.

Displaying data in the page

The "Form" page must display the product that is selected in the Looper control. In the page code, we are going to type the code used to:

- find the product to display.
- display the data in the page.

► To display data in the page:

1. Press the [F2] key. The code editor displays the different page processes.
2. In the "Global declarations" process, type the following code after the code that was written beforehand:

```
HReadSeekFirst(Product,ProductID,gnProductID)
IF HFound(Product) = False THEN
  // Display the list of products
  PageDisplay(PAGE_List_of_products)
END
FileToPage()
```

3. Let's study this code:

- **HReadSeekFirst** is used to find the first record in the Product file for which the ProductID item is equal to the value of gnProductID. gnProductID corresponds to the parameter passed to the page.
- **HFound** is used to check whether a record was actually found. This function is mainly required for the multi-user sites. It is used to avoid errors caused by the deletions performed by the other users. **HFound** returns False, the list of product is redisplayed.
- **FileToPage** is used to display in the controls the data found in the data file, for the current record. In our case, the current record corresponds to the record found by **HReadSeekFirst**.

4. Close the code editor.

Creating buttons

The "PAGE_Product_form" page will allow you to modify a product found in the list of products.

We are now going to add a cancelation button and a validation button:

- the cancelation button will redisplay the previous page.
- the validation button will check the data typed and save the modified data.

Cancel button

► To create the "Cancel" button:

1. On the "Creation" pane, in the "Usual controls" group, click "Button".
2. Click below the edit controls to create the button.
3. Modify the button caption (press the [SPACE] key to edit the caption): the new caption is "Cancel".

► Type the code associated with the "Cancel" button:

1. Select the button and press the [F2] key. The processes associated with the button are displayed.
2. Enter the following code in the server click code of button:

```
PageDisplay(PreviousPage())
```

3. Let's study this code:

- **PageDisplay** is used to display a specific page.
 - **PreviousPage** is used to find out the name of previous page.
 - This code allows you to display the previous page.
4. Save the modifications ( or CTRL S).
 5. Close the code editor.



Notes

By default, any button created in a page is a SUBMIT button: this button sends the value of page controls to the server. This is the most common case.

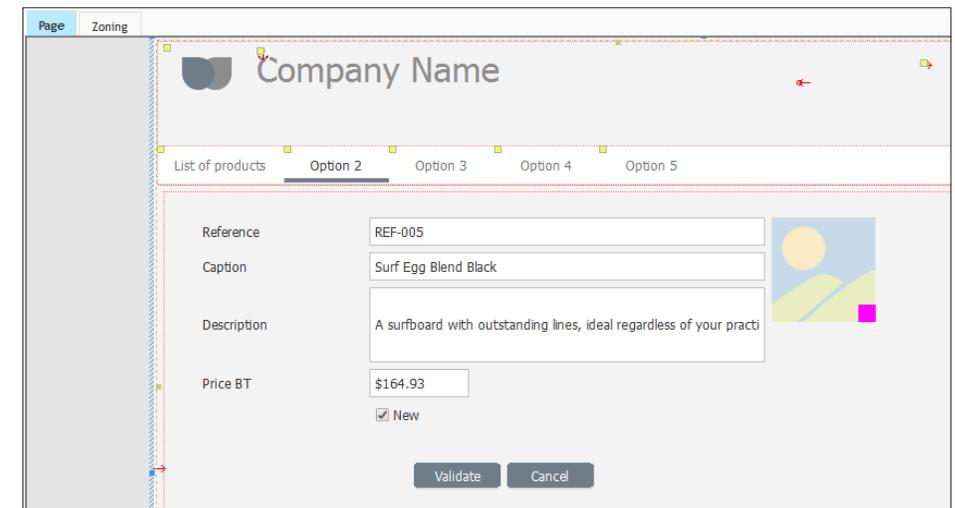
Two other modes are available:

- Reinitialize the page controls.
- No action. This mode must be used when the button should only have a "browser" action.

Validate button

► To create the "Validate" button:

1. On the "Creation" pane, in the "Usual controls" group, click "Button".
2. Click to the left of "Cancel" button then to create the button.
3. Modify the button caption: the new caption is "Validate".



The screenshot shows a web form editor interface. At the top, there's a header with a logo and "Company Name". Below that, there's a navigation bar with "List of products" and "Option 2" through "Option 5". The main form area contains several input fields: "Reference" (REF-005), "Caption" (Surf Egg Blend Black), "Description" (A surfboard with outstanding lines, ideal regardless of your practi...), and "Price BT" (\$164.93). There's also a "New" checkbox which is checked. At the bottom of the form, there are two buttons: "Validate" and "Cancel".

The validation button will be used to:

- check the data typed: this check consists in verifying that the different page controls have been filled by the user. This check doesn't require any server access and can be done in browser code.
 - save the data entered in the Product data file. This backup is performed in server code. The data is sent to the server then it is saved in the data file.
- Type the code associated with the "Validate" button:
1. Select the "Validate" button and press the [F2] key. The processes associated with the button are displayed.
 2. Type the following check code in the browser click code of the button:

```
IF EDT_Reference ~= "" THEN
    Error("Enter a reference.")
    ReturnToCapture(EDT_Reference)
END
IF EDT_Caption ~= "" THEN
    Error("Type a caption.")
    ReturnToCapture(EDT_Caption)
END
IF EDT_Description ~= "" THEN
    Error("Type a description.")
    ReturnToCapture(EDT_Description)
END
IF EDT_PriceBT = 0 THEN
    Error("Type a price.")
    ReturnToCapture(EDT_PriceBT)
END
```

3. Let's study this code:

- For each edit control found in our page, we are going to check whether a value was typed.
- The '~=' operator is used to check the equality while taking the case and punctuation into account.
- If no value is typed, an error message asks the Web user to perform an input (**Error**). The code execution is stopped and the input is forced in the relevant edit control by **ReturnToCapture**.

4. In the server click code of button, type the code for saving data:

```
PageToFile()
HModify(Product)
PageDisplay(PAGE_List_of_products)
```

5. Let's study this code:

- **PageToFile** is used to initialize the items with the values of linked controls, for the current record. This function is equivalent to the following code lines:

```
Product.Reference = EDT_Reference
Product.Caption = EDT_Caption
Product.Description = EDT_Description
...
```

Our page is using less than 10 controls and the benefit is already there; think of the pages containing a lot more controls: a single code line performs all the assignments!

- **HModify** is used to update the data found in the data file for the current record.
 - **PageDisplay** is used to display another page. in our case, we redisplay the "PAGE_List_of_products" page.
6. Save the modifications ( or CTRL S).
 7. Close the code window.

Managing the product image

In the Product file, an item lets you store the image associated with the product.

We currently have the Image control in our page, which allows us to see the image, but we're going to give the user the ability to change the image associated with the product.

To do so, we are going to allow the Web user to upload an image file found on his computer and to associate it with the item found in the data file. We're going to use an Upload control.



Notes

The upload consists in copying a file from the client computer to the server. Downloading, on the other hand, consists in copying a file from the server onto the client computer.



Notes

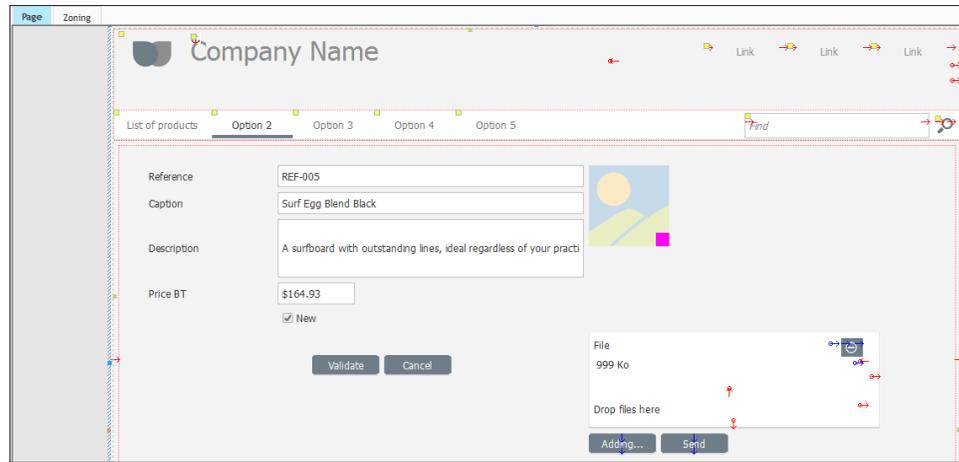
WEBDEV proposes to manage the file upload via two specific controls:

- an Upload control enabling the upload of a single file,
- an Upload control enabling the upload of multiple files.

In this example, the user will upload a single file at a time, therefore we'll use the single-file Upload control.

► To create the Upload control:

1. On the "Creation" pane, in the "Usual controls" group, expand "Button". The list of preset buttons is displayed.
2. Select "Uploading files".
3. Click the position where the control must be created in the page (below the Image control for example).



The Upload control includes:

- a cell containing:
 - a Static control,
 - a Looper control used to display the characteristics of the files to upload.
- a button allowing the user to select the file to upload,
- a button allowing the user to send the files to the server.

Now we're going to adapt the control's code to support the upload of the file in our site.

► To configure the Upload control:

1. Display the code of "Adding" button: select the control and press the [F2] key.
2. Different processes are associated with the Upload control. We are going to modify the "Receiving the uploaded files" process to copy the image into the data directory of the site.
3. Type the following code in the "Receiving the files uploaded" process:

```
UploadCopyFile(MySelf, fDataDir(), ...
  UploadFileName(Myself, False))
gsImagePath = fDataDir() + [fSep()] + ...
  UploadFileName(Myself, False)
IMG_Visual = gsImagePath
```



Notes

This code is using "...". These 3 dots are used to perform a line break in a code line. They're used here for page layout purposes. In the code editor, you can remove them and use a single code line.

4. Let's study this code:

- **UploadCopyFile** is used to save on the server the file uploaded by the user. In our case, the file is copied into the data directory (returned by **fDataDir**). The file name is kept.
- The path of uploaded image is stored in a global variable **gsImagePath**.

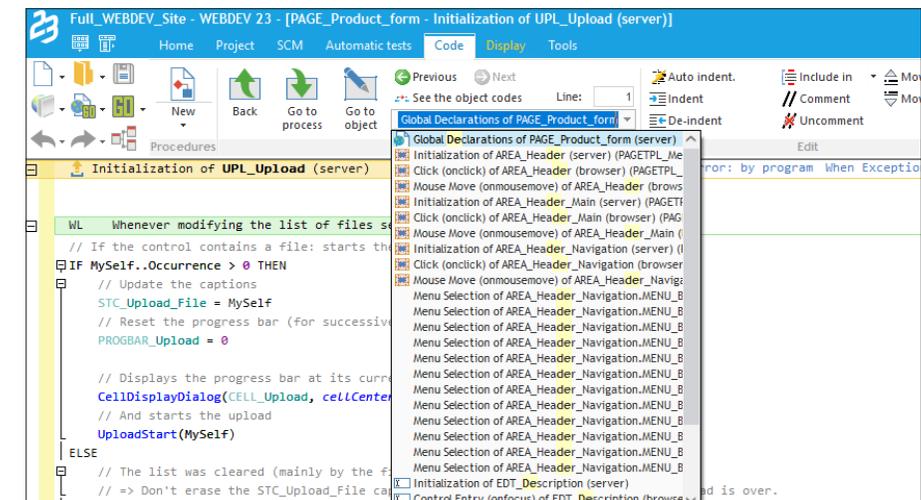


Notes

Why use a global variable?

A global variable is required because the image path will be used in the code of "Validate" button, to save the new image in the data file. For more details about using local and global variables, see the appendices.

- **fSep** allows you to retrieve the separator that will be used on the operating system of the server ("\" for Windows, "/" for Linux). Therefore, your site is independent of the server on which it is installed!
 - Then, the uploaded image is displayed in the IMG_Visual Image control.
5. When typing and saving the code, the **gsImagePath** variable is displayed in red and a compilation error occurs in the error pane: "'gsImagePath' identifier unknown or inaccessible". Indeed, this global variable was not declared.
 6. To declare the global variable:
 - Display the code for declaring global variables of the page (for example, in the code editor, on the "Code" pane, in the combo box that lists all the processes, select the "Global declarations" process).



- Type the following code after the procedure declaration:

```
gsImagePath is string
```

7. Save the modifications by clicking  among the quick access buttons. The compilation errors disappear.
8. Close the code editor.

Our image can now be uploaded on the server. All we have to do now is save the image in the database.

► To save the image in the Product file:

1. Display the code of "Validate" button:
 - Select the "Validate" button.
 - Press the [F2] key.
2. In the server click code of the button, type the following code AFTER the call to **PageToFile**:

```
IF gsImagePath<>" " THEN
    Product.Visual = fLoadBuffer(gsImagePath)
END
```

3. Let's study this code:

- This code checks the content of the global variable **gsImagePath**. If this variable does not correspond to an empty string, it means that the image was uploaded by the user. In this case, the "Visual" item of the Product file is filled with the binary content of the image. This content is retrieved via **fLoadBuffer**.
 - **HModify** (already found in the code) is used to save the changes in the data file.
4. Save the modifications ( or CTRL S).
 5. Close the code window.

Displaying the form from the list of products

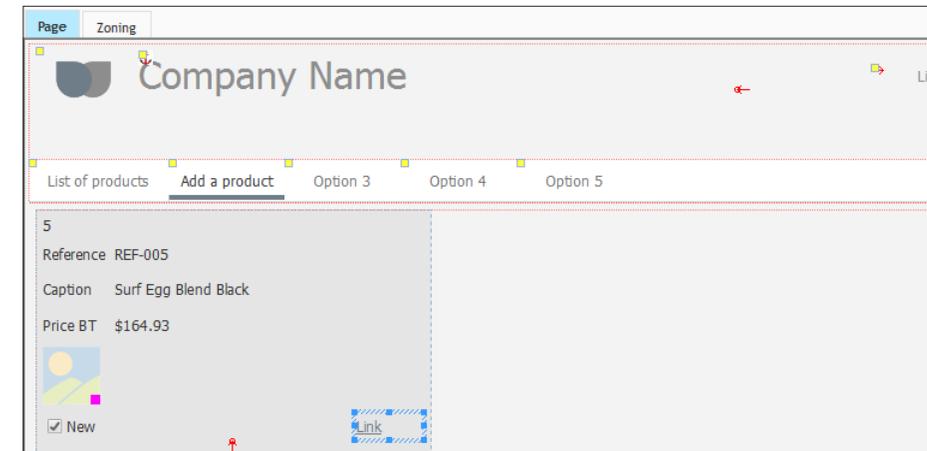
Let's see how to display the form of selected product in the list of products. The principle is straightforward: the user will select the product in the Looper control and he will display the details via a link.

► We are going to modify the "PAGE_List_of_products" page in order to create a modification link:

1. Position on the "List of products" page: click the "PAGE_List_of_products" button found in the button bar:



2. On the "Creation" pane, in the "Usual controls" group, click "Link".
3. Then click inside the Looper control to create the link (for example at the bottom right).



4. Modify the link caption (press the [SPACE] key to edit the caption): the new caption is "Modify".

► The "Modify" link must open the "PAGE_Product_form" page. We are going to open this page by programming.

1. Select the "Modify" link and display the associated processes ([F2] key).
2. In the code window that is displayed, type the following code in the server "Click" process:

```
PageDisplay(PAGE_Product_form,ATT_ProductID[LOOP_Product])
```



Notes

The assisted code input is going to help you: as soon as you type the opening bracket "(", a drop-down list proposes the name of all existing pages found in the project. Simply select the page using the keyboard or the mouse.
If the page name is not found in the list, it means that the page was not saved beforehand.

3. Let's study this code:

- **PageDisplay** is used to open the "PAGE_Product_form" page.
- The opened page expects in parameter the identifier of product to display. This identifier corresponds to the identifier of product selected in the Looper control. To get the identifier, you must specify the attribute that contains the identifier (ATT_ProductID) for the current row. The brackets are used to specify the row and LOOP_Product is used to get the current row in the looper.



Notes

By default, ATT_ProductID returns the attribute value for the row that was clicked. The code can be written as follows:
`PageDisplay(PAGE_Product_form,ATT_ProductID)`

4. Save the modifications ( or CTRL S).
5. Close the code window.

The different elements used to manage the product modification have been implemented, we are now going to run a test to check the operating mode.

- ▶ Run the project test ( among the quick access buttons).
 1. The editor is asking for the first project page. In our case, select the "PAGE_Product_List" page and validate.



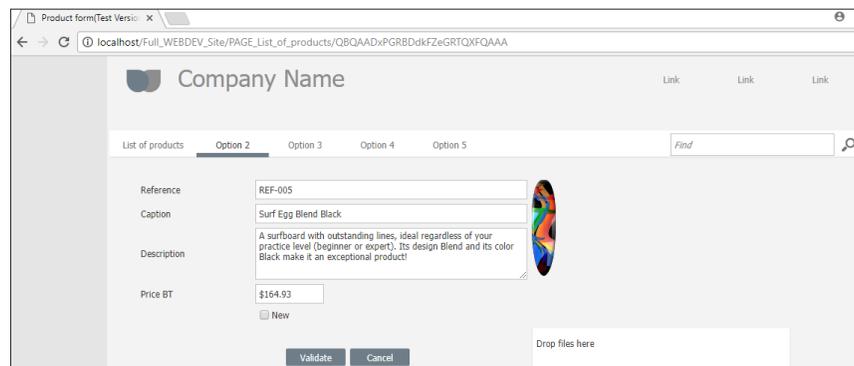
Notes

The first page of the project corresponds to the first page opened when starting the site.
 The first project page can be defined:

- during the project test.
- in the project explorer: all you have to do is select the requested page, display the popup menu and select "First dynamic project page".

When a first project page is defined, a small 1 in red is displayed in front of the page name in the project explorer.

2. The site is started.
3. In the list of products:
 - Click the "Modify" link.
 - The detailed page of the product is displayed.



The screenshot shows a web browser window with the URL `localhost/Full_WEBDEV_Site/PAGE_List_of_products/QBQAADxPGRBDdkFZeGRTQXFQAAA`. The page title is "Company Name". Below the title, there are three "Link" buttons. A navigation bar contains "List of products", "Option 2", "Option 3", "Option 4", "Option 5", and a "Find" search box. The main content area displays a product form for "Surf Egg Blend Black" with the following fields:

- Reference: REF-005
- Caption: Surf Egg Blend Black
- Description: A surfboard with outstanding lines, ideal regardless of your practice level (beginner or expert). Its design Blend and its color Black make it an exceptional product!
- Price BT: \$164.93
- New

At the bottom, there are "Validate" and "Cancel" buttons, and a "Drop files here" area.

- Modify the product price and validate.
 - The new product price is displayed in the list of products.
- ▶ Close the pages to stop the test.

Adding a new product via the "Product form" page

We just explained how to modify a product. Now, we want to be able to add a product. To do so, there is not need to re-create a new page: we will be using the "PAGE_Product_form" page that was created beforehand and adapt it to manage the addition.

First of all, we are going to modify the opening mode of "PAGE_Product_form" page:

1. Position on the "Product form" page: click the "PAGE_Product_form" button found in the button bar.
2. Click the [F2] key to display the page code.
3. In the "Global declarations" code, we are going to give a default value to the parameter passed to the page. Indeed, when modifying the record, the parameter passed always corresponds to the identifier of the product to modify. But when adding a record, the element identifier does not exist. To manage this case, we are going to use the default value "-1".
4. In the "Global declarations" process, replace the following code:

```
PROCEDURE MyPage (gnProductID is int on 8 bytes)
```

by the code:

```
PROCEDURE MyPage (gnProductID is int on 8 bytes = -1)
```

5. We must now manage the value "-1" (when adding a record). Replace the following code:

```
HReadSeekFirst (Product, ProductID, gnProductID)
IF HFound (Product) = False THEN
  // Display the list of products
  PageDisplay (PAGE_List_of_products)
END
FileToPage ()
```

by the code:

```
IF gnProductID = -1 THEN
  HReset (Product)
ELSE
  HReadSeekFirst (Product, ProductID, gnProductID)
  IF HFound (Product) = False THEN
    // Display the list of products
    PageDisplay (PAGE_List_of_products)
  END
END
FileToPage ()
```

Let's study this code:

- If the product identifier is set to -1, it means that we are adding a product. In this case, **HReset** is run. **HReset** initializes the variables of items found in the "Product" file with the default values to manage a new record.
- If the product identifier has a value other than -1, we retrieve the code allowing us to open the form in modification mode.

6. Close the code window.

► The record addition must also be supported by the validation button.

1. In the "PAGE_Product_form" page, select the "Validate" button.
2. Display the processes associated with the button ([F2] key).
3. The code found in the browser click process must not change: the checks to perform are always the same Only the server code must be modified.
4. In the server "Click" process, replace the code by the following code:

```
PageToFile ()
IF gsImagePath<>" " THEN
  Product.Visual = fLoadBuffer(gsImagePath)
END
IF Product..NewRecord THEN
  HAdd(Product)
ELSE
  HModify(Product)
END
PageDisplay(PAGE_List_of_products)
```

Let's study this code:

- **..NewRecord** is used to find out whether the current record must be created.
- If **HReset** was called beforehand, the property returns True (case of the new product") and the record must be created by **HAdd**.
- Otherwise, the current record already exists and it must be modified by **HModify**.
- **HAdd** adds the record into the data file. This function takes the values in memory and writes the content of file items into the data file itself. The indexes are immediately and automatically updated. In this case, it is the "Product" data file that is updated.



Notes

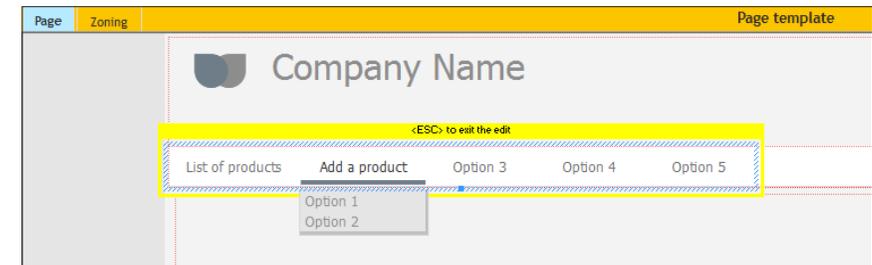
The test code of new record can be replaced by **HSave**. This function is used to check whether the record is already found in the data file, and it allows you to add it or to modify it. The code becomes:

```
PageToFile ()
IF gsImagePath<>" " THEN
  Product.Visual = fLoadBuffer(gsImagePath)
END
HSave(Product)
PageDisplay(PAGE_List_of_products)
```

5. Save the modifications ( or CTRL S).

6. Close the code window.

- We are now going to modify the menu in our page in order to allow the user to add a new product.
1. Position on the "List of products" page: click the "PAGE_List_of_products" button found in the button bar.
 2. Display the popup menu of a button and select "Open the template".
 3. Select the menu.
 4. Click "Option 2". A yellow border appears around the menu. This yellow border indicates that the menu is in "Edit" mode: it can be modified.
 5. Display the description window the option ("Option description" from the popup menu).
 6. In the "General" tab:
 - Type the option caption: replace "Option 2" by "Add a product".
 - In the Action area, select the "Display the PAGE_Product_Form page".
 7. Validate. The menu option appears:



8. In the orange bar of template, click  to propagate the template modifications to all pages that use the template.
9. In our case, the two project pages are proposed. Validate the template update window.
10. Close the page template.

We now can check the product addition.

Checking the product addition

- To check the product addition:
1. Run the project test ( among the quick access buttons).
 2. Click "Add a product".
 3. Enter a new product with the following characteristics:
 - Reference: REF-337
 - Caption: Surf 23
 - Description: Surf with WEBDEV colors
 - Price: 150

4. Validate. The new product appears in the list of products.
5. Click "Add a product" again.
6. Enter a new product with the following characteristics:
 - Reference: REF-337
 - Caption: Surf 23
 - Description: Surf with WEBDEV colors
 - Price: 150
7. Validate. A specific page is displayed:

ProductID	Caption	Description	PriceBT	Reference	Visual	NewProduct
338	Surf 23		150	REF-337	0	

This page informs the user that a duplicate was found: indeed, the reference (that is a unique key) is identical for two products. This page is used to modify the reference value (displayed in a red area): type "REF-338" for example.

This page is one of the pages for automatic management of HFSQL errors.



Notes

Several errors may occur when adding or modifying a record: duplicate error, integrity error, password error, ...

WEBDEV proposes several modes for managing these errors:

- **the automatic mode:** a specific page is displayed to the user whenever an error occurs when managing the database records. This page allows the user to modify his data directly.
- **the advanced programmed mode:** a procedure or a custom page for error management is called whenever an error occurs when managing the database records.

The "WW_Auto_Management_Errors" example, supplied with WEBDEV, is used to check these different modes for error management. This example can be opened via the home window of WEBDEV ("Open an example" option).

Viewing the records

The new records that we entered can be immediately viewed in the Looper control of the "PAGE_Product_list" page.

You can also view them using the WDMAP tool that we saw in the "My first pages" lesson.

PART 4

**Internet site
with data**



LESSON 4.1. OVERVIEW

This lesson will teach you the following concepts ...

- Overview of site created in this part



Estimated time: 5 mn

Overview of site created in this part

In this part, we will be using the project that was developed in part 3: we are going to create an Internet section in the Intranet site. The Internet section will be used to list the new products and to view their details.

This section will be created in AWP mode (Active WEBDEV Page).



Notes

Referencing

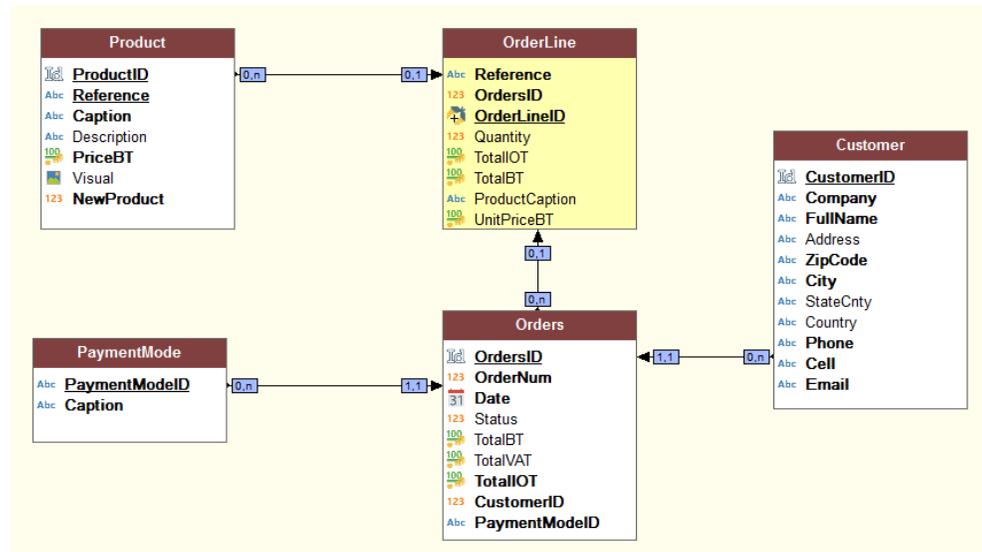
In a WEBDEV AWP site, all the site pages can be referenced.

Part 5 will allow you to continue the site development, by proposing the following features:

- Printing reports,
- Managing use levels,
- Sending emails,
- Managing the multilingual feature,
- ...

The development of this site will also allow us to talk about site deployment in part 6.

Reminder: We are going to work on a project already associated with an analysis and containing data files filled beforehand. Let's see the analysis used:



This analysis contains the description of 5 data files:

- Customer,
- Orders,
- OrderLine,
- Product,
- PaymentMode.

This analysis is straightforward and it is used to manage orders.

If you have followed part 3 of this tutorial, the operations of part 4 are performed in the same example.



Caution!

If you did not follow part 3 of this tutorial, you will not be able to perform the operation used to link the Internet and Intranet section of the site ("Linking the Internet site and the Intranet site", page 163).

LESSON 4.2. PRINCIPLE FOR DISPLAYING AN AWP SITE

This lesson will teach you the following concepts ...

- What is an AWP page?
- Operating mode of AWP sites.
- Sharing information between AWP pages.



Estimated time: 20 mn

Principle for displaying a WEBDEV AWP site

The Internet section of our site will be developed in AWP mode. Unlike the dynamic sites, the AWP mode allows for an easy referencing of the site pages.

Let's take a look at the operating mode of the AWP mode.

What is an AWP page?

An AWP page (Active WEBDEV Page) is a dynamic WEBDEV page **without persistent context** on the server. The AWP page context is temporary. It is created in a temporary session.

Reminder: In a dynamic WEBDEV site, each page displayed owns a persistent page context for the entire lifetime of the session on the server.

Operating mode of AWP sites

Are automatically created on the server whenever an AWP page is displayed in the browser:

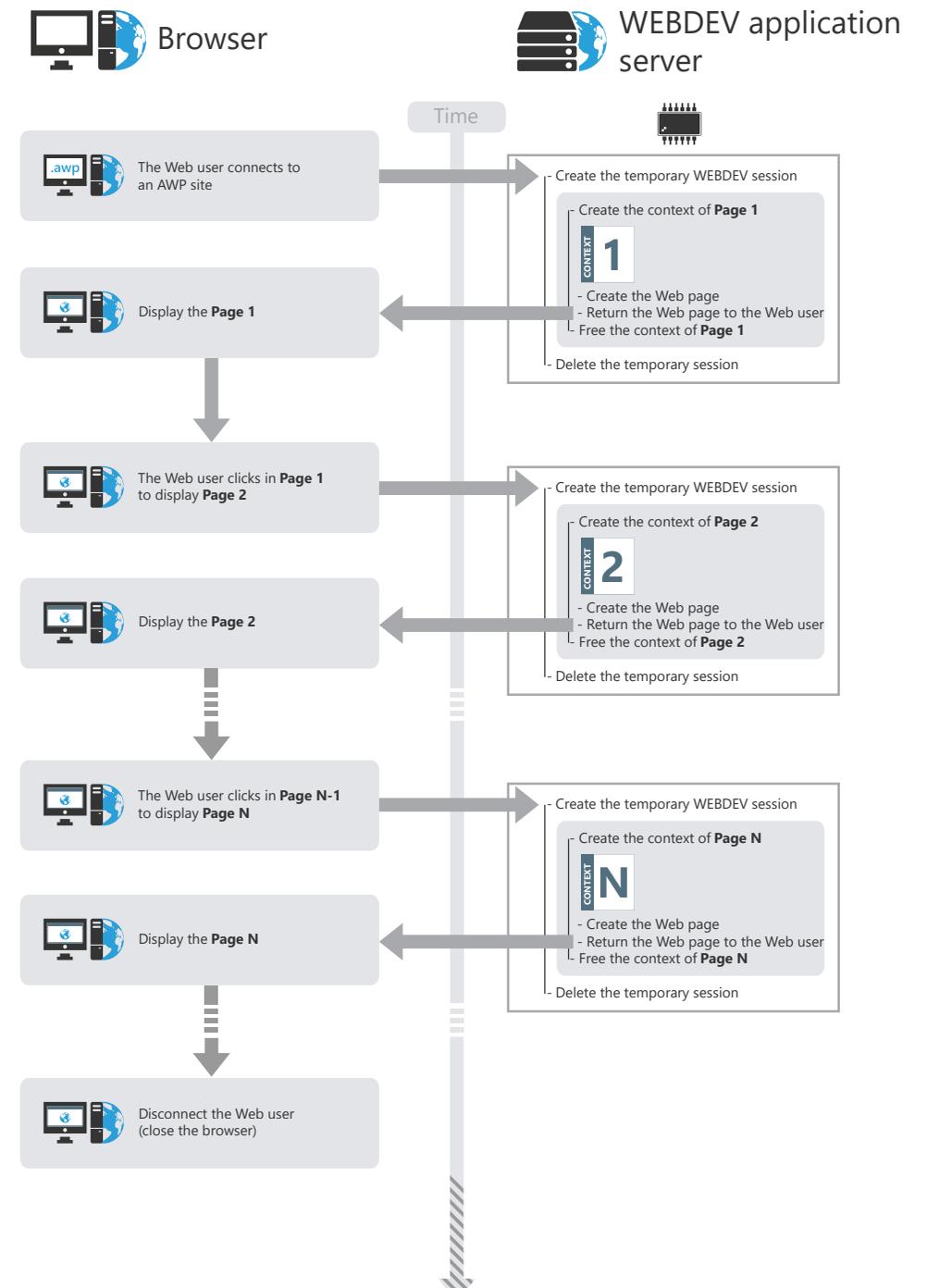
- a temporary session,
- a temporary AWP page context.

The **temporary session** contains the temporary AWP page context. When the AWP page was sent to the Web user, the temporary page context and the temporary session are destroyed. There is nothing left in memory on the server.

The **temporary AWP page context** contains all the elements that have been required to build the page viewed by the Web user:

- the local variables,
- the server processes,
- the connections to the databases,
- the file contexts, ...

When the AWP page was sent to the Web user, these elements are destroyed.



How to share information (values) between AWP pages?

Two methods are used to share information (values) between AWP pages:

- Passing information in the URL. This method allows for a better referencing.
- Saving information in the AWP contexts (by programming).

Passing information (values) between two pages in the URL

You have the ability to pass information from a page to another one via the URL. The URL has the following format: "http:\\Web server\\...\\mypage.awp?NameParam1=Value1&NameParam2=Value2".

This method allows for a better referencing of the page because the information passed in the URL is visible and analyzed by the referencing robots.

Saving information (values) in the AWP contexts (by programming).

You have the ability to store on the server values common to several AWP pages, via the **AWP contexts**. An AWP context is created on disk on the server. This context is available as long as AWP pages are displayed and as long as the time-out of AWP contexts is not exceeded. The time-out of AWP contexts is defined in the WEBDEV administrator ("Configuration" tab, "Duration of AWP contexts" option).

To manage the AWP contexts, you must use the WLanguage functions such as **DeclareAWPContext**, **FreeAWPContext**, ...

See the online help for more details.

LESSON 4.3. CREATING AWP PAGES

This lesson will teach you the following concepts ...

- Creating an AWP page used to list new products.
- Importing CSS styles.
- Importing CSS styles.
- Link between the Internet section and the Intranet section of the site.
- Creating a page for general sales conditions.



Estimated time: 50 mn

Overview

We are going to create the different AWP pages used to list, add and modify new products. These operations will allow you to discover several topics regarding the management of data files and they will also allow you to use some features of AWP mode.

- ▶ If the "Full WEBDEV Site" exercise was not opened in the previous lesson:
 1. Close (if necessary) the current project to display the home window.
 2. In the home window, click "Tutorial" and select "Full WEBDEV Site (Exercise)".



Answer

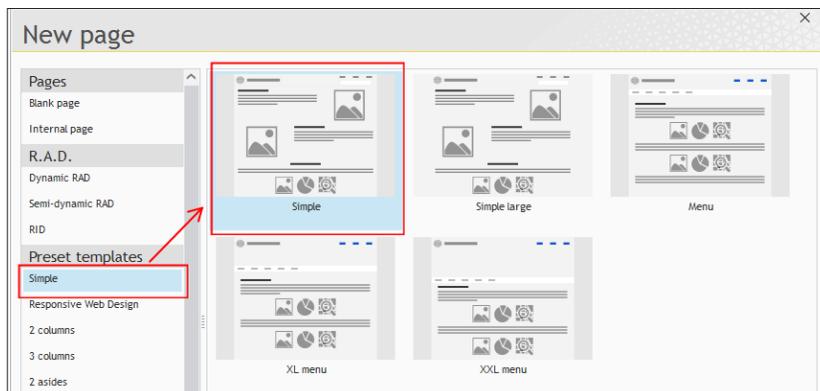
A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (Answer)".

Creating an AWP page used to list new products

We are now going to create a page used to list the new products in the "Full_WEBDEV_Site" project. This page will be a "Showcase" page and it will be accessible via Internet. This page must be referenced on the Internet, therefore it must use the AWP generation mode.

Creating the page

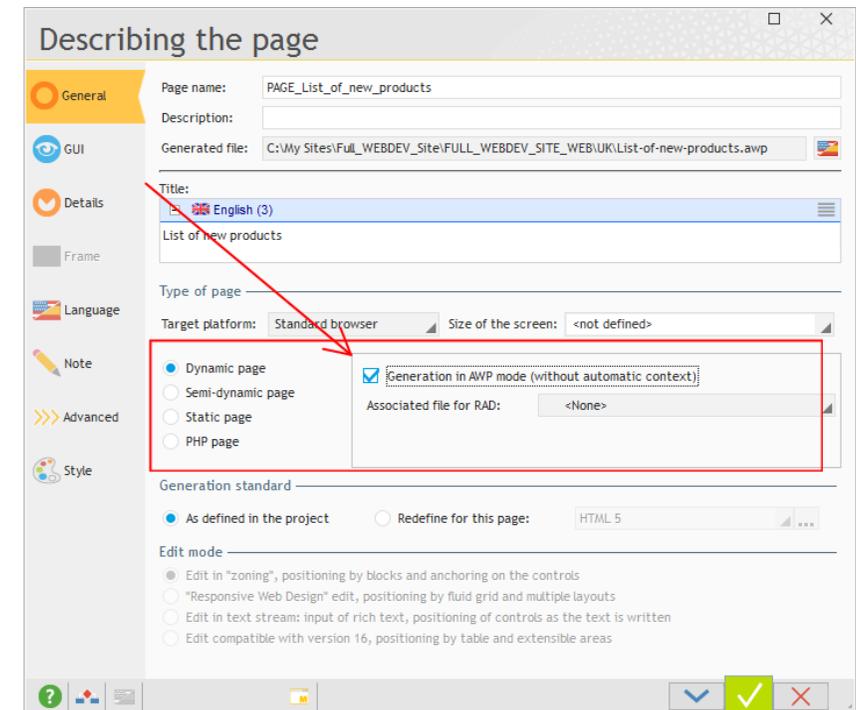
- ▶ To create a page used to list new products:
 1. Create a new page:
 - Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Page" then "Page".
 - In the wizard for page creation, click the "Simple" template and select "Simple".



- Make sure that "Generate a template and a page" is checked.
- Validate the wizard.

2. The new page appears in the editor. The backup window is displayed.
3. Type the page title: "List of new products". The page name ("PAGE_List_of_new_products") is automatically proposed.
4. Validate.

- ▶ The page was created with the controls proposed by default. Delete these controls:
 1. Select the controls found in the page (press CTRL A).
 2. Delete the controls ([Delete] key)].
- ▶ This page must be referenceable. To do so, it must be generated in AWP mode:
 1. Display the description window of page: display the popup menu and select "Description".
 2. Select the "Generation in AWP mode (without automatic context)".



3. Validate.
4. Save the modifications by clicking  among the quick access buttons.

Creating the list of new products

We are going to display the list of new products. To do so, we will be using a Looper control. We have already explained in the previous part how to create this type of control without programming, via the wizard. For this page, we are going to fill the Looper control by programming.

Creating the Looper control

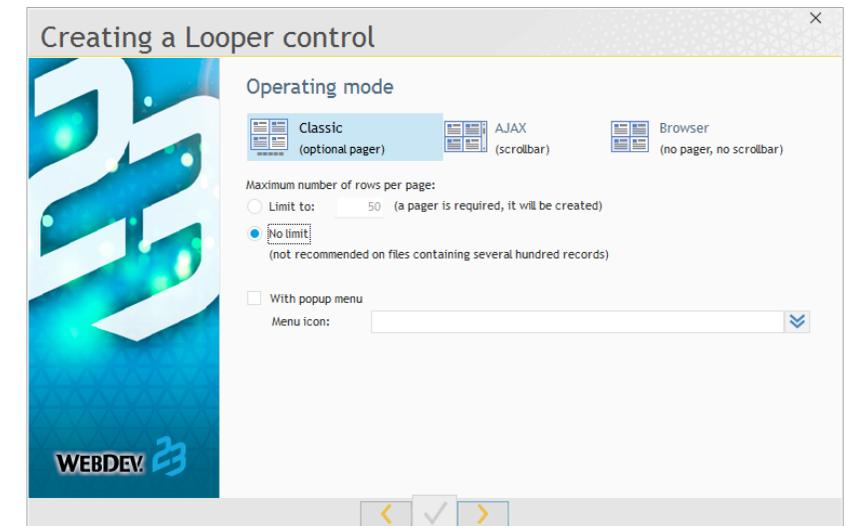
► To create the Looper control:

1. On the "Creation" pane, in the "Data" group, expand "Looper" and select "Looper". The control currently created follows the move of the mouse.
2. Click in the top left corner of the page: the wizard for creating the Looper control starts.
3. In the wizard, select "I want to fill the looper by programming". Go to the next step.



4. In the next step, we are going to select additional parameters for creating the Looper control:

- The Looper control is in Classic mode: the entire data will be displayed when loading the page. Select the "Classic" operating mode.
- The Looper control is using an unlimited number of rows. Indeed, all the products must be accessible in the page directly. The page will be enlarged in order to display all the products. Select "No limit" in the "Maximum number of rows per page" area.



5. Go to the next step. We are going to configure the display parameters.

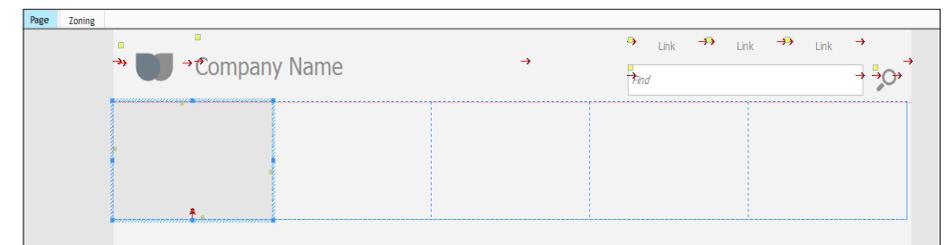
- Keep the proposed display direction: fixed number of columns, display in row.
- The Looper control will display the new products on 5 columns. In the "Number of columns" area, replace 2 by 5.

6. Go to the next step.

7. The last step is used to specify the control name (LOOP_NewProducts for example). Validate the wizard. The Looper control is automatically created in the page.

The created Looper control is empty. WEBDEV materializes the main cell (the one that will be edited) with a solid line and the other ones with a dotted line.

► Reduce the width of main cell in order for the 5 cells to fit across the page width.



We are now going to create the different controls that will be displayed in the Looper control.

Each row will display:

- An Image control displaying the product image.
- A Link control displaying the product name and used to open the detailed form.
- A Static control used to store the product identifier.

We are going to create these different controls and program the fill operation for the Looper control.

Creating the controls found in the looper

► To create the Image control:

1. On the "Creation" pane, in the "Usual controls" group, click "Image". The control currently created appears under the mouse cursor.
2. Hover the first row: a green border appears, allowing you to view the available area.
3. Click the top left corner of first row in the Looper control. The Image control is created.



4. Display the description of Image control to modify the control characteristics (double-click the control).
5. Modify:
 - the control name: IMG_Visual.
 - the type of image: "From a database: memo (medium speed)". Indeed, the product image is stored in memo format in an item of Product file.
 - the display mode of image: "Homothetic centered" with the "High-quality display" option checked.
6. Validate.

► To create the Link control:

1. On the "Creation" pane, in the "Usual controls" group, click "Link".
2. Click in the Looper control, below the image to create the link.



3. Display the description of Link control to modify its characteristics (double click the control).
4. This control is named "LINK_View_Product".
5. Validate the description window of control.
6. Select the Link control and increase its width (with the handles) so that it occupies the row width.

► To create the Static control containing the product identifier:

1. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Simple static".
2. Click in the Looper control, on the right of image to create the static control.



3. Display the description of Static control to modify its characteristics (double-click the control).
4. This control is named "STC_Product".
5. Validate the description window of control.

To fill the controls found in the looper and to modify their characteristics for each row, we are going to create attributes.



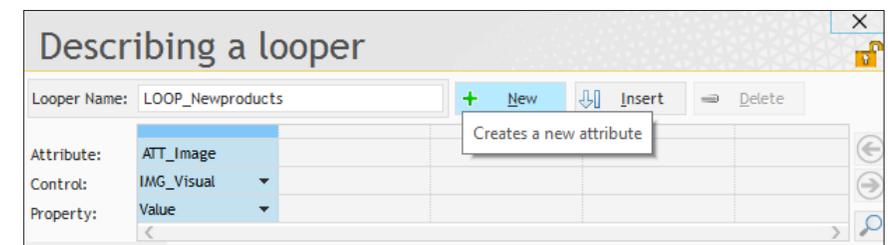
Notes

A Looper control includes:

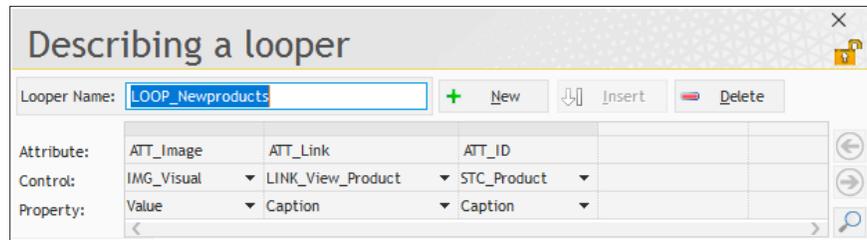
- controls, that are repeated on each row.
- attributes. An attribute is used to define the control characteristic that will be modified on each row. For example, if the value and color of PRICE control must change on each row, you will have to define two different attributes for the same control.

► To create attributes:

1. Display the description window of Looper control (double-click the control).
2. In the top section of window, a single attribute is created by default. We are going to define 3 attributes (one for each looper control).
3. Modify the attribute created by default:
 - Rename the attribute to "ATT_Image".
 - Select the "IMG_View" control.
 - Select the "Value" property. Indeed, the image value will be modified for each row.



4. Create a new attribute by clicking the "New" button:
 - Rename the attribute to "ATT_Link".
 - Select the "LINK_Product_View" control.
 - Select the "Caption" property. Indeed, the control caption will be modified for each row.
5. Create a new attribute by clicking the "New" button:
 - Rename the attribute to "ATT_ID".
 - Select the "STC_Product" control.
 - Select the "Caption" property. Indeed, the control caption will be modified for each row.



6. Validate the description window of Looper control.
7. Save the modifications ( or CTRL S).

The Looper control is now completed. We are now going to program the fill of Looper control.

Filling the looper

The Looper control is filled when initializing the page.

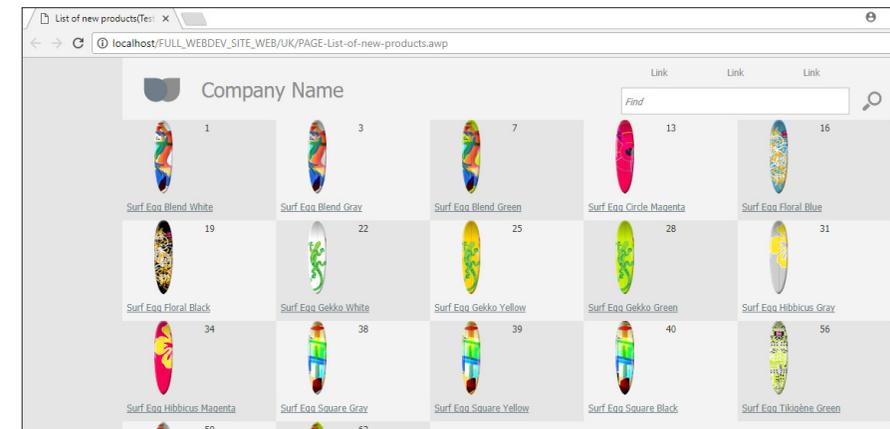
- ▶ To fill the Looper control:
 1. Display the processes associated with the page (click in the page and select "Code" from the popup menu or press the [F2] key).
 2. Type the following code in the initialization code of page:

```
FOR EACH Product WHERE NewProduct = True
  LooperAddLine (LOOP_NewProducts, Product.Visual, ...
  Product.Caption, Product.ProductID)
END
```

3. Let's study this code:
 - This code browses the Product file via the FOR EACH statement.
 - Only the records for which the "New" item is set to True are selected.

- **LooperAddLine** is used to add a new row into the Looper control. The row attributes are updated with the data of current record:
 - the value of "ATT_Image" attribute corresponds to the image found in the "Visual" item of "Product" file.
 - the caption of "ATT_Link" attribute has for value the content of "Caption" item of "Product" file.
 - the caption of "ATT_ID" attribute for the value the content of "ProductID" item of the "Product" file.
- 4. Save the modifications ( or CTRL S).
- 5. Close the code window.

- ▶ We are going to run the test of the page that was just created.
 1. Click  among the quick access buttons.
 2. The page is automatically displayed in the browser.



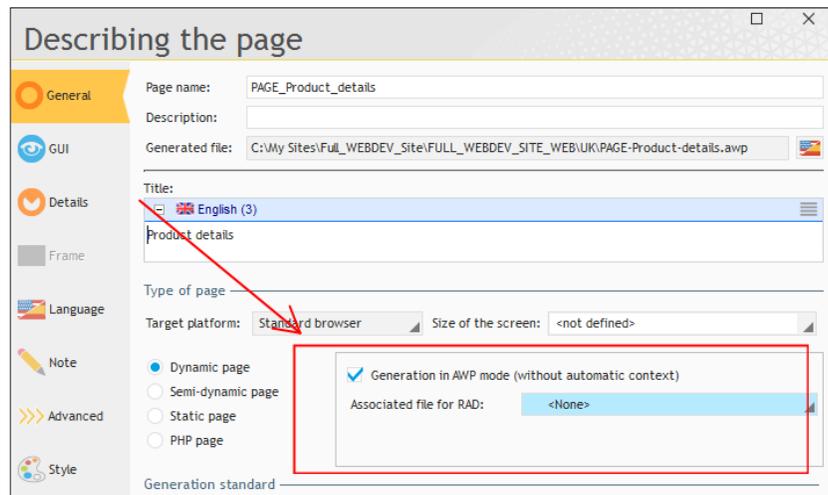
- ▶ Close the browser. The WEBDEV editor is redisplayed.

Viewing the details of a new product

Displaying the list of new products is a good thing. We are now going to create a "form" page used to see the product details. This page will be displayed when the user will click the link displaying the product in the Looper control.

Creating the page

- ▶ To create a page used to display the product details:
 1. Create a new page.
 - Click  among the quick access buttons.
 - The window for element creation is displayed: click "Page" then "Page".
 - The wizard for page creation starts.
 - In the "Based on a project template" section, choose "PAGETPL_Simple" and validate the wizard.
 2. The backup window of page is displayed. Type the title: "Product details". The name ("PAGE_Product_details") is automatically proposed.
 3. Validate.
- ▶ Like the page used to list new products, this page must be generated in AWP mode:
 1. Display the description window of page (select "Description" from the popup menu).
 2. Select the "Generation in AWP mode (without automatic context)".



3. Validate.
4. Save the modifications ( or CTRL S).

What should this page do?

This page is intended to display the characteristics of the product currently selected in the Looper control..

In our case, this page will be used to display the content of different items found in the "Product" data file.

First, we are going to indicate to the page the product that will be displayed. To do so, all you have to do is declare a parameter in the page. This parameter will be passed on the URL.

- ▶ To declare a parameter in the page:
 1. Press the [F2] key. The code editor displays the different page processes.
 2. Type the following code in the "Global declarations" process:

```
PROCEDURE MyPage (gnProductID is int on 8 bytes)
```

3. Let's study this code:
 - The PROCEDURE keyword in the "Global declarations" process is used to associate a procedure with the page opening.
 - The procedure is named "MyPage". At run time, this keyword will be automatically replaced by the name of current page.
 - The procedure expects the gnProductID variable (that is an 8-byte integer) in parameter. This variable corresponds to the product identifier that will be displayed in the page. The type of this variable is identical to the type of corresponding ProductID item described in the Product file. This parameter contains the value that will be found in the URL.
4. Close the code editor. We are now going to create the different page controls.

Creating controls

We are now going to create the edit controls used to display information about the selected product in the page.

In the previous part, we explained how to create controls by Drag and Drop from the analysis pane. In this section, we're going to create the controls one by one, then associate them with the corresponding item in the data file.

We are going to create the following controls:

- an Image control.
- simple static controls for the product caption, price and reference.
- a rich static for the product description.

► To create the Image control:

1. On the "Creation" pane, in the "Usual controls" group, click "Image".
2. Click in the page (top left corner for example).



3. Display the description of Image control to modify the control characteristics (double-click the control).

- This control is named "IMG_Product_Image".
- Its type is "From a database: memo (medium speed)".
- The display mode: "Homothetic centered without enlargement".



Notes With the "Homothetic centered without enlargement" display mode, the image size will homothetically adapt to the area defined for the image. The proportions will be respected and the image will not be enlarged.

4. Validate the description window of control.

► To create the simple Static control used to display the product caption:

1. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Simple static".
2. Click in the page (on the right of image for example).



Notes To simplify the positioning of controls, press the [F7] key. Pressing this key allows us to see the area taken by the control, pressing this key a second time displays a light border around the control (only in edit mode). This allows you to view the border of control and to easily position the controls in relation to the other ones. This feature is very useful for the controls without border.

3. Rename the Static control: "STC_Title".

4. Enlarge the control (with the sizing handles).

► We are going to display the product description in a rich Static control (called Rich Text Area). To create this control:

1. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Rich Text Area". You also have the ability to click the "Text" icon.
2. Click inside the page (to the right of image for example, below the STC_Title control).

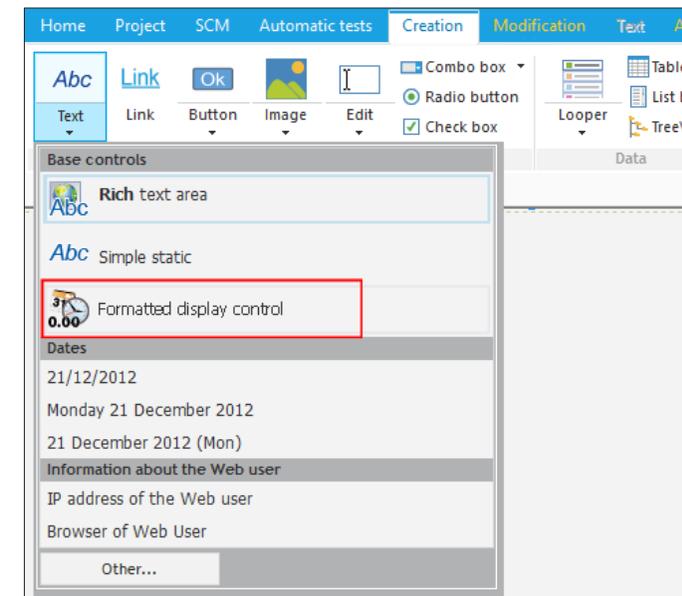


3. Rename the control: "RTA_Description".

We are going to display the product price in a formatted display control. This control can be used to display dates, times and currency values while respecting the corresponding display format.

► To create a formatted display control:

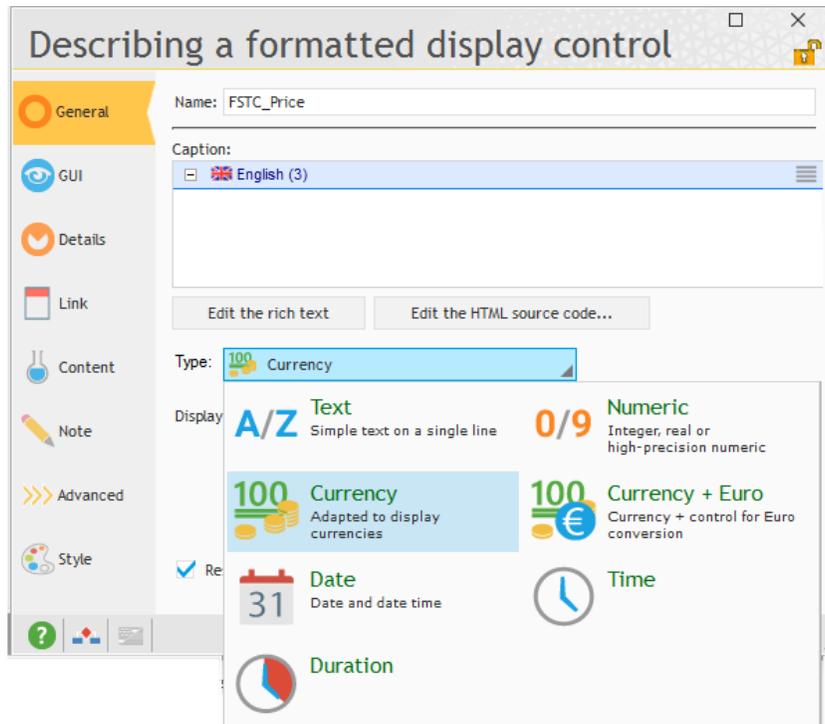
1. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Formatted Display control".



2. Click in the page, below the product description: the control is automatically created.

► We are going to modify the characteristics of this control:

1. Display the description window of the control that was just created: double-click the control for example.
2. In the "General" tab:
 - Modify the control name: "FSTC_Price",
 - Delete the caption.
 - Modify the control type: select "Currency".



3. Validate.

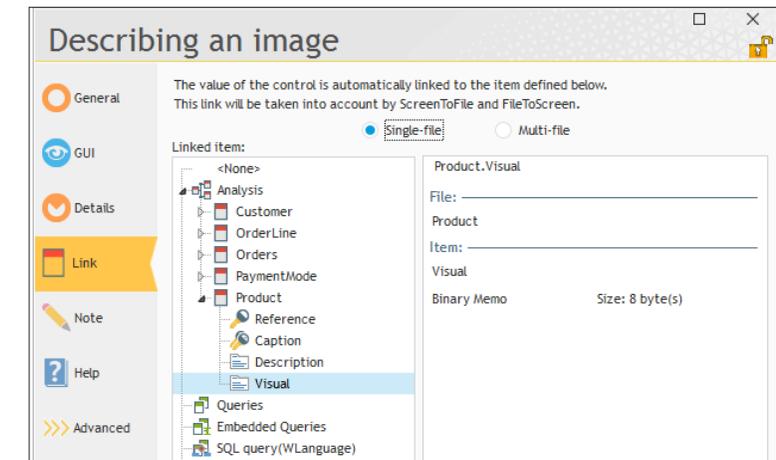
► Create a simple Static control to display the product reference. This control is named "STC_Reference" and its caption is "Reference". This control is positioned below the price.

All the necessary controls have been created. We are now going to associate each control with the corresponding item. This association is performed in the "Link" tab of the description window of the control.

► To link the "IMG_Product_Image" control with the corresponding item:

1. Select the "IMG_Product_Image" control.
2. Display the popup menu (right mouse click) and select "Description".
3. Display the "Link" tab of the description window. This tab allows you to see that the current control is linked to no item. We're going to link it to the View item of the Product data file.

4. In the "Linked item" area, expand "Analysis" then expand "Product". The list of items from the Product data file comes up.
5. Select the "View" Item of the "Product" data file.



6. Validate the description window.

- Perform the same operation to create the following links:
 - STC_Title control linked to the "Caption" item of "Product" file.
 - RTA_Description control linked to the "Description" item of "Product" file.
 - FSTC_Price linked to the "PriceBT" item of "Product" file.
 - STC_Reference control linked to the "Reference" item of "Product" file.
- Save the page.

Displaying data in the page

The "Form" page must display the product that is selected in the Looper control. In the page code, we are going to type the code used to:

- find the product to display.
- display the data in the page.
- To display data in the page:
 1. Press the [F2] key. The code editor displays the different page processes.
 2. In the "Global declarations" process, type the following code after the code that was written beforehand:

```
HReadSeekFirst (Product, ProductID, gnProductID)
IF HFound (Product) THEN
    FileToScreen ()
END
```

3. Let's study this code:

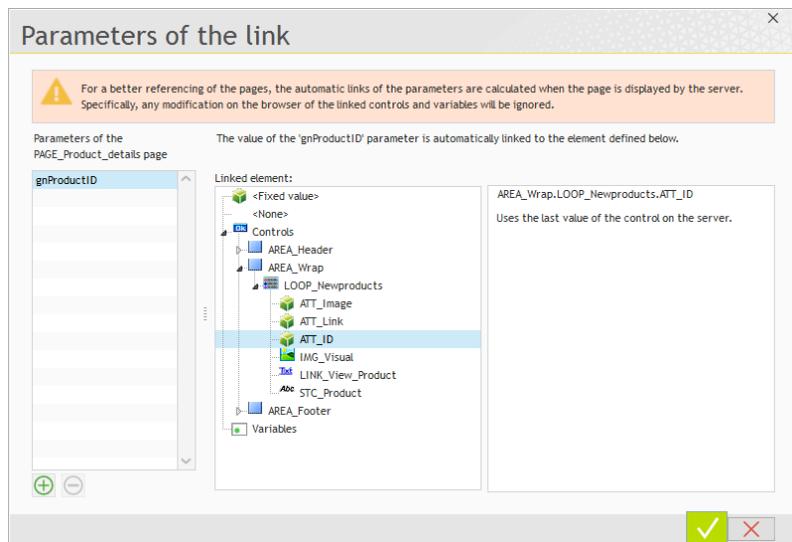
- **HReadSeekFirst** is used to find the first record of Product file for which the ProductID item is equal to the value of gnProductID, corresponding to the parameter passed to the page.

- **HFound** is used to check whether a record was actually found. This function is mainly required for the multi-user sites. It is used to avoid errors caused by the deletions performed by the other users.
 - **FileToScreen** is used to display in the controls the data found in the data file, for the current record. In our case, the current record corresponds to the record found by **HReadSeekFirst**.
4. Close the code editor.

Displaying the form from the list of products

Let's see how to display the form of selected product in the list of products. The principle is straightforward: the user will select the product in the Looper control and he will display the details via a link. This link already exists in the looper.

- First of all, we are going to modify the "PAGE_List_of_new_products" page in order for the link to open the description page of product.
 1. Position on the "List of products" page: click the "PAGE_List_of_new_products" button found in the button bar.
 2. Display the description window of Link control: double-click the control.
 3. In the "General" tab of description window, in the "Action" area, select "Display the PAGE_Product_details page".
 4. Click the "Parameters" button. We are going to define the parameter that will be passed to the page to display the details of selected product.



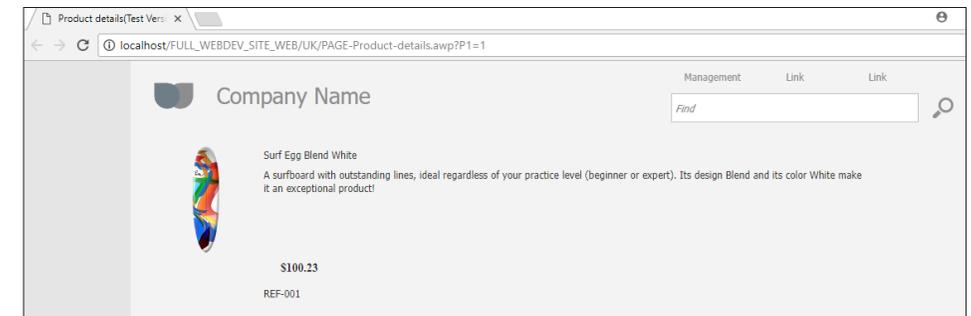
5. In the window that is displayed, you will find the name of the parameter that was declared in the "Global declarations" process of page (gnProductID). All you have to do is select the control containing the parameter value. In our case, the product identifier is found in the "ATT_ID" attribute.
6. Select "ATT_ID" and validate.



Notes

Caution: In a looper, the value is not contained in the control but in the attribute that is associated with it for the value characteristic.

7. Validate the description window of Link control.
 8. Save the modifications (or CTRL S).
- We are going to run a test to make sure that everything is operating properly.
 1. Position (if necessary) on the "PAGE_List_of_new_products" page.
 2. Click among the quick access buttons.
 3. The page is automatically displayed in the browser.
 4. Click the link to display the product details.



- Close the pages to stop the test.

Managing styles

To improve our page, we are going to modify the styles of controls. If you are working with a graphic designer or if a style book was defined for your site (style book of the company for example), the styles have been defined in a CSS style sheet beforehand. We are going to import this style sheet into our WEBDEV project in order to use these styles.



Notes

WEBDEV proposes two types of styles:

- the WEBDEV styles.
- the CSS styles.

A WEBDEV style is a set of CSS styles used to define a global style on high-level objects.

For example, a WEBDEV style for an edit control will contain two CSS styles:

- a style for the caption.
- a style for the text in the input area.



Notes

Reminder: When creating the site, you have the ability to choose a skin. The skins are used to standardize the visual aspect of a site and they allow you to easily change style.

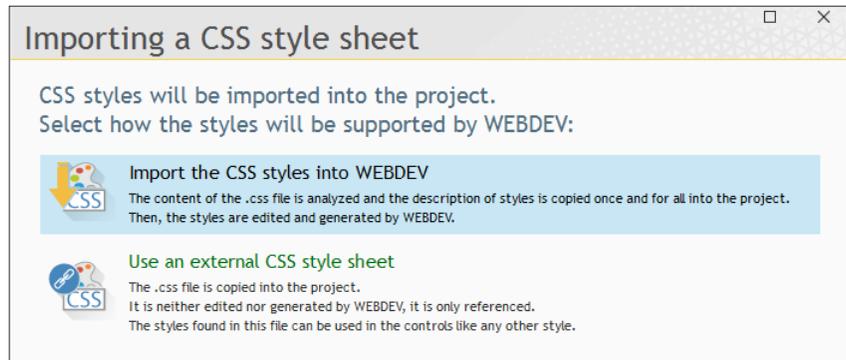
The skin provides the images, the fonts, the texture, the shape of buttons and the styles available for the project.

You create sites with a professional style from the styles supplied in standard. Importing CSS styles is used for example to add a new style found on Internet.

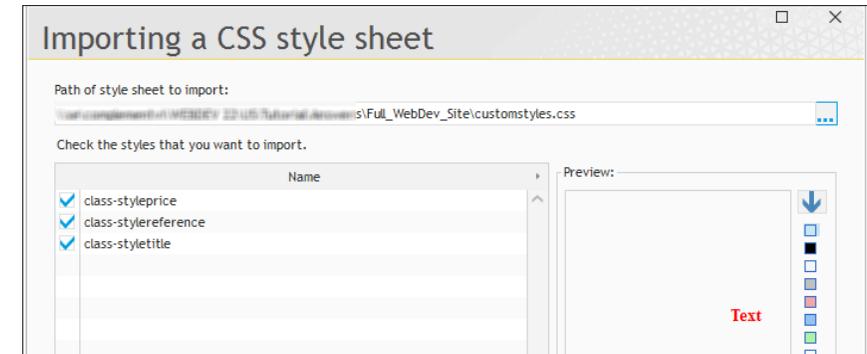
Importing a CSS style sheet

► To import the CSS style sheet into the project:

1. In the ribbon, on the "Project" pane, in the "Project" group, expand "Import" and select "A CSS style sheet".
2. The import wizard starts and proposes two options:



- Import the CSS styles: This option imports the CSS styles into the WEBDEV project. Then, the styles can be modified in WEBDEV.
 - Use an external CSS style sheet: This option allows you to use an existing style sheet. This option must be chosen when a style sheet is defined for a company and when it must be shared between several sites (company style book).
3. Select "Import the CSS styles into WEBDEV" and click the yellow arrow.
 4. Select the "customstyles.css" file. This file is found in the "\\Tutorial\Exercises\Full_WEBDEV_Site" directory. The various styles found in the style sheet are displayed.



- A preview of the selected site is displayed on the right.



Tip

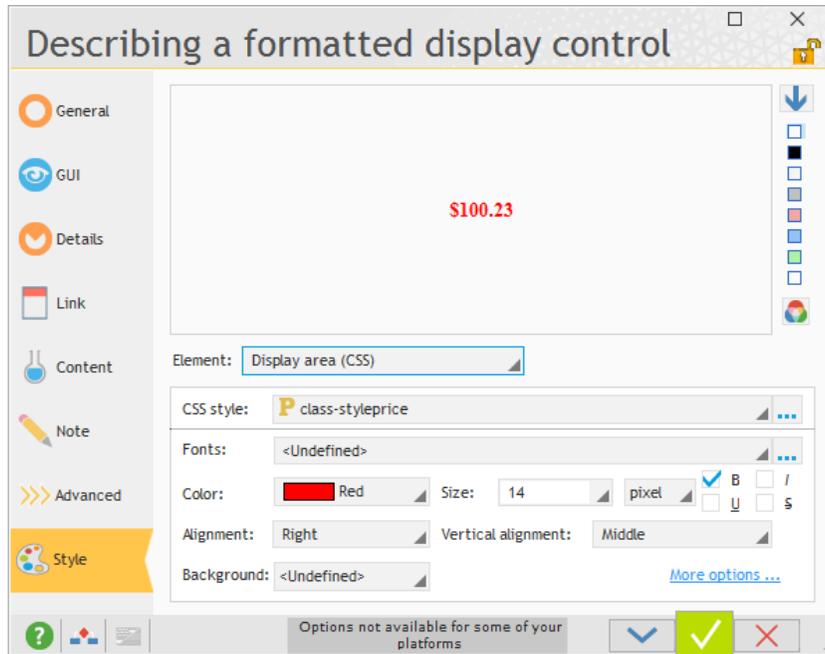
The rectangles of colors found in the preview are used to change the background color. This allows you to get a preview on another color than white.

- The "class-" prefix means that it is a CSS class. This attribute means that this style will be applied to the elements that have the HTML "class" attribute.
5. Keep all the proposed styles and validate. The styles are imported and they are available in WEBDEV.

Applying the CSS styles

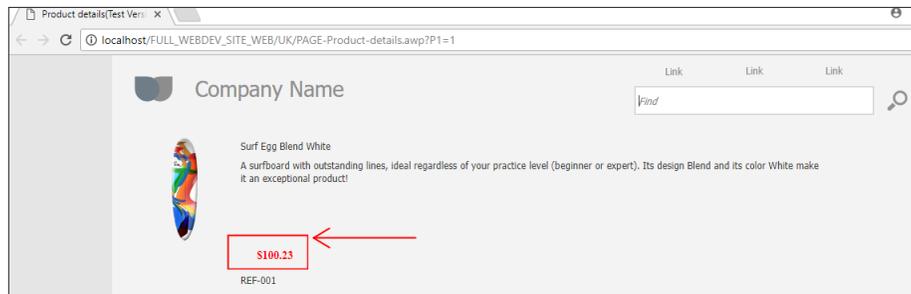
We are now going to apply the different styles to the controls found in the "PAGE_Product_details" page.

1. Position on the "PAGE_Product_details" page.
2. Display the description window of "FSTC_Price" control that displays the product price (double-click the control).
3. In the "Style" tab:
 - Select the "Display area (CSS)" element.
 - Select the "class-styleprice" style in the "CSS style" list:



4. Validate. This style is automatically applied to the control.
5. Enlarge the control if necessary (with the sizing handles) to adapt its size to its content.
6. Save the modifications ( or CTRL S).

- ▶ We are going to run a test to make sure that everything is operating properly.
 1. Position (if necessary) on the "PAGE_List_of_new_products" page.
 2. Click  among the quick access buttons.
 3. The page is automatically displayed in the browser.
 4. Click the link to display the product details. The price is now displayed in red.



- ▶ Close the pages to stop the test.

Linking the Internet site and the Intranet site

Until now, we have created Intranet pages used to manage the products and Internet pages allowing the Web users to see the new products.

We are now going to link these pages by proposing, from the Internet pages, a "Management" link used to access the Intranet pages.

To do so, we are going to modify the template for the Internet pages.



Caution!

The following operations can be performed only if you have followed part 3 of this tutorial regarding the creation of Intranet pages.

- ▶ To create a "Management" link:
 1. Position on the "List of new products" page.
 2. Display the popup menu of one of the links found at the top of the page and select "Open the template".
 3. Display the description of Link control (double-click the control).
 4. Modify the control caption: Management.



Tip

In your programs, we advise you to choose meaningful names for your elements (in this case, the link is named "LINK_Management"). Therefore, the code is more readable.

5. Select the operation on the controls: "None".
6. Validate.



- ▶ Now let's type the code associated with this link:
 1. Display the processes associated with the Link control ([F2] on the control).
 2. Type the following code in the browser click code:

```
DynamicSiteDisplay("Full_WEBDEV_Site")
```

3. In this code, **DynamicSiteDisplay** is used to display the first dynamic page of the project (in our case, it will be the "PAGE_List_of_Products" page).
4. Close the code editor.
5. In the orange bar of template, click  to propagate the template modifications to all the pages that use this template.

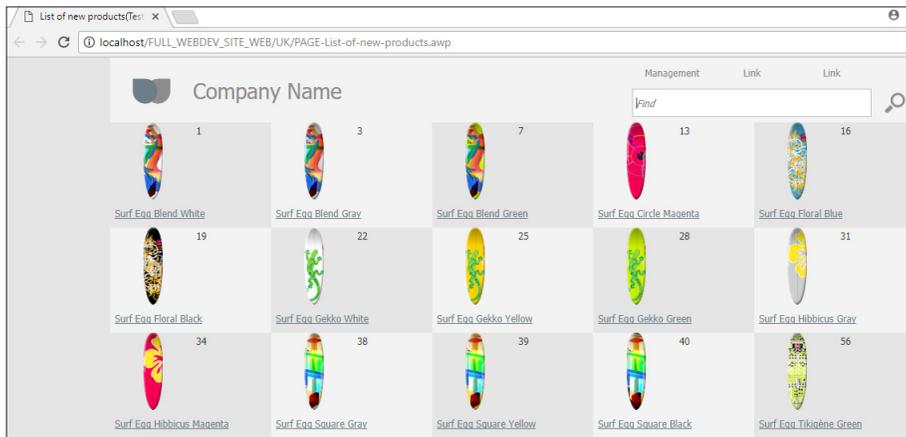
6. In our case, the two AWP pages of the project are proposed. Validate the template update window.
7. Close the page template.

► We are now going to modify the "PAGE_List_of_Products" page so that it can be opened by **DynamicSiteDisplay**:

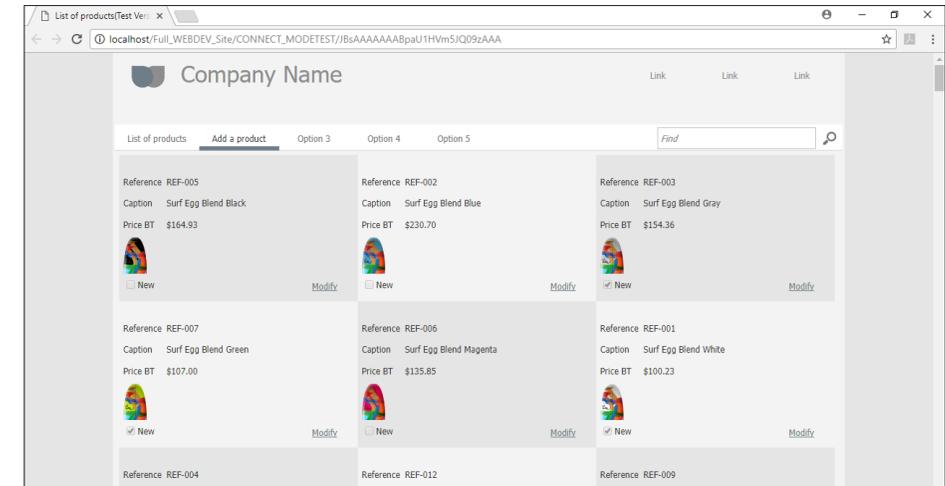
1. Display the "PAGE_List_of_products" page in the editor.
2. Display the description window (select "Description" from the popup menu).
3. In the "Advanced" tab, check "Allow the access via DynamicSiteDisplay()".



4. Validate.
- Now we're going to start the project via the AWP pages.
1. Position on the "List of new products" page.
 2. Click among the quick access buttons.
 3. The page is automatically displayed in the browser.



4. Click the "Management" link. The page for product management is displayed.



- Close the pages to stop the test.

Creating a page used to display the general sales conditions

We are now going to create a type of page that is often used in the e-commerce sites: a page displaying the general sales conditions. To do so, we will be using the Rich Text Area control, allowing us to quickly format a text.

Creating the page

- To create a page used to display the general sales conditions:
1. Create a new page.
 - Click among the quick access buttons.
 - The window for creating a new element is displayed: click "Page" then "Page".
 - The wizard for page creation starts.
 - In the "Based on a project template" section, choose "PAGETPL_Simple" and validate the wizard.
 2. The backup window of page is displayed. Type the title: "General sales conditions". The name ("PAGE_General_sales_conditions") is automatically proposed.
 3. Validate.

Creating the control

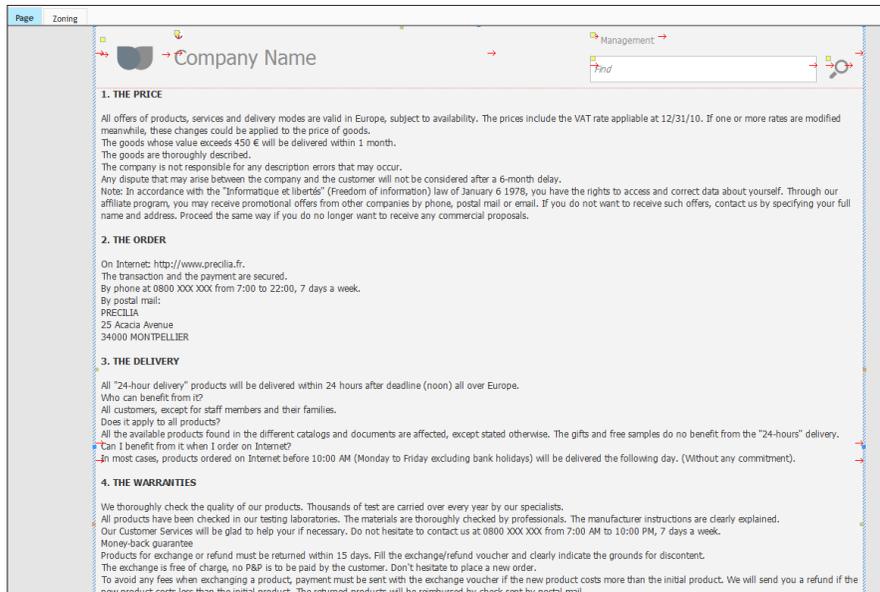
- To create a rich Static control:
1. In the ribbon, on the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Rich Text Area".
 2. Click in the page (top left corner for example). The control is automatically created.

► This type of control automatically adapts its height to its content. We are going to define a specific width:

1. Select the control that was just created.
2. Increase the control in width (via the handles) until it occupies the entire page width.

► Now, we are going to display the general sales conditions in this control. The file containing the text was prepared beforehand.

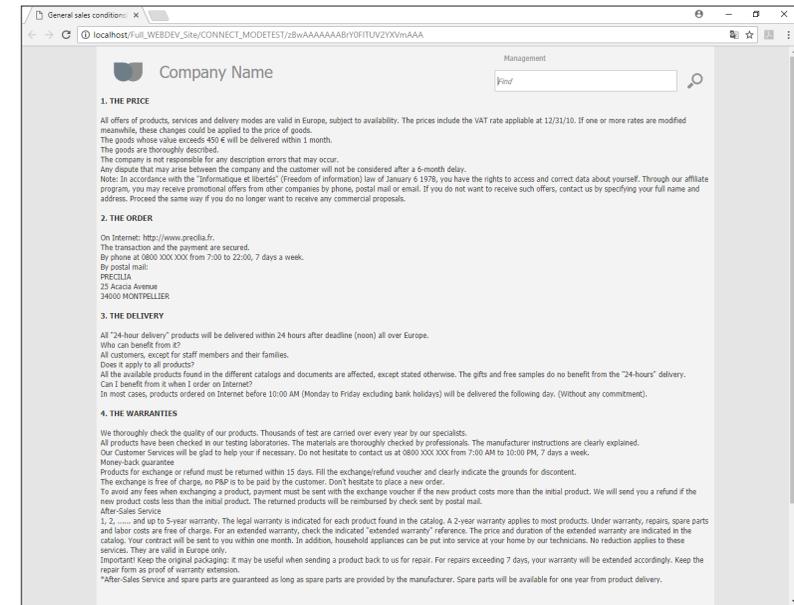
1. Copy the text found in the Gsc.txt file. This file is found in the "\\Tutorial\Exercices\Full_WEBDEV_Site" directory of setup directory of WEBDEV.
2. Select the "Rich Text Area" control in the page.
3. Press the [SPACE] key on the keyboard: the control becomes editable. A yellow border is displayed around the control.
4. Replace the existing text by the text found in the clipboard (CTRL V). The text comes up.
5. Perform some modifications in the text:
 - Add Carriage Returns before and after the titles.
 - Write the titles of different points in bold.



6. Try the different options: all the options available for the text are available in the ribbon.

7. Click  among the quick access buttons.

8. The page is automatically displayed in the browser.



Conclusion

The last two parts have allowed us to see the specific features of an Intranet site and the ones of an Internet site.

In the next part, we will present several topics that can be included in an Internet site and in an Intranet site:

- Multicriteria search,
- Printing,
- Managing the multilingual feature, ...

PART 5

Let's continue
the development



LESSON 5.1. MULTICRITERIA SEARCH

This lesson will teach you the following concepts ...

- Creating a query with parameters.
- Creating the interface for selecting the search criteria.
- Passing parameters to a query.
- Displaying the result of a query in a Table control.
- Optimizing the page refresh by enabling Ajax



Estimated time: 1h30 mn

Overview

In the previous parts, we have created a WEBDEV project used to develop a site containing both an Internet section and an Intranet section.

We will still be working with the "Full_WEBDEV_Site" project that was used in parts 3 and 4.

► If the "Full_WEBDEV_Site" project was not opened in the previous lesson:

1. Close (if necessary) the current project to display the home window.
2. In the home window, click "Tutorial" and select "Full WEBDEV site (With pages)".



Answer

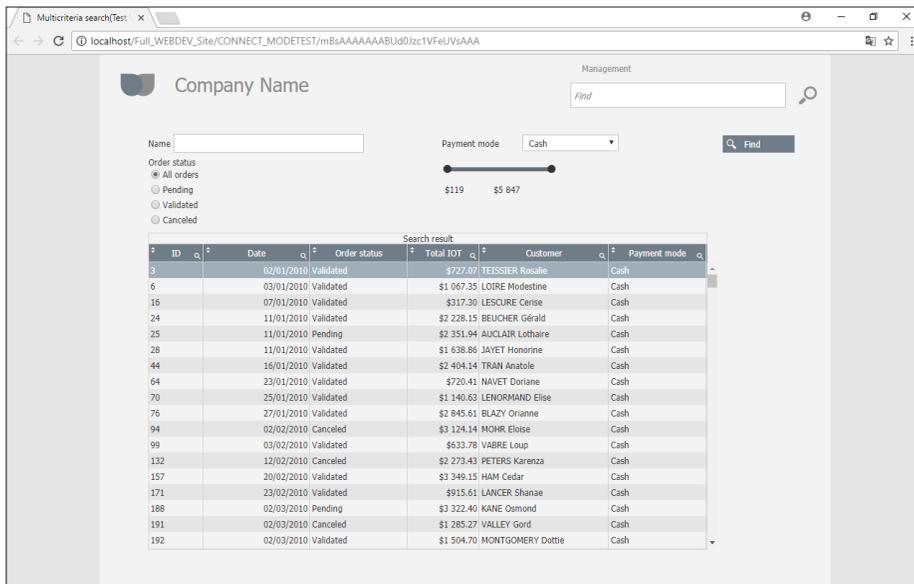
A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site".

In this lesson, we will give the user the ability to perform a multicriteria search.

In our example, this search will be performed on the "Orders" file. The user will be able to select:

- the customer name,
- the order status,
- the payment mode,
- the order price.

The interface of the "PAGE_Multicriteria_search" page will be as follows:



ID	Date	Order status	Total IOT	Customer	Payment mode
3	02/01/2010	Validated	\$272.07	TEISSIER Rosalie	Cash
6	03/01/2010	Validated	\$1 067.35	LOIRE Modestine	Cash
16	07/01/2010	Validated	\$317.30	LESCURE Cécile	Cash
24	11/01/2010	Validated	\$2 228.15	BEUCHER Gérard	Cash
25	11/01/2010	Pending	\$2 351.94	AUCLAIR Lothaire	Cash
28	11/01/2010	Validated	\$1 638.86	JAVET Honorine	Cash
44	16/01/2010	Validated	\$2 404.14	TRAN Anatole	Cash
64	23/01/2010	Validated	\$720.41	NAVET Doriane	Cash
70	25/01/2010	Validated	\$1 140.63	LENORMAND Elise	Cash
76	27/01/2010	Validated	\$2 845.61	BLAZY Orienne	Cash
94	02/02/2010	Cancelled	\$3 124.14	MOHR Eloise	Cash
99	03/02/2010	Validated	\$633.78	VABRE Loup	Cash
132	12/02/2010	Cancelled	\$2 273.43	PETERS Karenza	Cash
157	20/02/2010	Validated	\$3 349.15	HAM Cedar	Cash
171	23/02/2010	Validated	\$915.61	LANCER Shanae	Cash
188	02/03/2010	Pending	\$3 322.40	KANE Osmond	Cash
191	02/03/2010	Cancelled	\$1 285.27	VALLEY Gord	Cash
192	02/03/2010	Validated	\$1 504.70	MONTGOMERY Dottie	Cash

This interface includes:

- controls used to select the search criteria.
- buttons used to start the search or to print the result.
- a Table control used to display the search result. This Table control is based on a query. This query will be used to select the records to display. The Table control will list the search results.

The first step consists in creating the query for selecting records.



Notes

What is a select query?

A selection query is a query that will only catch the records that meet specific criteria.

This type of query is called a select query because the SELECT command is used in SQL language.

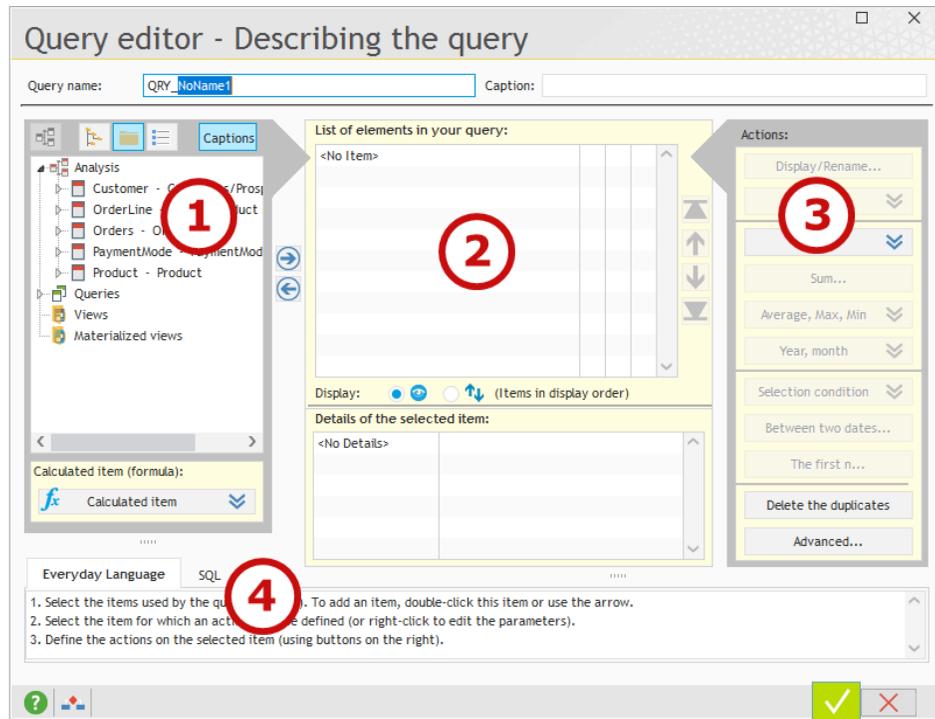
Creating the query used to find orders

Creating the query

► The query editor will be used to create the query.

1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Query". The wizard for query creation starts.
2. Select the "Select" option. Go to the next step.
3. The description window of query is displayed.

Let's stop for a minute on the interface of the query editor:



This interface includes:

1. an overview of the analysis (the description of the project's database).
2. the items that must be included in the query
3. the options for selecting records
4. the query in SQL code or in natural language

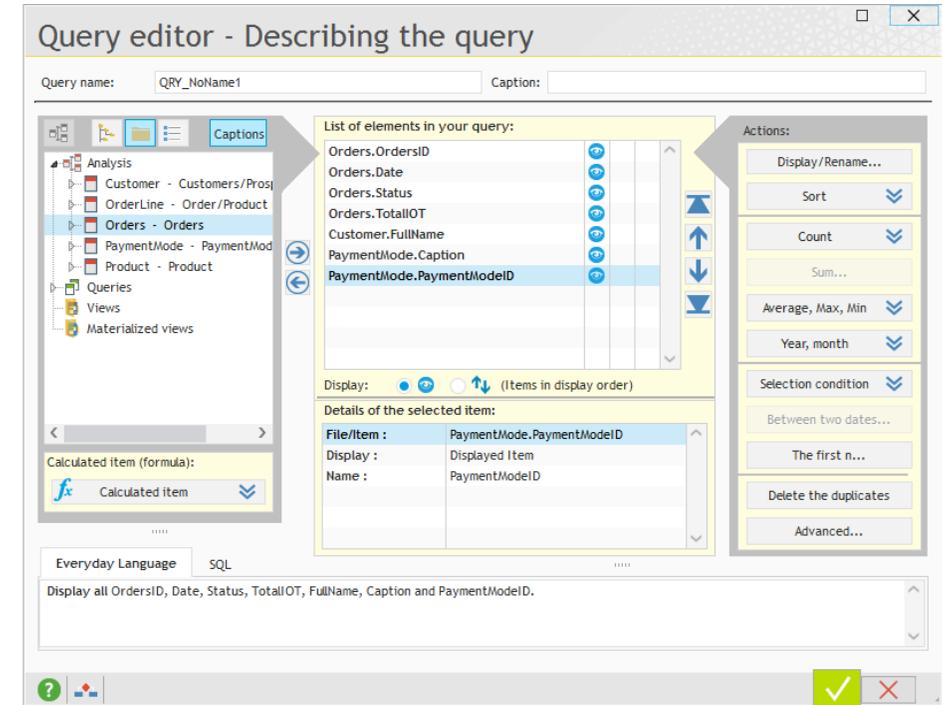
► To build the query, we are going to select the elements that will be displayed in the result.

1. Double-click the items found in the analysis on the left of the description window. The items taken into account are displayed in the middle of the screen.

We want to display:

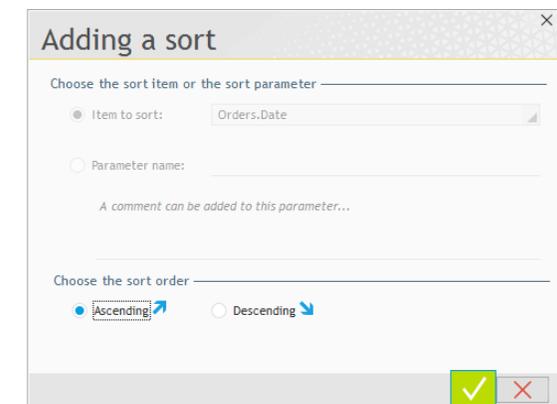
- information regarding the order. In the "Orders" file, double-click the items: OrdersID, Date, Status and TotalIOT.
- information regarding the customer who placed the order. In the "Customer" file, double-click the "FullName" item.
- information regarding the payment mode of order. In the "PaymentMode" file, double-click the "Caption" and "PaymentModeID" items.

The description window of query is as follows:



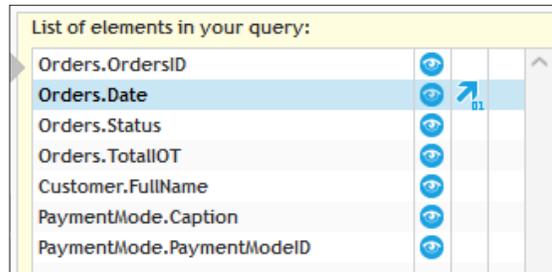
2. The data will be sorted by date.

- Select the "Orders.Date" item, then click the "Sort" button and select "Sort on the selected item".



- In the window that is displayed, specify an ascending sort on the item and validate.

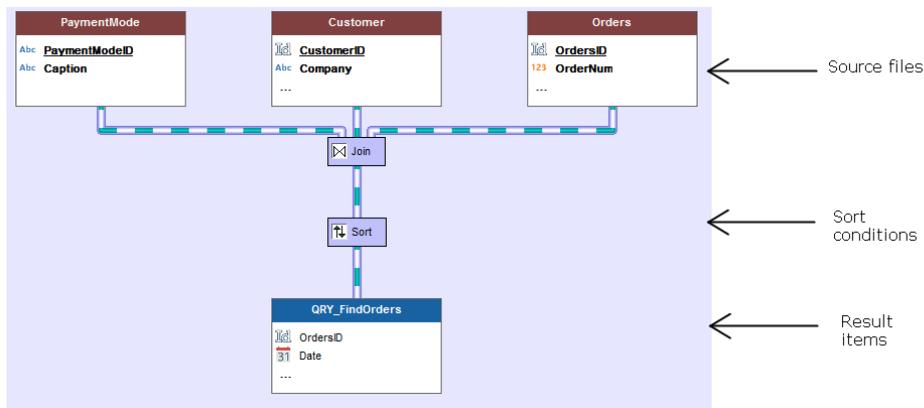
3. A red arrow with the number 01 appears on the right of "Orders.Date" item. This arrow indicates that an ascending order is performed on this item. The number "01" indicates that this sort will be performed first.



4. Give a name to the query: type "QRY_FindOrders" instead of "QRY_NoName1" in the "Query name" area:



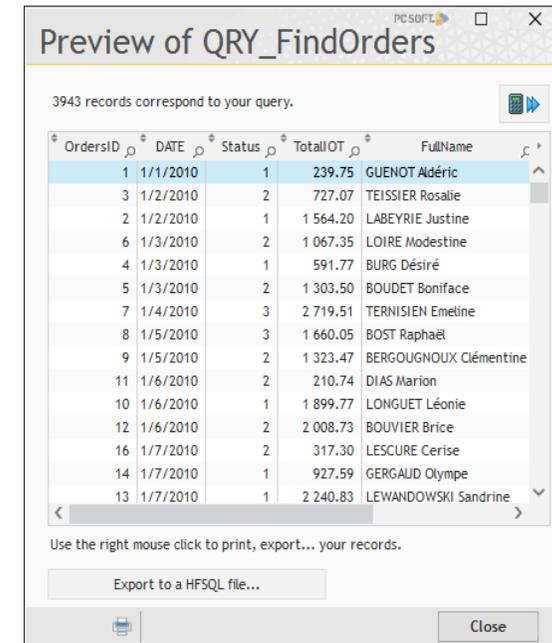
- 5. Validate the description window of query (green button at the bottom of the screen).
- 6. Validate the backup window of query.
- 7. The graphic query representation is displayed:



Query test

Like all the elements of a WEBDEV project, you have the ability to immediately run the test of the query that was just created:

- 1. Click .
- 2. The result is displayed in a window:



Notes
 A popup menu is displayed when a right click is performed on the query result. The result can be exported to:

- an XLS file (Excel).
- an XML file (eXtensible Markup Language).
- a Word or Open Office file.

- 3. The result lists ALL the orders. In our case, we want to display the orders corresponding to the search criteria only. To do so, we must use a query with parameters.
- 4. Close the window.

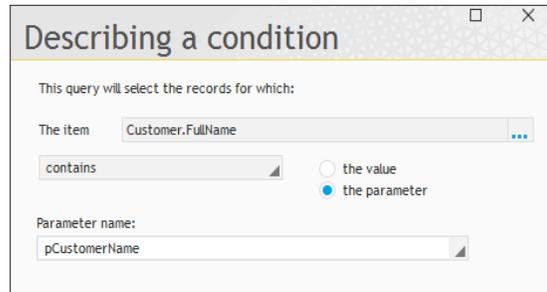
Using parameters in the query

In our example, the user will be able to select a value for the following search criteria:

- Customer name.
- Order status.
- Payment mode of order.
- Order price.

We must modify the query in order for these search criteria to correspond to the query parameters.

- ▶ To define the query parameters, display the description window of query: double-click the background of graphic query representation (or select "Description" from the popup menu).
- ▶ To manage the "Customer name" parameter:
 1. Select the Customer.FullName item in the middle of the screen.
 2. Expand the "Selection condition" button and select "New condition".
 3. In the window that is displayed, we are going to specify that the selection condition corresponds to a parameter:



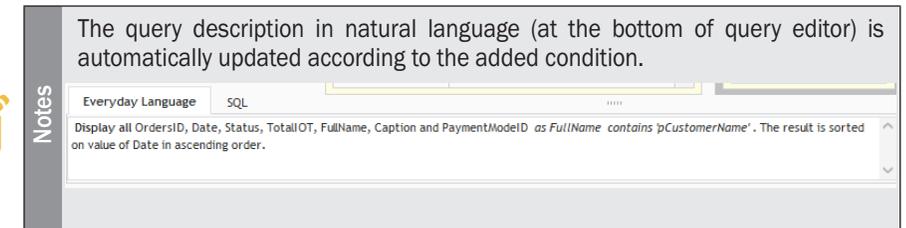
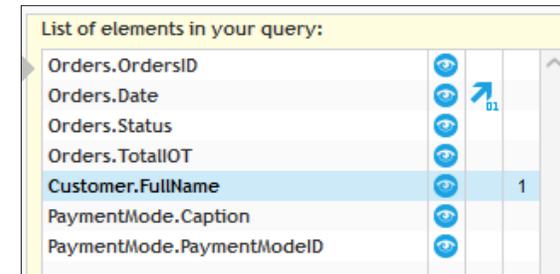
- Select "Contains".
- Check "the parameter".
- Specify the parameter name: "pCustomerName".



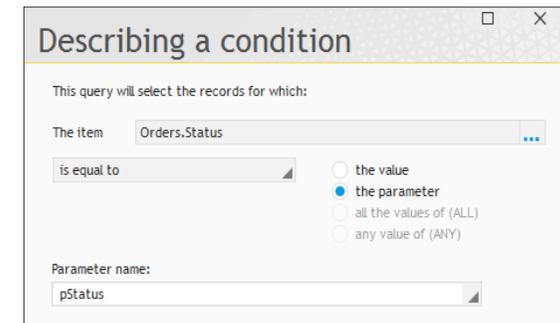
Notes

We advise you prefix the parameters of queries with the letter "p". This allows you to easily find them in the code editor. When you're looking for a query parameter, simply enter 'p' and the editor's auto-complete feature will offer you all the parameters.

4. Validate the description window of condition. The number "1" appears on the right of the "Orders.CompleteName" item, indicating that a selection condition has been defined.



- ▶ Let's now see the second parameter: the order status:
 1. Select the Orders.Status item (in the middle of the screen).
 2. Expand the "Selection condition" button and select "New condition".
 3. In the window that is displayed, we are going to specify that the selection condition is equal to a parameter:



- Select "Is equal to".
- Select "the parameter".
- Specify the parameter name: "pStatus".
- 4. Validate the description window of condition. The number "1" appears on the right of "Orders.Status" item, indicating that a selection condition was defined.

► We are now going to define a condition on the payment mode. The "PaymentMode.PaymentModelID" item must not be displayed in the result of the query but a condition must be applied to it. To do so, it will be made invisible.

1. To avoid displaying the "PaymentMode.PaymentModelID" item in the result:
 - Click the eye (👁️) corresponding to the "PaymentMode.PaymentModelID" item in the list of query elements (in the middle of the screen).
 - In the menu that is displayed, select "Don't display".
2. To define a selection condition on the "PaymentMode.PaymentModelID" item:
 - Select the "PaymentMode.PaymentModelID" item (in the middle of the screen).
 - Expand the "Selection condition" button and select "New condition".
 - In the window that is displayed, specify that the selection condition corresponds to a parameter:
 - Select "Is equal to".
 - Select "the parameter".
 - Specify the parameter name: "pPaymentModelID".
3. Validate the definition of selection condition.

Orders.OrdersID	👁️		
Orders.Date	👁️	👉	
Orders.Status	👁️		1
Orders.TotalIOT	👁️		
Customer.FullName	👁️		1
PaymentMode.Caption	👁️		
PaymentMode.PaymentModelID	👁️		1

► The last selection condition to define affects the order amount. In fact, we are going to define two selection conditions in order to specify a minimum price and a maximum price.

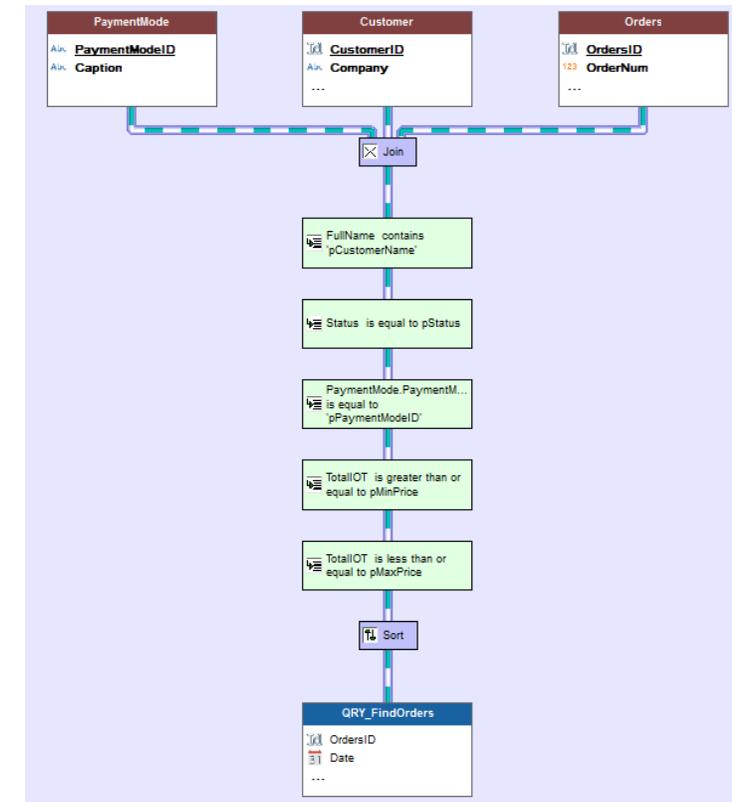
1. Select the "Orders.TotalIOT" item in the list of query elements.
2. Display the popup menu of item (right mouse click) and select "Selection condition .. New condition".
3. In the window that is displayed:
 - Select the condition "Is greater than or equal to".
 - Click "to parameter".
 - Specify the parameter name: "pMinPrice".
4. Validate the definition of selection condition.
5. Define once again another condition on the same "Orders.TotalIOT" item: display the popup menu of item (right mouse click) and select "Selection condition .. New condition".
6. In the window that is displayed:
 - Select the condition "Is less than or equal to".
 - Click "to parameter".
 - Specify the parameter name: "pMaxPrice".
7. Validate the definition of selection condition.



Notes

The definition of the two selection conditions on the item (is greater than or equal to and is less than or equal to) can be replaced by a single condition "Is included between". However, using a selection condition such as "Is found between" means that the two values must be typed (which is not the case if two different conditions are defined).

8. Validate the description window of query. The query graph is modified to take into account the selection conditions that have been defined.



9. Save the query by clicking among the quick access buttons.

Test of query with parameters

► To run the test of query with parameters:

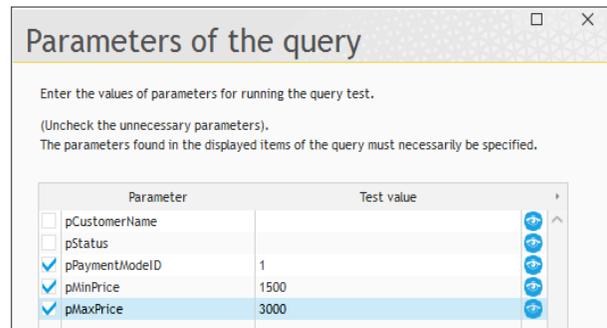
1. Click .
2. A window is displayed, allowing you to type the different query parameters.



Notes

The parameters can be ignored by unchecking the box in front of their name.. In this case, the selection condition associated with the parameter is ignored. For example, if the pCustomerName parameter is ignored, the orders placed by all customers are taken into account by the query.

Note: If all the parameters are unchecked, the query returns the entire result.



Parameter	Test value
<input type="checkbox"/> pCustomerName	
<input type="checkbox"/> pStatus	
<input checked="" type="checkbox"/> pPaymentModelID	1
<input checked="" type="checkbox"/> pMinPrice	1500
<input checked="" type="checkbox"/> pMaxPrice	3000

3. Type the following data:

- Uncheck the pCustomerName and pStatus parameters.
- Select the pPaymentModelID parameter. In the lower section of the screen, enter "1".
- Select the pMinPrice parameter. In the lower section of the screen, enter "1500".
- Select the pMaxPrice parameter. In the lower section of the screen, enter "3000".

4. Validate the window. The query result corresponding to the specified parameters is displayed.

5. Close the window.

We are now going to create the page used to:

- specify the parameters of this query,
- run this query,
- display the result of the query.

Creating the page proposing the multicriteria search

Creating a page

► To create a page used to display the result of the multicriteria search:

1. Create a new blank page.
 - Click  among the quick access buttons.
 - The window for creating a new element is displayed: click "Page" then "Page".
 - The wizard for page creation starts.
 - In the "Based on a project template" section, choose "PAGETPL_Simple" and validate the wizard.
2. The backup window of page is displayed.
3. Type the title: "Multicriteria search". The name ("PAGE_Multicriteria_search") is automatically proposed.
4. Validate.

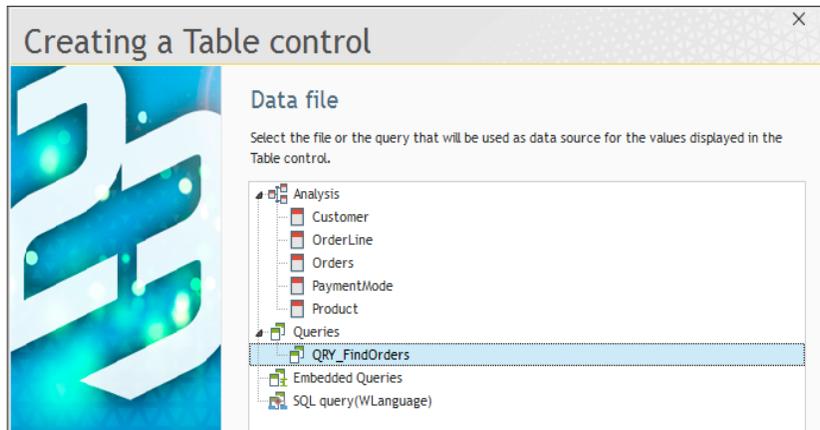
Creating the controls used to configure the criteria and to display the result

We are going to create a Table control based on the query then the different controls allowing the user to select the search criteria.

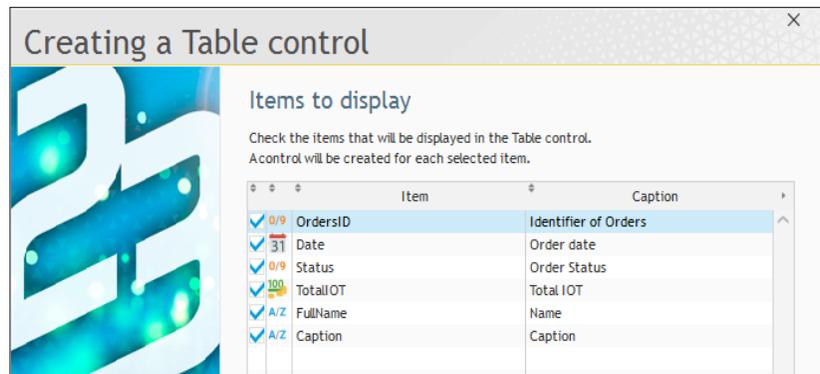
Creating the Table control

► To create the Table control used to display the search result:

1. Create a Table control: on the "Creation" pane, in the "Data" group, expand "Table" and select "Table".
2. Click in the "PAGE_Multicriteria_search" page: the wizard for creating the Table control starts.
3. The Table control will be based on the "QRY_FindOrders" query (that was created beforehand). Select "Display the data found in a file or in an existing query". Go to the next wizard step.
4. Select the query that will be used as data source for the Table control:
 - Expand the "Queries" group.
 - Select the "QRY_FindOrders" query.
 - Go to the next wizard step.



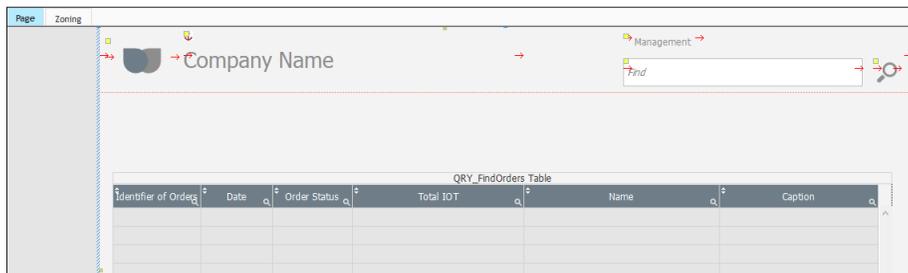
5. Select all the proposed items: all of them will be displayed in the Table control. Go to the next wizard step.



6. Keep the default options in the different wizard screens and validate the creation of Table control.

7. The Table control is automatically created in the page.

8. Modify (if necessary) the position of Table control so that it is entirely displayed in the page.



► For better readability, we are going to modify the description of Table control.

1. Display the description of Table control (double-click the control).

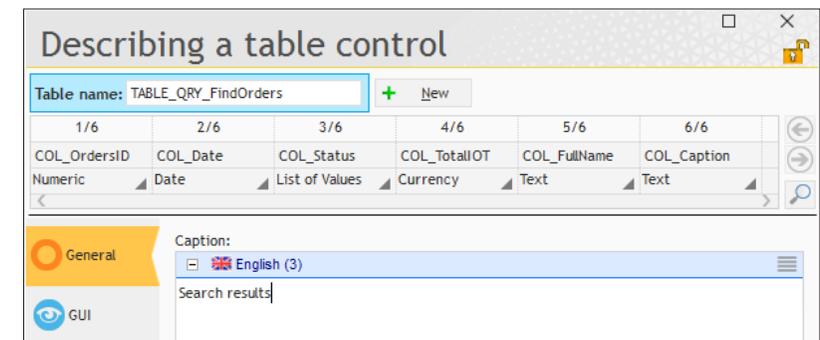


The description window of a Table control includes two sections:

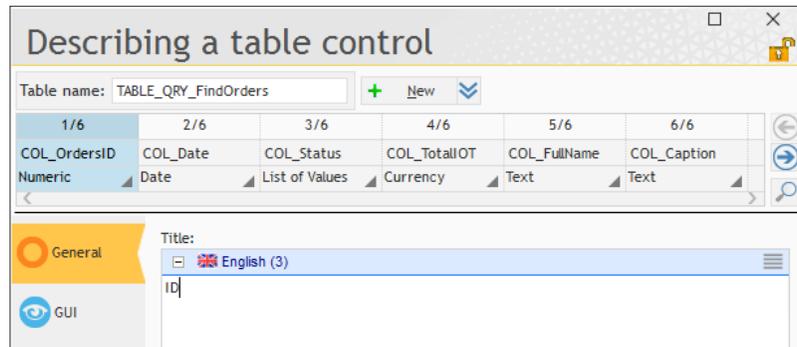
- the upper section, presenting the control name, the name of columns as well as their type.
- the lower section, containing the different description tabs.

If the name of Table control is selected, the lower section presents the characteristics of Table control.
If a column is selected, the lower section presents the characteristics of columns.

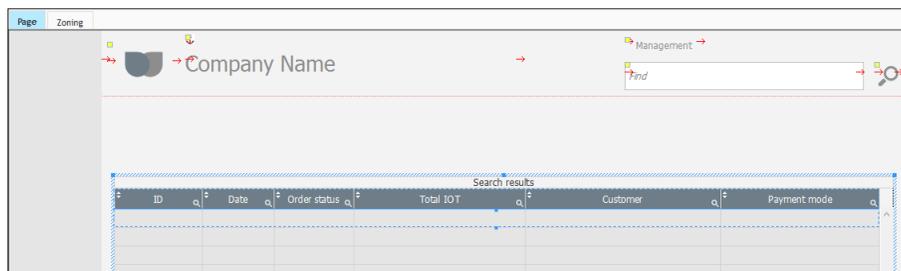
2. Click the name of the Table control. The caption of the Table control is displayed in the lower section. We are going to modify this caption. Replace the proposed caption by "Search results".



3. Click the "COL_OrdersID" column. The column title is displayed in the lower section of the screen. Replace the "Order identifier" caption by "ID".



4. Click the "COL_FullName" column. Replace the "Name" caption by "Customer".
5. Click the "COL_Caption" column. Replace the "Caption" caption by "Payment mode".
6. Validate the description window of Table control. The control is automatically updated with the modifications performed.
7. Reduce the size of "ID" column in order for all the columns to be displayed in the Table control.
8. Increase the size of the Customer and PaymentMode columns in the Table control.



9. To save the page, click among the quick access buttons. We are going to check the sizes of columns by running the page.



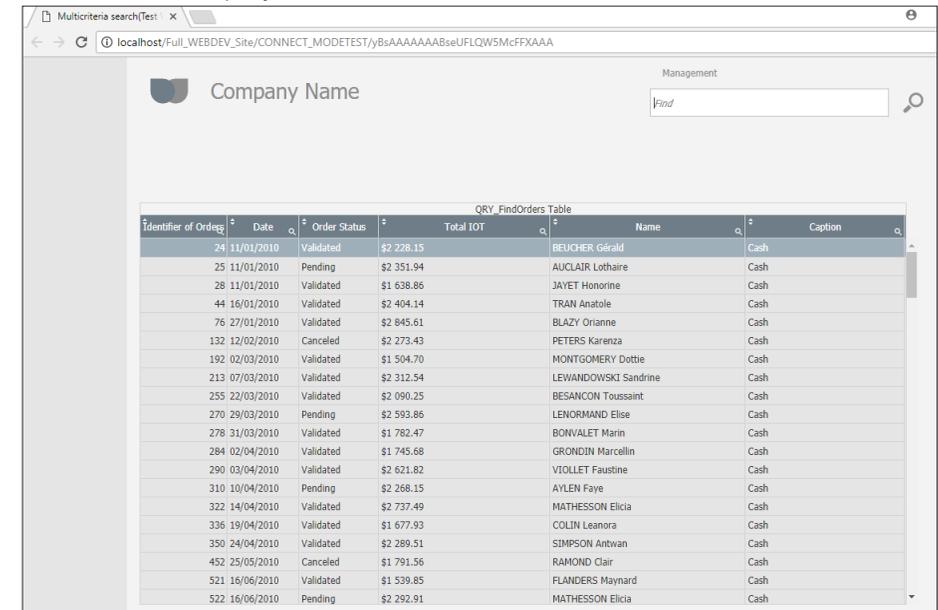
Notes

Live Data and controls based on queries

The Live Data is not displayed in the controls that use a query as data source for the following reason: the data displayed in the control depends on the query result therefore it is known at run time only.

► We are going to run a first test of this page:

1. Click among the quick access buttons.
2. Only orders paid in cash and whose price is found between \$1,500 and \$3,000 are displayed, like in the last query test run in the editor, when the parameters were specified in the test window of query.



3. Close the browser to go back to the editor.

► Let's study the processes associated with the Table control:

1. Select the Table control and press the [F2] key.
2. The "Initializing TABLE_QRY_FindOrders" process contains the following code:

```
//MySource.pCustomerName = <Value of pCustomerName parameter>
//MySource.pStatus = <Value of pStatus parameter>
MySource.pPaymentModeID = "1"
MySource.pMinPrice = "1500"
MySource.pMaxPrice = "3000"
```

The test parameters have been retrieved as default parameters for the execution. We are now going to modify the page in order for the parameters to be entered by the user, via controls.

3. To build our page, we are going to switch all the code lines corresponding to parameters into comments:

- Select the "MySource ..." code lines.
 - Use the [CTRL] + [/ from numerical keypad] key combination.
4. Close the code editor (click the cross at the top right of the editor).

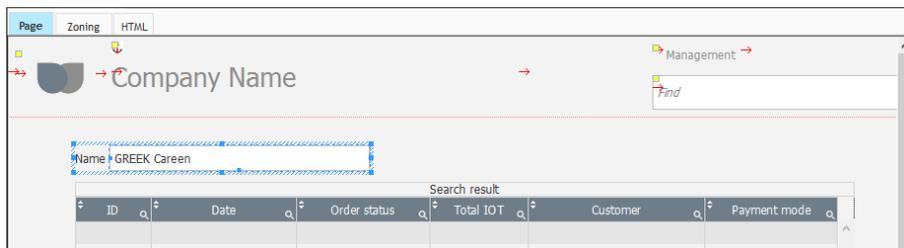
We are now going to create in our page the different controls allowing the user to select the query parameters. These controls will be positioned above the Table control.

- ▶ If necessary, move the Table control in the page and reduce its height in order to leave space for creating the different controls for criteria selection.

First parameter: Customer name

In order for the user to type a customer name to find, we are going to create an edit control.

- ▶ To create the edit control:
 1. Display the "Analysis" pane if necessary: on the "Home" pane, in the "Environment" group, expand "Panels" and select "Analysis". The different data files described in the "Full_WEBDEV_Site" analysis are displayed in the pane.
 2. Click  on the left of "Customer" file: the items found in the data file are listed.
 3. Select the "Full Name" item found in the Customer file and Drag and Drop this item to the "PAGE_Multicriteria_search" window.
 4. The edit control is automatically created. Position this control above the Table control.



- ▶ We are now going to pass the value typed in the edit control in parameter to the query:
 1. Select the Table control and press the [F2] key.
 2. In the initialization process of Table control, replace the line:

```
//MySource.pCustomerName = <Value of pCustomerName parameter>
```

by

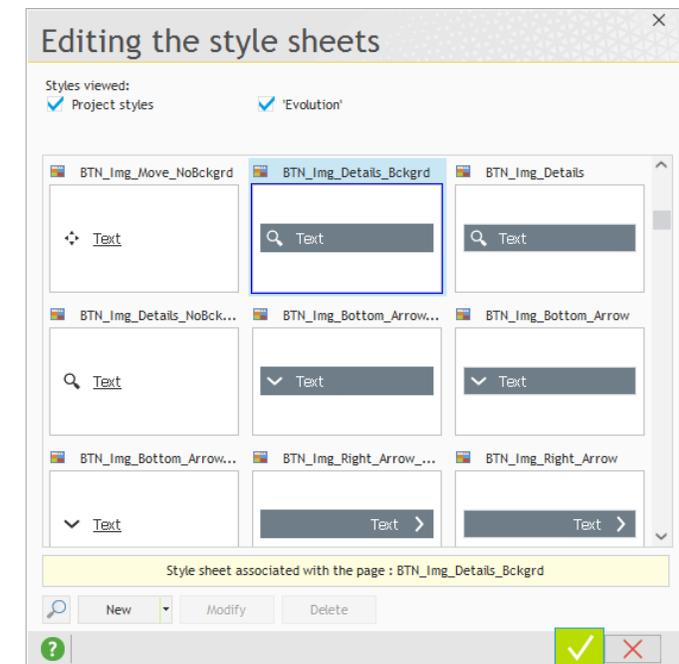
```
MySource.pCustomerName = EDT_FullName
```

In this code, EDT_FullName is the name of the edit control that was just created. The value of this control is associated with the pCustomerName parameter expected by the query.

3. Close the code editor.

- ▶ Before running the test, we are going to create a button used to re-display the content of Table control according to the value selected in the Radio Button control:

1. Create a Button control:
 - on the "Creation" pane, in the "Usual controls" group, click .
 - then, click in the top right corner of the page.
2. Modify the control caption (press the [ENTER] key on the keyboard). The new caption is "Find".
3. Modify the control style:
 - In the popup menu of control (right mouse click), select "Choose a WEBDEV style".
 - In the window that is displayed, select "BTN Details (With background)" and validate.



4. Resize the control if necessary:
 - select the control.
 - select "Adapt the size" from the popup menu.
5. Display the code associated with this control: press the [F2] key.
6. Type the following code in the "Click (server)" process:

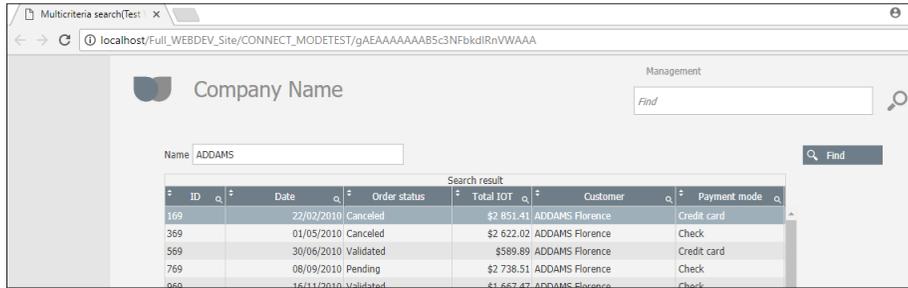
```
// Refreshes the display of Table control
TableDisplay(TABLE_QRY_FindOrders, taInit)
```

In this code, the **taInit** constant is used to re-run the initialization process of Table control (the process in which the parameters are passed to the query).

7. Close the code editor.

► We are now going to check how the first parameter is passed:

1. To save the page, click among the quick access buttons.
2. Click among the quick access buttons.
3. In the page that is displayed, type the customer name ("ADDAMS") and click the "Find" button. The content of Table control is modified: all the orders placed the customer "ADDAMS" are displayed.



4. Close the test page.

Second parameter: Order status

Three states can be assigned to an order:

- pending,
- paid,
- canceled.

In our analysis, the order status is saved in the "Status" item found in the "Orders" file. This item is a radio button.

To allow the user to select one of these three states, we are going to use the Radio Button control associated with the "Status" item found in the "Orders" file.



Notes

The radio buttons are also called "option boxes". They are used to select a single option among the proposed ones.

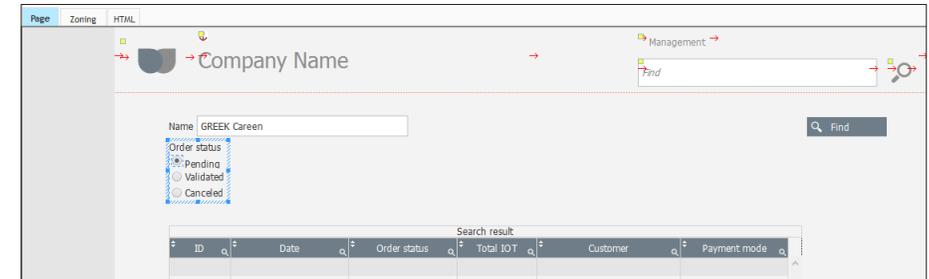
How to differentiate between a radio button and a check box?

The radio button is used to select a single option.

The check box is used to select several options.

► To create the Radio Button control:

1. In the "Analysis" pane, click the "+" icon on the left of "Orders" file: the items found in the data file are listed.
2. Select the "Status" item found in the Orders file then Drag and Drop this item into the "PAGE_Multicriteria_search" window.
3. The Radio Button control is automatically created. Position this control above the Table control.



► We are now going to pass the value selected in the Radio Button control in parameter to the query:

1. Display the processes associated with the Table control:
 - Select the Table control.
 - Select "Code" from the popup menu (right mouse click).

2. In the initialization process of Table control, replace the line:

```
//MySource.pStatus = <Value of pStatus parameter>
```

by

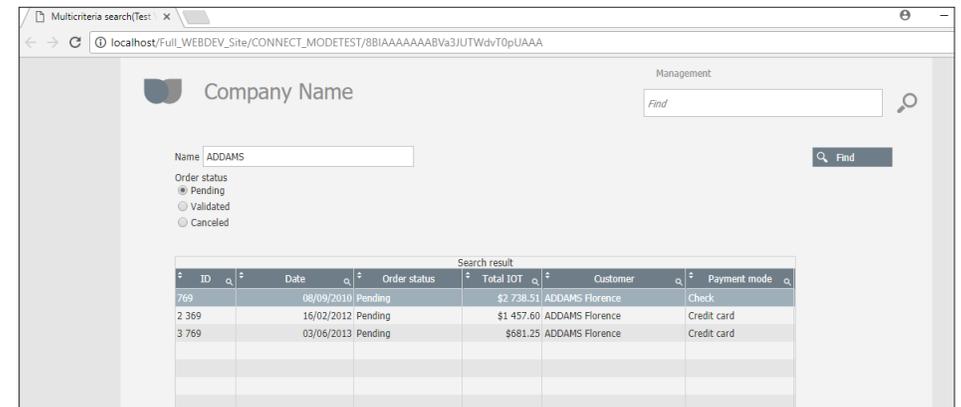
```
MySource.pStatus = RADIO_Status
```

In this code, RADIO_Status is the name of the Radio Button control that was just created. The value of this control is associated with the pStatus parameter expected by the query.

3. Close the code editor.

► We are now going to check how the first two parameters are passed:

1. Click among the quick access buttons.
2. Type the name "ADDAMS" and select "Pending".
3. Click the "Find" button: only the orders placed by the customer ADDAMS and waiting for payment are listed.



- ▶ In this example, only the orders placed by a customer and in a specific status can be listed. It may also be interesting to list all the orders placed by a customer regardless of their status. We are going to modify our page in order to perform this process.

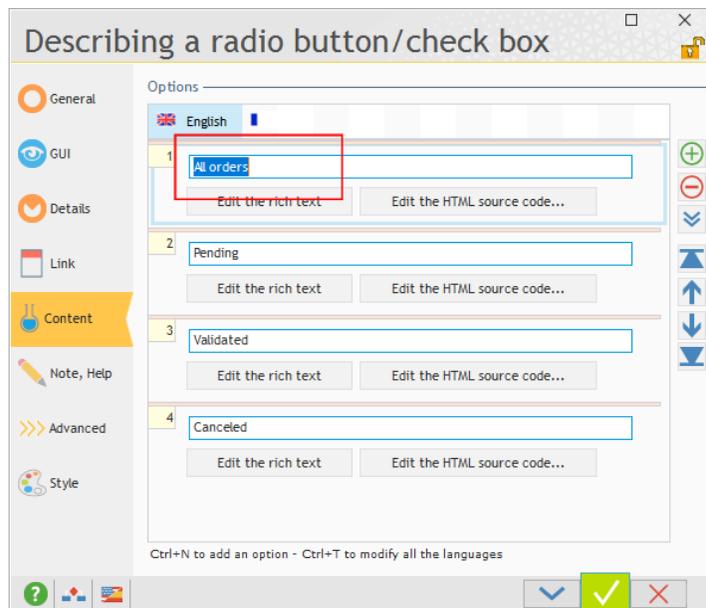
To manage this case, we must:

- Add a report that will be used to display all the orders,
- Manage the additional report.

- ▶ Close the browser.

- ▶ To add an option into the Radio Button control:

1. Select the radio Button control that was entered beforehand.
2. Display the description window of control (double-click the control for example).
3. In the "Content" tab, in the list of options, insert the "All orders" option at the top of the list:
 - Click the "+" button.
 - Type "All the orders".
 - Use the arrow pointing to the top (on the right) to move the option up.



4. Validate the description window of control.
5. Enlarge the control in the editor in order for all the options to be displayed (move the Table control if necessary).

- ▶ The new option "All orders" means that the "pStatus" parameter of the query will be ignored. To ignore a query parameter, all you have to do is assign the NULL value to this parameter. We are going to manage this case for the pStatus parameter.

1. Select the Table control in the editor and display its processes (press [F2] or select "Code" from the popup menu of control).

2. In the initialization process of Table control, replace the line:

```
MySource.pStatus = RADIO_Status
```

by

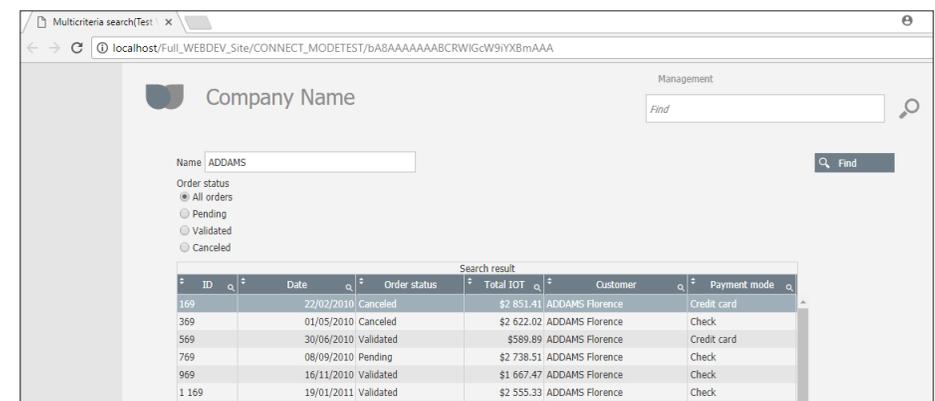
```
SWITCH RADIO_Status
// All orders
CASE 1
  MySource.pStatus = Null
OTHER CASE
  MySource.pStatus = RADIO_Status-1
END
```

In this code, if the RADIO_Status control corresponds to 1 (case of "All orders" option), the parameter of the associated query has "NULL" for value. Otherwise, the parameter has for value the number of the selected option minus 1 (that corresponds to the option that was just added).

3. Close the code editor.

- ▶ Let's test our page right now.

1. To save the page, click  among the quick access buttons.
2. Click  among the quick access buttons.
3. Type the name "ADDAMS" and select "All orders".
4. Click the "Find" button: all the orders placed by the customer ADDAMS are listed, regardless of their status.



5. Close the browser.

Third parameter: Payment mode

Several payment modes can be used for an order: cash, checks, ... The available payment modes are stored in the "PaymentMode" data file.

We are going to use a Combo Box control based on this data file to allow the user to select the requested payment mode.



Notes

The "Combo box" control is used to display a list of elements and to select an element from this list.

Unlike a list box, a combo box displays a single element at a time: during a click on the control, the combo box is expanded and it proposes to select another element. A single element is selected.

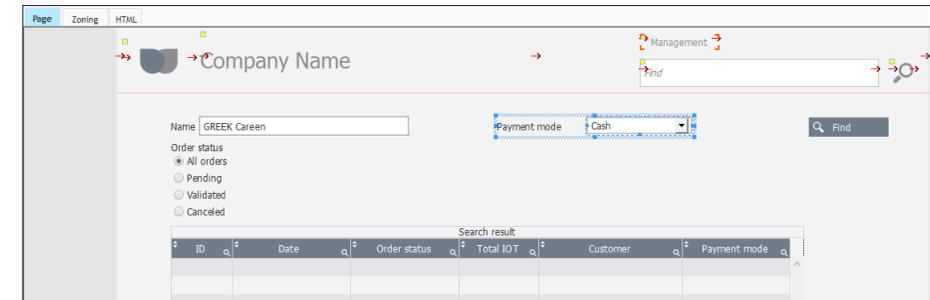
The elements displayed in a combo box can be defined when creating the control in the editor. These elements:

- are defined by programming.
- come from a data file or from a query.

► To create a Combo Box control:

1. On the "Creation" pane, in the "Usual controls" group, click "Combo box".
2. Click the location where the control must be created in the window (beside the radio button that was just created for example).
3. The wizard for creating the Combo Box control starts.
4. Select "Display the data coming from a file or from an existing query" and go to the next step.
5. Select the "PaymentMode" data file and go to the next step.
6. The item that will be displayed in the Combo Box control is "Caption". Select "Caption". Go to the next step.
7. Select the search key: "Caption". Go to the next step.
8. Select the return value: "PaymentModeID". This return value is very important because it will be passed in parameter to the query. Go to the next step.
9. Keep the default options in the different wizard screens and validate the creation of the Combo Box control.
10. The Combo Box control is automatically created in the page.

► Modify the caption of Combo Box control (press the [ENTER] key). The new caption is "Payment mode".



► We are now going to pass the value selected in the Combo Box control in parameter to the query:

1. Display the processes associated with the Table control: select the Table control and press the [F2] key.
2. In the initialization process of Table control, replace the line:

```
//MySource.pPaymentModeID = 1
```

by

```
MySource.pPaymentModeID = COMBO_PaymentMode
```

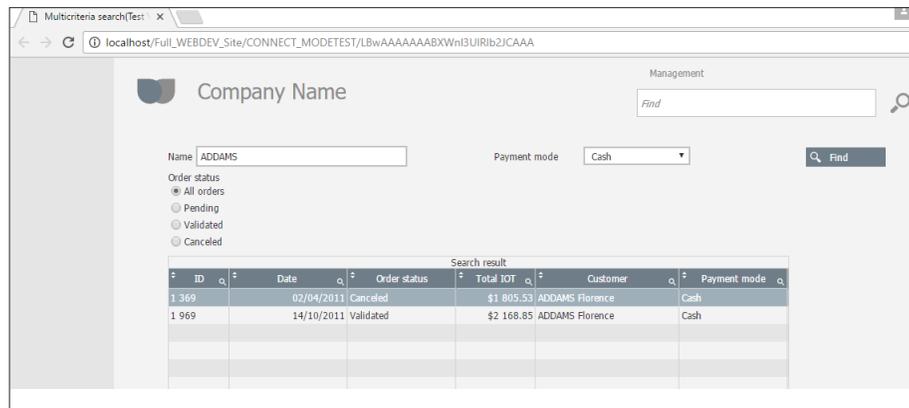
In this code, COMBO_PaymentMode is the name of the Combo Box control that was just created. The value of this control is associated with the pPaymentModeID parameter expected by the query.

3. Close the code editor.

4. Save the page (or CTRL S).

► We are now going to check how the parameters are passed:

1. Click  among the quick access buttons.
2. Enter the name "ADDAMS".
3. Change the status of orders via the radio button and modify the payment mode via the Combo Box control, then click the "Find" button. The content of Table control is modified.



4. Close the browser.

Last parameter: Total amount of order

The last query parameter corresponds to the order amount taken into account. We have a parameter that corresponds to the minimum amount and a parameter that corresponds to the maximum amount. The interface must allow the user to type a price interval. To do so, we are going to use a Range Slider control.

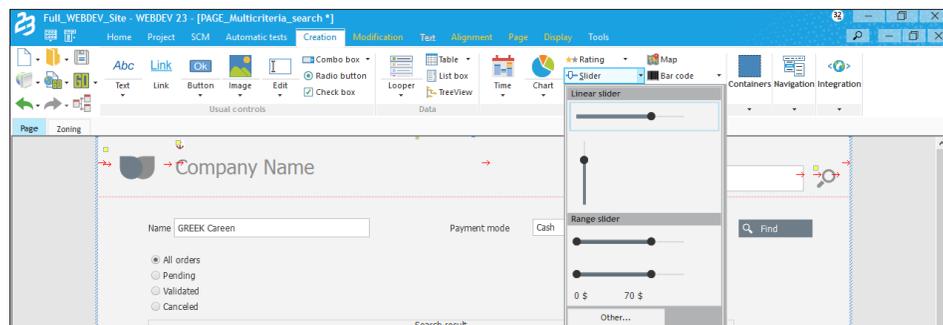


Notes

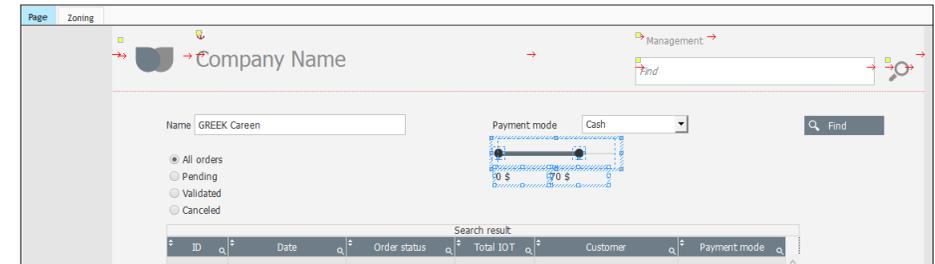
A Range Slider control is a graphic control that can be easily included in a WEBDEV site and that is used to select the bounds of an interval (start value and end value).

► To manage the product price:

1. On the "Creation" pane, in the "Graphic controls" group, expand "Slider" and select the type of range slider with the prices.



2. Click inside the page, between the "Payment mode" control and the "Find" button. The range slider is automatically created.



- The control that was just created is an advanced control that includes the slider and two controls displaying the bounds. The programming required to display the bounds is already performed in the control. We are going to initialize the Slider control so that it proposes the bounds corresponding to the data found in the Orders file.
1. Select the Slider control and display its processes ([F2] key).
 2. Insert the following code (before the existing code lines) into the initialization process of control:

```
// The lower value is the least expensive amount
HReadFirst(Orders, TotalIOT)
MySelf..MinValue = Orders.TotalIOT
MySelf..LowerValue = MySelf..MinValue

// The upper value is the most expensive amount
HReadLast(Orders, TotalIOT)
MySelf..MaxValue = Orders.TotalIOT
MySelf..UpperValue = MySelf..MaxValue
```

Let's study this code:

- **HReadFirst** is used to read the first record of "Orders" file according to the search key, TotalIOT in this case. This function will allow us to read the record corresponding to the lowest amount.
- Then, the amount read in the record is associated with the lower bound of Slider control as well as its minimum value.
 - The lower bound is used to specify the minimum value that can be selected by the user.
 - The lower value is used to specify the minimum value currently selected.
- The same principle will be used to find the greatest amount. The only difference is the name of the function used: **HReadLast**, that is used to read the last record of Orders file according to the amount, which means corresponding to the greatest amount.

3. Close the code window.

- We are now going to pass the selected amounts in parameter to the query:
 1. Display the processes associated with the Table control: select the Table control and press the [F2] key.

2. In the initialization process of Table control, replace the lines:

```
//MySource.pMinPrice = "1500"
//MySource.pMaxPrice = "3000"
```

by:

```
MySource.pMinPrice = RSLI_NoName2..LowerValue
MySource.pMaxPrice = RSLI_NoName2..UpperValue
```

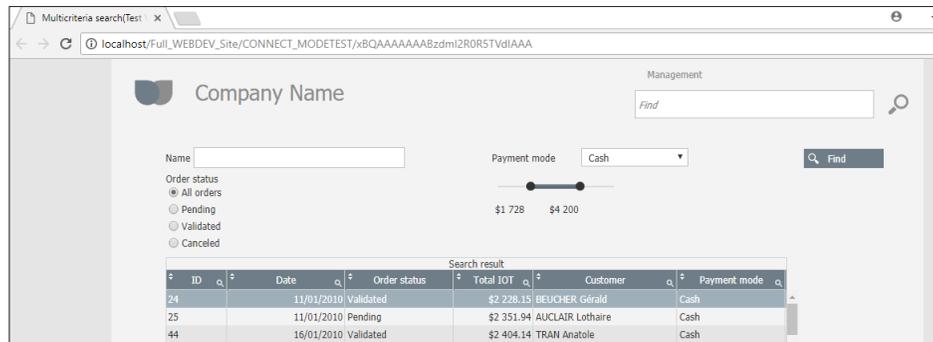
In this code, the parameters corresponding to the prices are equal to the lowest value and greatest value of slider.

3. Close the code editor.

4. Save the page ( or CTRL S).

- We are now going to check how the parameters are passed:

1. Click  among the quick access buttons.
2. Define the different search parameters:
 - Customer name,
 - Status,
 - Payment mode,
 - Amount.
3. Click the "Find" button. The content of Table control is modified.



The screenshot shows a web application interface with a search form and a table of search results. The search form includes a "Company Name" field, a "Find" button, and a "Payment mode" dropdown menu set to "Cash". Below the search form, there is a table with the following data:

ID	Date	Order status	Total IOT	Customer	Payment mode
24	11/01/2010	Validated	\$2 228.15	BEUCHER Gérald	Cash
25	11/01/2010	Pending	\$2 351.94	AUCLAIR Lothaire	Cash
44	16/01/2010	Validated	\$2 404.14	TRAN Anatole	Cash

4. Close the browser.

Optimizing the page display

By default, during the click on the "Find" button, the entire page content is returned by the server. To optimize the page display, you have the ability to enable the Ajax mode on this button. In our example, only the content of the Table control will be returned by the server.



Notes

WEBDEV proposes several methods for using the Ajax technology in a site:

- **Ajax in one click:** all you have to do is click in the code editor to transform a process into Ajax process.
- **Ajax controls** (Table control, Looper control). The Ajax controls are used to display data dynamically loaded from the server. No need to download everything at once!
- **Ajax by programming** via specific WLanguage functions: *AjaxExecute*, *AjaxExecuteAsynchronous*.

- To implement the Ajax mode on the "Find" button:

1. Select the "Find" button and display its processes ([F2] key).
2. In the server click process of button, the AJAX link appears to be struck out in the code bar.

```
Click of BTN_Find (server) AJAX If Error:
// Refreshes the display of the Table control
TableDisplay(TABLE_QRY_FindOrders, taInit)
```

3. Click the AJAX link: the "JAVA enabled" link appears, indicating that the process is automatically changed into an Ajax process.

```
Click of BTN_Find (server) AJAX enabled
// Refreshes the display of the Table control
TableDisplay(TABLE_QRY_FindOrders, taInit)
```

4. Close the code editor.

- Run the page test:

1. Click  among the quick access buttons.
2. Define the different search parameters:
 - Customer name,
 - Status,
 - Payment mode,
 - Amount.
3. Click the "Find" button. Only the content of Table control is re-displayed.

LESSON 5.2. PRINTING AN INVOICE

This lesson will teach you the following concepts ...

- Creating a report based on a query
- Printing a report based on a query with parameters



Estimated time: 45 mn

Overview

We are going to give the user the ability to print the invoice corresponding to the sought order.



Answer

If the pages have not been created in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (With pages)".

A full corrected project is also available: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (Answer)".

Principles for printing on Internet

Actually, we should not talk about "printing" on Internet. Indeed, when a document must be "printed", a file is generated beforehand (in HTML, PDF or XML format) and it is transmitted to the browser. As soon as the file transfer is completed, the Web user decides whether the document that was received must be printed or not.

However, "printing" on the server remains possible. However, the document that is printed (on the server printer or on a shared network printer) will not be accessible to the Web user.

The printed or generated document is prepared and formatted by the report editor. The information found in the document can come from a database.

Two types of prints are available:

- The direct print (on the server printer).
- The generation of various documents (HTML, PDF, ...).

Direct print

The direct print consists in printing on a printer directly. This printer is plugged into the server or accessible from the network.

The direct print with WEBDEV is recommended for Intranet or Extranet only. This type of print is used to print the logs for customer connections, to directly print the order on the printer of sales department.

Generating documents in HTML, PDF, XML, ...

Printing from HTML, PDF or XML documents consists in creating a document and in displaying it on the browser of Web user. To create this document, all you have to do is use the report editor of WEBDEV. A formatted file can be "generated" by your WEBDEV site from a data source. The file can be in HTML, PDF, RTF or XML format.

The main benefit of this method is that the generated file can be transmitted to the browser. The Web user can print the document or store it on his computer.

Printing in a file can be used for an Internet site and for an Intranet/Extranet site (for example: to transmit an order form in PDF format to the Web user, ...).

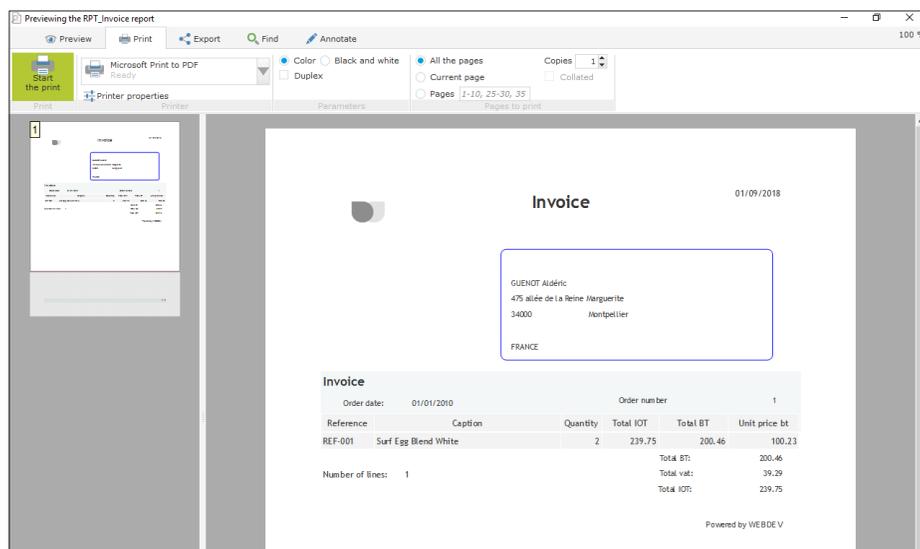
What is a report made of?

In our example, the invoice that must be printed will correspond to a report. A report is a project element used to group and format the data to print.

To create and use reports, WEBDEV proposes a specific editor: the report editor.

A report includes:

- different blocks. The blocks are used to define the areas where the data will be displayed. The available blocks are as follows:
 - Start of document: block displayed on the first page only.
 - Page header: block displayed at the top of each page.
 - Body: block containing the data. This block is printed as long as there is data to print.
 - Page footer: block displayed at the bottom of each page.
 - End of document: block displayed on the last page only.
 - controls used to display the data.
- Let's see the report that we want to create:



Creating the "Invoice" report

First of all, we are going to list the information that will be displayed in the report:

- The order characteristics: date and order number.
- The personal details of customer: name, address, zip code, city and country.
- The characteristics of order lines:
 - Quantity ordered
 - Product reference
 - Product caption
 - Total BT
 - Total IOT

To easily create this report, the data to print will be grouped in a query. This query can be used by the report or by any other element of the WEBDEV project (Table control, Looper control, ...).



Note

WEBDEV proposes to create reports from several data sources: data files, queries, controls, text files, ...

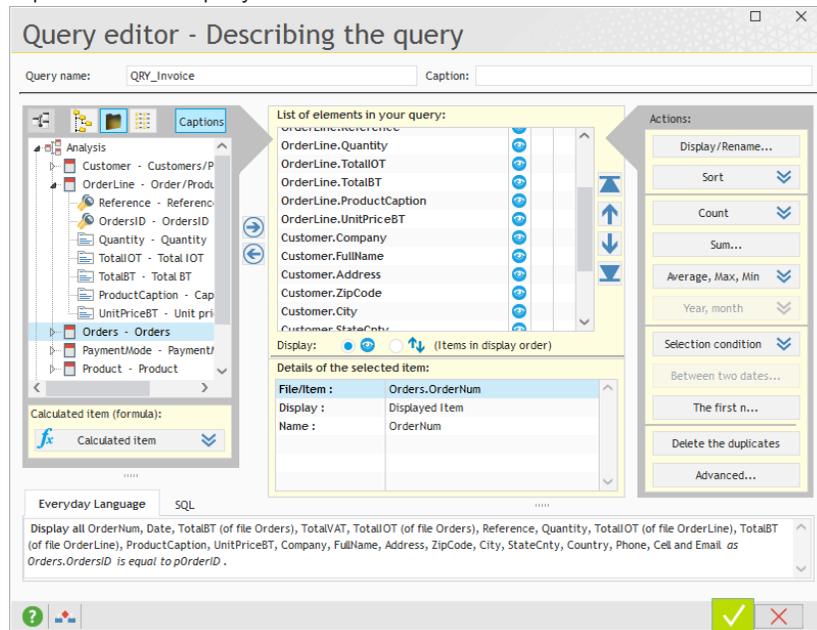
In most cases, we advise you to group the data to print via a query and to create a report based on this query. To add an information into the report, all you have to do is add the corresponding item into the query.

The reports based on data files must be simple reports, which means reports that display data coming from a single data file.

Creating the query

- ▶ The query editor will be used to create the base query of the report.
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Query". The wizard for query creation starts.
 2. Select the "Select" option. Go to the next step.
 3. The description window of query is displayed.
- ▶ First of all, give a name to the query: type "QRY_Invoice" instead of "QRY_NoName1" in the "Query name" area.
- ▶ To build the query, we are going to select the elements that will be displayed in the result. The query will contain the content of Orders file, the content of OrderLine file and the content of Customer file.
 1. In the left section of the screen, select the Orders file and click the blue arrow (): the items of the Orders file are displayed in the middle of the screen.
 2. Repeat this operation for the OrderLine and Customer files.

The description window of query is as follows:



At this time, this query is used to select all the orders and the corresponding order lines. We are now going to reorganize the query items. Indeed, this order will be used to create the different controls of the report based on this query. We want the product caption to be positioned just after the product reference.

- ▶ To reorganize the order of items:
 1. Select the ProductCaption item of the OrderLine file.
 2. Click the "top arrow" button on the right of the list of items. The item moves up.
 3. Position the item just after the product reference (OrderLine.Reference).

We want to select the data corresponding to a single order whose identifier is known. Therefore, we are going to define the order number in parameter.

- ▶ To manage the "Order identifier" parameter:
 1. Select the Orders.OrdersID item (in the middle of the screen).
 2. Click in the fourth column: select "New condition".



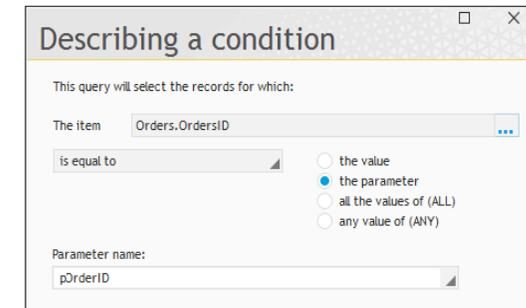
Note

When describing a query, the section that lists the query elements includes 4 columns:

- The item name,
- The ability to display (or not) the item in the query result,
- The sort management for the item,
- The number of conditions associated with the item.

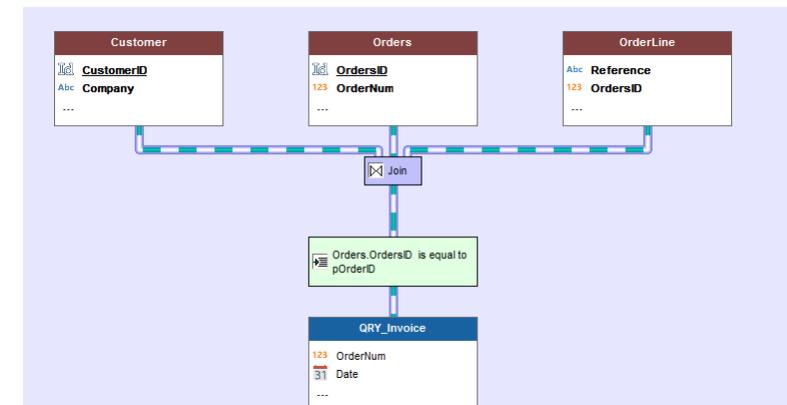
To access one of these characteristics, all you have to do is click in the corresponding column.

3. In the window that is displayed, we are going to specify that the selection condition corresponds to a parameter:



Perform the following operations:

- Select "Is equal to".
 - Check "the parameter".
 - Specify the parameter name: "pOrderID".
4. Validate the description window of condition. The number "1" is displayed on the right of Orders.OrdersID item, indicating that a selection condition was defined.
 5. The order identifier will not be viewed so let's make it invisible: click the eye found on the line of the item and select "Don't display".
 6. Similarly, make the following items invisible:
 - Orders.Status,
 - Orders.CustomerID,
 - Orders.PaymentModelID,
 - OrderLine.OrdersID,
 - Customer.CustomerID.
 7. Validate the description window of query (green button at the bottom of the screen).
 8. Validate the backup information.
 9. The graphic query representation is displayed:



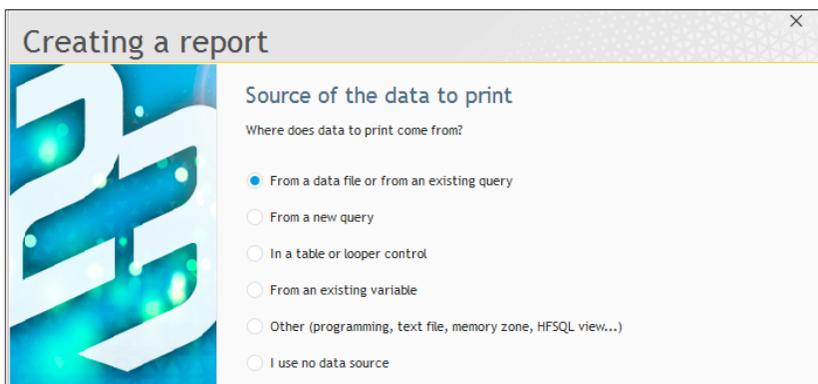
Creating the report based on a query

► To create a report:

1. Click among the quick access buttons.
2. The window for creating a new element is displayed: click "Report" then "Report". The wizard for report creation starts.
3. The wizard for report creation proposes several types of reports:



4. Select "Table". Go to the next step.
5. Select the data source of report. The report will be based on the query that was just created. Select "From a data file or from an existing query". Go to the next step.

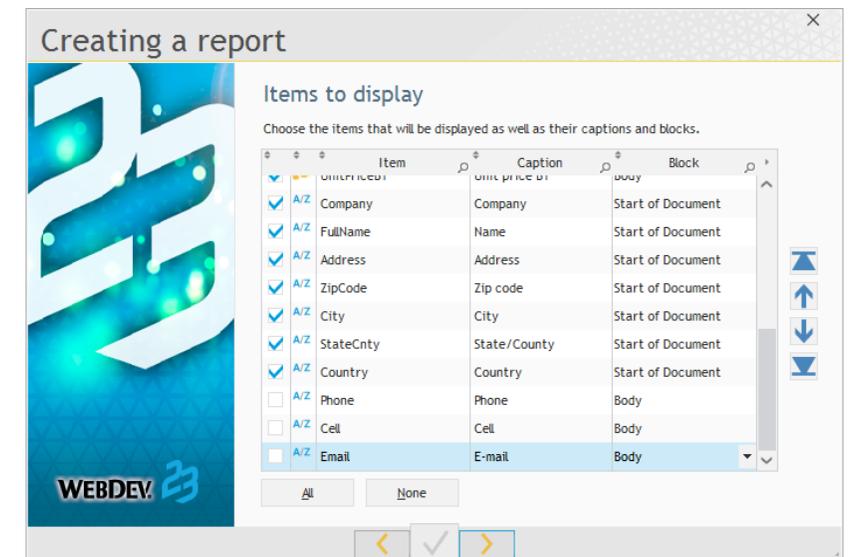


6. In the list of data files and queries, select the "QRY_Invoice" query. Go to the next step.

7. The wizard asks you to specify whether a break is required. No break will be used in this report. This concept will be presented later in this tutorial. Answer "No". Go to the next step.

8. You are going to specify the order in which the items will be printed and how they will be distributed in the different blocks:

- The items regarding the customer will be displayed in the "Start of document" block. Indeed, this information must not be repeated on each order line.
- The items regarding the order will be displayed in the "Page header" block. Indeed, this information must not be repeated on each order line.
- The items regarding the order lines will be displayed in the report body. These items will be displayed for all the order lines of order.
- The items regarding the totals of the order will be displayed in the "End of document" block. Indeed, this information must not be repeated on each line of the order.



The following table presents the different assignments of items in the order presented in the wizard:

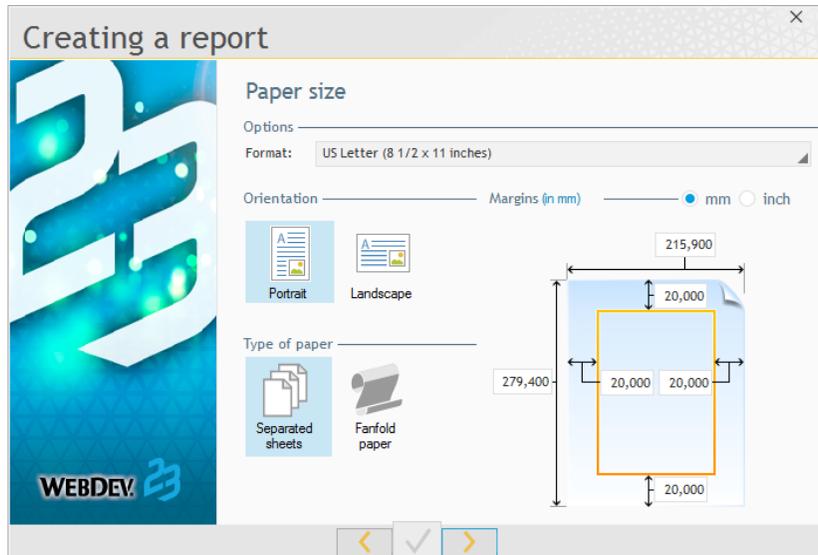
Item	Block
OrderNum	Start of document
Date	Page header
TotalBT	End of document
TotalVAT	End of document
TotalIOT	End of document
Reference	Body
Quantity	Body
TotalIOT_Ca	Body
TotalBT_Ca	Body

ProductCaption	Body
UnitPriceBT	Body
Company	Start of document
FullName	Start of document
Address	Start of document
ZipCode	Start of document
City	Start of document
StateDep	Start of document
Country	Start of document
Phone	Uncheck
Cell	Uncheck
Email	Uncheck

9. Go to the next step.

10. The wizard proposes to create a counter, a sum or a mean on the numeric items found in the report. In this report, the calculations are performed by the query. Click the "No calculation" button. Go to the next step.

11. This screen is used to define the report layout.



We will keep the default values with the "Portrait" orientation.



Notes

Print margins

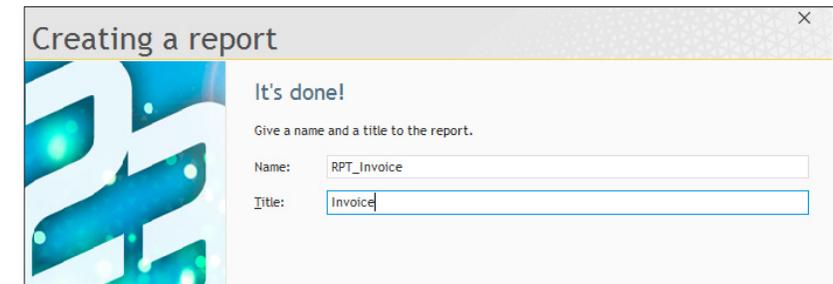
When choosing the print margins, don't forget to take into account the physical margins of printers. The physical margins of the printer are margins where no print is allowed. Additionally, physical margins are different depending on the printer models.

12. Go to the next step.

13. This screen allows you to select the skin template used for the report. We recommend that you use the same skin template as the one used for the pages. In our case, select the "Evolution" skin template for example and go to the next step.

14. Let's give a name and a title to the report.

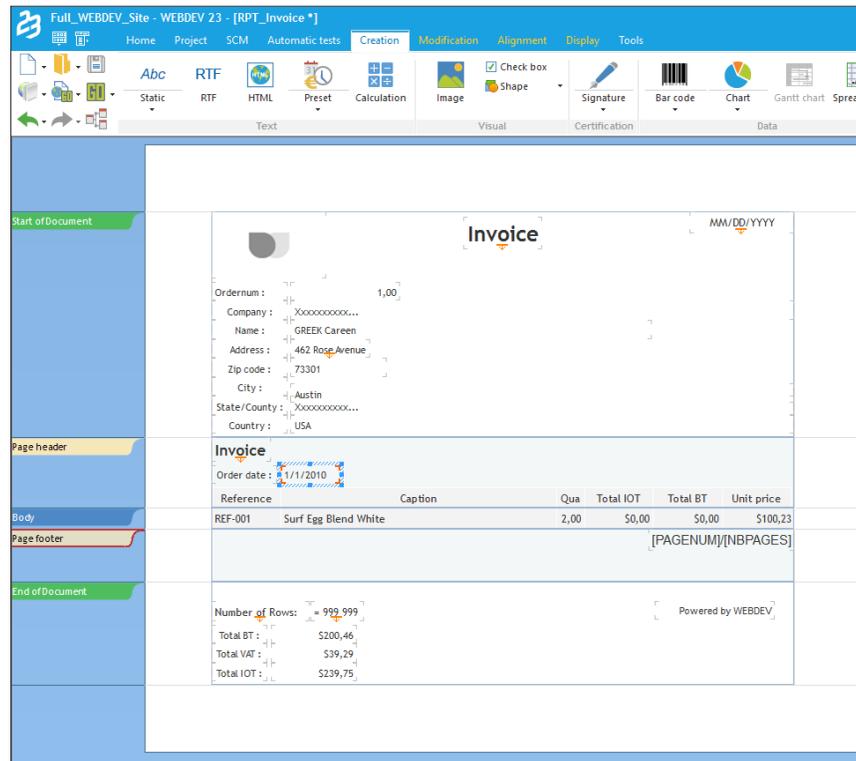
- Type the title: "Invoice".
- Enter the name: "RPT_Invoice".



15. Validate.

16. Validate the backup information.

17. The report is displayed in edit in the report editor:



18. The different order lines are grouped in a table.

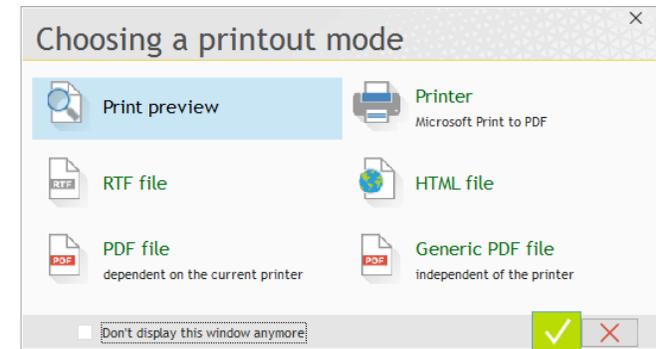
► To run this report, click among the quick access buttons.

1. The report editor asks for the print destination. The print destination can be:
 - Print preview: the report is displayed on the screen in a specific window.

Notes

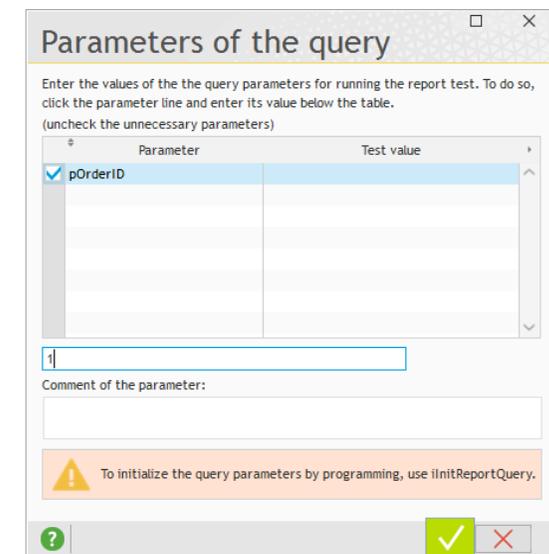
The print preview is available when developing the site only. At run time, no print preview will be displayed to the user. The only available preview will be the display of generated file by the browser.

- Printer: the report is printed on the default printer.
- Specific file: the report is printed in a file to the selected format.



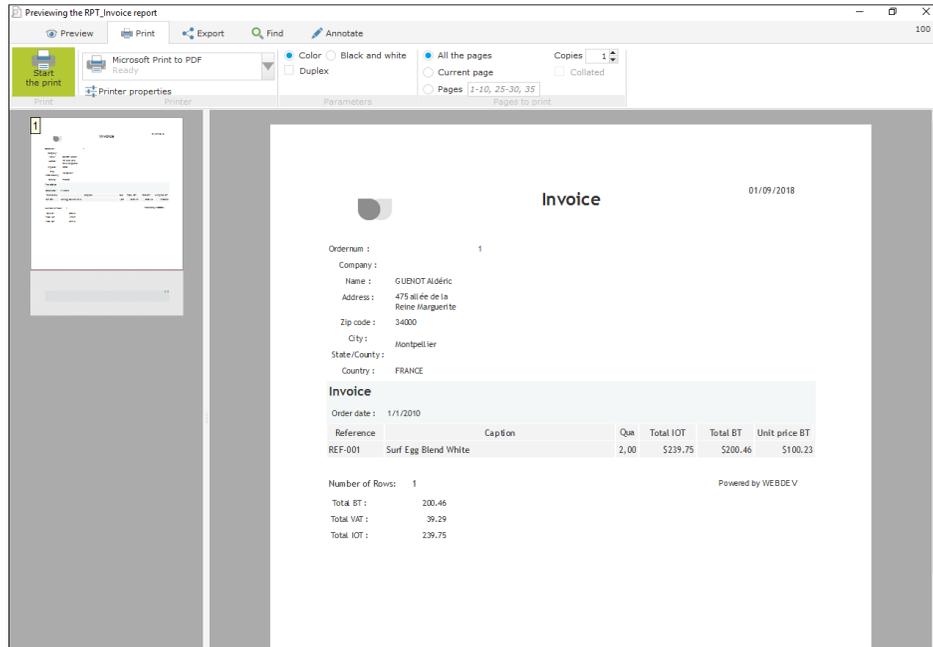
Choose "Print preview" and validate.

2. The report editor asks for the parameters of query used by the report. Don't forget that a parameter was used to specify the number of the order to print. For the example, enter the test value "1".



Validate.

3. The report is displayed on the screen.

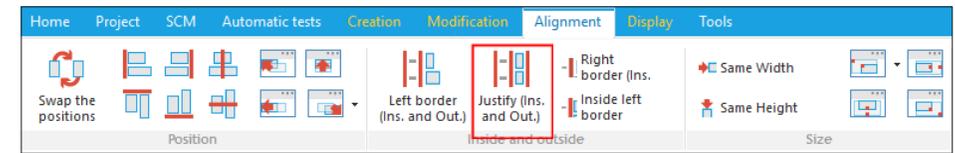


Modifying the "Invoice" report

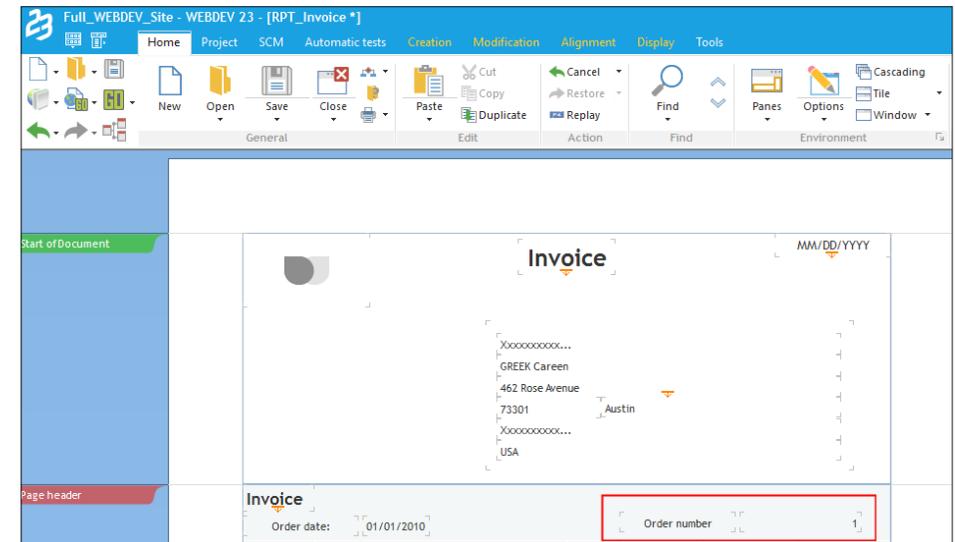
We are going to perform modifications in the report that was just created.

- ▶ We want to position the information regarding the customer and the order in the page header:
 1. Delete the captions found in front of the customer information (Name, ...).
 2. Position the item containing the city beside the zip code.
 3. Enlarge (with the handles) the control containing the company name: the control size must be identical to the size of the Zip Code and City controls.
 4. Align the controls:
 - Select the Company control.
 - Press the [CTRL] key and select (with the mouse) the controls containing the address, the state and the country.

- On the "Alignment" pane, click "Justify (Ins. and Out.)".

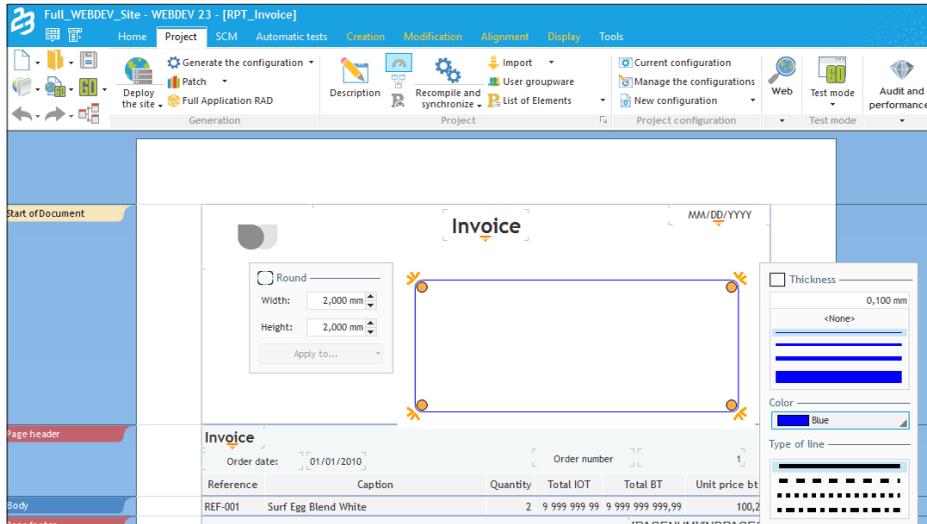


5. Select the information regarding the customer and move it with the mouse to the right of report.
6. Move the order number up (to the top of "Page header" block).



- ▶ We are going to create a border around the customer details:
 1. Create a Static control: on the "Creation" pane, in the "Text" group, click "Static".
 2. Click in the page header, at the location where the customer details are displayed.
 3. Press the [SHIFT] key and enlarge the static control (with the handles) so that it contains all the customer details. This operation is used to enlarge the static control without moving the controls found below.
 4. In the popup menu of the Static control, select "Edit the caption" and delete the caption. Click inside the report to validate the modification.

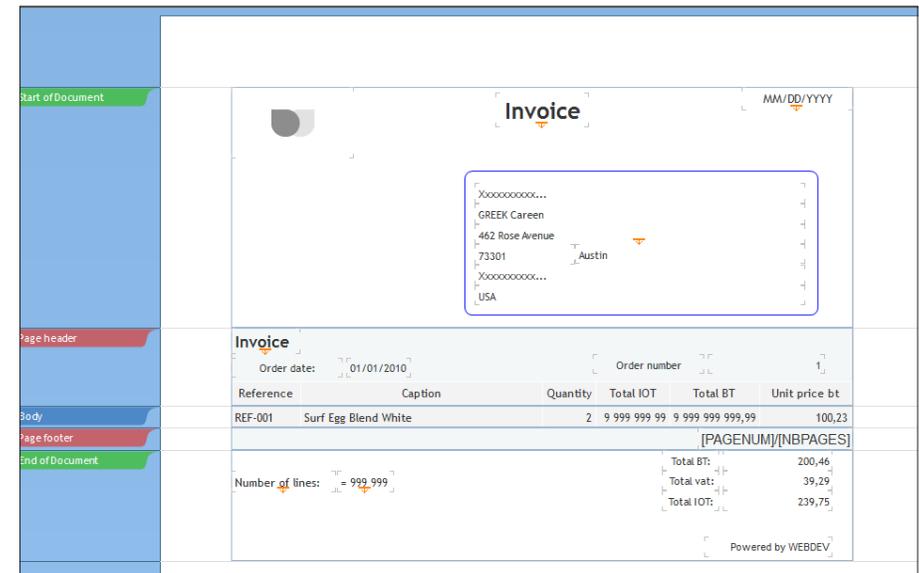
- ▶ To configure the border:
 1. Select the Static control that was just added.
 2. Press the [R] key and click the picto that appears in the top right corner of the control.
 3. Select "Edit the border". The configuration options are displayed:



4. Select:
 - dark blue as the color in the color selection.
 - a simple border (fine line).
 - a rounding set to 2 mm in width and in height.

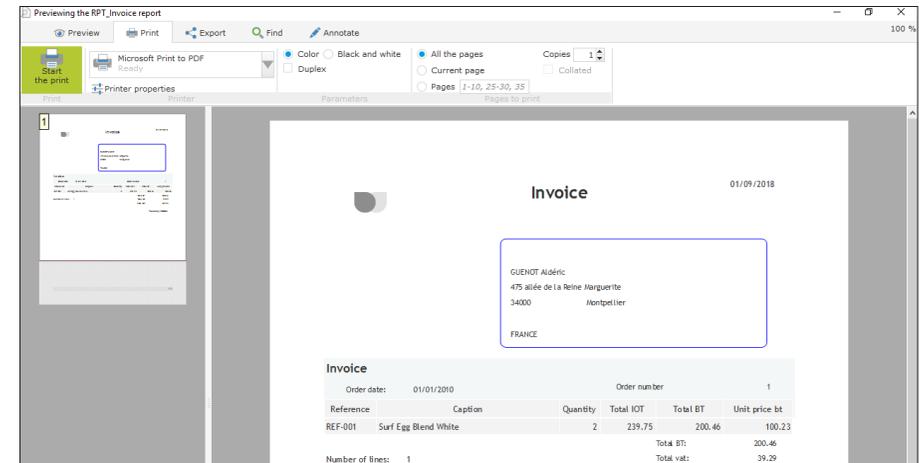
- ▶ We are going to position the totals properly in the "End of document" block:
 1. Select the controls corresponding to the totals found in the "End of document" block.
 2. Position these controls with the mouse in the bottom right corner of table.

3. The report is displayed in the report editor:



4. Our "Invoice" report is completed. Save the report by clicking among the quick access buttons.

5. Check the modifications performed by running the report in "Preview" mode (click among the quick access buttons).



6. Close the preview window.

Displaying the printed report from a button

As already seen at the beginning of this lesson, the site being run on a server, the document will be printed on a printer connected to the server (and therefore inaccessible to the user).

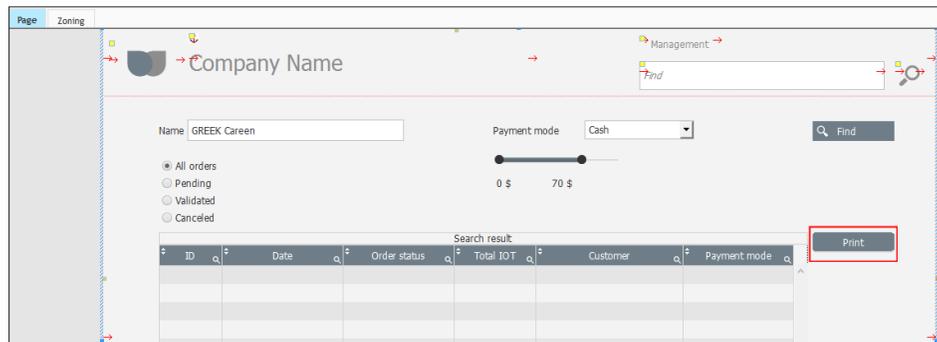
Therefore, the print will be done in PDF format on the server and the generated PDF document will be downloaded or displayed on the computer of the Web user. The document can then be printed by users on their printer.

In our site, the "RPT_Invoice" report will be printed from a button found in the page used to find an order. This button will be used to print the invoice of selected order.

Implementing the print

To print the "RPT_Invoice" report:

1. Position on the "PAGE_Multicriteria_Search" page: click the "PAGE_Multicriteria_Search" button found in the button bar.
2. On the "Creation" pane, in the "Usual controls" group, click "Button".
3. Click on the right of Table control to create the button.
4. Select the button and press the [ENTER] key on the keyboard. The caption becomes editable.
5. Replace "Button" by "Print" and validate.



6. Display the code of this control ([F2] key) and type the following server click code:

```
iDestination (iGenericPDF)
iInitReportQuery(RPT_Invoice, ...
    TABLE_QRY_FindOrders.COL_OrdersID[TABLE_QRY_FindOrders])
iPrintReport(RPT_Invoice)
FileDisplay(iLastFile(), "application/pdf")
```

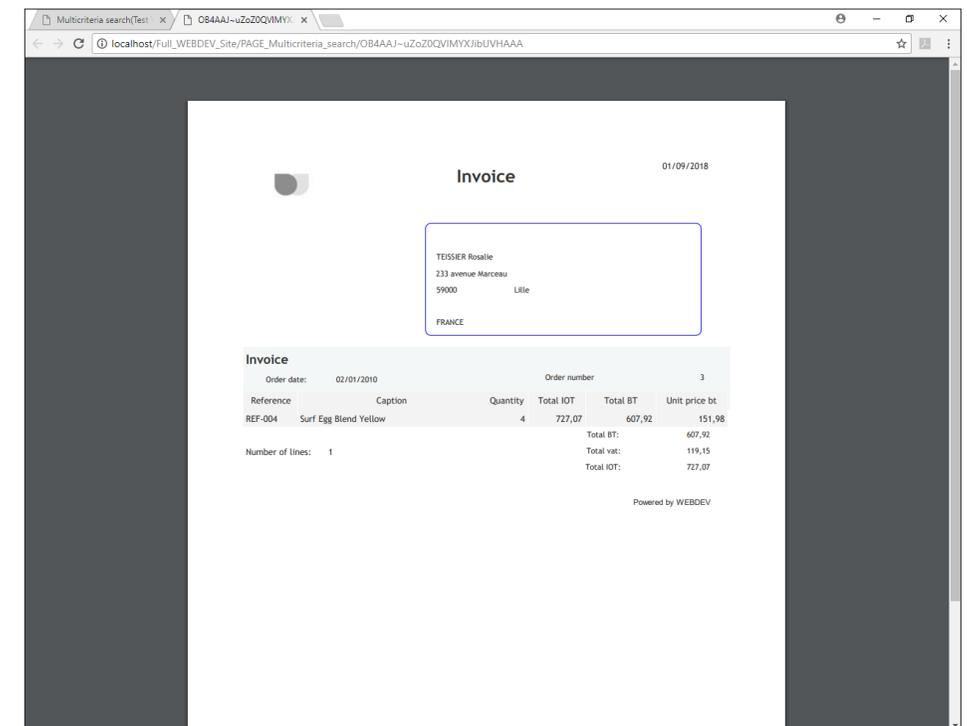
Let's study this code:

- **iDestination** allows you to define the print format used (PDF in this case).

- The "RPT_Invoice" report being based on a query with parameters, the parameter must be passed to the query before running the report. This operation is performed by **iInitReportQuery**. In our case, the parameter corresponds to the number of the current order, displayed in the Table control.
 - **iPrintReport** is used to trigger the generation of "Report_Invoice" report in the specified format (PDF in this case).
 - **FileDisplay** is used to display the invoice to the user:
 - **iLastFile** is used to get the path of last file generated by a report.
 - "application/pdf" if the MIME type of returned file. When this type is specified, the browser can display the file directly or it can choose the application that will be used to perform the display on the computer of Web user.
7. Close the code editor.
 8. Save the page ( or CTRL S).

Print test

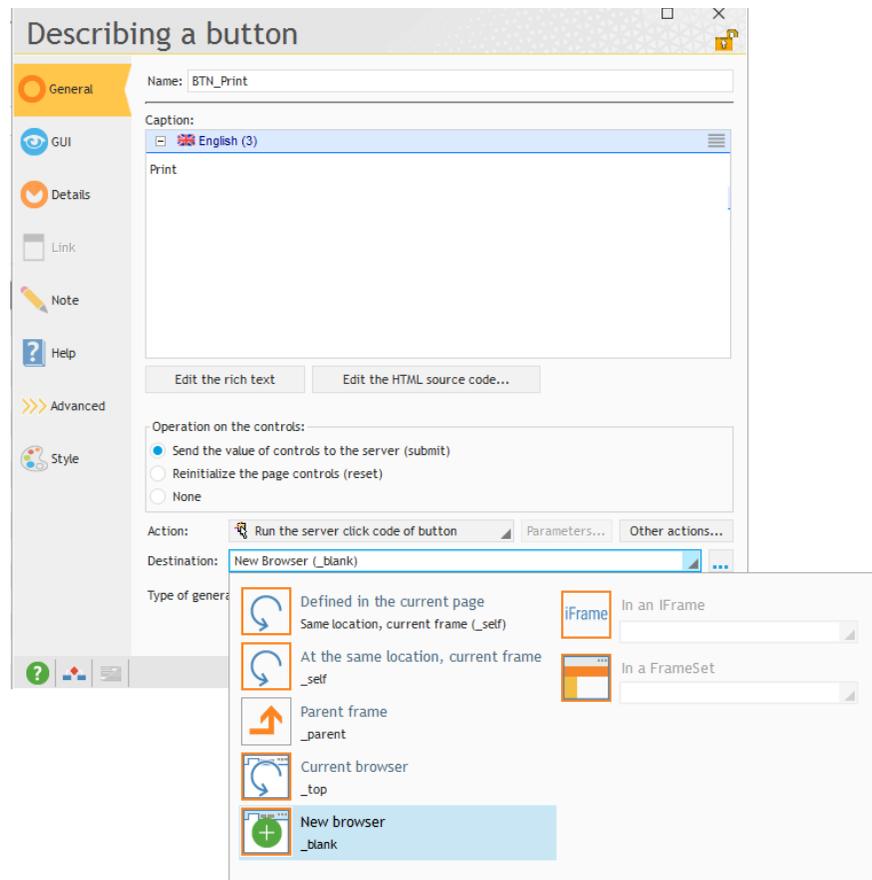
- ▶ Now, all we have to do is run a real test:
 1. Run the test of "PAGE_Multicriteria_Search" page.
 2. Specify the criteria and start a search.
 3. Select one of the orders displayed in the Table control.
 4. Print the order via the "Print" button.



5. The browser opens the PDF file. The browser has replaced the current page by the PDF file. We are going to change this behavior.
6. Close the browser.

► To open the PDF file in another tab or browser:

1. Double-click the "Print" button in the "PAGE_Multicriteria_Search" page. The description window is displayed.
2. In the "General" tab, in the Destination field, select "New browser".



3. Validate



Notes

New tab or new window?

The display in a new tab or in a new window can be chosen neither by the user nor by the site developer. It is the browser that chooses whether a new tab or a new window is opened. The behavior can differ according to the browser used!

LESSON 5.3. SENDING AN EMAIL

This lesson will teach you the following concepts ...

- How to send an email from a WEBDEV site



Estimated time: 20 mn

Overview

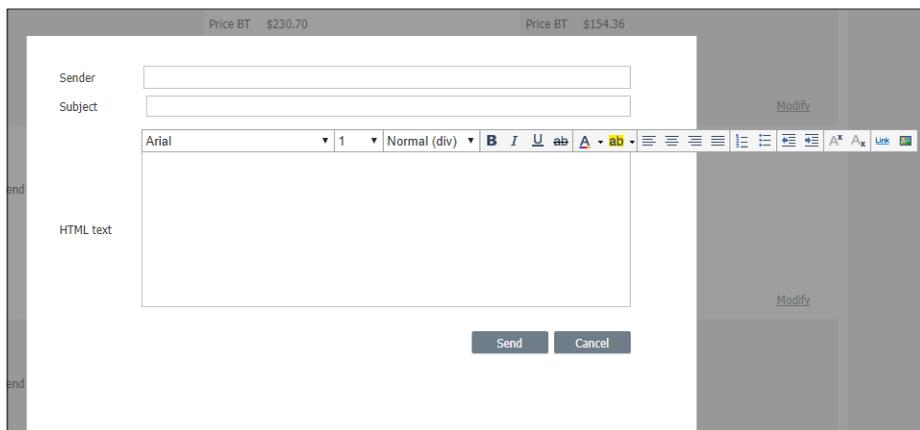
Several WLanguage functions allow you to manage the incoming and outgoing emails. You also have the ability to access the email characteristics:

- sender, recipients,
- outgoing date, subject, message,
- attachments ...

WEBDEV proposes several methods for managing the emails:

- Management via Lotus Notes, Outlook or MS Exchange:
 - **The Lotus Notes or Outlook messaging software:** these programs allow you to send and receive emails.
 - The **"Simple Mail API"** (also called SMAPI or Simple MAPI): this management mode of emails is used by most of the Microsoft applications, especially by Microsoft Exchange.
- Management via the POP3, IMAP and SMTP protocols:
 - The **POP3** protocol: this protocol for receiving emails is recognized by all the service providers. It is used to communicate with the server directly, available at your ISP. This protocol is used to list the incoming messages and to read them.
 - The **IMAP** protocol: this protocol for receiving emails allows you to leave the emails on the server so that they can be viewed from different messaging clients or webmail.
 - The **SMTP** protocol: this protocol for sending emails is recognized by all the service providers.

In this lesson, we are going to create a Popup page allowing the user to email a suggestion to the site developer. This popup is as follows:



To do so, we will be using the SMTP protocol. Indeed, this mode is commonly used all over the world.

See the online help for more details about the other methods.



Answer

If the pages have not been created in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (With pages)".

A full corrected application project is also available: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (Answer)".



Example

The "Sending emails" example (unit example), supplied with WEBDEV, provides more details about the management of emails. This example can be accessed from the home window of WEBDEV.

A popup page to send emails

The popup page that we want to create will contain all the controls allowing the user to type the different email elements. A "Send" button will group all the processes used to send the email.

Creating the popup page

- ▶ To create the popup page:
 1. Open the "Full_WEBDEV_Site" project if necessary.
 2. Display the "PAGE_List_of_products" page.
 3. On the "Creation" pane, in the "Containers" group, click "Popup".
 4. The Popup page is displayed in the editor.
 5. Enlarge the Popup page in width and height via the handles.
 6. Save the page ( or CTRL S).

Creating controls used to type the email characteristics

To write an email, the user must have:

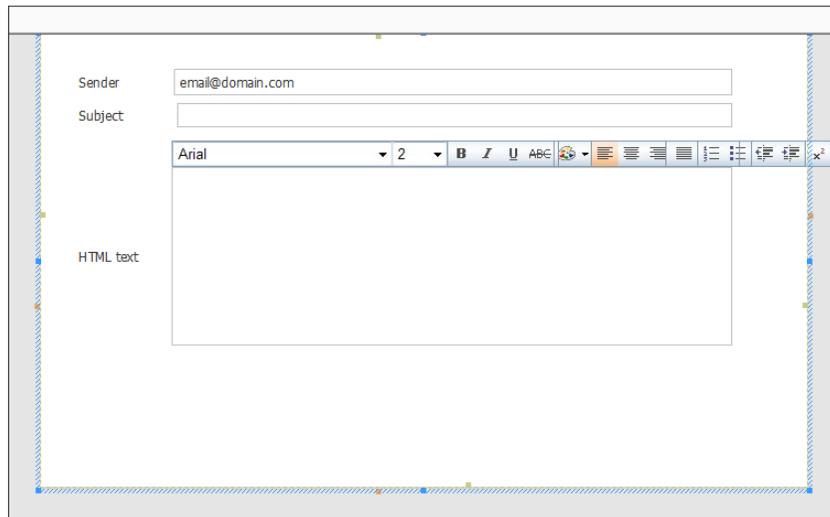
- A control used to type the sender address.
- A control used to type the email subject.
- A control used to enter the email text.

We are now going to add all these controls into our page.

Note: the recipient being the site developer, the corresponding address will be found in the send code of email. Similarly, the parameters of SMTP server will be typed in the code directly.

- ▶ To create the edit control corresponding to the sender address:
 1. On the "Creation" pane, in the "Usual controls" group, expand "Edit".
 2. Select the preset "Email" edit control.
 3. Click in the page: the edit control is automatically created.
 4. Modify the control caption: "Sender".

- ▶ To create the edit control corresponding to the email subject:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click below the "Sender" control: the edit control is automatically created.
 3. Modify the control caption: "Subject".
- ▶ For the message body, we will be using an HTML edit control: the user will have the ability to format the email text via a specific toolbar.
 1. On the "Creation" pane, in the "Usual controls" group, expand "Edit".
 2. Select a preset "HTML text" edit control.
 3. The control shape appears under the mouse cursor.
 4. Click below the "Subject" control: the edit control is automatically created.
 5. Enlarge the control in order for several lines to be visible.
 6. Display the description window of control (double-click the control).
 - On the "General" tab, modify the mode for displaying the HTML toolbar. This toolbar must always be visible.
 - Validate the description window of control.
 7. Reposition the control if necessary.
- ▶ Align the different controls found in the page.



We are now going to create the button for sending emails.

Sending the email

- ▶ To create the send button:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click the location where the button must be created (at the bottom of the page for example).
 3. Select the button and modify its caption (press the [ENTER] key for example). The new caption is "Send".
 4. Edit the code of this button ([F2] key).
 5. Type the following code in the server "Click" process:
 - the code for opening the connection and for starting the SMTP session:

```
MySession is emailSMTPSession
MySession.ServerAddress = "smtpserver"
// Type the address of the SMTP server
MySession.Name = "user_name"
// Type the user name (if necessary)
MySession.Password = ""
// Type the password (if necessary)

// Starts the SMTP session
IF EmailStartSession(MySession)=False THEN
  ToastDisplay("Unable to connect to the SMTP server.",...
    ErrorInfo())
RETURN
END
```

- ▶ This code is using an advanced *EmailSMTPSession* variable. The different properties of this variable are used to define the characteristics of SMTP session. Then, *EmailStartSession* associated with this variable is used to start the session.
 - the code for preparing the email:

```
Suggestion is Email
Suggestion.Sender = EDT_Sender
Suggestion.Subject = EDT_Subject
Suggestion.HTML = EDT_HTML_Text
Suggestion.Message = HTMLToText(EDT_HTML_Text)
Suggestion.Recipient[1] = "developer@mysite"
// Type the recipient address for the suggestions
```

This code is using an Email variable. The different properties of this variable are used to define the characteristics of email to send. This code associates the content of different page controls to the properties of Email variable.

- the code for sending the email:

```
// Send the email
IF EmailSendMessage(MySession,Suggestion) = False THEN
  ToastDisplay("Sending error",ErrorInfo())
ELSE
  ToastDisplay("Thanks for your suggestion.")
END
```

The email is sent by **EmailSendMessage**. All you have to do is pass in parameter the variable containing the characteristics of SMTP session and the variable containing the characteristics of email to send.

If the email is sent, a Toast message is displayed, indicating that the email was sent. A Toast message corresponds to a stealth message.

- the code for closing the SMTP session:

```
// Close the SMTP session
EmailCloseSession(MySession)
```

This code closes the session with **EmailCloseSession**.

6. Close the code editor.

- ▶ Save the page and its code ( or CTRL S).



Notes

The process for sending emails is given for information only. In a real site, we would have to check the parameters typed, process the errors, save a log file, ...

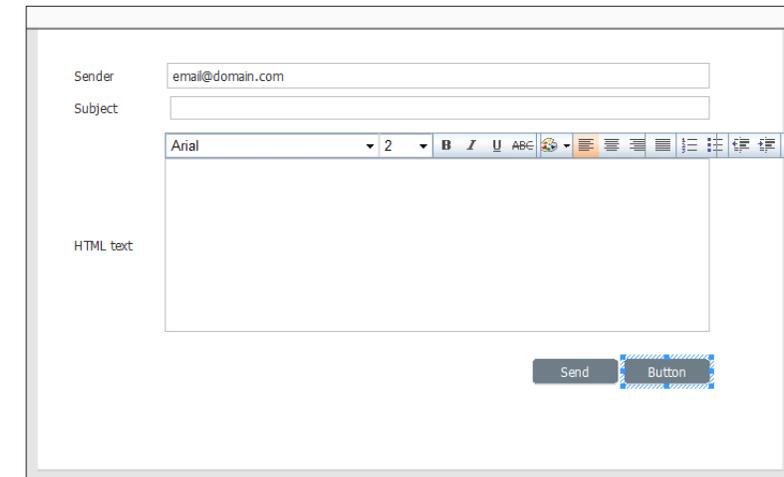
Page improvements

We are going to improve our popup page:

- Add a closing button.
- Start the popup page from the "PAGE_Product_List" page.

Closing the popup page

- ▶ To add a button used to close the popup page:
 1. On the "Creation" pane, in the "Usual controls" group, click "Button".
 2. Click the position where the control will be created in the page (bottom right of "Send" button for example).



3. Select the control and press the [ENTER] key on the keyboard. The caption becomes editable. Enter "Cancel" and validate.

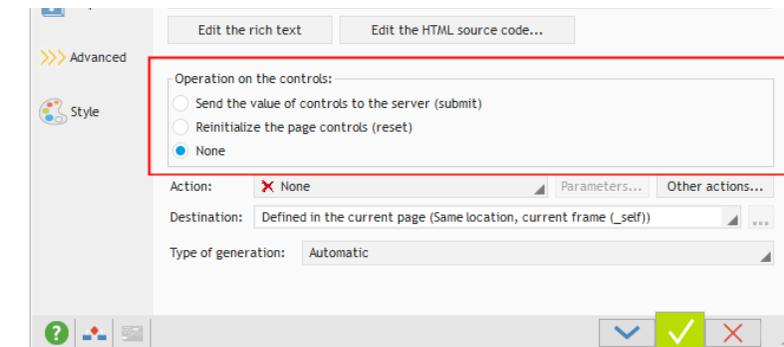
4. Display the processes associated with the button ([F2] key).

5. Type the following code in the browser click code:

```
PopupClose ()
```

PopupClose is used to close the popup.

- ▶ This function is a Browser function, run on the browser only: no return to the server is required. Therefore, we are going to modify the type of "Cancel" button consequently:
 1. Display the description window of "Cancel" button (double-click the control for example).
 2. In the "General" tab, in the "Operations on the controls" area, select "None".



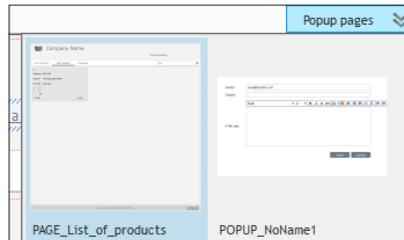
3. Validate the description window of control.

Let's now see how the popup page can be opened.

Opening the popup page

The popup page for sending an email will be opened from the menu of "PAGE_List_of_products" page.

- ▶ Display the "PAGE_List_of_products" page:
 1. In the editor, in the page bar, expand "Popup pages".



2. In the list that is displayed, click "PAGE_List_of_products".

- ▶ We are now going to create a link for sending a suggestion:
 1. On the "Creation" pane, in the "Usual controls" group, click "Link".
 2. Then, click in the top section of the page (above the search control): the Link control is created.
 3. Modify its caption: "Send a suggestion" and validate.
 4. Display the description window of control (double-click the control).
 5. In the "Operations on controls" area, select "None".
 6. In the "Action" area, select "None".
 7. Validate the description window of control.
 8. Display the processes associated with the button ([F2] key).
 9. Type the following code in the browser click code:

```
PopupDisplay(POPUP_NoName1)
```

PopupDisplay is used to display the popup.



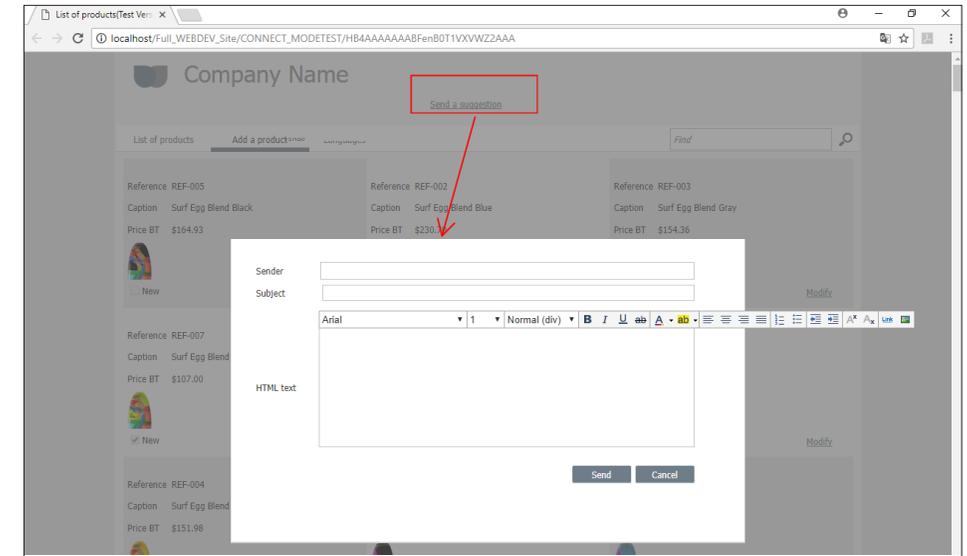
Notes

Why didn't we open the popup from the menu found in the page template?

The popup page that was created is linked to the "PAGE_List_of_products" page. It can only be used from this page.

To use a popup from a page template, the popup must be created from the page template.

- ▶ Save the page and its code ( or CTRL S).
- ▶ Run the page test ( among the quick access buttons) and open the popup for sending a suggestion.



LESSON 5.4. IDENTIFYING THE USER: THE USER GROUPWARE

This lesson will teach you the following concepts ...

- What is the user groupware?
- Integrating the user groupware
- Configuring the user groupware
- Checking the user groupware



Estimated time: 20 mn

Overview

A site can be used by different contributors with different profiles. It is sometimes necessary to define different access levels according to the Web user (customer, salesman, manager for example).

Let's take a simple example: when implementing an e-business site, the site proposes the following features:

- Viewing the price list,
- Modifying the price list,
- Entering orders,
- Managing customers.

The accesses differ according to the Web user. Some examples:

- the Web users can see the price list and place orders.
- the sales people can see the price list, place orders and create new customers.
- the sales directors have access to all the options.

WEBDEV allows you to manage these different access levels via the user groupware.

We are going to include the user groupware in our "Full_WEBDEV_Site" site and to configure it to limit the accesses to the page for product addition.

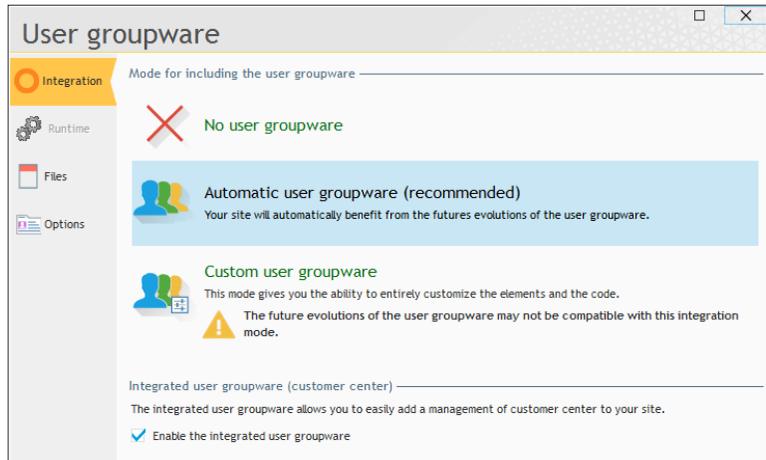


Answer

If the pages have not been created in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (With pages)".

Integrating the user groupware

- To include the user groupware in the "Full_WEBDEV_Site" project:
1. On the "Project" pane, in the "Project" group, click "User Groupware". The window for configuring the user groupware is displayed.



2. In the "Integration" pane, select the following options:
 - Automatic user groupware.
 - Enable the integrated user groupware.



Note

When the integrated user groupware is not used, a standard connection page is proposed. If the user types his login and password, he is allowed to access the site.

When the integrated user groupware is used, a specific control template is made available to the developer. This control template is used to include in the page a link allowing the user to connect. We are going to present this solution in this lesson because it is flexible and it can be easily included.

3. Validate. A message is displayed, indicating that a "supervisor" user is created by default.



Note

A single user exists by default, the supervisor. During the first site start, connect yourself by using the: "supervisor" login and the "supervisor" password.

4. Validate this message. The user groupware is integrated in the project.



Note

In this example, we will keep all the default options. Several options can also be configured. See the online help for more details.

Configuring the user groupware

The user groupware is configured when running the site. This configuration consists in defining the users and their rights on the site pages and controls.



Note

The configuration of users can be performed:

- when developing the application. The necessary data files (regarding the users and their rights) can be installed with the site.
- when the site is installed, by the administrator.

Including the customer section in the site

In this example, we're going to add the connection link to the "PAGE_Product_list" page. The connection link is provided in the form of a control template.



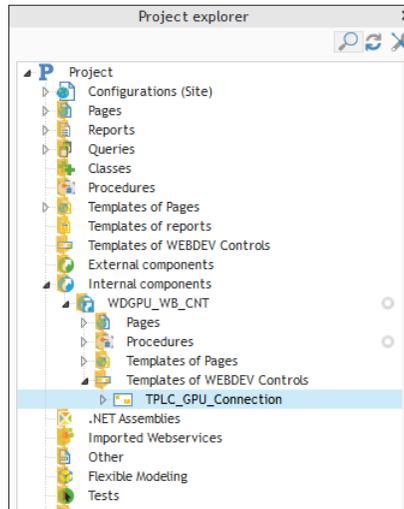
Notes

A control template is a specific page containing several controls. All types of controls can be found in this page. A control template is a file whose extension is "WDT".

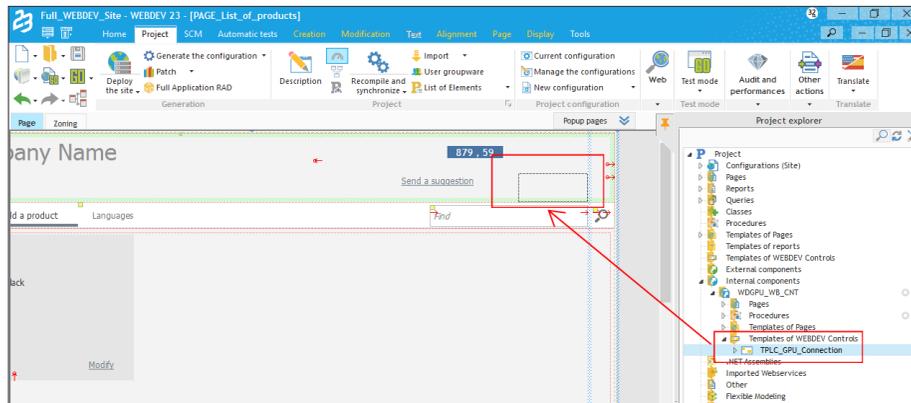
The first benefit of a control template is the reusability. A control template found in a project can be re-used in any page of the project. The modifications are performed once only in the control template and the modifications are automatically applied by WEBDEV.

The control templates can be overloaded: code can be added, the controls can be moved in the page that is using the control template. The controls can be modified.

- To include the connection link:
1. Display the "PAGE_List_of_products" page in the editor.
 2. In the project explorer, in the "Internal components" folder, expand "WDGPU_WB_CNT". This internal component was included in the project when implementing the integrated groupware. It contains all the necessary elements for its management, including the control template used to connect.
 3. Then, expand "WEBDEV control templates".



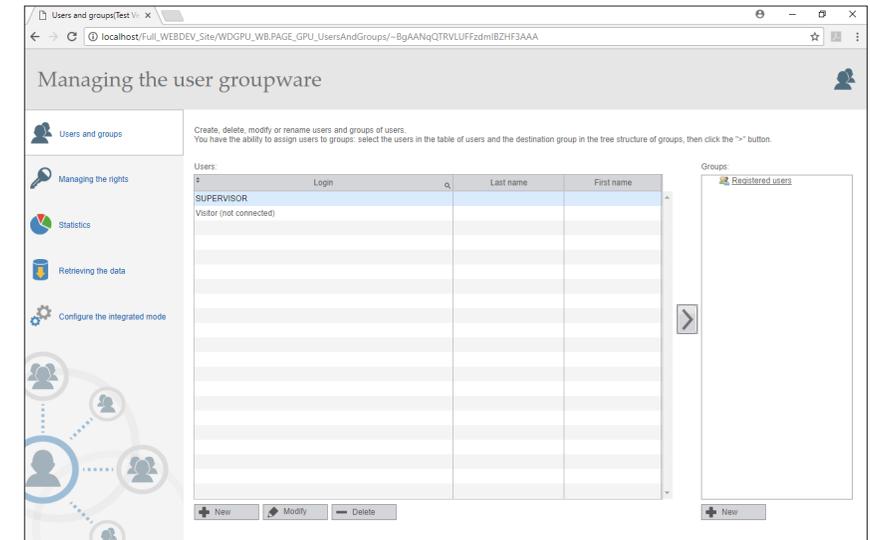
4. Select the control template named "TPLC_GPU_Connection" and perform a Drag and Drop into the "PAGE_List_of_products" page. Drop the control template above the search control.



5. The control template is integrated. The "Connection" link appears.

Site test

- ▶ Let's now run the test of our site:
 1. Run the project test (GO among the quick access buttons).
 2. The page corresponding to the list of products is displayed.
 3. Click the "Connection" link. The connection page is displayed.
 4. Connect yourself as supervisor:
 - Login: supervisor
 - Password: supervisor
 5. Validate.



6. The page for groupware configuration is displayed.

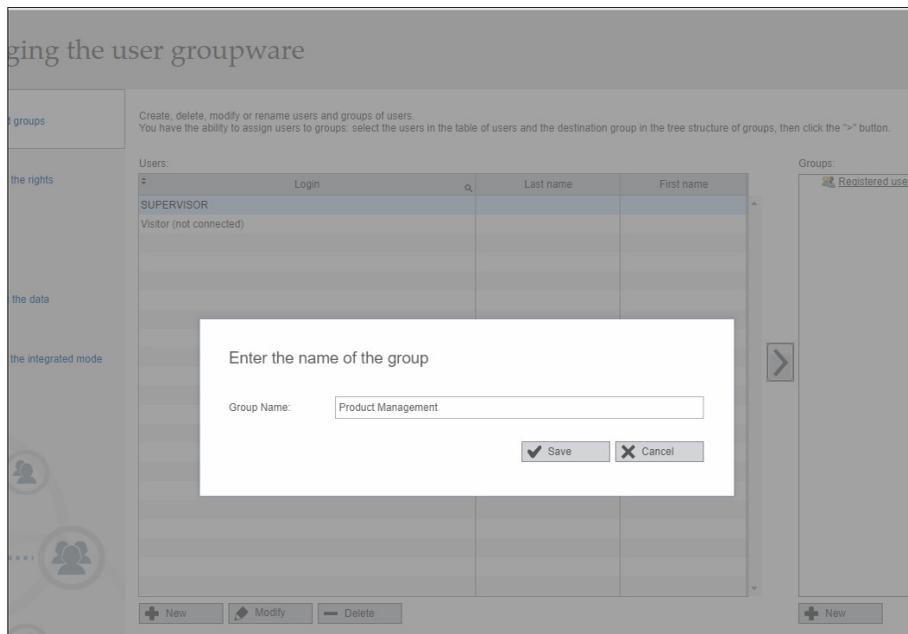
Creating users and groups

To configure the user groupware, we are going to create a "Product Management" group. This project will group all the users authorized to modify and add products in the site.

We then going to create a user named Alice, associated with the "Product management" group.

► To create a new group of users:

1. Click the "New" button found below the "Groups" area. The screen for entering a new group is displayed.
2. Type the group name: "Product Management".



3. Click the "Save" button. The "Product Management" group appears in the list of groups defined for the user groupware.

► To create a user:

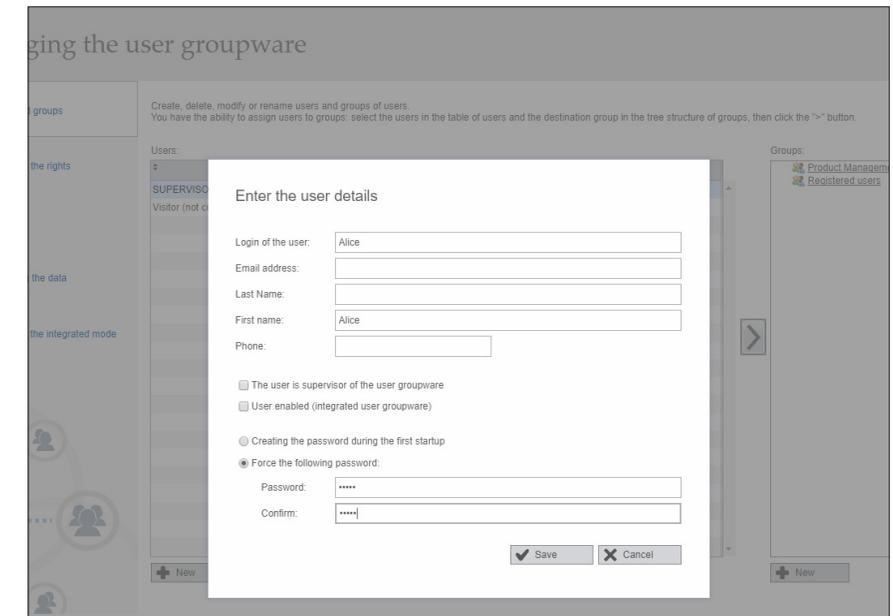
1. Click the "New" button found below the "Users" area. The screen for entering a new user is displayed.
2. Type the following information:
 - Login: Alice
 - FirstName: Alice
 - User enabled (the user account is automatically enabled in the site and it can be used immediately).
 - Password: Alice. You can give the user the ability to choose his password during the first connection.



Note

Information regarding the user:

- In the information regarding the user, only the login is required.
- You also have the ability to define that the user is a groupware supervisor. In this case, he or she will be authorized to modify the users, the groups and the rights.



3. Click the "Save" button. The "Alice" user appears in the list of users defined for the user groupware.

► To associate the user named "Alice" with the "Product Management" group:

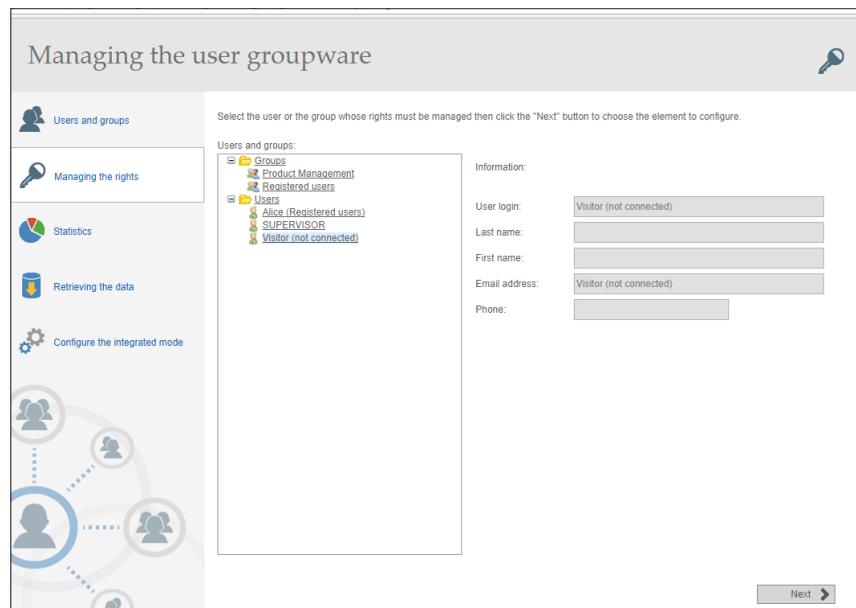
1. Select the "Alice" user in the page.
2. Select the "Product Management" group.
3. Click the addition arrow.
4. The association is performed.

Defining rights

We are now going to define the access rights to the menu for product addition.

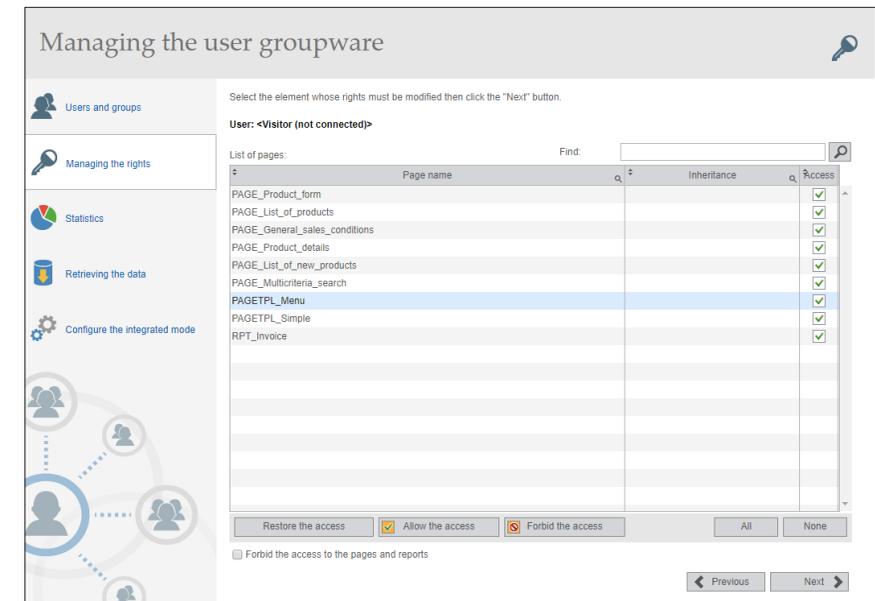
A good security practices consists in denying access by default and in allowing access to the selected groups.

- ▶ We are going to define rights for the unconnected users (which means not identified by the groupware). These rights will be used by default when starting the site, as long as the user is not connected.
- ▶ To define the rights:
 1. Click "Manage the rights" on the left of the page.
 2. To prevent all users from accessing the page for product addition, select the "Guest (not connected)" user. Each unconnected user is automatically associated with this user.

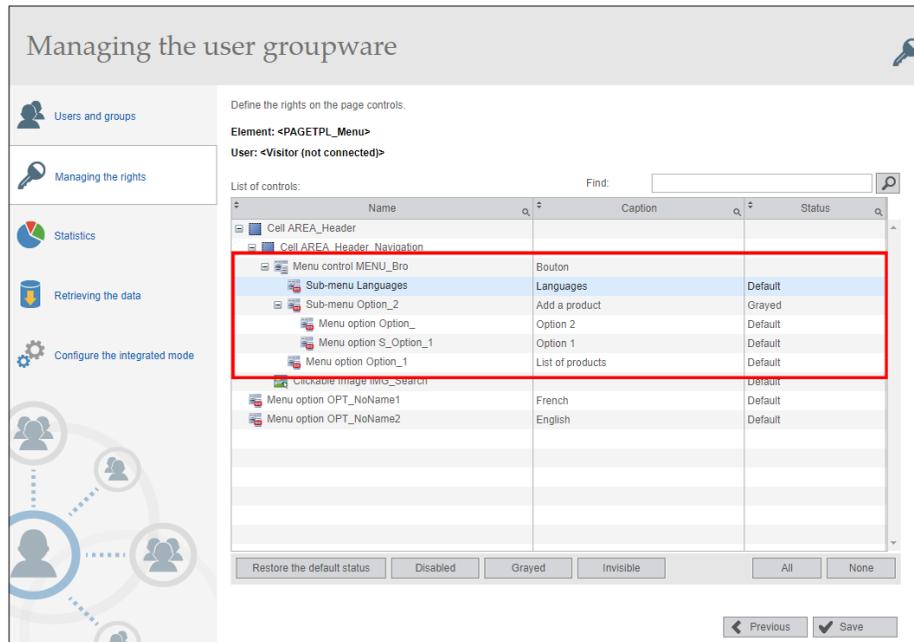


3. Click "Next".

4. The page that is displayed is used to select each page, page template or application report.



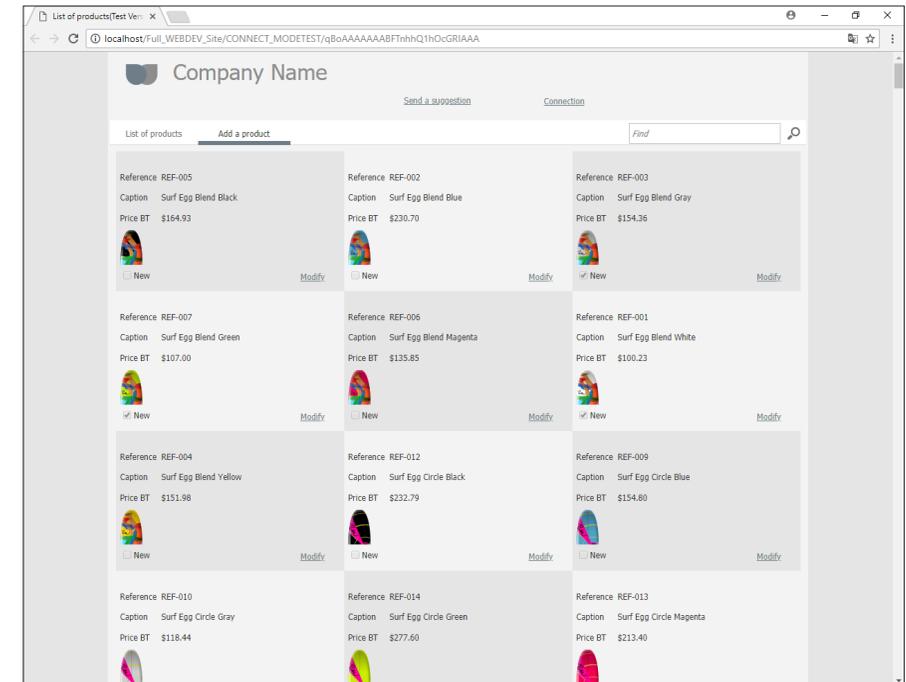
- For each page, page template or report, you can specify whether the element will be accessible or not by the group.
 - For each page or page template, you can say whether the controls will use the site's behavior (default) or will be inactive, hidden or grayed out.
- ▶ In our case, the "Add a product" link is found in the "PAGETPL_Menu" page template:
1. Select the "PAGETPL_Menu" page template in the list. The rights defined on the page template will be applied to all the pages that use the template.
 2. Click "Next".
 3. The window for configuring rights on the page controls is displayed.
 4. Select the menu option to configure: "Sub-menu Option_2"



5. Click the "Grayed" button.
6. Save the modifications by clicking "Save".
7. Close the browser.

Site test

- We are now going to run the site test.
 1. Run the project test ( among the quick access buttons).
 2. By default, the "Add a product" option is grayed.
 3. Click "Connection" and use the login "Alice" (and the password "Alice"). Validate.
 4. The "Add a product" option becomes visible.



5. Close the browser.

Disabling the management of user groupware

The user groupware will no longer be used in the rest of this tutorial. Therefore, you can disable the management of user rights in this project:

1. On the "Project" pane, in the "Project" group, click "User groupware".
2. In the window that is displayed, in the "Integration" tab, select "No user groupware".
3. Validate.

LESSON 5.5. MANAGING THE MULTILINGUAL FEATURE

This lesson will teach you the following concepts ...

- What is a multilingual site?
- Creating a multilingual site, step by step.



Estimated time : 50 mn

What is a multilingual site?

A multilingual site is a site that proposes an interface in several languages: English, French, German or any other language.

The same site created with WEBDEV can propose up to 64 different languages.

We are going to handle a project that can be run in English or in French, according to the user's choice.

The following steps are used to transform a site into a multilingual site:

- Choosing the project languages.
- Localizing the project elements (pages, reports, controls, ...).
- Localizing the messages found in the code.
- Programming the change of language in the site.

We are going to apply these different steps to the "Full_WEBDEV_Site" project. This project, available in English, will be translated in French.

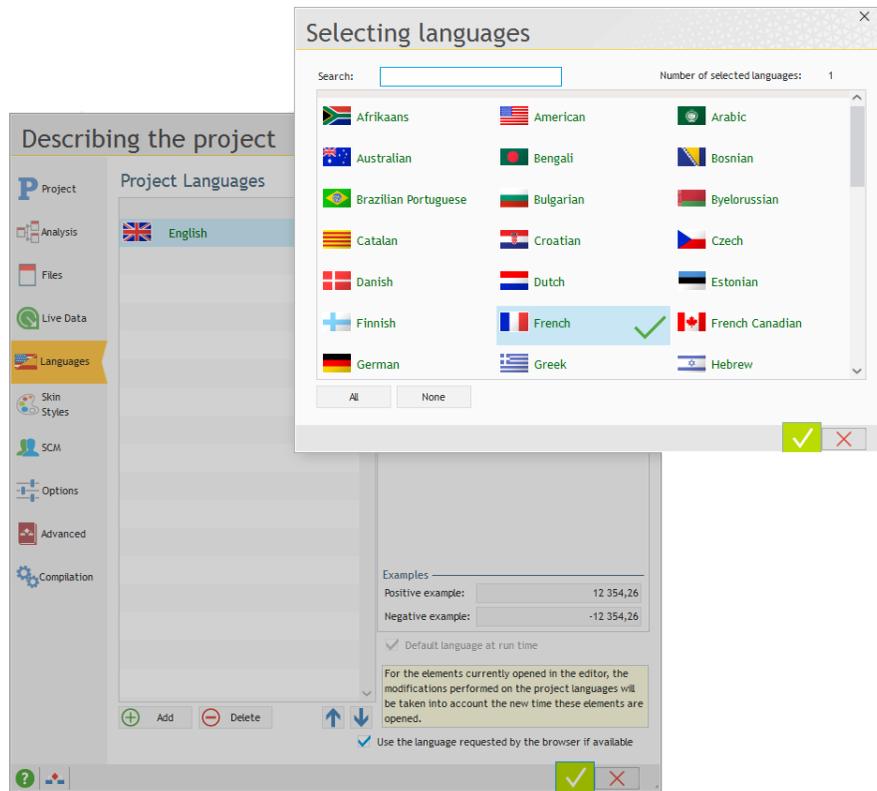


Answer

If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (With pages)".

Choosing the project languages

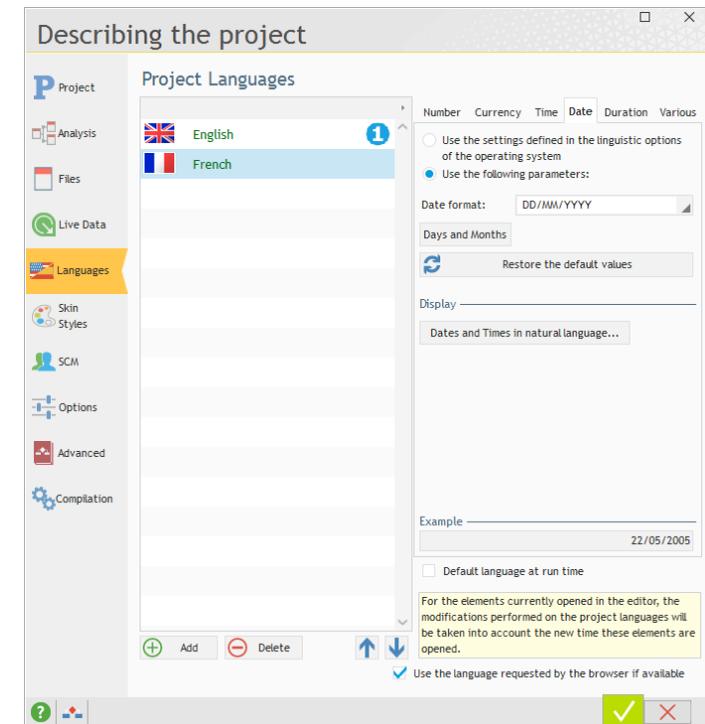
- ▶ The first operation consists in choosing the project languages.
 1. Display the project description: on the "Project" pane, in the "Project" group, click "Description".
 2. Click the "Languages" tab. Our project will support English and French.
 3. Click the "Add" button. The window for selecting languages is displayed.



4. Click "French". A checkmark is displayed on the right of language.
5. Validate. The "French" language appears in the list of project languages.

► The "Languages" tab can also be used to configure the linguistic options regarding the numbers, the currencies, the dates, ... for the selected language. Let's see an example:

1. Click the "French" language.
2. Select the "Date" tab.
3. Specific linguistic options are used by default: you have the ability to define the date format as well as the translation used for the days and months. If you select "Use the parameters defined in the linguistic options of operating system", the parameters used in deployment will be the server parameters (and not the user parameters).
4. Keep "Use the following parameters".



Notes

In the linguistic options, you have the ability to choose the text direction for the language ("Various" tab, "Text direction" option). This allows you to create interfaces with a language written from right to left.

- Validate. A message proposes to synchronize the different project elements. Answer "Yes". All the project elements opened in the editor (pages, reports, ...) are closed and the additional languages are added to these elements.



Notes

GUI errors may occur. We will ignore them for now. These errors will be processed later in this tutorial.

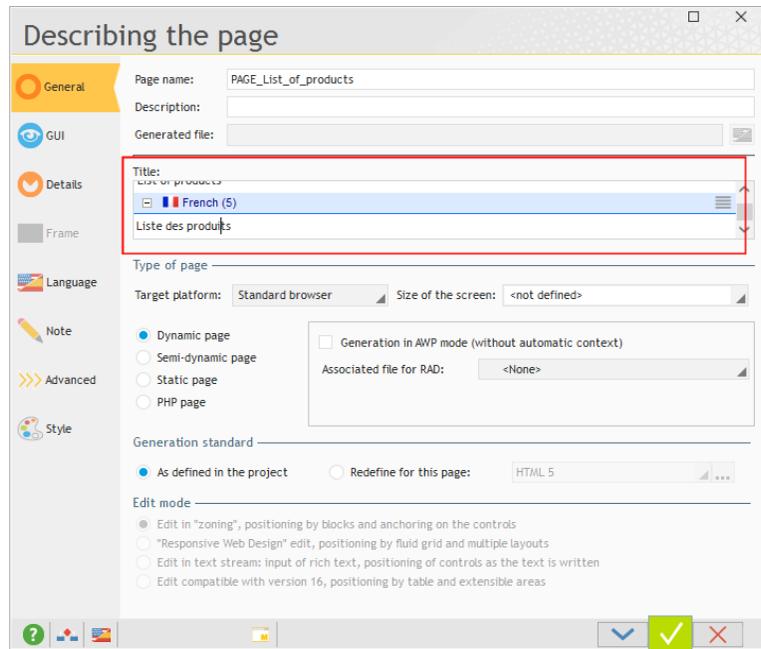
Localizing the project elements

All the project elements can become multilingual elements: pages, reports, ...

We are going to modify some elements in the "PAGE_List_of_products" page to present the different methods that can be used.

We are going to see how to modify:

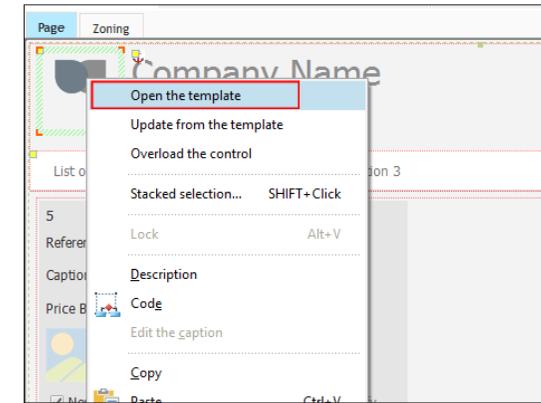
- the logo image used in the PAGE_List_of_products page.
 - the captions of controls in the PAGE_List_of_products page.
 - the menu options.
 - a message displayed by the WLanguage code.
- ▶ Open the "PAGE_List_of_products" page in the editor (double-click its name in the "Project explorer" pane for example).
 - ▶ Validate the update of template if necessary.
 - ▶ First of all, check whether the PAGE_List_of_products page is associated with the different languages defined in the project:
 1. Display the page description ("Description" from the popup menu of page).
 2. Select the "Language" tab: the two languages selected in the project are displayed.
 3. Select the "General" tab: the page title must be translated.



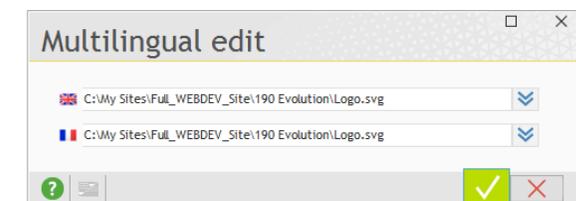
4. Replace the text found in the French section by "Liste des produits".
5. Validate the window.

Localizing an image

- ▶ To change the logo image used in the PAGE_List_of_products page according to the runtime language:
 1. Open the "PAGE_List_of_products" page if necessary.
 2. The logo image is found in the template associated with the page. Therefore, the associated page template must be opened:
 - Click the logo and display the popup menu.
 - Select "Open the template".



- The page template appear inside an orange border.
3. Display the description window of logo (double-click the control).
 4. In the "General" tab, on the right of "Image" area, click the button . Select "Multilingual" from the menu that is displayed.
 5. The window for managing multilingual images is displayed.



6. A different image can be used for each language:
 - Click the button .
 - Select "Browse" from the menu that is displayed.

This feature is very useful if you are using images containing text. The same image is used in our case.

7. Validate the window for multilingual edit.
8. Close the description window of control.
- ▶ As we are positioned in the page template, we are going to translate the "List of products" menu option.

Localizing the menu

The menu options can be translated like the other controls via the description window of option, or from the page editor.

- ▶ For our example, we are going to translate the "List of products" menu option.
 1. Select then click the menu found in the page template.
 2. The menu becomes editable and a yellow border appears.
 3. Select the "List of products" option.
 4. Display the description window of option: display the popup menu and select "Option description".
 5. In the description window, type the option in French: "List of products".
 6. Validate.

The options can also be translated in the page editor.

- ▶ To translate the "Add a product" option:
 1. On the "Display" pane, in the "Options" group, expand "Language displayed" and select the language that must be viewed in the editor (French in our case).
 2. The menu options are displayed in the selected language. If no translation corresponds to the selected language, the menu options are displayed in English.
 3. Select the "Add a product" option.
 4. Press the [SPACE] key on the keyboard: the caption becomes editable.
 5. Type the caption in French: "Ajouter un produit".
 6. Press the [ENTER] key to validate the caption.
 7. Press the [ESC] key to exit from the edit mode.
 8. Save the page template ( or CTRL S).
 9. Restore the English display mode: on the "Display" pane, in the "Options" group, expand "Displayed language" and select "English".
 10. Update the pages that use the page template by clicking  in the orange bar. Validate the update window.
 11. Close the page template displayed in the editor.

Localizing controls

A control can display various information to the user:

- a caption,
- an image, ...

This information must be translated. This information is accessible in the different tabs of the description window of control.

- ▶ To translate the "Modify" link found in the "PAGE_List_of_products" page:
 1. Select the "Modify" link.
 2. Display the description window of control ("Description" from the popup menu).
 3. Type the caption in French: "Modifier".
 4. Validate.
 5. Save the page ( or CTRL S).

Localizing a programming message

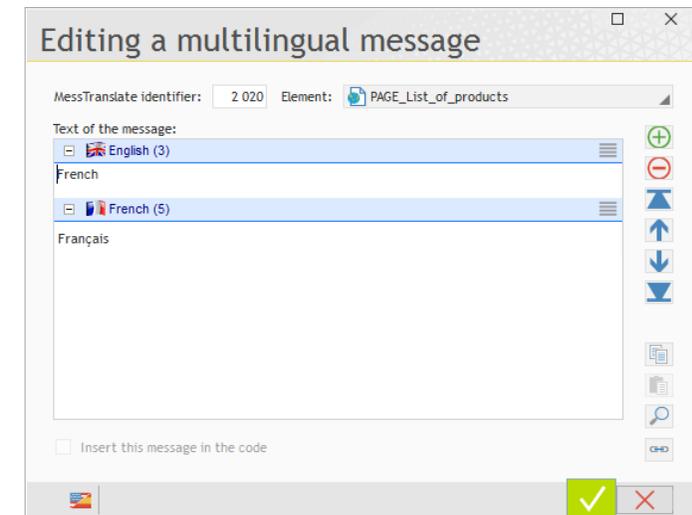
All the messages found in your program can be translated into several languages.

Let's see how to translate a programming message:

1. Display the code editor (press [F2] on the PAGE_List_of_products" page).
2. Type the following code in the "Global declarations" process:

```
sMyString is string = "French"
```

3. To translate this type of message, position the cursor in the "French" string and press the [CTRL T] keys. You also have the ability, on the "Code" pane, in the "Languages" group, to expand "Translate the strings" and select "Translate the messages".
4. The window for editing a multilingual message is displayed. This window allows you to translate all the messages of your program into all project languages.
5. In the "French" area, type "Français" and validate.



6. The icon  as well as a digit appear in the code editor. These icons indicate that the multilingual message exists in 2 languages.
7. Close the code editor.

The translation tools

Some elements of our application have been translated manually.

Several methods can be used to translate this information:

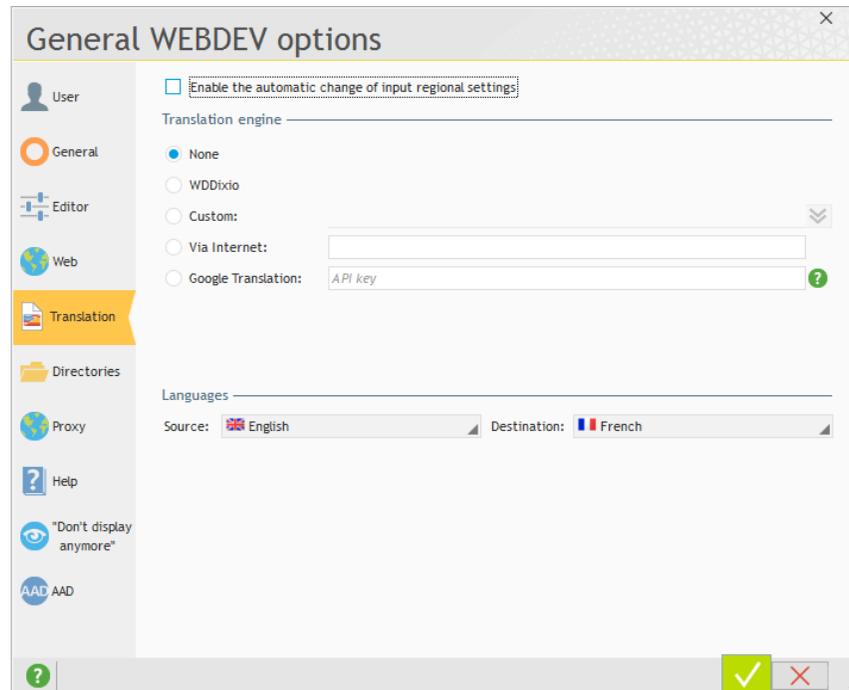
- a direct translation of messages performed in the different editors: this is the feature that was just used to translate some elements of our site. This translation can possibly use a translation tool, Google Translate (providing that you own a license), ...
- an "industrialized" translation performed via an external tool (WDMMSG and WDTRAD).

Direct input of translations

The translations are typed in the product interface directly: this is the method that was used until now.

If you want to use a translation software or a translation site, WEBDEV can be configured to use this software:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
2. Display the "Translation" tab.



3. Specify:

- Whether the regional settings must be automatically enabled according to the language used for the input. In this case, if the language requires a specific character set, this character set will be automatically selected.

- The software or the site that will be used for translation. You have the ability to use WDDixio, translation dictionary supplied with WDMMSG (see next paragraph), a specific translation software or site, or Google Translate (see the online help for more details).
- The supported languages.

4. When the translation parameters are defined, you have the ability to use the button  found in the different description windows of project elements: this button allows you to use the software defined for the translation.

Translation with WDMMSG and WDTRAD

An optional tool named **WDMMSG** is available, allowing you to:

- check out all the project messages (caption of controls, code message, title of windows, ...) in order to translate them,
- check in the translated messages.

The messages to translate are checked out:

- in a text format that can be configured to be used by most of the translation tools.
- in HFSQL format.

WDMMSG is also supplied with a tool for computer-assisted translation, WDTRAD. WDTRAD is used to easily enter all the translations for the multilingual information of a project.

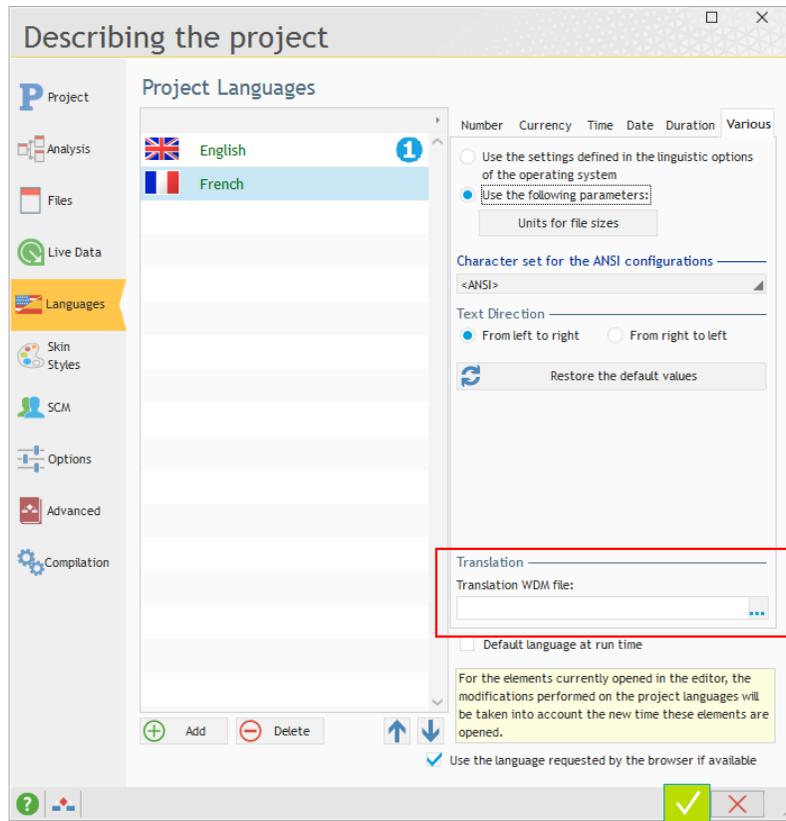
Contact PC SOFT Sales Department for more details about **WDMMSG** and **WDTRAD**.

Other elements to translate: the framework messages

Various information and messages are found in the WEBDEV framework. For example, the names of days and months used by the functions for date management come from the WEBDEV framework. To translate one or more libraries of this framework, you must use WDINT (optional tool supplied with WDMMSG).

This software is used to get a WDM extension file containing all the translations of libraries. To use this file in your application:

- you can use **LoadError**.
- you have the ability to include the file to the project description in the "Languages" tab. All you have to do is select the requested language and select the "Various" tab.



Contact PC SOFT Sales Department for more details about WDINT.

Programming the change of language

By default, the project is run in the runtime language defined for the project, in the "Languages" tab of project description ("Description" in the "Project" pane).

In a site, the language can be chosen via a menu option. **Nation** used in the process associated with the menu option allows you to change the language for the application currently run.

Adding a menu option

► To add a menu option:

1. Open (if necessary) the "PAGE_List_of_products" page in the editor (double-click its name in the project explorer).
2. Click the menu and select "Open the template" from the popup menu.
3. In the page template, select then click "Option 3".
4. The menu becomes editable and a yellow border appears.
5. Press the [SPACE] key on the keyboard: the caption becomes editable.

6. Type the "Languages" caption and validate.
7. Select the "Languages" option that was just created.
8. Display the popup menu (right click) and select "Insert a sub-menu".
9. Press the [Enter] key to edit the caption.
10. Type the caption of first sub-option: "English".
11. Press the [ENTER] key
12. Press the [SPACE] key and type the caption of second sub-option: "English".

► As we are positioned in the page template, let's delete the last two menu options:

1. Select the "Option 5" option. Display the popup menu of option and select "Delete". The option is deleted.
2. Repeat this operation to delete the option 4.

We are now going to type the WLanguage code required to change language.

Programming

► To type the code for managing languages:

1. Select "Languages .. English" in the page template displayed in the editor.
2. Display the popup menu (right mouse click). Select "Code".
3. Type the following code in the server code of menu option:

```
Nation(nationEnglish)
PageUse(CurrentPage())
```

4. Close the code window.
5. Select "Languages .. French" in the page template displayed in the editor.
6. Display the popup menu (right mouse click). Select "Code".
7. Type the following code:

```
Nation(nationFrench)
PageUse(CurrentPage())
```

In this code:

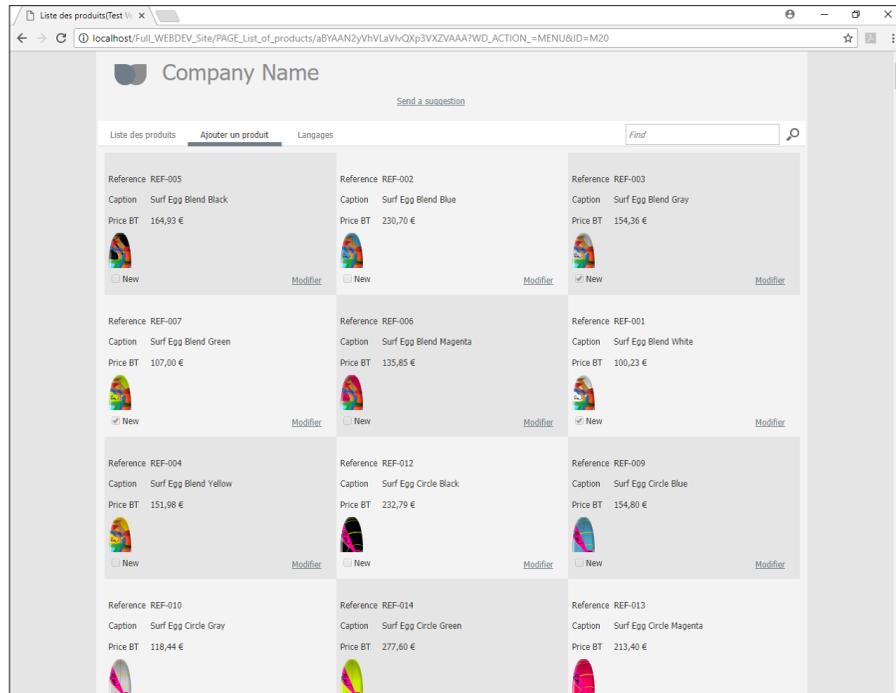
- **Nation** is used to change the runtime language of site. The constants passed in parameter allow you to specify the language to use.
 - **PageUse** is used to reload a page (in our case, the current page) in order to take the change of language into account.
8. Close the code window.
 9. Press the [ESC] key to exit from the edit mode.
 10. Save the page template ( or CTRL S).
 11. Update the pages that use the page template by clicking  in the orange bar. Validate the update window.
 12. Close the page template displayed in the editor.

Project test

Some application elements being translated, let's now check the change of language.

► To run the site test:

1. Run the project test ( among the quick access buttons). The page is displayed in test mode in English.
2. Select "Languages .. French".
3. The elements that have been translated are displayed in French:



4. Close the browser

LESSON 5.6. THE REFERENCING

This lesson will teach you the following concepts...

- Referencing wizard
- Properties of pages
- Site map path
- Physical naming of pages



Estimated time : 20 mn

Overview

This lesson presents the best practices to adopt in order for your Internet site to be properly referenced by the search engines. Indeed, having an Internet site is a good thing, but having a well-referenced site is even better!



Note

This section concerns the developers of Internet sites only. The referencing is not essential if you are developing an Intranet site.



Answer

If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (With pages)"

Reminder : AWP mode

At the beginning of this tutorial, we presented the differences between a WEBDEV site in Classic mode and a WEBDEV site in AWP mode.

Only the AWP mode allows you to have page URLs that are independent, fixed and directly addressable.

To be referenced, an Internet site must necessarily be developed in AWP mode.

The available methods

To help you optimize the referencing of your AWP pages, WEBDEV proposes several tools:

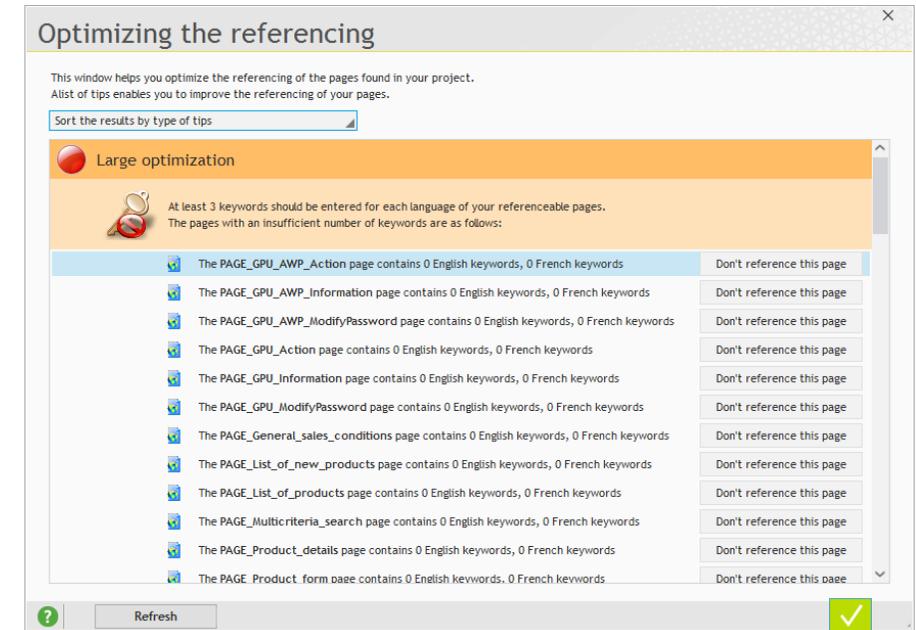
- A referencing wizard.
- The page properties.
- The site map path.

We are going to present these different methods.

Referencing wizard

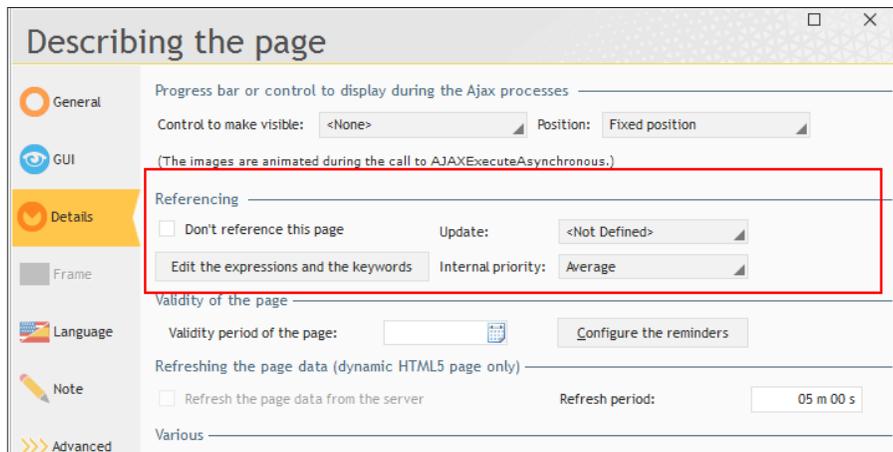
WEBDEV is supplied with a referencing wizard.

- ▶ To start this wizard, in the ribbon, on the "Project" pane, in the "Web" group, expand "Referencing" and select "Optimize the referencing".



The wizard proposes tips for optimizing site referencing. The most important tips are as follows:

- **Title of pages:** The title of pages must reflect the content and it must contain the important keywords. The page title is defined in the description window of the page, for each language.
- **Keywords of pages:** We advise you to associate keywords to a page. To define these keywords:
 - Display the description window of "PAGE_List_of_new_products" page: display the popup menu and select "Description".
 - Display the "Details" tab.
 - In the "Referencing" section, click "Edit the expressions and keywords".



- **Alternative text:** The alternative texts are texts that replace images when these ones are not displayed (no matter whether the Web user explicitly asked no to display the images, or whether the images are not loaded yet, or whether the Web user is using a browser in text mode or a screen reader for poor-sighted persons).

The alternative texts describe the image and they are indexed by the search engines. Therefore, they must be chosen with great care. The alternative text of an image is defined in the description window of Image control, in the "Help" tab.



Properties of pages

As already seen, the referencing wizard indicates several modifications that can be performed in the description window of pages:

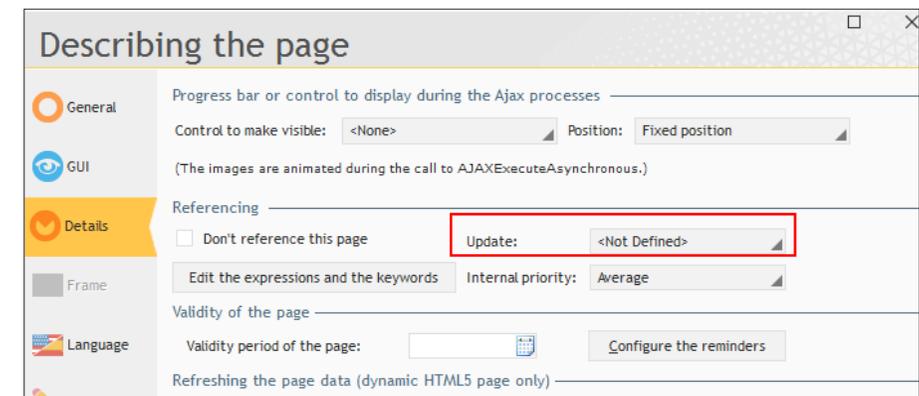
- Page title,
- Keywords of pages, ...

Other properties that influence the referencing can be defined in the description window of the page:

- Frequency of update,
- Importance in the site.

Frequency of update

The update frequency tells the search engine how often the page should be reindexed.



Whenever possible, you should indicate a frequency close to reality:

- If the page changes more often that specified, some versions may not be indexed.
- If the page changes less often, the search engine crawl will add an unneeded burden to the server and indexing mat be penalized.

Importance in the site

This criterion lets the search engine know which page to offer first to the user when several pages of the site match a search.

Site map path

The site map path defines the organization of your site. In referencing, it has two uses:

- allow you to display the "Site map" control to guide the Web users within the site.
- generate the sitemap.xml file.

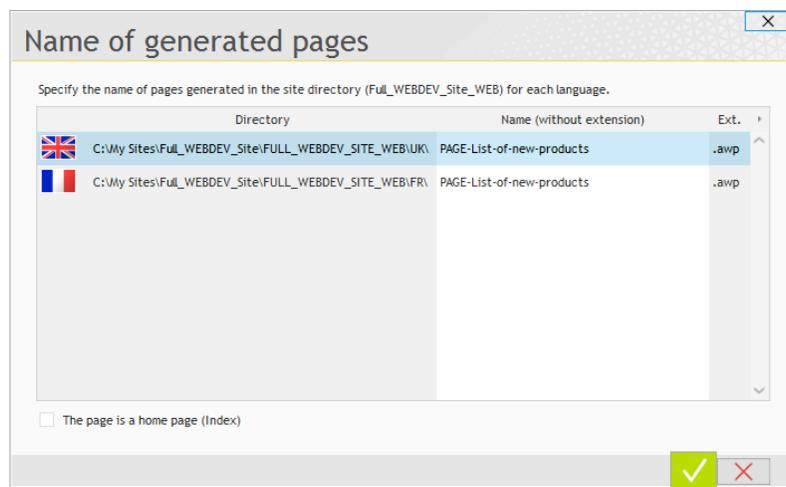
- ▶ To define the site map path:
 1. In the ribbon, on the "Project" pane, in the "Web" group, click "Site map path".
 2. A window is opened, allowing you to build the tree structure of your site. An automatic build mode can help you create a tree structure by automatically detecting the links between pages. The site map path can be configured later.
 - ▶ The sitemap file is automatically built by WEBDEV from the site map path. It is generated in the _WEN directory of your site and is named _sitemap.xml.
- The sitemap.xml file must be provided to the webmaster's search engine tools so they can index the content of your site.

Physical naming of pages

To further improve the referencing, you can set the physical name of the AWP pages so they're different for each language (and different from the logical name as well).

For example, a page displaying a sales dashboard whose logical name (the name used in your programs) would be PAGE_SalesDash can be generated in English in the 'sales-dashboard.awp' file and in the 'tableau-de-bord-ventes.awp' file in French.

- ▶ To configure the generation name of a page:
 1. Display the description window of "PAGE_List_of_new_products" page.
 2. In the "General" tab, click the "Language" button  on the right of "Generated file" control.
 3. A window used to configure the name of generated pages is displayed:



4. You can change the page name and validate.

LESSON 6.1. DEPLOYING A SITE

This lesson will teach you the following concepts...

- How to deploy?
- Required configuration



Estimated time : 30 mn

Overview

When the WEBDEV site is developed, it must be deployed on a server in order to make it accessible to the Web users.

We are going to present the different steps required to deploy a WEBDEV site.



Note

This section presents the deployment on a Windows server. If you want to use a Linux server, see the documentation about the WEBDEV application server for Linux or the online help.

Several methods can be used to deploy a dynamic WEBDEV site:

- Deployment by physical media (CD, ...). This deployment can be stand-alone: in this case, a Web server and a limited WEBDEV application server are also installed by the setup.
- Remote deployment from the development computer (by FTP).
- Remote deployment from a management computer (by FTP) via a "deployment package".
- Deploying the site via the test hosting service of PC SOFT.
- Deployment in PC SOFT Cloud.

Required configuration

For the deployment on the server, the following elements must have been installed and configured:

- A Web server,
- An FTP server,
- A WEBDEV application server (a 10-connection version is supplied with WEBDEV).

When installing WEBDEV, you have the ability to install the test version of WEBDEV application server. This gives you the ability to test the deployment of your dynamic sites.

To simplify the deployment operations and to allow you to directly test the deployment of your Web site, we will:

- Install a WEBDEV Application Server 10 connections on a Windows computer. This application server must be installed on a computer other than the development computer. WEBDEV Development must not be installed on this computer.
- Deploy the site remotely (by FTP).



Note

If you already have the parameters for accessing the server on which your site will be deployed, there is no need to install the WEBDEV Application Server 10 Connections. The setup can be directly performed on the server by FTP.

Installing and configuring the "WEBDEV Application Server - 10 Connections"

The setup program of "WEBDEV Application Server - 10 connections" is available :

- for download on the www.windev.com site, in the "Download" section, "WEBDEV upgrades", "Application server (deployment)".
- on the setup DVD of WEBDEV: start "Menu.exe" and select "Install a WEBDEV Application Server - 10 connections"..

Installing the "WEBDEV Application Server - 10 Connections"

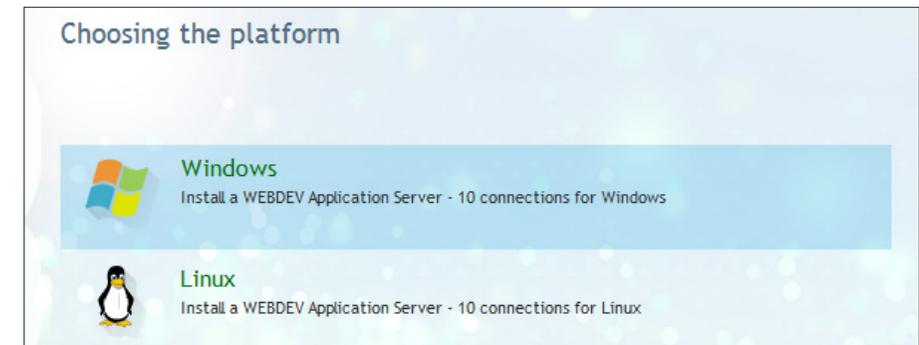
The setup steps are as follows:

1. Accept the license agreement. Go to the next wizard step.
2. Choose the "Windows" platform. Go to the next step.

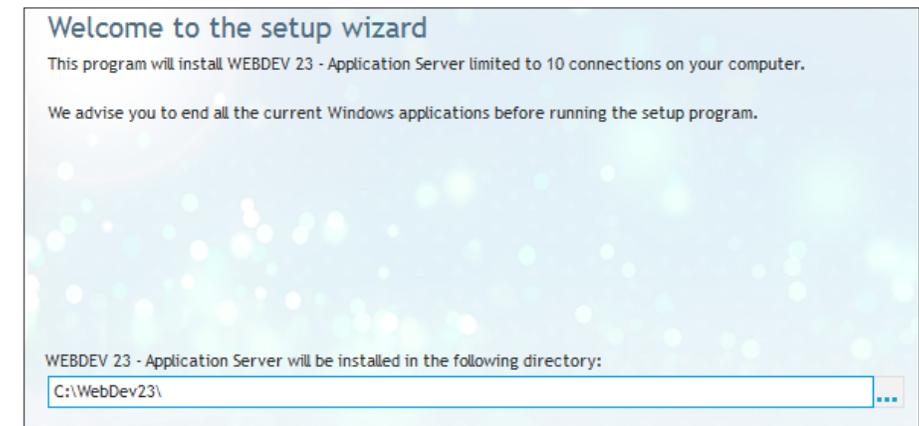


Note

This section presents the deployment on a Windows server. If you want to use a Linux server, see the documentation about the WEBDEV application server for Linux or the online help.



3. Select the setup path of application server ("C:\WEBDEV23" by default).



4. The next step is used to specify the advanced parameters of WEBDEV application server. Keep the option "Use the default parameters".



5. If an earlier version of WEBDEV application server was already installed on the same computer, the wizard is used to define the management of sites and webservices in earlier versions. Keep the option "Allow the deployment of sites in version 23" and go to the next step.
6. The next step is used to manage the hosting of SaaS sites. The SaaS activation automatically installs a site and a Webservice to manage your SaaS sites. See the online help for more details. Validate this step.
7. The next step requires a secure connection for accessing the SaaS administrator.
8. Validate the different setup steps.
Note: The IIS Web server is automatically installed if no valid Web server is detected by the setup (in case of failure, the Apache Web server will be proposed for setup).
9. Select the virtual Web servers on which the WEBDEV application server must be installed. The default Web site is sufficient. Validate.
10. Keep the selected options and check "Start the WEBDEV administrator". Validate.
11. The administrator is automatically started.



Note

The PDF file named "WEBDEVDeployment.pdf" is installed with the WEBDEV application server (10 connections). This file may help you solve the configuration problems linked to the access rights on the server.

Note: If "Place the icons in the Start menu" was checked at the end of setup, this file will be accessible via the "Start" menu.

Configuration via the Hosting Control Center

The Hosting Control Center is a tool designed to automatically configure the Application Server, the Web Server (IIS) and the FTP Server (IIS).

To use the Hosting Control Center:

1. Start the Hosting Control Center from the "Start" menu of Windows.
2. In the "Hosting parameters" group, click "Directories". The different options of this group allow you to define the parameters that will be used by default for your hosting server.

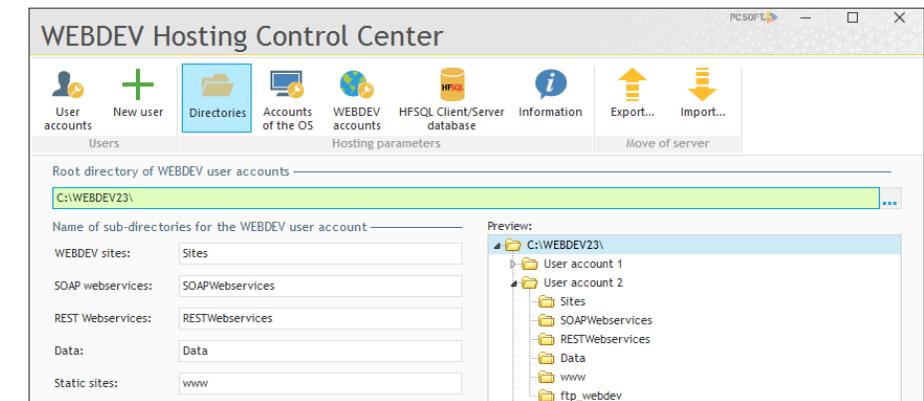
3. Indicate where the sites will be installed. Choose the root directory where the sub-directories of WEBDEV accounts will be created.



Note

Use a directory local to the computer. If you want to use a network directory, a UNC path must necessarily be specified. The Internet guest of the computer must have access to this path without having to authenticate.

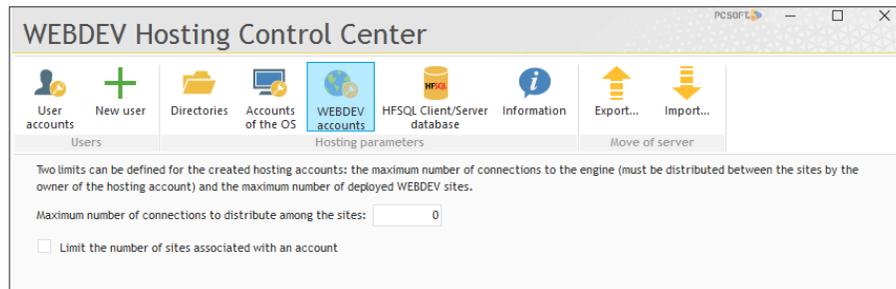
You can choose the name of sub-directories that will contain the sites, the webservices and the data.



4. Click "Accounts of the OS" and define the groups where the Windows users created for the deployment will be assigned.
For the deployment, you can create a group or use the standard "Power Users" group.
For running sites, a good practice consists in using the "IIS_IUSRS" group (on the Windows versions where it exists).

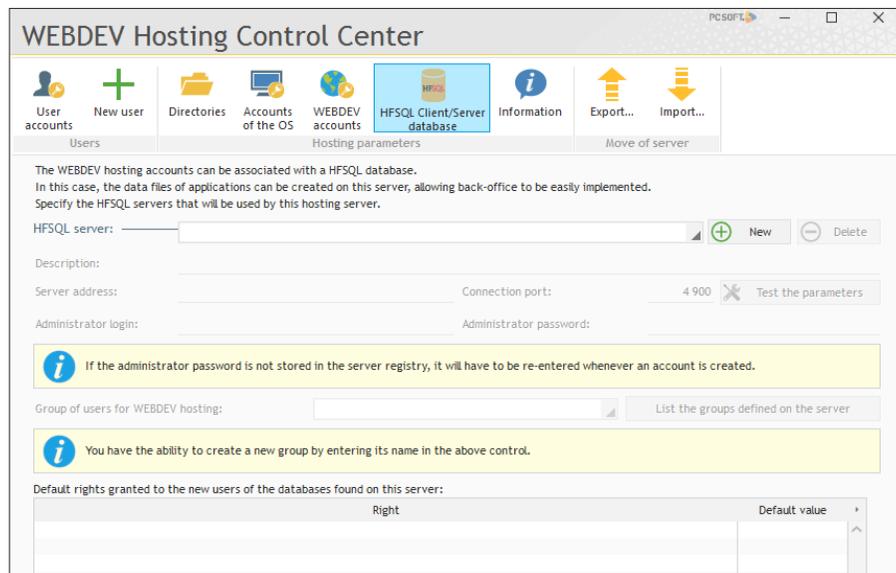


5. Click "WEBDEV accounts" to specify the limitations of resources that will be applied to the WEBDEV accounts:



- Maximum number of connections that must be shared between the sites (0 corresponds to an unlimited number),
- Limitation regarding the number of sites to associate with an account, ...

6. Click "HFSQL Client/Server database" to configure the creation of a HFSQL database whenever a user is created.



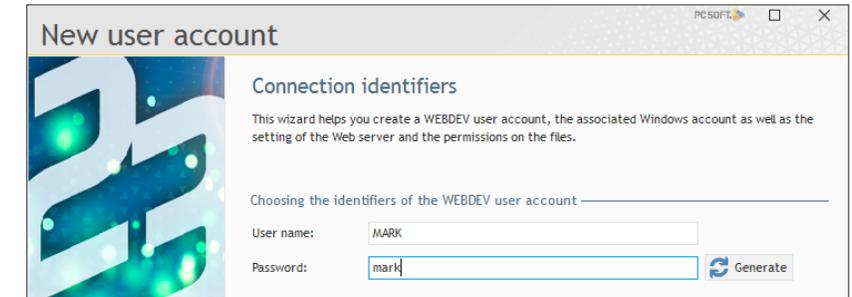
7. Once the hosting is configured, click "Apply" then the "Refresh" button (if this button is proposed).

Creating a deployment account

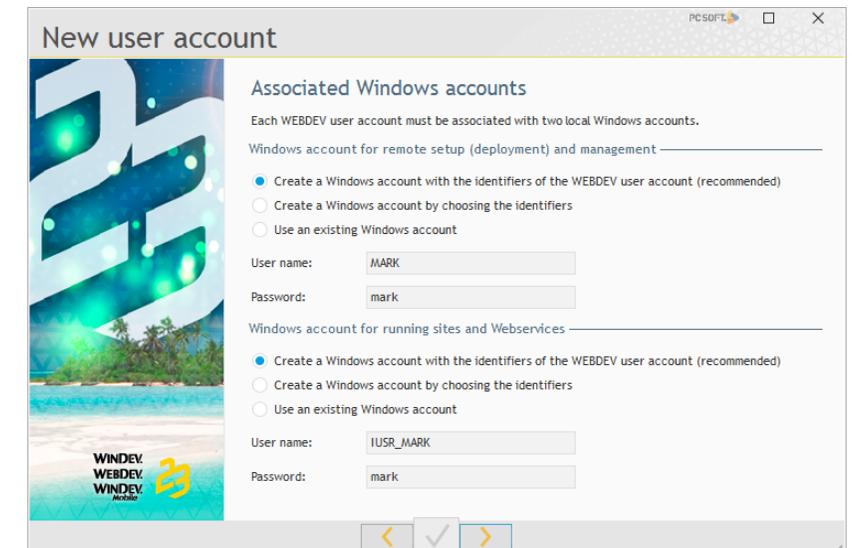
The Hosting Control Center also allows you to create a deployment account.

To create a deployment account:

1. Click "User accounts".
2. Click the "New user" button. The wizard for creating a new user starts. All you have to do is follow the different steps.
3. Type the user name and password (you also have the ability to generate the password. In this case, don't forget to write it down !). Go to the next wizard step.

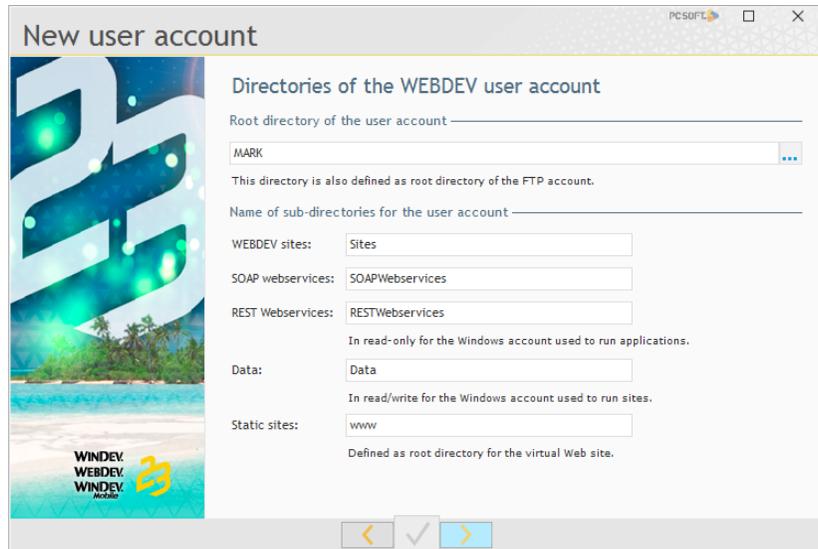


4. The wizard proposes to create the necessary Windows accounts. Keep the default choices and go to the next step.



5. Enter the information about the user. Go to the next step.

6. The directories of the user account are automatically filled according to the specified data.



7. Continue with the wizard until you reach the "Virtual Web server" step.
 - If you choose to create a new virtual site, all you have to do is specify the DNS name that will lead to this site (the DNS must be configured accordingly).
 - If you choose to use an existing virtual site, its configuration will be replaced.
8. Go to the next step.
9. Select an FTP site. Go to the next step.
10. The wizard is ended. Check all the choices. You have the ability to uncheck some operations if you don't want the wizard to perform them on your behalf.
11. Validate the wizard. Your server is now ready to receive WEBDEV sites.

The different deployment modes

Now that our Web server was configured, we are ready to deploy our site. Several methods are available. The choice of a method mainly depends on the technical constraints (ability to use an FTP server or not, access to the server, ...).

A dynamic WEBDEV site can be deployed according to one of these methods:

- **Deployment by physical media** (CD, ...). The deployment by physical media creates a setup support (like an application) that will have to be run on the server directly. This deployment mode avoids using an FTP server. It will operate in Windows only.
- **Remote deployment from the development computer** (by FTP). We are going to deploy our site according to this method.
- **Remote deployment from a management computer** (by FTP) via a "deployment package". This deployment mode operates like the deployment by FTP. The only difference: the deployment is not performed from the development environment of WEBDEV. In this mode:
 - the developer generate a deployment package.
 - The site manager (who can be a person other than the developer) deploys the package on the remote server (by using the same mechanism as the deployment by FTP) via WDDeploy.

WDDeploy is a freely distributable tool that is supplied with WEBDEV.

- Deploying the site via the test hosting service of PC SOFT.
- Deployment in PC SOFT Cloud.

Let's take a closer look at the deployment by FTP.

A detailed example: deployment by FTP

We are going to deploy the "Full_WEBDEV_Site" project that was used in part 3 of this tutorial. A corrected version is available if you did not use this project yet.

- ▶ To open this project in WEBDEV:
 1. Close (if necessary) the current project to display the home window.
 2. In the home window, click "Tutorial" and select "Full WEBDEV Site (Exercise)".



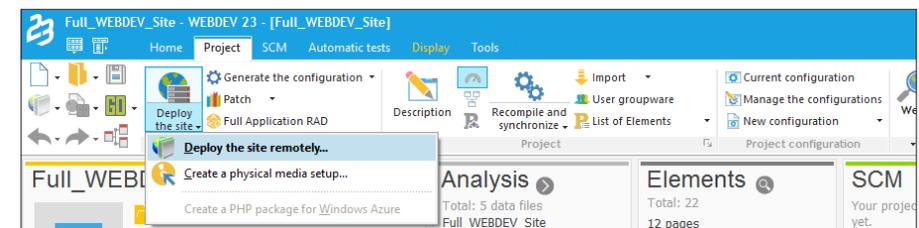
Answer

If you did not perform the operations in the previous parts, open the corrected project. This project contains the different pages created in the previous parts. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (Answer)".

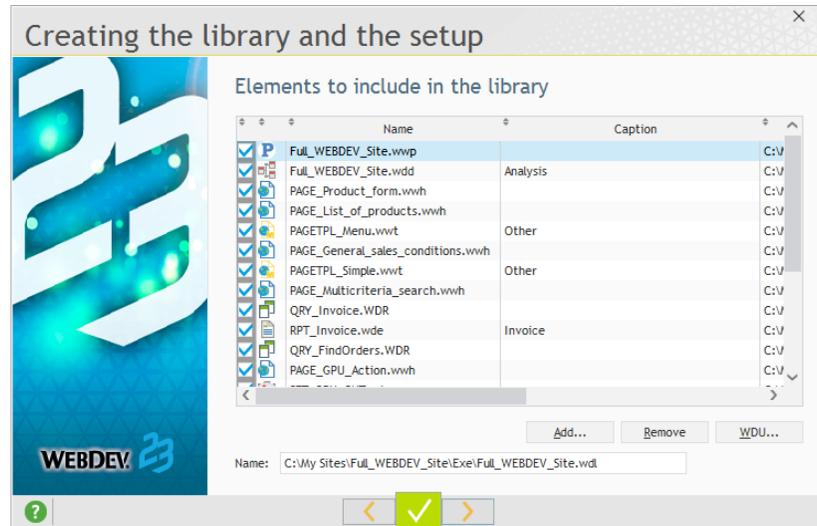
Preparing the setup

A setup wizard is supplied with WEBDEV ; this wizard allows you to easily install your site at the hosting company (Internet or Extranet site for example) or on one of your servers dedicated to WEBDEV hosting (Intranet site for example).

- ▶ We are going to use this wizard:
 1. In the ribbon, on the "Project" pane, in the "Generation" group, expand "Deploy the site" and select "Deploy the site remotely".



2. Display the next step.



3. Before performing the setup, all the elements found in your site must be included in a library. A library is a file that groups all the elements created during the development steps (description of the database, pages, reports, queries, ...). The HTML pages and the images are not included in the library.

Go to the next step.

4. Several languages can be included in the library. In our example, we will keep the default options. Go to the next step.

5. The information about the library version is used to enter the elements that will be displayed in the file properties in the Windows explorer. Go to the next step.

6. Don't save the project and validate the library creation.

Setup

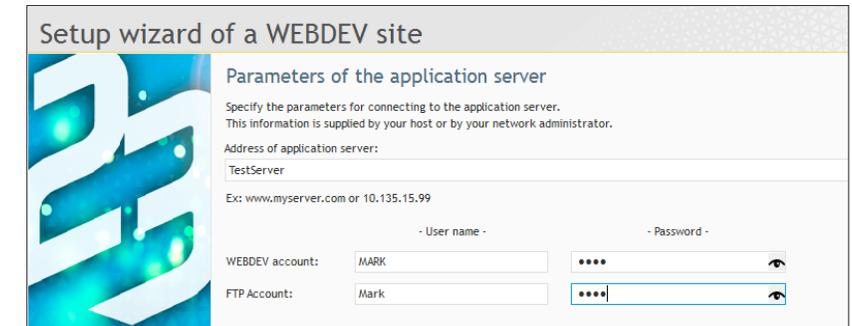
► The setup wizard will now ask you some questions to define how your site will be deployed. In our case, we are going to perform a remote setup by FTP:

1. Select the first option "Deploy the WEBDEV site on a remote WEBDEV Application Server". Go to the next step.



2. To define the server parameters, the following information must be supplied by your hosting company. We are going to enter the information corresponding to the setup that was performed beforehand:

- Server address (in our example, name of computer where WEBDEV Application Server 10 connections was installed). The name can be:
 - a computer name accessible by network (for example: "TestServer"),
 - an IP address (for example: 192.168.15.99),
 - an Internet address (for example: www.myserver.eu).



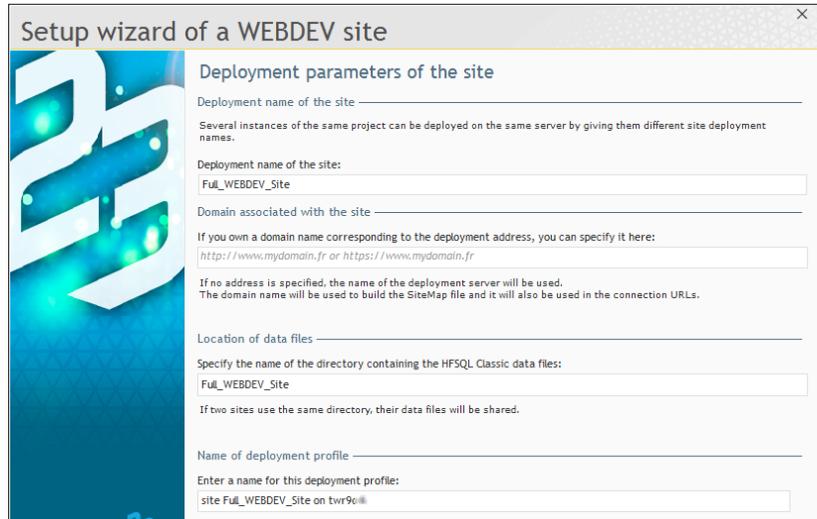
- Characteristics of user account typed by the hosting company in the WEBDEV account manager.
- Characteristics of FTP account typed by the hosting company when creating an FTP account.



Note

Caution: the user name can be preceded by the domain name to avoid confusions.
For example: "mycomputer\test" or "mydomain\test"

- When the information regarding your WEBDEV account and your FTP account was entered, go to the next step.
- Type the parameters for site deployment. We will keep the default options. Go to the next step.



Setup wizard of a WEBDEV site

Deployment parameters of the site

Deployment name of the site _____

Several instances of the same project can be deployed on the same server by giving them different site deployment names.

Deployment name of the site:
Full_WEBDEV_Site

Domain associated with the site _____

If you own a domain name corresponding to the deployment address, you can specify it here:
<http://www.mydomain.fr> or <https://www.mydomain.fr>

If no address is specified, the name of the deployment server will be used.
The domain name will be used to build the SiteMap file and it will also be used in the connection URLs.

Location of data files _____

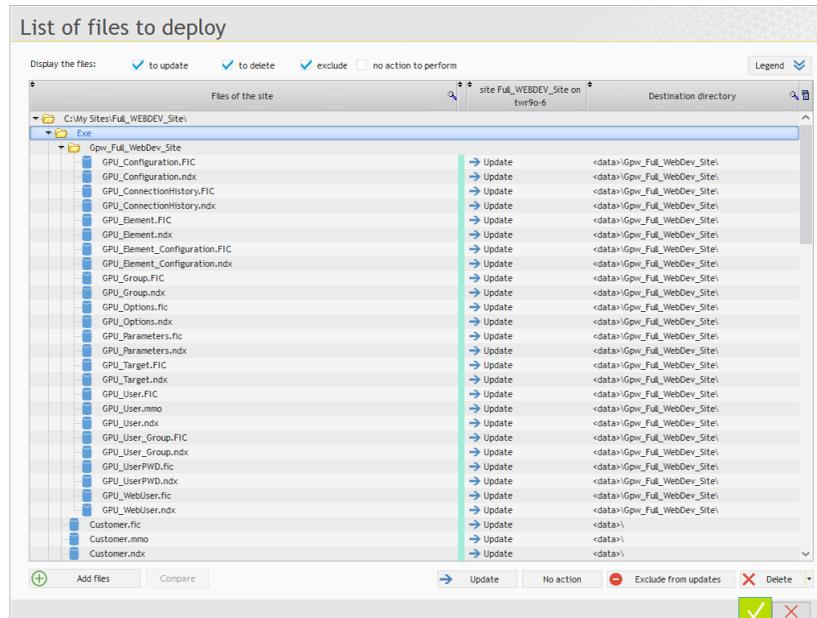
Specify the name of the directory containing the HFSQL Classic data files:
Full_WEBDEV_Site

If two sites use the same directory, their data files will be shared.

Name of deployment profile _____

Enter a name for this deployment profile:
site Full_WEBDEV_Site on twr90

- WEBDEV establishes the connection and it summarizes the operations to perform (number of files to update, number of files to delete, ...). To get the details of operations and to modify them if necessary, click the "Edit the list of files" button.



List of files to deploy

Display the files: to update to delete exclude no action to perform

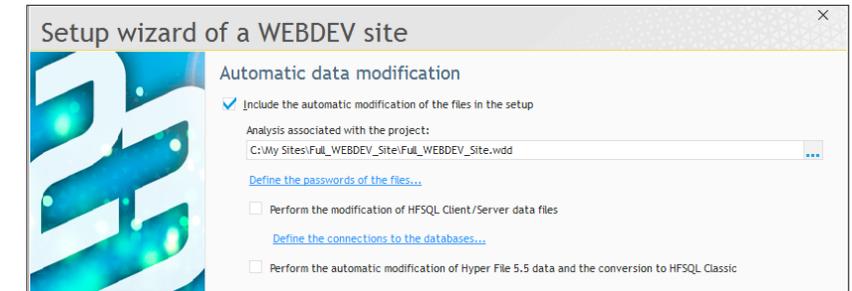
Files of the site: site Full_WEBDEV_Site on twr90-6

Files of the site	Operation	Destination directory
GPU_Configuration.FIC	Update	<data>Gpwr_Full_WebDev_Site
GPU_Configuration.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_ConnectionHistory.FIC	Update	<data>Gpwr_Full_WebDev_Site
GPU_ConnectionHistory.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_Element.FIC	Update	<data>Gpwr_Full_WebDev_Site
GPU_Element.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_Element_Configuration.FIC	Update	<data>Gpwr_Full_WebDev_Site
GPU_Element_Configuration.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_Group.FIC	Update	<data>Gpwr_Full_WebDev_Site
GPU_Group.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_Options.fic	Update	<data>Gpwr_Full_WebDev_Site
GPU_Options.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_Parameters.fic	Update	<data>Gpwr_Full_WebDev_Site
GPU_Parameters.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_Target.FIC	Update	<data>Gpwr_Full_WebDev_Site
GPU_Target.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_User.FIC	Update	<data>Gpwr_Full_WebDev_Site
GPU_User.mmo	Update	<data>Gpwr_Full_WebDev_Site
GPU_User.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_User_Group.FIC	Update	<data>Gpwr_Full_WebDev_Site
GPU_User_Group.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_UserPWD.fic	Update	<data>Gpwr_Full_WebDev_Site
GPU_UserPWD.ndx	Update	<data>Gpwr_Full_WebDev_Site
GPU_WebUser.fic	Update	<data>Gpwr_Full_WebDev_Site
GPU_WebUser.ndx	Update	<data>Gpwr_Full_WebDev_Site
Customer.fic	Update	<data>
Customer.mmo	Update	<data>
Customer.ndx	Update	<data>

Buttons: Add files, Compare, Update, No action, Exclude from updates, Delete

- Go to the next step.

- The wizard proposes to include the automatic modification of data files in the setup. You also have the ability to configure the elements required to use a HFSQL Client/Server database. Keep the default options and go to the next step.



Setup wizard of a WEBDEV site

Automatic data modification

Include the automatic modification of the files in the setup

Analysis associated with the project:
C:\My Sites\Full_WEBDEV_Site\Full_WEBDEV_Site.wdd

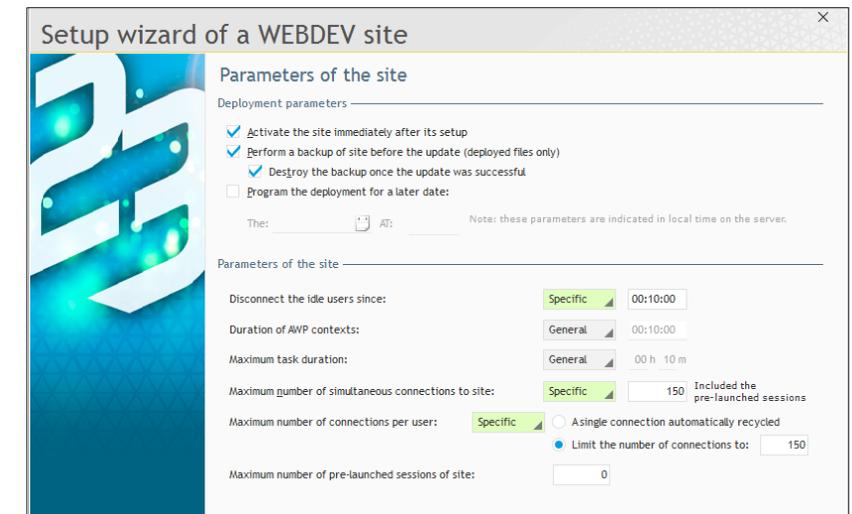
[Define the passwords of the files...](#)

Perform the modification of HFSQL Client/Server data files

[Define the connections to the databases...](#)

Perform the automatic modification of Hyper File 5.5 data and the conversion to HFSQL Classic

- Specify the site parameter:



Setup wizard of a WEBDEV site

Parameters of the site

Deployment parameters

Activate the site immediately after its setup

Perform a backup of site before the update (deployed files only)

Destroy the backup once the update was successful

Program the deployment for a later date:

The: _____ AT: _____ Note: these parameters are indicated in local time on the server.

Parameters of the site

Disconnect the idle users since: **Specific** 00:10:00

Duration of AWP contexts: **General** 00:10:00

Maximum task duration: **General** 00 h 10 m

Maximum number of simultaneous connections to site: **Specific** 150 Included the pre-launched sessions

Maximum number of connections per user: **Specific** A single connection automatically recycled Limit the number of connections to: 150

Maximum number of pre-launched sessions of site: 0

You can modify:

- The maximum number of connections to the site: if this value is set to "5" for example, only 5 Web users will be able to connect to your site at the same time.
- The maximum number of connections per Web user: if this value is set to "5" for example, a Web user will be able to start your site up to 5 times.
- The amount of idle time before user disconnection: this option is used to free all the resources occupied by the session of Web user if this one has performed no action since the specified duration.

- By default, your site is immediately enabled after setup. The users will have no access to your site if "Activate the site immediately after its setup" is unchecked.

10. The wizard proposes to perform:

- an immediate setup: the files will be immediately transferred to the server and your site will be immediately installed.
- a delayed setup: the files will be immediately transferred to the server but your site will be installed at the specified date and time ("Program the deployment for a later date" option).

11. Go to the next step.

12. The wizard allows you to define the parameters for site security: Change of IP, ... Go to the next step.

13. The wizard proposes to automatically generate the statistical files for the site. These statistics affect for example the actions performed on the site, the origin of Web users, ...

Keep the options proposed by default and go to the next step.

14. Display the next screen and validate the setup. The setup wizard transfers the files.

During the file transfer, the wizard compresses and encrypts the transferred data. Your data is transferred with a high-security level.

At the end of setup, a link allows you to immediately start the site.

Correspondence between the directories of the development computer and the deployment

The distribution of site files is slightly different between the development and the deployment server. Let's see a summary of the distribution automatically proposed by WEBDEV.

Directory on the development computer	Directory on the deployment server
<Project name>\<Project name>_WEB	<site>\<site name>\<site name in uppercase>_WEB
<Project name>\Exe	For a site: <ul style="list-style-type: none"> • data files (.fic, .ndx, .mmo, .ftx): <data>\site name> • other files: <site>\<site name> For a Webservice: <ul style="list-style-type: none"> • data files (.fic, .ndx, .mmo, .ftx): <data>\site name> • other files: <webservice>\<site name>

where

- <site>, <data> and <webservice> are the directories defined when creating the hosting account (see above).
- <site name> is the name of site.

The distribution of files can be configured in the deployment wizard.

LESSON 6.2. MANAGING A SITE

This lesson will teach you the following concepts...

- Local WEBDEV administrator
- Remote WEBDEV administrator



Estimated time : 10 mn

Overview

Two tools can be used to manage the WEBDEV sites (and the Webservices):

- the WEBDEV administrator.
- the remote WEBDEV administrator.

Local WEBDEV administrator

The WEBDEV administrator is an application, installed with the Application Server.

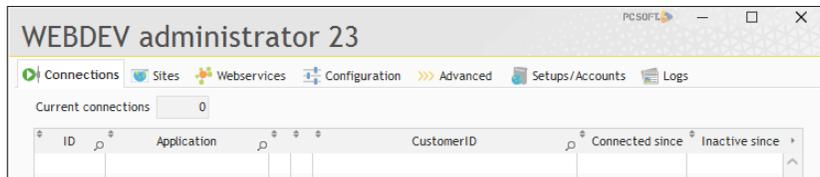


Note

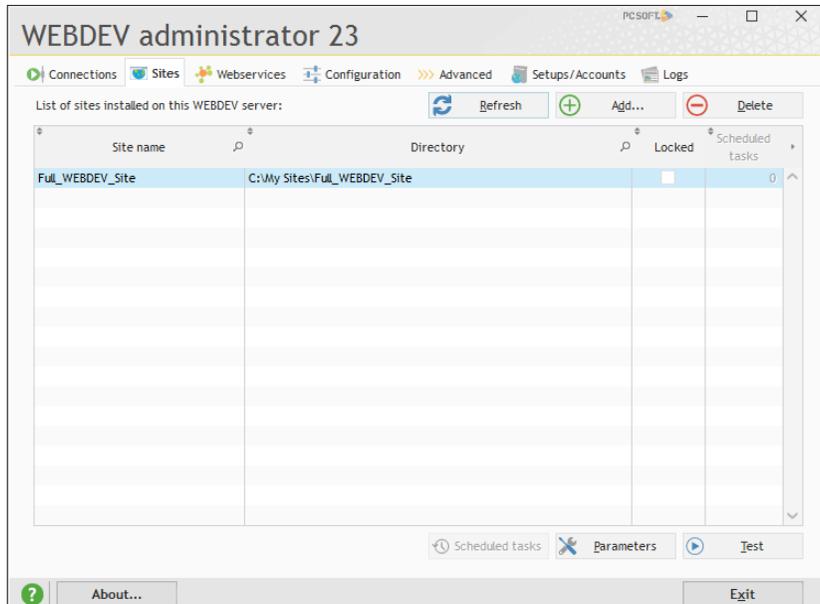
The WEBDEV administrator can be used only if you can directly access the Web server on which your site is deployed.

This application includes seven parts, represented by seven tabs:

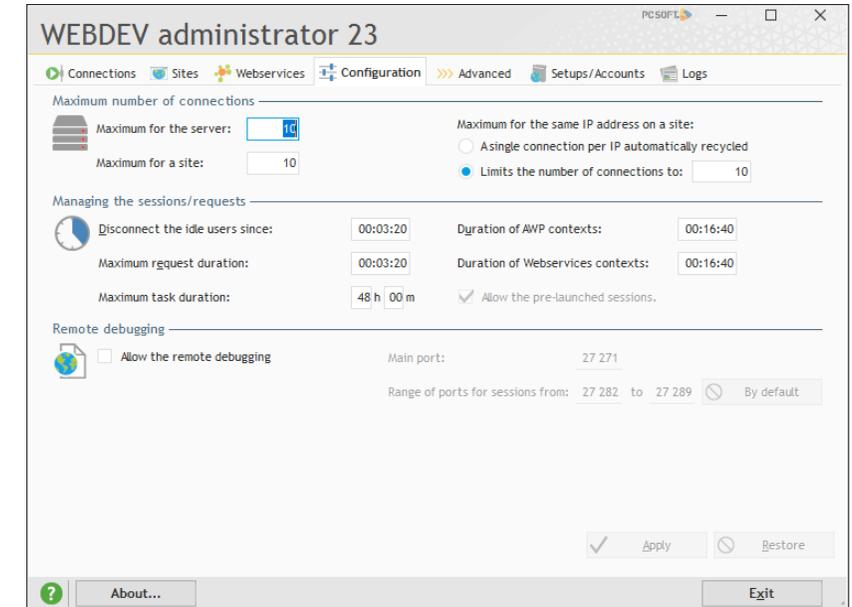
- Connections: Displays the list of users connected to the sites and allows you to disconnect a user.



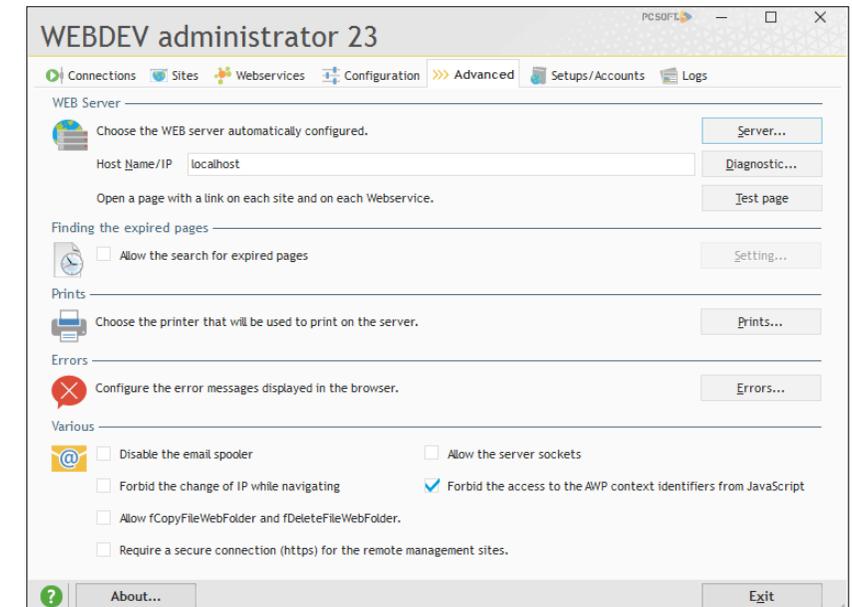
- Sites : Lists the sites deployed on the server as well as their parameters:



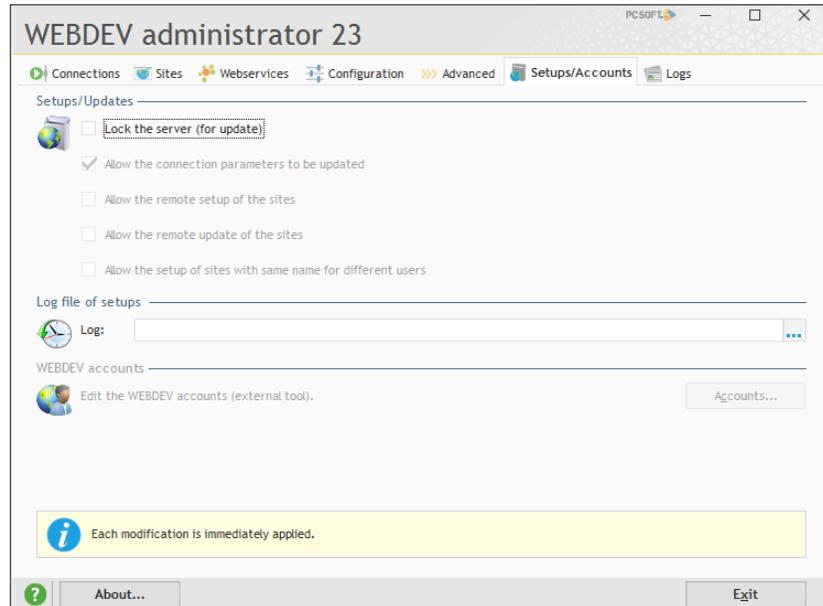
- Webservices : Lists the webservices deployed on the server as well as their parameters.
- Configuration: Used to manage the server configuration.



- Advanced : Configures the additional server parameters such as print management, email spooler ...



- **Setups/Accounts:** Used to lock the server for maintenance, to allow or forbid the remote setups and to configure the log of setups. This log is used to trace the elements installed in a site. It is especially useful before contacting our Technical Support if you encounter errors during the deployment.



- **Logs:** Used to see the logs for the sites or webservices for a specific period. The elements displayed are the elements for which errors occurred during the specified period and for which the logs are enabled.

Remote WEBDEV administrator

The remote administrator is a WEBDEV site that provides features similar to the WEBDEV administrator but that can be used to manage the WEBDEV sites remotely.



Caution! If you are using the remote administrator, the security configuration is important, we advise you to access it via an HTTPS connection and to choose strong passwords.

Note: This tool is not available with the WEBDEV application server 10 connections.

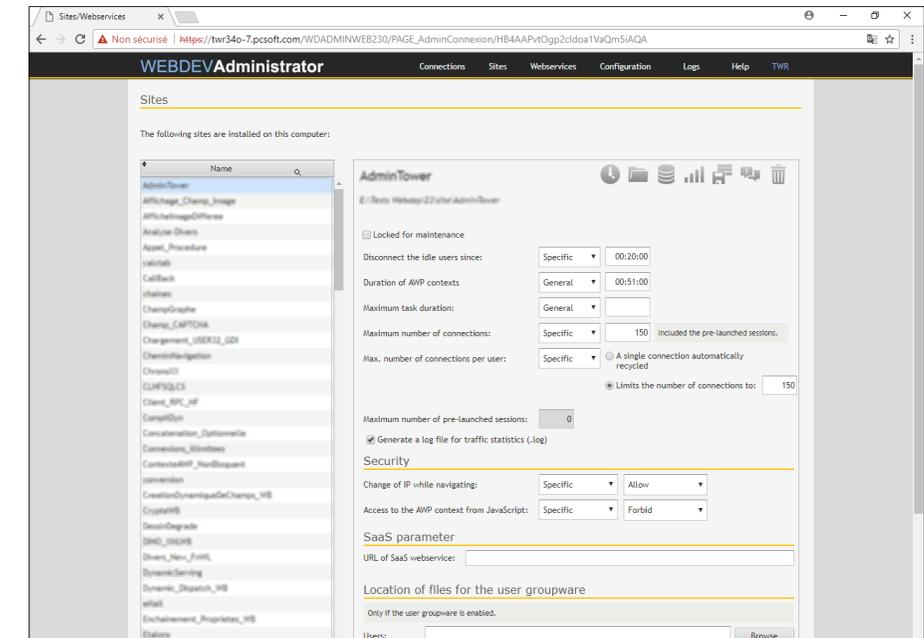
When deploying your dynamic site at a hosting company (and if this hosting company allows the remote management of your site), this tool can be used to update the parameters of your site.

- To start WAdminWeb230:
 1. Open your favorite browser (Internet Explorer for example) on your computer.
 2. Type the following URL in the address bar of browser (while respecting the case):

```
http://computer/WAdminWeb230
```

where "computer" is the name of deployment server.

3. Type the login and password used to connect to the remote WEBDEV administrator.
4. The administrator starts.



PART 7

Specific Web features



LESSON 7.1. THE STYLES

This lesson will teach you the following concepts...

- The CSS styles
- The WEBDEV styles



Estimated time : 1 h

Overview

WEBDEV allows you to "design" your sites. To help you with this task, WEBDEV proposes to use two types of styles:

- The CSS styles,
- The WEBDEV styles.

Using styles for your controls presents a huge benefit: a modification performed in a WEBDEV or CSS style is automatically applied to all the controls that use this style.

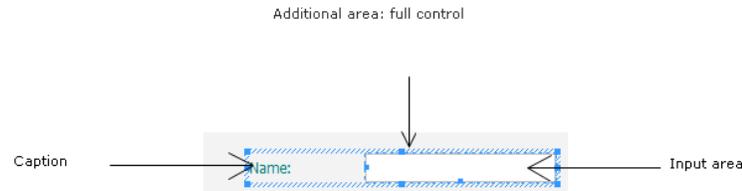
Discovering styles

Before using the styles, let's see their application support: the controls.

A control: several elements

Each WEBDEV control includes several elements,

- ▶ Let's consider the edit control for example. This control includes three elements:
 - the caption.
 - the input area.
 - an additional area containing the caption and the input area.



- ▶ How are the styles applied to the edit control ? The mode for applying the styles depends on the type of style used (WEBDEV or CSS).

- **The CSS styles in WEBDEV will be applied to the control elements.**
For the edit control, a specific CSS style can be applied to the caption of edit control or to the input area.
- **The WEBDEV styles will be applied to the entire control.**
A WEBDEV style includes several CSS styles. For example, on the edit control, the WEBDEV style contains:
 - The CSS style for the caption,
 - The CSS style for the input area,
 - The different style options for the global control area.

Discovering the styles via an example

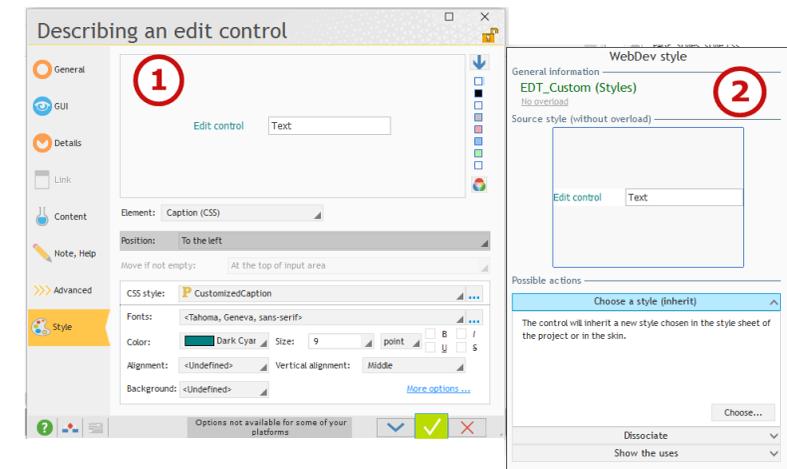
- ▶ To discover the styles in WEBDEV, a simple example was prepared for you:
 1. Start WEBDEV 23 (if not already done). Close (if necessary) the current project to display the home window.
 2. Open the "Styles" project. To do so, in the home window, click "Tutorial" and select the first "Styles (Answer)" project.



Tip

If the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Styles (Answer)".

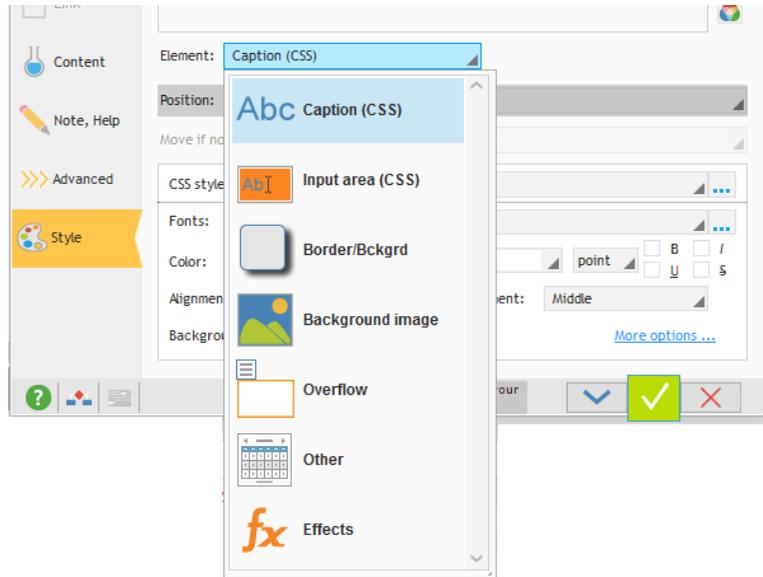
3. Open the "PAGE_Styles" page in the editor.
4. Display the description window of "Name" edit control (display the popup menu of control and select "Description").
5. Select the "Style" tab. This tab is used to define the style characteristics of control.
6. The following window is displayed:



This window allows you to manage the CSS styles (section 1 of window) and the WEBDEV styles (section 2 of window) for the control. Let's take a look at the features of this window.

Selecting the CSS style of a control

- To select the CSS styles of a control:
 - In the "Style" tab of the description window of control, in the "Element" combo box, choose the element with the requested style.

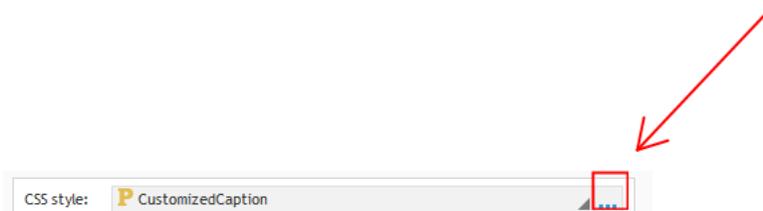


Note: All the elements flagged "(CSS)" can use a CSS style directly: all you have to do is select its name.

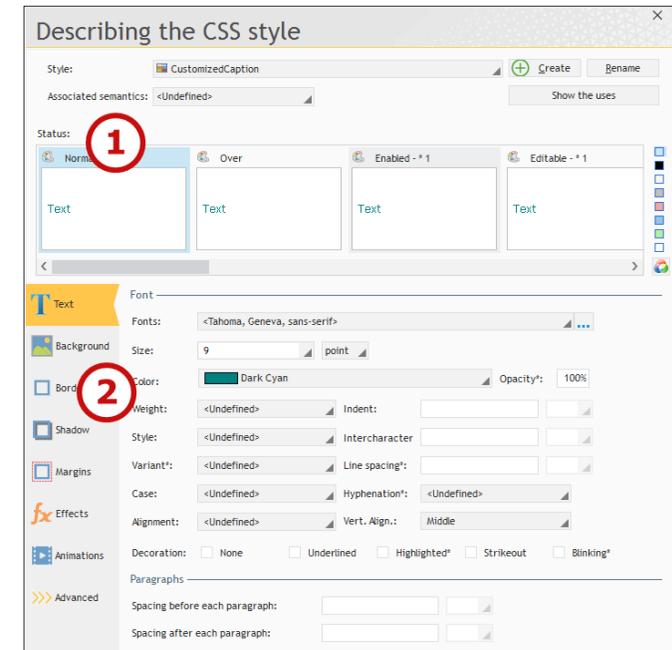
- The "CSS style" combo box is used to choose the CSS style of the element: this style will be applied to the element.

Editing the CSS style of a control

- To create or edit a CSS style:
 - In the "Style" tab of the description window of control, click the "..." found beside the "CSS style" combo box.



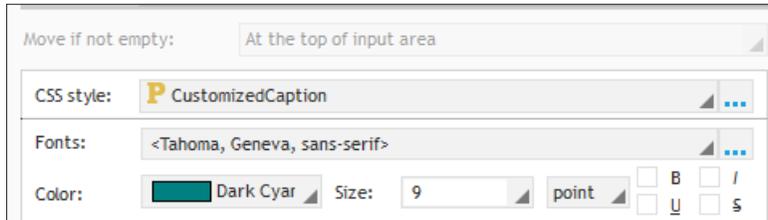
- The window for editing the CSS styles is opened.



- This window allows you to create or modify all the CSS styles of the project.

- Let's study this window. This window includes 2 sections:
 - Section 1. The status of control onto which the style will be applied: normal, rollover, active, ...
 - Section 2. The CSS properties associated with the selected status. Each CSS property can be modified for each status.
- Let's check the characteristics of the CSS style named "MyCustomStyle_InputArea":
 - Select the "MyCustomStyle_InputArea" style in the combo box of section 1 if necessary.
 - Select the "Normal" status (the first status of section 1).
 - Click the "Background" tab: the background color is white.
 - Select the "Editable" status (the 4th status of section 1).
 - In the "Background" tab, the background color of input area is pastel yellow. **The CSS style contains the style for each one of the states.**
 - Close the description window of CSS style.

► The "Style" tab found in the description window of control is redisplayed. In the bottom section of the window, you have the ability to quickly access the style elements that are frequently modified:



- Font,
- Size,
- Bold, Italic, Underlined, Strikethrough,
- Vertical and horizontal alignment,
- Background color.



Caution

These options are used to quickly overload the CSS style for the current control. In this case, the modifications performed in these options will not be applied to the other controls that use this CSS style. The "More options" link is used to overload all the other properties of the CSS style.

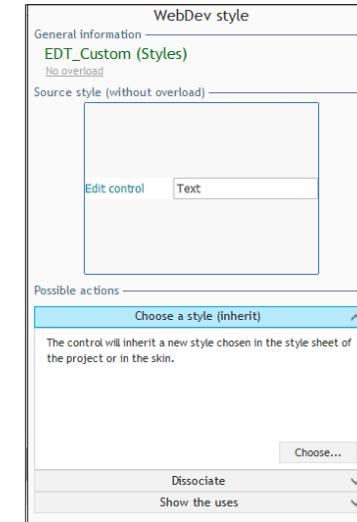
Let's now discover the WEBDEV styles.

The WEBDEV styles

A WEBDEV style includes several CSS styles. For example, on the edit control, the WEBDEV style contains:

- The CSS style for the caption,
- The CSS style for the input area,
- The different style options for the global control area.

► To manage the WEBDEV styles, use the panel displayed on the right of "Style" tab of control:



The name of the WEBDEV style appears at the top of this window. You can:

- Choose an existing style,
- Add the style of current control to the style sheet of project. This option allows you to re-use this style in other project controls.
- Dissociate this control. This option is not recommended: we advise you to use style overloads.

The project skin defines a WEBDEV style for each type of control. This WEBDEV style can be used "as it is" or it can be modified on some control elements.

For example, to specify that the caption must be bold on an edit control (and on this one only), you must:

- Select the "Caption (CSS)" element in the bottom area of "Style" tab.
- Select "Bold" in the bottom area.

This modification will not dissociate the WEBDEV style from the control: the Bold property will be overloaded in relation to the initial style.

Therefore, if the initial style is modified (to change the background color for example), this modification will be applied to this control while keeping the overload.



Note

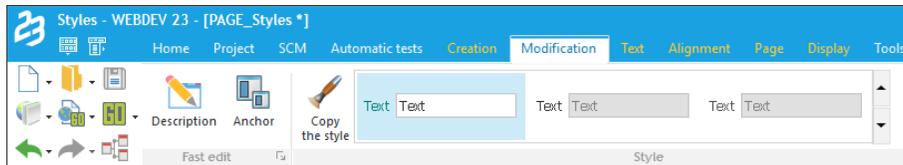
To see the overloaded elements, all you have to do is click at the top of the panel for managing the WEBDEV styles on the link that indicates the number of overloads performed for the control.

Implementing styles: a practical example

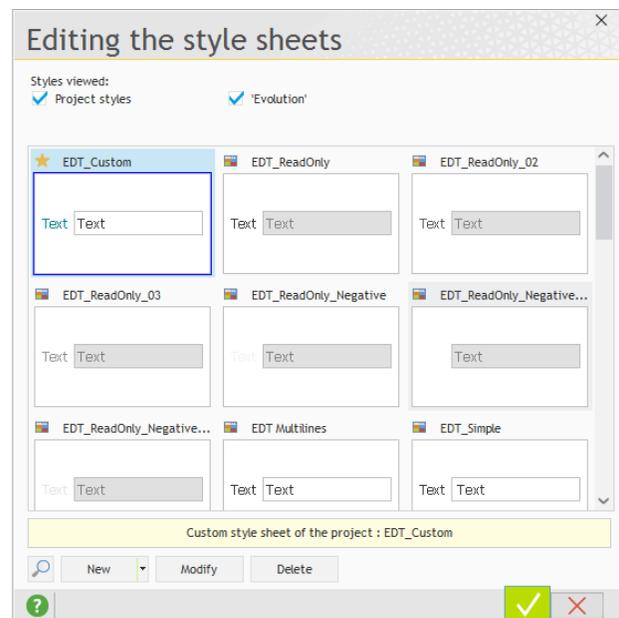
Enough theory, let's get down to work. We are going to create an edit control and to modify its styles in the "PAGE_Styles" page of "Styles" project.

Note: We will be using an edit control but the principle for handling styles can be applied to all the controls available in WEBDEV.

- ▶ To create the edit control:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. This control is associated with a WEBDEV style (it is the default WEBDEV style defined in the skin chosen when creating the project).
- ▶ Three methods can be used to select the WEBDEV style of a control:
 - Via the "Style" tab found in the control description of control (as already seen).
 - Via the "Modification" pane of ribbon: a preview of different WEBDEV styles available for the control is displayed.



- By displaying the popup menu of control and by selecting "Choose a WEBDEV style".



- ▶ This last method will be used to associate the control that was just created with the "EDT_Custom" style. This style is used by all the other page controls.
- ▶ Validate the window for style selection. The color of control caption changes: the style is not applied.
- ▶ Save the page (CTRL S) and run the page test ( among the quick access buttons).
 1. The page is displayed in the browser.
 2. Enter in input in the new edit control: the input area is colored in yellow.
 3. Close the browser.

Overloading a CSS style for a control

- ▶ We are now going to modify the style of the input area found in the control:
 1. Display the description window of created control.
 2. In the "Style" tab, choose the "Input area (CSS)" element.
 3. Click the "More options" link to overload the CSS options of this element.
 4. Select the "Editable" status.
 5. In the "Background" tab, change the background color. Choose a mauve color for example.
 6. Validate. In the pane of WEBDEV styles, the link (top right) indicates "1 overload".
 7. By clicking it, you can see that the background color of editable inside area was overloaded.
 8. Validate the description window of control.
- ▶ Save the page (CTRL S) and run the page test ( among the quick access buttons).
 1. The page is displayed in the browser.
 2. Move the cursor from a control to another one: the background color is yellow for all the controls except for the control that was just created.
 3. Close the browser.

Modifying a CSS style for all the controls

- ▶ We are now going to modify the CSS style of input area in order to change the text color. This modification will be performed for all the edit controls of the page.
 1. Display the description window of created control (ALT + Enter).
 2. In the "Style" tab, choose the "Input area (CSS)" element.
 3. Click the [...] button on the right of the name of CSS style. The window for editing the CSS style is displayed.
 4. Select the "Normal" status.
 5. In the "Text" tab, change the color. Choose a green color for example.
 6. Validate then validate the description window of control.

- ▶ Save the page (CTRL S) and run the page test ( among the quick access buttons).
 1. The page is displayed in the browser.
 2. Type text in the different edit controls of the page: the text is displayed with the new selected color. This modification was taken into account even on the control that was created beforehand and whose style was overloaded.
 3. Close the browser.

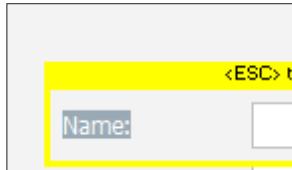
Styles and rich text

You also have the ability to define rich text in the elements!

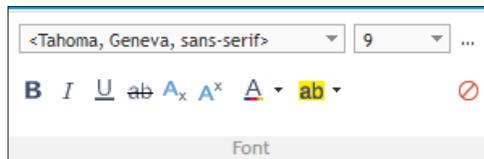
The rich text allows you to use different styles for the different words found in the control caption. Therefore, for the caption of an edit control, you have the ability to define that part of the caption will be using the default color while another part will be using a different color.

Let's take a simple example: adding a red asterisk to specify that a control is a mandatory control.

- ▶ In our example:
 1. Select the "Name" control.
 2. Press the SPACE or ENTER key on the keyboard.
 3. The caption text becomes editable.



4. Add a star at the end of caption.
5. Select this star.
6. On the "Text" pane, in the "Font" group:



- Click the button **A^x** move the star in superscript.
 - Change the text color to red with the button **A ▾**.
7. Exit from the edit mode (ESC key).
 8. The red asterisk appears in the control caption.

LESSON 7.2. THE COOKIES

This lesson will teach you the following concepts...

- What is a cookie?
- How to manage the cookies?



Estimated time : 10 mn

What is a cookie?

A cookie is an easy way to temporarily store an information on the computer of Web user. This information can be read again later by the site that created it.

This allows you to avoid asking for details already supplied during a previous visit and to propose custom pages.



Internet

A cookie has an expiration date (30 days after its creation by default). It is automatically destroyed by the browser of Web user when its lifetime is exceeded. Caution: using cookies is possible only if the browser of Web user is configured to accept cookies.

A cookie is used to store on the computer of Web user various information such as the user name, the pages displayed by the user, the date of his last connection, the backup of his options, ...

This information, saved in the format of cookies, will be read by the site during the next connection of Web user. Therefore, the site will be able to propose custom information to the user:

- advertising banner related to subjects looked up during the last connection,
- custom home page with the user name and the date of last connection,
- special offers corresponding to the searches performed during the last visit, ...



Note

The cookies are not encrypted when they are stored: we advise you not to use them to store sensitive information.

What is a cookie made of?

A cookie is a text file stored on the computer of Web user (in most cases, in the Internet "cache" of browser) during a specific duration. The cookie is created by the browser or by the server.

The following elements are required to store information in a cookie:

- Name of cookie, used by the site to identify the cookie.
- Text of cookie, corresponding to the information written by the site: pages displayed, details supplied by the Web user, ...
- Expiration date after which the cookie is not valid anymore (it will be automatically deleted).
- Name of Internet domain that created the cookie.

Practical example

- ▶ To check the management of cookies, we are going to import the unit example named "The cookies" into the "Full_WEBDEV_Site" project.
- ▶ To open a unit example:
 1. Display the home window of WEBDEV (CTRL <).
 2. Click "Open an example".
 3. In the search control, type "Cookies". The unit example named "The Cookies" appears in the window.
 4. Click the name of the unit example.
 5. The page corresponding to the unit example is displayed in the editor.
 6. Save the page.

How to use the cookies?

The cookies can be used according to two different modes:

- Cookies used in Server code.
- Cookies used in Browser code.

To use the cookies, WEBDEV includes 2 functions that can be used both in server code and in browser code:

- **CookieWrite**: is used to send a cookie when displaying the HTML page in the browser of Web user.
- **CookieRead**: is used to retrieve the value of a cookie saved on the computer of Web user.

The "PAGE_Cookies" page presents an example of read and write operations on a cookie in server code and in browser code.

See the online help for more details.

LESSON 7.3. SECURE TRANSACTIONS AND PAYMENT

This lesson will teach you the following concepts...

- Securing the information and the pages via TSL/SSL
- Secure payment



Estimated time : 15 mn

Securing the information and the pages via TSL/SSL

Overview

By default, the data exchanged between the computer of the Web user and the Web server is not protected. This data flows in clear on the network.

Several systems can be used to secure the data. A common system consists in using the TLS (Transport Layer Security) / SSL (Secure Socket Layer) protocol. <https://customers.mywebdevsite.com/customers>).

Implementing secure transactions via the TLS/SSL protocol

To implement secure transactions via TLS/SSL, you must install a certificate on the Web server and configure the Web server.

Two different methods can be used to get a certificate:

1. A purchase beside a certified organism.
2. The generation of a self-signed certificate.

See the online help for more details, keyword: "SSL".

Transactions secured by TLS\SSL in a WEBDEV site

The secure mode is implemented when displaying the page that requires to be secured (page for typing a credit card number for example).

All you have to do is call **SSLActive** in the browser code of the button that opens this page.

As soon as the secure page is opened, all the actions will be performed in secure mode (which means encrypted), regardless of the objects used (link, table, loop, clickable image, ...).

See the online help for more details, keyword: "SSLActive".

Secure payment

Overview

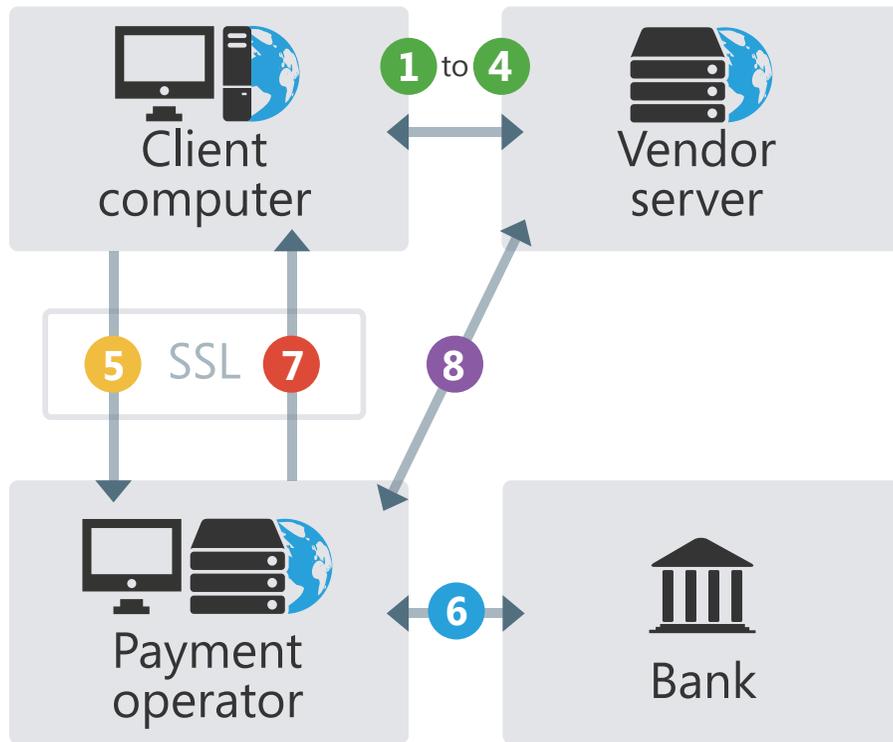
Most of the business sites allowing the Web users to perform an online purchase are using a system for online payment by credit card.

The secure payment is an essential feature for an e-commerce site. The payment solution must reassure the Web user (the "customer") and must guarantee the payment to the e-commerce site.

The data exchanged during this transaction must be secured (via the SSL protocol that was described in the previous paragraph for example).

A wide range of payment solutions is available (PayPal Paybox, ...).

The principle for secure payment may slightly change from a provider to another one but it is overall the same:



- | | |
|--|--|
| 1 View the site
Fill the basket | 5 Enter the credit
card number |
| 2 Identification
(input of personal details) | 6 Bank authorization |
| 3 Check the
order | 7 Response of the bank |
| 4 Redirection to the
secure payment | 8 Result of the transaction |

• 1 to 4. Preparing the order on the e-commerce site: the Web user places his order on the site. During the payment operation, the e-commerce site transfers the information used to identify the order (vendor number, invoice amount, ...) to the payment operator

- 5 to 7. Typing and checking the credit card number: the user types his credit card number in a page for secure payment. The data transmission is protected via SSL to ensure the confidentiality of data. The business site does not know the credit card number typed by the Web user.
- 8. Back to the e-commerce site: the payment operator indicates to the e-commerce site whether the payment was validated, canceled or refused.



Caution!

Important : this domain evolves very quickly. Before implementing a payment solution, always check the latest solutions proposed by the different providers!

System for secure payment in a WEBDEV site

General procedure

In most cases, the following operations must be performed when implementing a solution for secure payment:

1. Request a development kit from the payment operator (PayBox, ATOS, SIPS, CyberMut, ...).
2. Contact the payment operator to establish a contract for remote sales.. At this step, the bank provides a vendor number.
3. Contact the payment operator to establish a contract specifying the parameters for the remote sales contract.
4. Implementing the e-commerce site.

Using the component supplied with WEBDEV: "Secured Payment component"

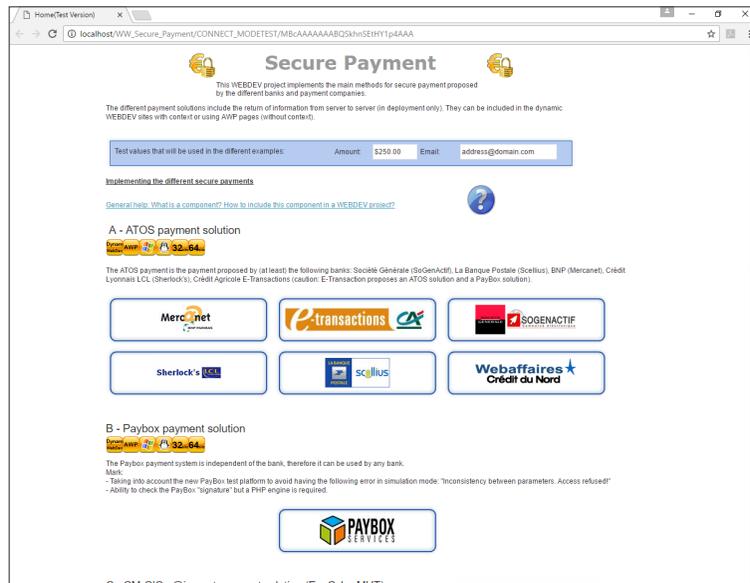
Several components used to perform online payment are supplied with WEBDEV.

The "Secure payment" component includes several modes for secure payment. The "Secure payment" component is supplied with its source code and with a use example.

► To open the example for using the "Secure payment" component:

1. Display the home window of WEBDEV (CTRL <).
2. Click "Open an example".
3. In the search area, type "Payment".
4. Click the link corresponding to the "WW_SecurePayment" example: the corresponding project is automatically opened in the editor.
5. In the project explorer, select the "Example" configuration.

6. You can test this example.



LESSON 7.4. ANCHORING AND ZONING

This lesson will teach you the following concepts...

- What is an anchor?
- Creating a page in Zoning mode.
- Implementing the anchors.



Estimated time : 30 mn

Overview

WEBDEV allows you to manage two types of anchors:

- The anchor to content: An anchor to content is used to define the behavior of a control or area when modifying its content. Will the control keep the same size? Will it adapt to its content?
- The browser anchor: A browser anchor is used to define the behavior of a control or area when resizing the browser. Will the control be enlarged? Will it move?

The browser anchor is used to define the behavior of pages during the display on specific resolutions (tablets, phones, ...).

Anchors are easily set up, thanks to an option in the control popup menu.

We are now going to use the anchors via an example. The browser anchors will be used in this lesson.

We are going to create a project containing a page used to implement the browser anchors and to present their benefits.

► To create a project in WEBDEV:

1. Start WEBDEV (if not already done). Close the current project if necessary.
2. In the home window, click the "Create a project" button and select "Site". The wizard for project creation starts.
3. Enter the name of the project ("Anchors") and validate the different wizard screens with the default options.



Notes

A corrected example can be access at any time to check the validity of operations performed.

To open this corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Anchors (Answer)".

Creating a page in zoning mode

► To create a page used to implement the management of anchors:

1. Create a new blank page.
 - In the window for creating a new element, click "Page" then "Page".

Note: To display the window for creating a new element, click  among the quick access buttons.
 - The wizard for page creation starts.
 - Click "Blank - Simple layout" and validate the wizard.
2. The backup window of page is displayed.
3. Type the title: "Anchor". The name ("PAGE_Anchor") is automatically proposed.
4. Validate.

We are now going to divide our page into 3 sections:

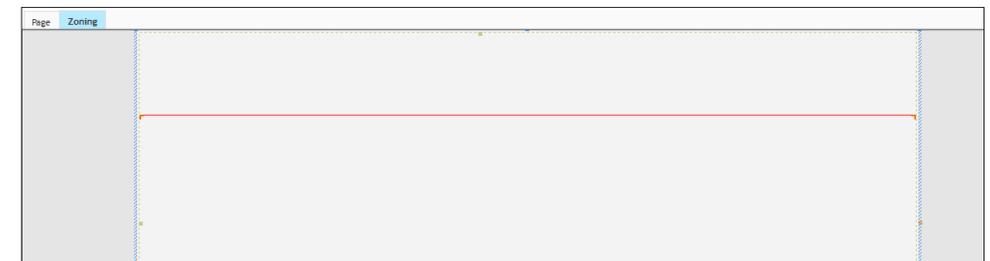
- A header (at the top on the entire width).
- A menu on the left (on the entire height).
- A content area in the middle.

Each area will have a specific behavior when resizing the page.

To create the areas, we will be using the zoning feature.

► To create a first title area at the top of page:

1. On the "Page" pane, in the "Edit" area, expand "Zoning" and select "Split the layout".
2. The mouse cursor turns into a pen.
3. Click at the top of page and draw an horizontal line. The area is created when releasing the mouse button.



► To view this area, we are going to associate it with a background color:

1. Display the description window of area: select "Area description" from the popup menu.
2. In the "Style" tab, select the "Border/Background" element if necessary and select a background color (tooltip yellow for example).
3. Validate the description window of area.



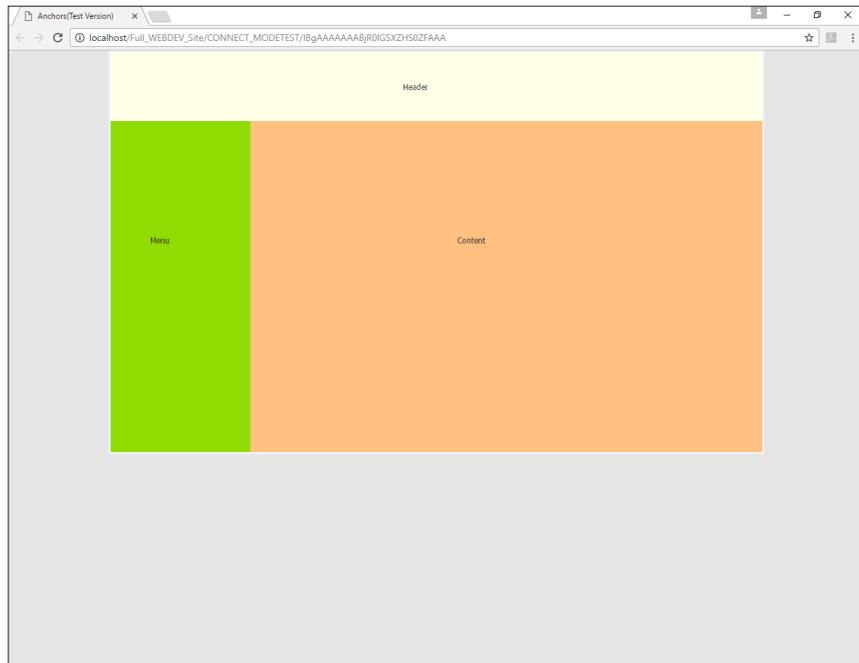
► We are going to perform the same operations to split the bottom area of our page:

1. On the "Page" tab, in the "Edit" area, expand "Zoning" and select "Split the layout".
2. The mouse cursor turns into a pen.
3. Click in the middle of the bottom area and draw a vertical line.

- ▶ To visualize these areas, associate the color green to the bottom area on the left and orange to the bottom area on the right.
- ▶ We are now going to create Static controls in each area:
 1. Exit from the "Zoning" edit mode by clicking the "Page" tab at the top of the page (below the ribbon).



2. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Simple static".
 3. Click inside the yellow area. The Static control is automatically created.
 4. Modify the caption displayed ([SPACE] key on the keyboard): the control caption is "Header".
 5. Repeat the operations 2 to 4 to create:
 - a Static control displaying "Menu" in the green area.
 - a Static control displaying "Content" in the orange area.
- ▶ Save the page (CTRL S) and run the page test (📄) among the quick access buttons.
1. The page is displayed in the browser.
 2. Resize the browser: the page remains centered but the areas are not resized.



We are going to change this operating mode in order for the areas to adapt to the browser size. We will be using the anchors.

- ▶ Close the browser.

Implementing the anchors

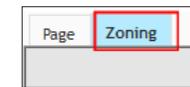
First of all, we are going to implement the anchoring of areas, then the anchoring of controls in the areas.

We want to get the following behavior when enlarging the browser:

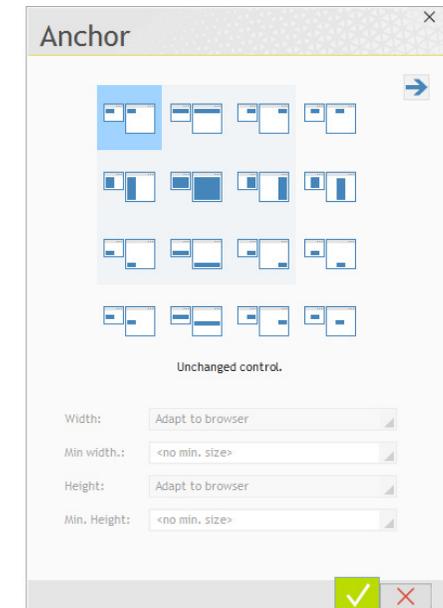
- The header area (yellow) must be enlarged in width.
- The menu area (green) must be enlarged in height.
- The content area (orange) must be enlarged in width and in height.
- For the controls, we would like that they remain centered within their area.

We are now going to perform all the necessary operations.

- ▶ To implement the anchoring of areas:
 1. Display the page in zoning mode: click the "Zoning" tab at the top of the page.



2. Select the yellow area.
3. Display the popup menu and select "Anchor". The window for anchor management is displayed.



4. The area must be enlarged in width: select "Width" (📏) and validate.

5. Select the green area, display the popup menu and select "Anchor".
6. The area must be enlarged in height: select "Height" () and validate.
7. Select the orange area and display the window for anchor management. The area must be enlarged in width and in height: select "Width and height" () and validate.

► To implement the anchoring of controls:

1. Display the page in Page mode: click the "Page" tab at the top of the page.



2. Select "Header".
3. Display the popup menu and select "Anchor".
4. The control must be centered in width: select "Centered horizontally" () and validate.
5. Repeat the same operations for the "Menu" control and for the "Content" control:
 - The "Menu" caption must be centered in height ().
 - The "Content" caption must be centered in width and in height ().

► Save the page (CTRL S) and run the page test () among the quick access buttons).

1. The page is displayed in the browser.



2. Resize the browser: the areas adapt to the available space in the browser.

► Close the browser.

PART 8

Responsive Web Design

LESSON 8.1. RESPONSIVE WEB DESIGN

This lesson will teach you the following concepts...

- What is Responsive Web Design?
- Benefits
- Concepts



Estimated time : 5 mn

What is Responsive Web Design?

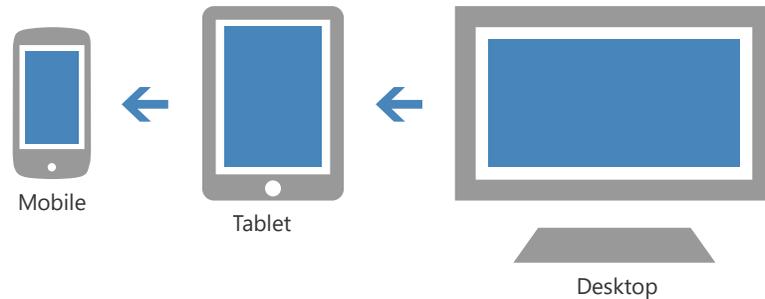
Responsive Web Design is a method for developing Web sites allowing the user to get the best possible reading quality regardless of the platform used (mobile devices, tablets, PC, ...).

Benefits

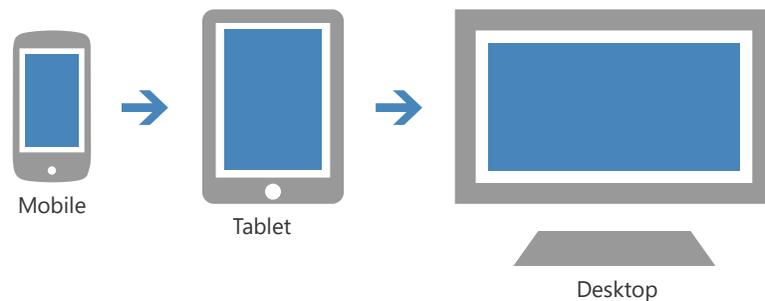
- The site can be displayed with a high-quality level on any platform (from mobile device to desktop via the tablet)
- The site is better referenced. Indeed, the search engines give priority to the sites that are properly displayed on all platforms.
- The maintenance is simplified: there is a single page to maintain for all devices.

The concepts in WEBDEV

The principle of Responsive Web Design development with WEBDEV consists in defining the page for the Desktop and in adapting it to smaller screen sizes. It is the principle of "Desktop First".



However, you also have the ability to operate in reverse direction: design the Mobile site and modify it by adding features to get the Desktop site. It is the principle of "Mobile First".



Several WEBDEV tools are available to help you. We will discover them in the next lesson.

LESSON 8.2. RESPONSIVE WEB DESIGN AND WEBDEV

This lesson will teach you the following concepts...

- Available WEBDEV tools
- Positioning grid
- Resolution blades
- Overloads



Estimated time : 30 mn

Overview

An example project will allow you to discover the methods that can be used in WEBDEV to create sites in Responsive Web Design mode. This allows you to perform the operations and to better understand the available tools.

- ▶ A simple example was prepared for you:
 1. Start WEBDEV 23 (if not already done). Close (if necessary) the current project to display the home window.
 2. Open the "WW_Responsive" project. To do so, in the home window, click "Tutorial" and select the project named "Responsive Web Design (Exercise)".



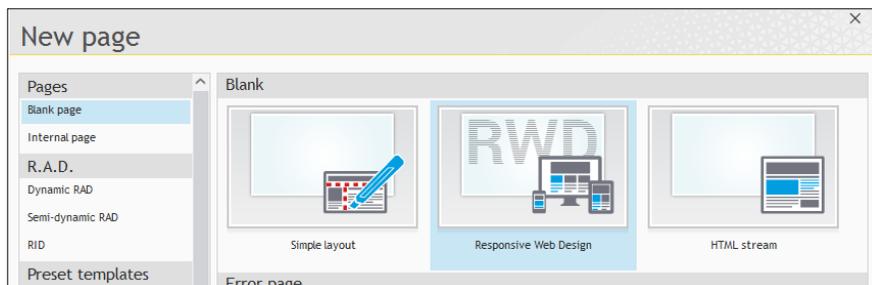
Tip

If the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Responsive Web Design (Exercise)".

My first page in Responsive Web Design mode

Creating a page

- ▶ In WEBDEV, the method for creating a page in Responsive mode is identical to the method for creating the other pages:
 1. Create a new blank page.
 - In the window for creating a new element, click "Page" then "Page".
Note: To display the window for creating a new element, click  among the quick access buttons.
 - The wizard for page creation starts.



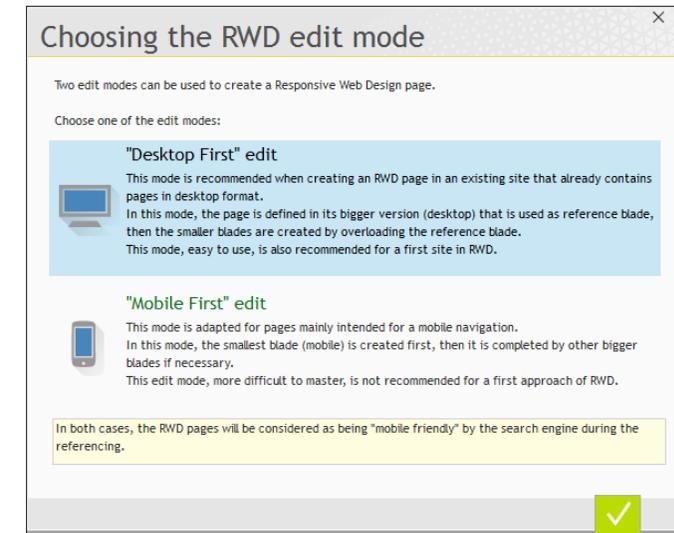
- Click "Blank Page - Responsive Web Design" and validate the wizard.



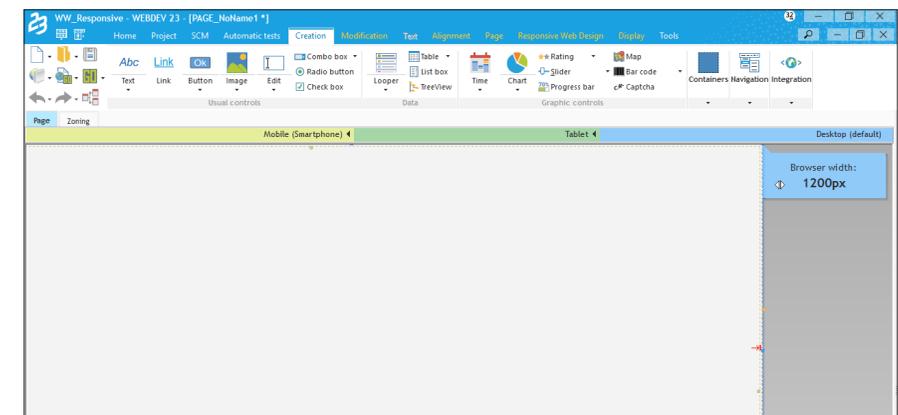
Notes

You also have the ability to use one of the Responsive page templates supplied with WEBDEV. We are now going to create a blank page to understand the principle of Responsive Web Design.

2. WEBDEV displays a window allowing to choose the main width used for Responsive Web Design. In this example, we are going to create a page for the "Desktop" mode.



3. Select "'Desktop First' edit" and validate.
4. The backup window is displayed. Type the page title: "Responsive". The name ("PAGE_Responsive") is automatically proposed.
5. Validate.
6. The page is created in the editor. This page is displayed in "Desktop" mode.



Notes

The new page is displayed in "Desktop first" mode. A menu option allows you to use the "Mobile first" mode if necessary.

Fluidity and Responsive Web Design

Let's see how the page behaves in the WEBDEV editor by creating a Static control.

► To create the Static control:

1. On the "Creation" pane, in the "Usual controls" group, click "ABC".
2. Then, click inside the page to create the control.



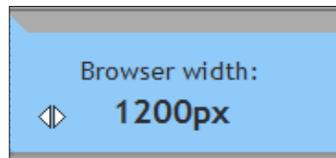
3. Press the F7 key three times to display the border of controls.



The first principle of responsive is fluidity: when the browser is reduced, the page controls are reduced accordingly.

► To check this fluidity in the editor:

1. Use the cursor found at the top of the page to reduce the browser size.



2. The Static control that was just created is reduced accordingly.

Resolution blades

The cursor that was just used belongs to the resolution blades, second principle of WEBDEV Responsive.

The resolution blades are used to materialize the resolutions of different devices.

► Let's study the operating mode of these resolution blades.

1. In the "WW_Responsive" project, open the PAGE_Adaptations page.
2. Three blades appear above the page:



- In blue, the Desktop blade, for the displays performed on a desktop,
- In green, the Tablet blade corresponding to the display on tablets,
- In yellow, the Mobile blade for the display on Smartphones.

The resolution blades allow you to define the layouts that will be used for each type of device.



Notes

You have the ability to add custom resolution blades to manage the different devices more efficiently.
To change resolution blade, you can use the cursor or click the resolution blade directly.



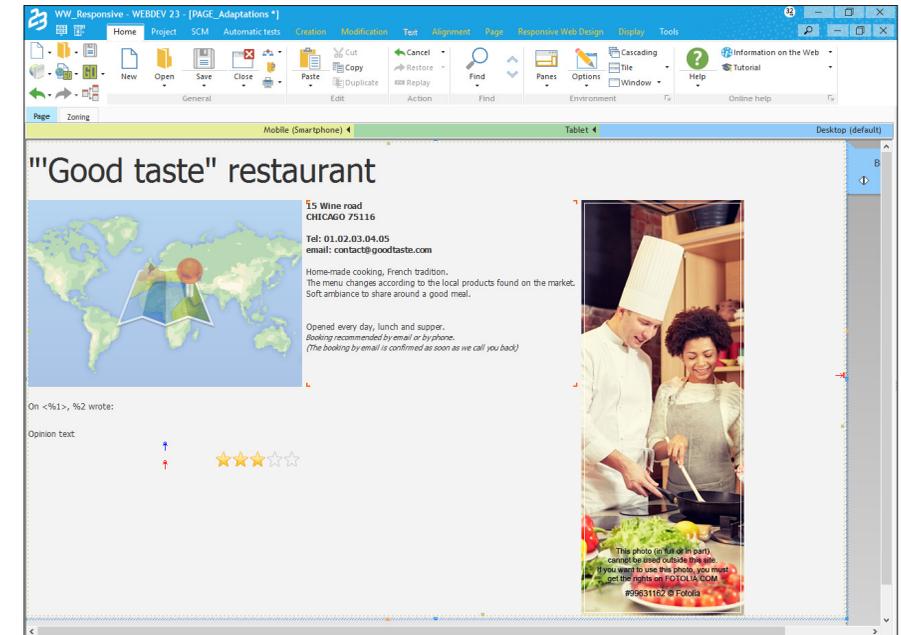
Caution

Responsive WEB Design with WEBDEV is using the concept of "Desktop First" by default. In this mode, the controls are always created in the Desktop blade. Then, the interface must be adapted to smaller blades.

The Desktop blade corresponds to the base of the page, without adaptation.

To perform adaptations in the other resolution blades: all you have to do is perform the modifications in the editor directly.

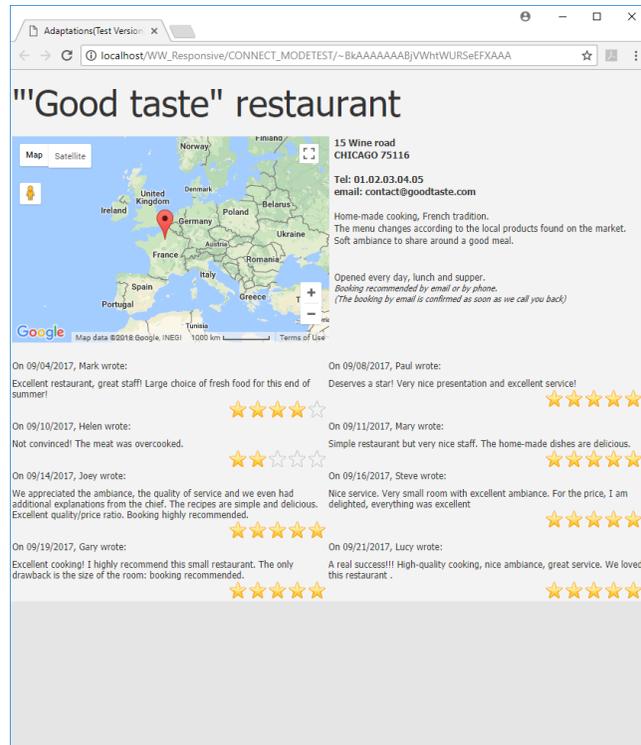
► Let's study the PAGE_Adaptations page.



► This page contains different controls:

- a title at the top,
- a Map control to indicate the restaurant location,
- a Static control containing the restaurant address,
- an Image control on the right to display an advertising banner,
- a Looper control at the bottom to display the opinions.

- ▶ Run the page test (🖥️ among the quick access buttons).
 - The page appears in test as it was in the editor in Desktop mode. This page is using a Map control. In GO and in deployment, a Google key must be used. Otherwise, an error occurs in the Map control. See the online help to get this key. If you own a key, specify it in the "Project" tab of the description window of project.
 - Reduce the size of your browser to see the change of blade.
- ▶ Some page controls change position or disappear according to the resolution blade displayed. For example, the advertising image is found in the Desktop blade but it becomes invisible in the smaller blades.



How to adapt the controls to the different blades?

Performing adaptations regarding the positioning and/or the size is child's play! The modifications are performed in the editor directly (like any other modification). In fact, these adaptations are overloads.



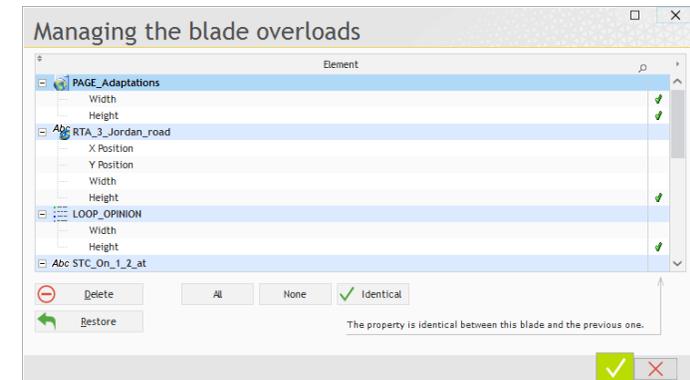
Caution

In order for the sizes to be respected when reducing the browser, we advise you to click the resolution blade to modify **before** any modification. The default size of the selected blade will be taken as reference and no error will occur at run time.

Several properties can be overloaded for each resolution blade, they can be identified in the description window of controls via the picto 🖥️.

For example, for the Image control, a different image file can be used according to the resolution blade.

- ▶ To overload the size and position of controls, you must:
 1. Select the requested blade.
 2. Perform the modification in the editor.
- ▶ To see the list of overloads for a resolution blade.
 1. Select the requested resolution blade (for example, for the tablet blade, click inside the tablet blade).
 2. Display the popup menu of blade bar (right mouse click) and select "List of overloads".
 3. The list of overloads performed is displayed:



Note: This window allows you to delete the overloads in order to restore the default operating mode of control.

Special case: Visibility of control

The visibility of control is defined in the "GUI" tab of the description window of control. In Responsive mode, this option cannot be overloaded by blade.

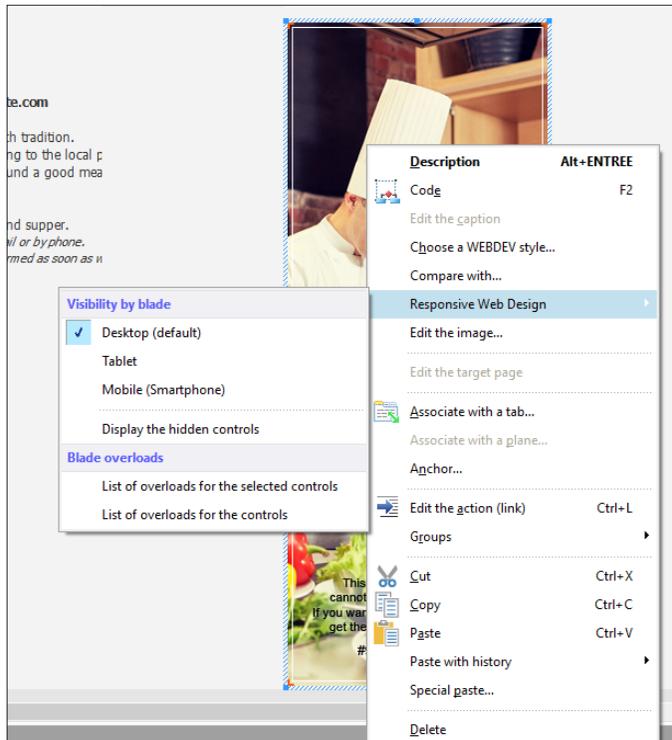
- ▶ To define the visibility of a control by blade, you must:
 1. Select the control.
 2. Display the popup menu of control (right mouse click).
 3. Select "Responsive Web Design .. Visibility by blade".

LESSON 8.3. ADVANCED CONCEPTS

This lesson will teach you the following concepts...

- Specific control: Navigation Bar control with "Hamburger" menu
- The anchors
- Managing the order of controls

 Estimated time : 30 mn



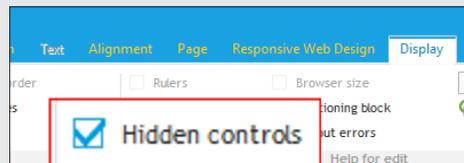
4. Keep checked the blades for which the control must remain visible.

Note: Only the visibility defined in the description window of control can be modified by programming. The visibility overloads performed by resolution blade cannot be modified by programming.



Notes

To see the hidden controls (to modify their position if stacked for example), on the "Display" pane, in the "Help for edit" group, check "Hidden controls".



Controls specific to Responsive Web Design

WEBDEV includes a specific control for Responsive Web Design: the Navigation Bar control. The Navigation Bar control is used to create a menu area that can easily adapt to the browser size. This area can contain all types of controls.

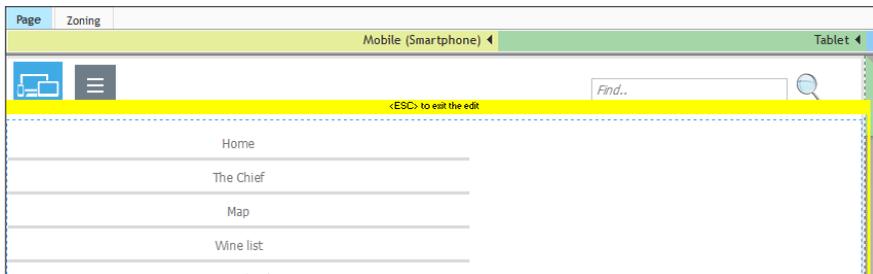
- ▶ Let's study the operating mode of Navigation Bar control.
 1. Open the "WW_Responsive" project if necessary. To do so, in the home window, click "Tutorial" and select the project named "Responsive Web Design (Exercise)".
 2. Open the PAGE_NavigationBar page.



- ▶ The Navigation Bar control is the control found at the top that contains the menu. This Navigation Bar control includes 3 sections:
 - On the left, the site logo.
 - In the middle, the site menu
 - On the right, a search control with a button used to start the search.
- ▶ Let's switch to Tablet mode:
 1. Click the header of Tablet blade.
 2. Click the Navigation Bar control to select it: the middle section is hatched.



- ▶ From the Tablet blade, the Navigation Bar control makes the middle section invisible and adds a Hamburger menu in the left section. The "hamburger" menu expands a popup containing the controls of middle section. This popup can be customized, all you have to do is click the central pane. Then, this popup can be modified according to your own needs.
- ▶ Let's study the content of the middle section:
 1. Click inside the hatched middle section. The popup is displayed:

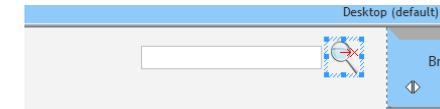


2. In our example, we can see the site menu. The menu is vertical now. Indeed, the Navigation Bar control automatically overloads the Menu control so that it is displayed vertically.
3. To exit from the edit mode of popup, press the ESC key.

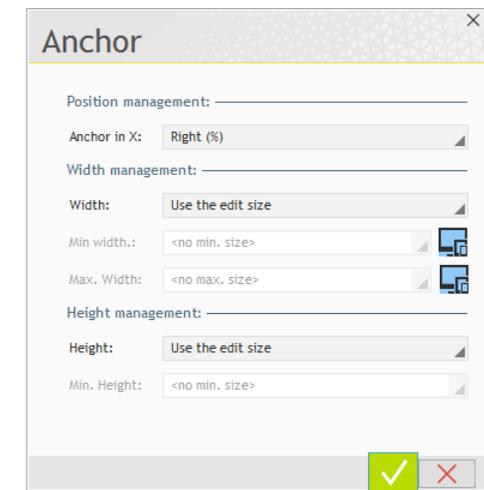
The anchors

The management of anchors was already presented in the previous chapters. However, the anchors operate differently in a Responsive Web Design page.

- ▶ Let's study the operating mode of anchors in Responsive Web Design mode.
 1. Open the "WW_Responsive" project if necessary. To do so, in the home window, click "Tutorial" and select the project named "Responsive Web Design (Exercise)".
 2. Open the PAGE_Anchor page.
 3. Select the search button at the top right:



4. Display the popup menu (right mouse click) and select "Anchoring".
5. The window for anchor management is displayed:



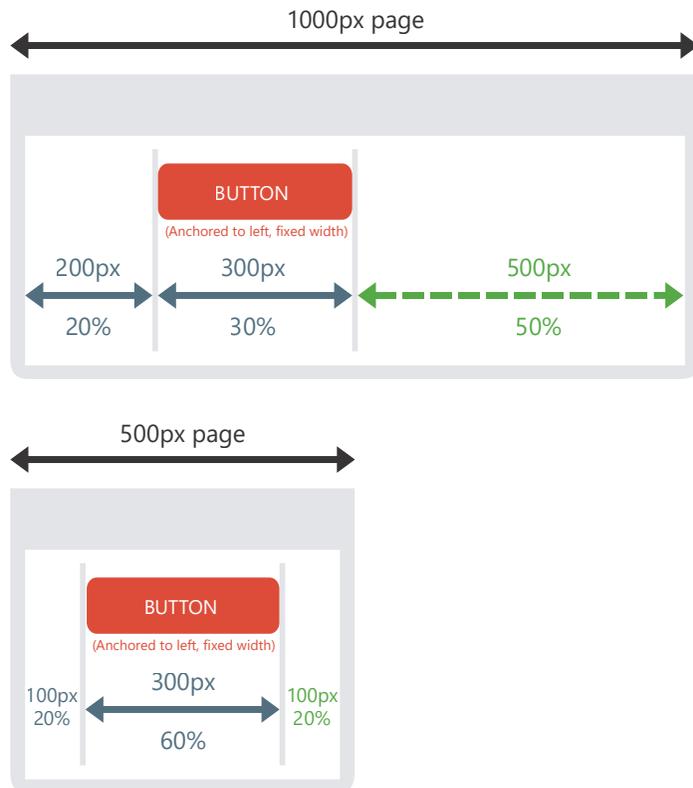
WEBDEV proposes two types of anchor:

- the positioning anchors,
- the size anchors.

Positioning anchors

The positioning anchors correspond to the "Anchor in X" option. This option proposes the following choice:

- An anchor to left.
 - A centered anchor.
 - An anchor to right.
- ▶ Let's study these different possibilities:
- The centered anchor must be selected in order for the control to remain centered when resizing the browser.
 - The anchor to left is used to specify that the size between the control and the left border of the page will always be the same proportionally to the page size. The section found on the right of control can be truncated if the browser reduction is too important.

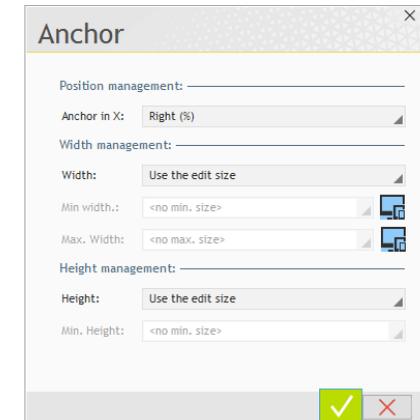


- The anchor to right performs the same action as the anchor to left, but on the right.
- ▶ In our example, the search button found at the top right of the page is anchored to right. Meaning that the spacing between the right border of control and the right border of page will always be the same proportionally to the page width.

- ▶ Run the page test ( among the quick access buttons).
 - The page appears in test as it was in the editor in Desktop mode.
 - Reduce the size of your browser.
 - The area on the left of search button is reduced while the area on the right remains the same.
- ▶ Close the browser.

Size anchors

- ▶ Let's now study the size anchors.
 1. In the PAGE_Anchor page, select the edit control named "Find" beside the search button.
 2. Display the popup menu (right mouse click) and select "Anchoring".
 3. The window for anchor management is displayed. The size anchors are grouped in the "Width management" and "Height management" sections.
- ▶ Let's study the different possibilities:



- Regarding the width, you can:
 - **use the edit size:** the control will neither be enlarged nor reduced: it will keep the width that was defined in the editor.

Note: If the control content is modified by programming and if it exceeds the edit size, the control will be enlarged.
 - **adapt the width to the grid:** when the page is reduced, the control is reduced proportionally. In this case, you have the ability to specify the minimum width and the maximum width. These widths can correspond to:
 - a specific value, expressed in pixels.
 - no minimum or maximum size.
 - the size defined in edit (which means the control size in the page editor).
 - **adapt the control to its content:** The control width depends on its content only. In this case, you have the ability to specify the minimum width and the maximum width. These widths can correspond to:
 - a specific value, expressed in pixels.
 - no minimum or maximum width.
 - the size defined in edit (which means the control size in the page editor).

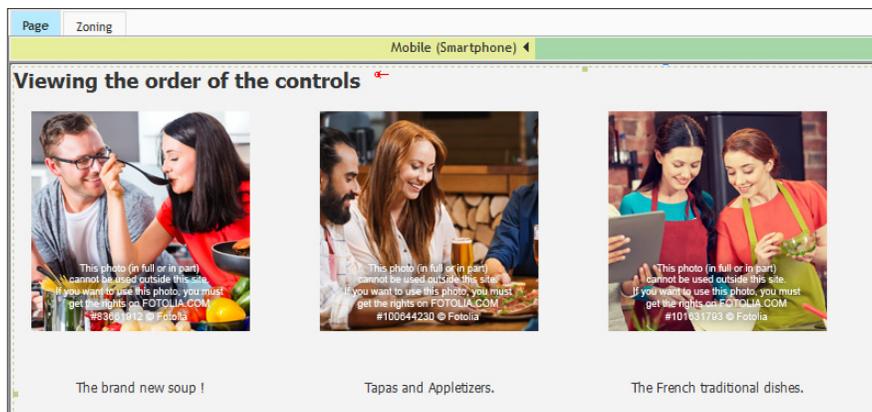
- Two possibilities are available for the height:
 - adapt the control height to its content. In this case, a minimum height must be defined.
 - use the edit size.
- ▶ In our example, the edit control and the search buttons are defined as keeping their edit size.
- ▶ Run the page test ( among the quick access buttons).
 - The page appears in test as it was in the editor in Desktop mode.
 - Reduce the size of your browser.
 - The edit control and the search button are not resized.
- ▶ Close the browser and go back to the editor.

Order of controls in Responsive Web Design mode

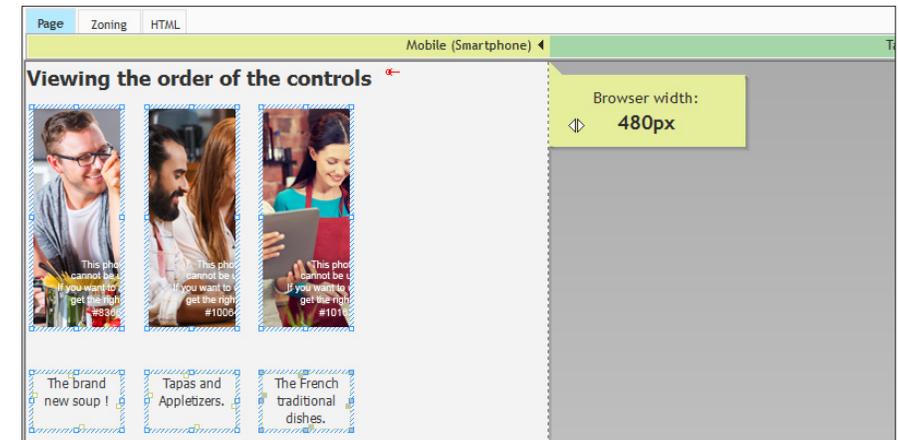
We have seen that the pages in Responsive Web Design mode adapt according to the browser size. The overload mechanism by blade is used to modify the interface according to the requested blade, especially by modifying the position of controls.

You must pay great attention to this modification of position. Indeed, in a responsive page, you cannot change the order of controls in the page.

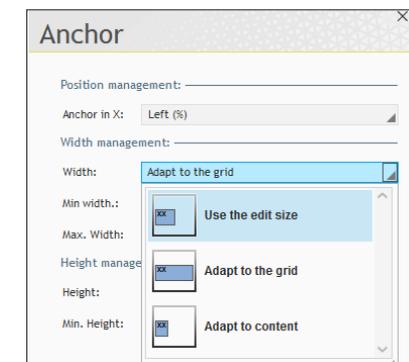
- ▶ To understand the management of order of controls:
 1. Open the "WW_Responsive" project if necessary. To do so, in the home window, click "Tutorial" and select the project named "Responsive Web Design (Exercise)".
 2. Open the PAGE_ControlOrder page.
- ▶ Let's study this page:
 1. In the Desktop blade, 3 products are presented horizontally.



2. Click the header of Mobile blade.



3. In the Mobile blade, the images of products are very reduced.
 - ▶ On Mobile, the controls should be displayed with the same size as in the Desktop blade and an image should be moved.
 - ▶ Go back to the Desktop blade to modify the anchor of controls.
 1. Select the Image control and the associated Static controls.
 2. Display the popup menu and select "Anchor".
 3. In the "Width" option, select "Use the edit size"

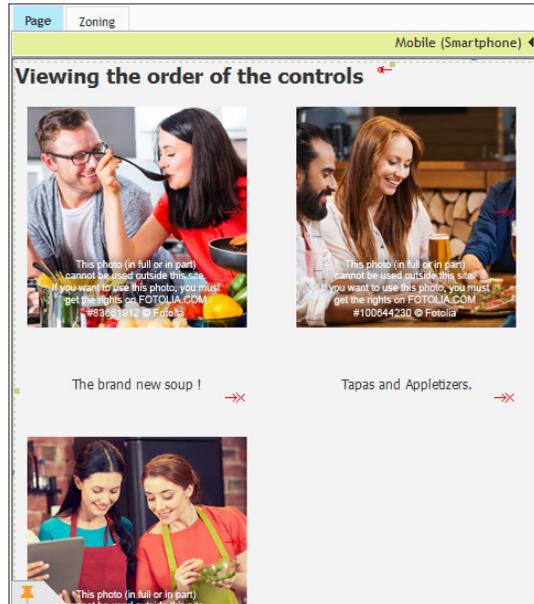


4. Validate the anchoring window.
5. Save the page (CTRL S). Two compilation errors occur.



These errors indicate that the `Img_Product_3` and `RTA_The_traditional_French_dishes` controls are not displayed inside the page at the end of Tablet blade.

- Click the Tablet blade. Move the "Browser width" cursor to the left. Indeed, from a particular browser dimension, the image and caption of 3rd product do not fit in the page anymore.
- Select these two controls in the Tablet blade and move them below the two first ones.



- Save the page (CTRL + S).
- New GUI errors occur in the pane of compilation errors.



- These errors are errors regarding the display order.

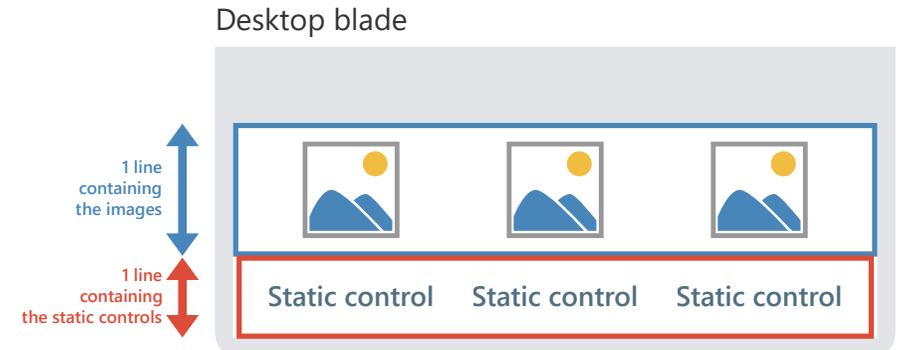
► Why do these errors occur?

This use case is quite common and it is important to understand the principle used.

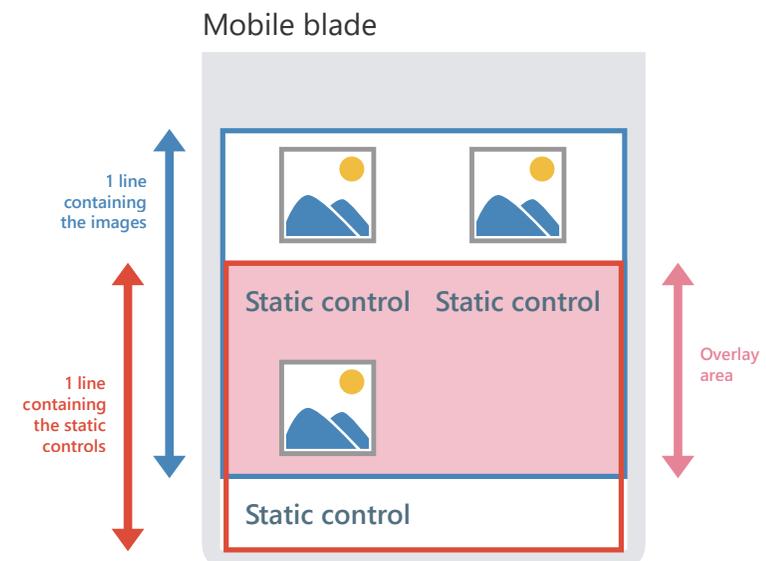
Each control is positioned in relation to its previous control.

In our example, each image is positioned in relation to the previous image. It is the same for the Static controls.

The first static on the left (number 4 in the example) is positioned in relation to the last image (number 3 in our example).

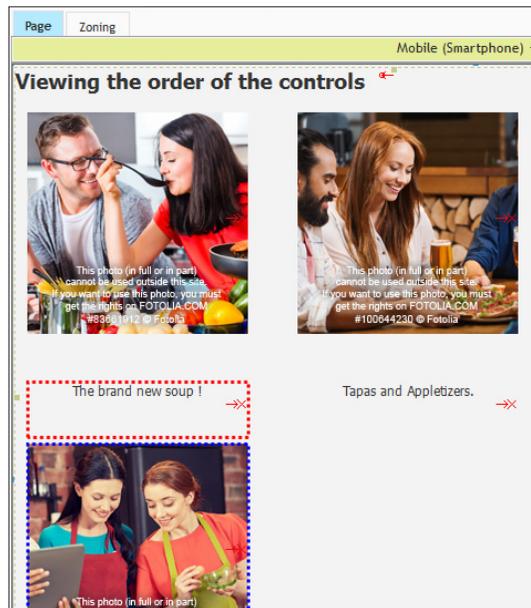


When the image and its associated static have been moved into the Tablet blade (image 3 and static 6 in our example), the static 4 was positioned "above" its "reference" control (the image 3 beside which it should be).



This positioning is not possible in Responsive Web Design. Therefore, new GUI errors have been displayed.

To guide you, the area where the change of display order is performed is bordered by red dots.



- To fix this problem, all you have to do is specify the logical formatting to WEBDEV. In our case, each Image control must be linked to its static: all you have to do is create cells around each "product" (image + caption). The cell must have an anchor such as "Use the edit size".



Notes

In most cases, in Responsive Web Design mode, to group controls and to get the same operating mode for each group control, we advise you to include them in a cell.

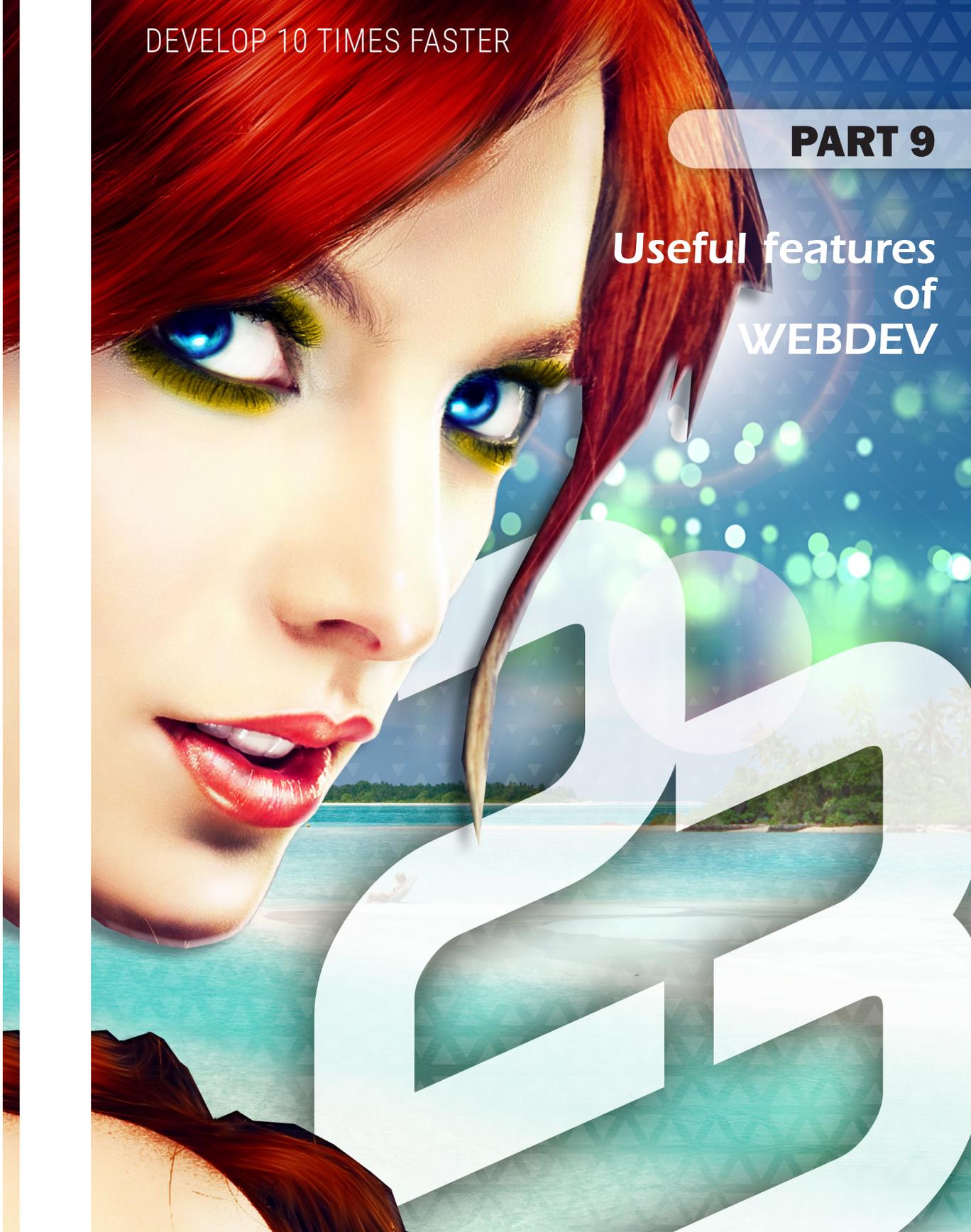
- To perform these corrections in our page, we are going to delete all the overloads performed in the Tablet blade:
 1. Display the popup menu of Tablet blade (right-click the section header).
 2. Select "List of overloads".
 3. In the window that is displayed, select all the overloads and click the "Delete" button.
 4. Validate the window.
- To create a cell containing one or more existing controls:
 1. Display the Desktop blade if necessary.
 2. Select the controls (the first image on the left and its static for example).
 3. On the "Modification" pane, in the "Transformations" group, expand "Refactoring and permutations" and select "Create a cell from the selection".
 4. Select the cell. Select "Anchor" from its popup menu.
 5. In the anchoring window, select "Use the edit size" for the anchor in width.

6. Validate.
7. Perform these operations again for the other two groups of controls.

- We are now going to reposition the controls in Tablet blade:
 1. Switch to Tablet blade.
 2. Select the cell containing the last two controls and move this cell below the first two controls.
 3. Save the page (CTRL S). No compilation error is displayed.

PART 9

**Useful features
of
WEBDEV**



LESSON 9.1. THE INTERNAL COMPONENTS

This lesson will teach you the following concepts...

- What is an internal component?
- Creating an internal component, step by step.
- Distributing and using an internal component.



Estimated time : 30 mn

Overview

An internal component is a group of project elements. This group is used to:

- organize a project,
- Share elements between different projects (mainly via the Source Code Manager).

When an internal component is included in a project, all the elements of the component are included in the project. The public elements can be handled in the editor directly. Furthermore, the internal component can be debugged from the project that is using it.

The projects that use an internal component have access in the WEBDEV editor to the name of objects, procedures or methods made visible by the component creator.

Creating a component is **child's play**.

How do I proceed? Don't change anything, create your pages, procedures, classes. Then, when it's done, choose the option for creating a component and that's it!

An internal component can contain code, pages, an analysis, data files, etc.

Step by step

Step 1: Creating an internal component

We are going to create an internal component used to subscribe and unsubscribe to an information letter in a Web site.

This component includes:

- a page, used to view the list of persons who subscribed to the information letter,
- a Web control template that contains the different controls (edit of the email address, validation button, ...),
- an analysis describing the data file of subscribers.

To avoid having to develop the code required for the component to operate, all the necessary elements have been grouped in a project named "WW_Internal_Component". We will be using this project to create our internal component. A new project will be created later to use this internal component.

► To open the example project:

1. Close the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the project named "Internal Component (Exercise)".



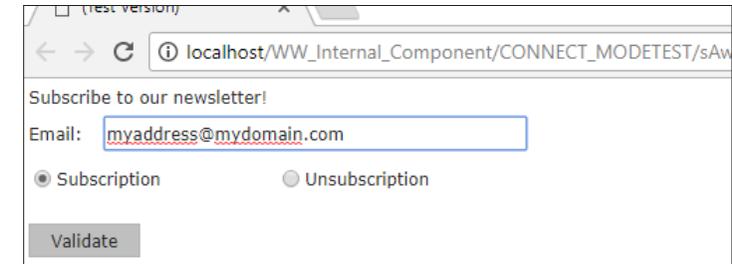
Notes

Tip: if the home window is not displayed: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Internal component (Exercise)".

3. The project is loaded

► We are going to test the control template of this project:

1. Display the Web control template named "TPLC_NewsletterRegistration" (double-click its name in the project explorer).
2. Run the test of the Web control template ( among the quick access buttons).
3. Enter an email address.

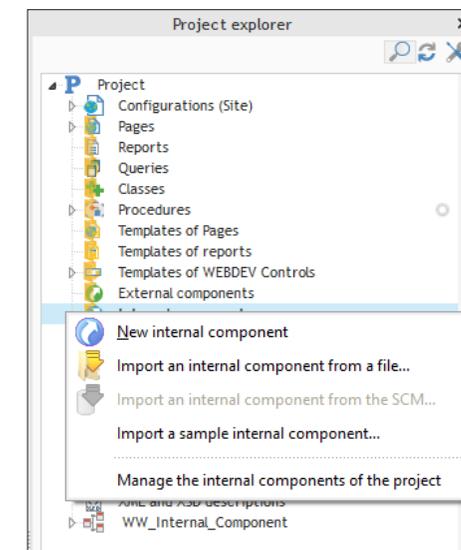


4. Click "Validate" to add the address into the data file.
5. Close the browser and go back to the editor.
6. The file content can be checked by WDMaP (accessible in the "Tools" pane of ribbon).

We are now going to create our internal component.

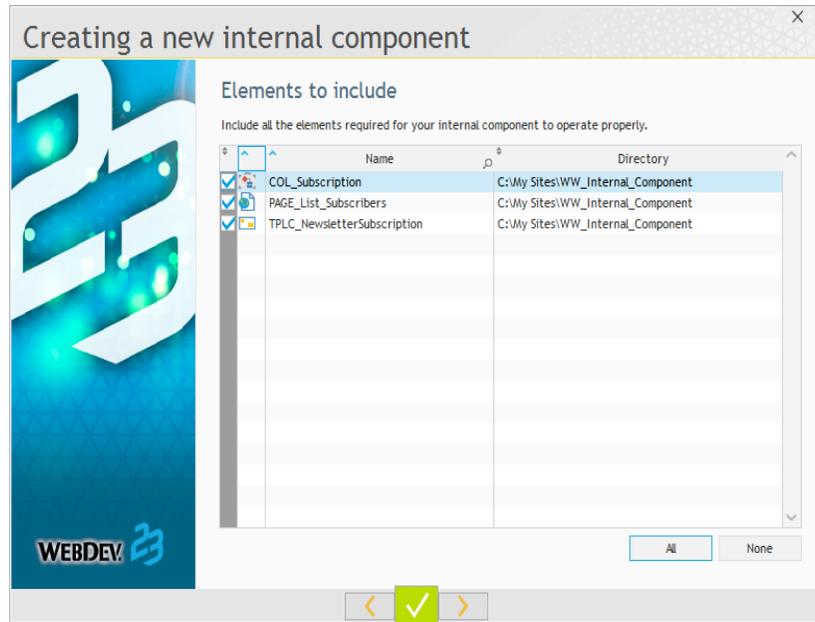
► To create an internal component:

1. In the project explorer:
 - Select the "Internal components" folder.
 - Display the popup menu (right mouse click).

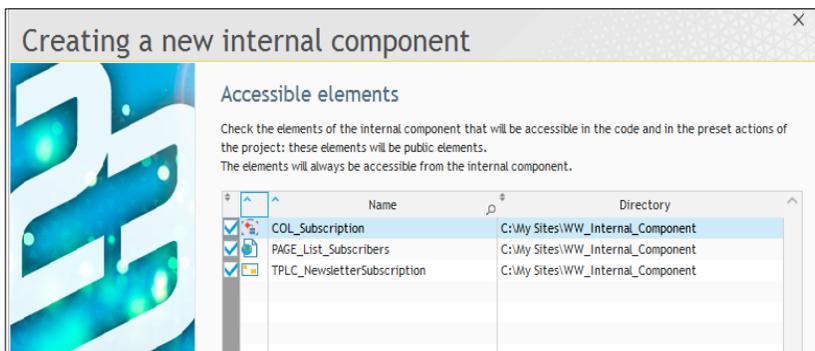


- Select "New internal component".
- The wizard for creating an internal component starts.

2. Go to the next wizard screen.
3. Identify your component: type the "InternalComponentNewsletterRegistration" name. The component caption is automatically proposed.
4. Go to the next step.
5. Select the elements that will be included in the component. In our case, all the elements must be selected.

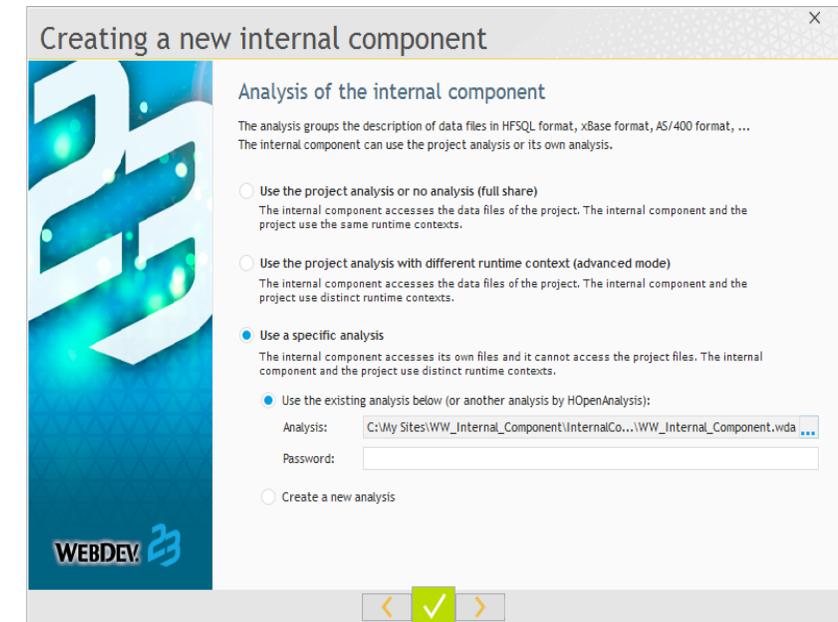


6. Go to the next step.
7. The wizard asks you to select the component elements that will be accessible from the client project. In our case, all the elements must be selected.

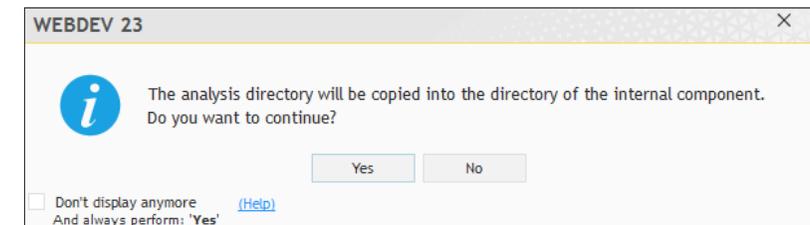


8. Go to the next step.

9. This step is used to specify the mode for managing the component data. In our case, the internal component is using its own analysis.



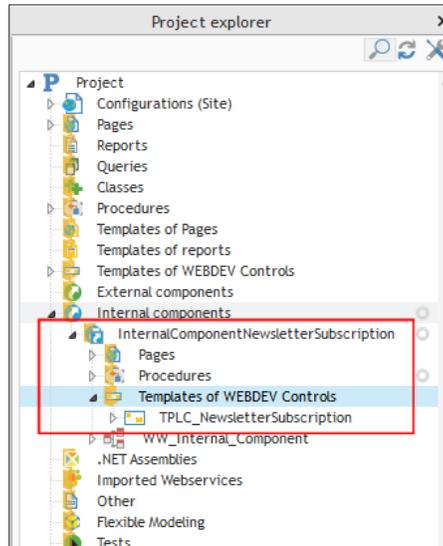
10. Select "Use a specific analysis" then, in the "Analysis" control, select the analysis corresponding to the current project ("WW_Internal_Component.wda" file found in the "WW_Internal_Component.ana" sub-directory of project). WEBDEV proposes to copy the analysis directory into the directory of internal component.



Accept.

11. End the wizard. WEBDEV will create the internal component in a specific directory of your project.

In the project explorer, the "TPLC_NewsletterSubscription" Web control template is no longer found in the list of Web control templates of project. On the contrary, the internal component that was just created is listed in the "Internal Component" folder, along with its different elements: analysis and Web control template.



Step 2: Using the internal component

Once created, your internal component can be used in any other WEBDEV project. Let's now see how this component can be re-used.



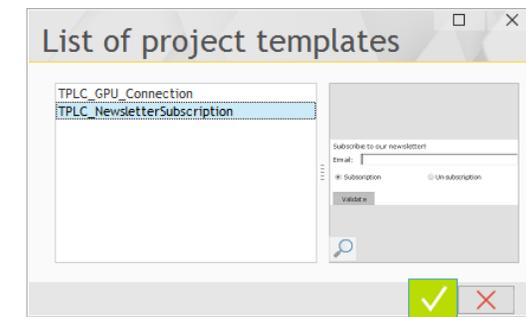
Notes

In this example, we will present a "direct" use of the internal component. To share resources, we recommend that you use the internal components via the Source Code Manager (SCM). See the online help for more details (keyword: "Internal component").

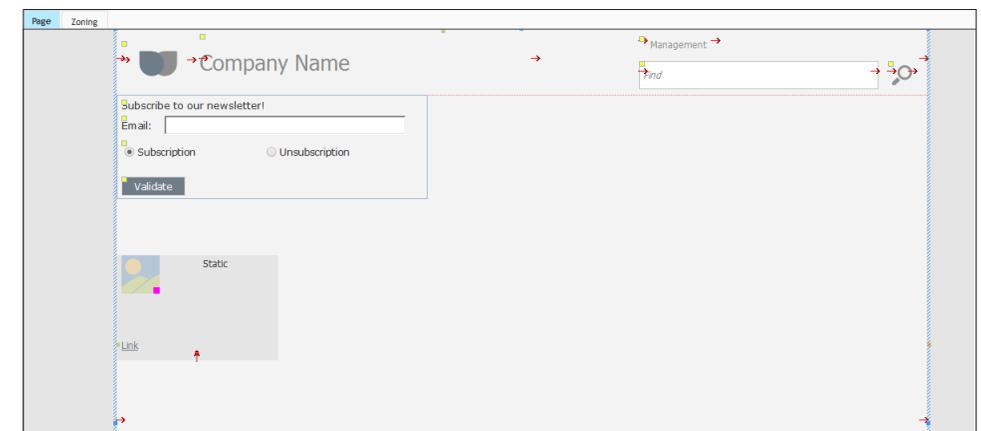
In our example, we are going to import the internal component into the "Full_WEBDEV_Site" project.

- ▶ Open the "Full_WEBDEV_Site" project.
 1. Close the current project if necessary. The home window is displayed.
 2. In the home window, click "Tutorial" and select "Full WEBDEV site (With pages)". Tip: if the home window is not displayed: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (With pages)".

- ▶ Include the internal component in the project.
 1. In the ribbon, on the "Project" pane, in the "Project" group, expand "Import" and select "An internal component .. From a file".
 2. Select the "InternalComponentNewsletterSubscription.wci" file (found in the "WW_Internal_Component\InternalComponentNewsletterSubscription" sub-directory of WEBDEV tutorial).
 3. The internal component is included in the project.
- ▶ To use the internal component, we are going to instantiate the Web control template supplied by the internal component in the home page of the site. To do so:
 1. In the editor, display the "PAGE_List_of_new_products" page (double-click its name in the project explorer).
 2. Move the looper found in the page to the bottom in order to insert the control template of internal component.
 3. Select the "Creation" pane of ribbon and in the "Containers" group, click "Control template".
 4. The window for selecting the template to instantiate is displayed.

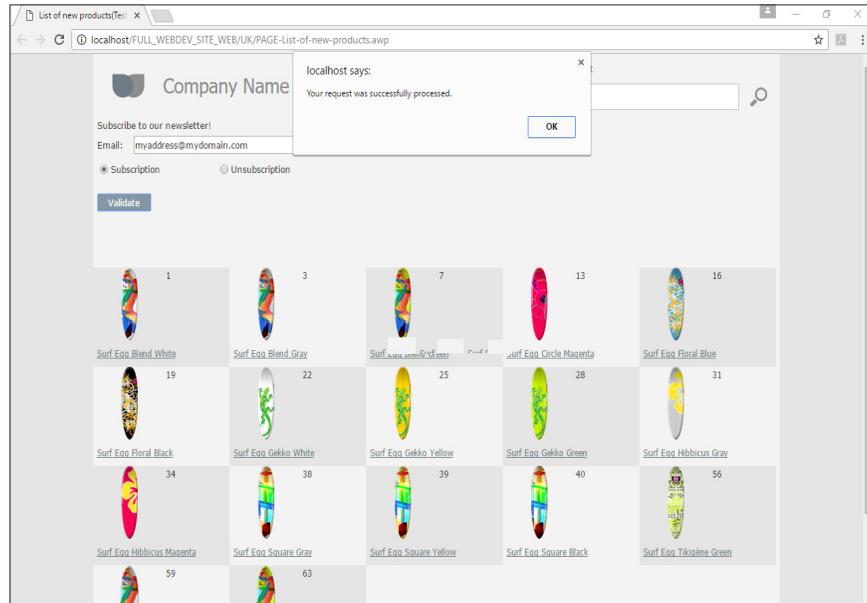


5. Select the template coming from the internal component and validate.
6. Click the location where the Web control template must be positioned in the page.



► We are going to check the operating mode of the control template:

1. Run the page test (Go among the quick access buttons).
2. Enter an email address.
3. Click "Validate".
4. A message confirms the registration.



LESSON 9.2. AUTOMATIC MANAGEMENT OF ERRORS

This lesson will teach you the following concepts...

- What is the automatic management of errors?
- Using the automatic management of errors.



Estimated time : 10 mn

Overview

WEBDEV can manage the errors automatically. This feature helps you reduce the number of code lines while centralizing the management of errors.

The use of this feature also makes the code easier to read.

Operating mode

Two operations are performed when an error is detected by a WLanguage function:

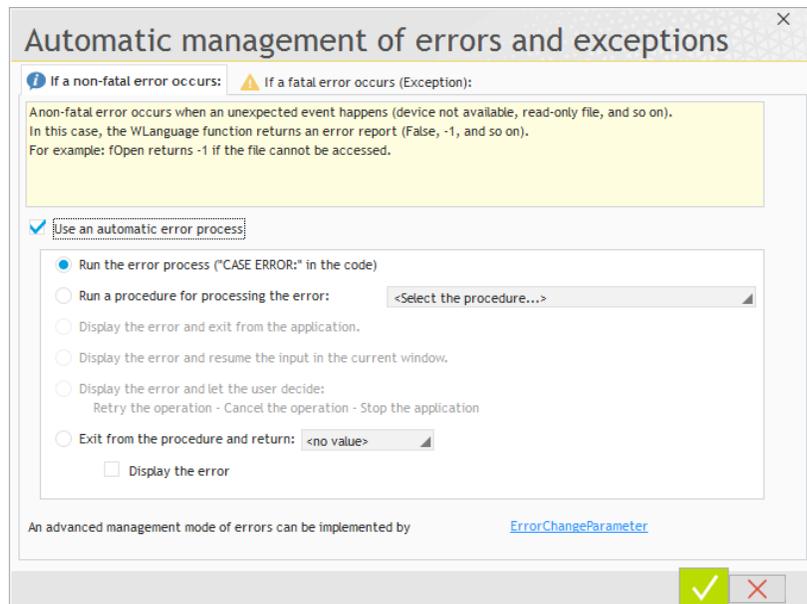
- a return value of error is returned by the function (**fOpen** returns "-1" if the specified file was not opened for example).
- the error is detected by WLanguage (the **ErrorDetected** variable is set to True) and the error details are returned by **ErrorInfo**.

This second operation can be automatically managed via the error management of WEBDEV.

Implementation

The automatic management of errors can be configured:

- in the code editor: all you have to do is click the link "If error: By program" in the code editor:



- by programming with **ErrorChangeParameter**.

Types of affected errors

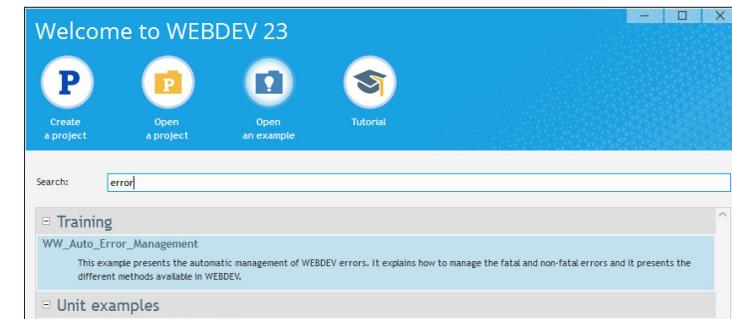
Two types of errors can occur in WLanguage:

- the "non-fatal" errors (also called runtime errors): in most cases, these errors are managed in the code and they do not stop the application. For example, opening a file that cannot be accessed or an archive that does not exist.
- the "fatal" errors (also called programming errors): in most cases, these errors are linked to a development problem (access to a non-declared file, use of non-existing controls, ...). A "fatal" error can also occur after a "non-fatal" error that was not processed properly. In this case, the application will be stopped.

The mechanism for managing the errors is used to manage these two types of errors according to different methods in order to specify behaviors adapted to the errors that occur.

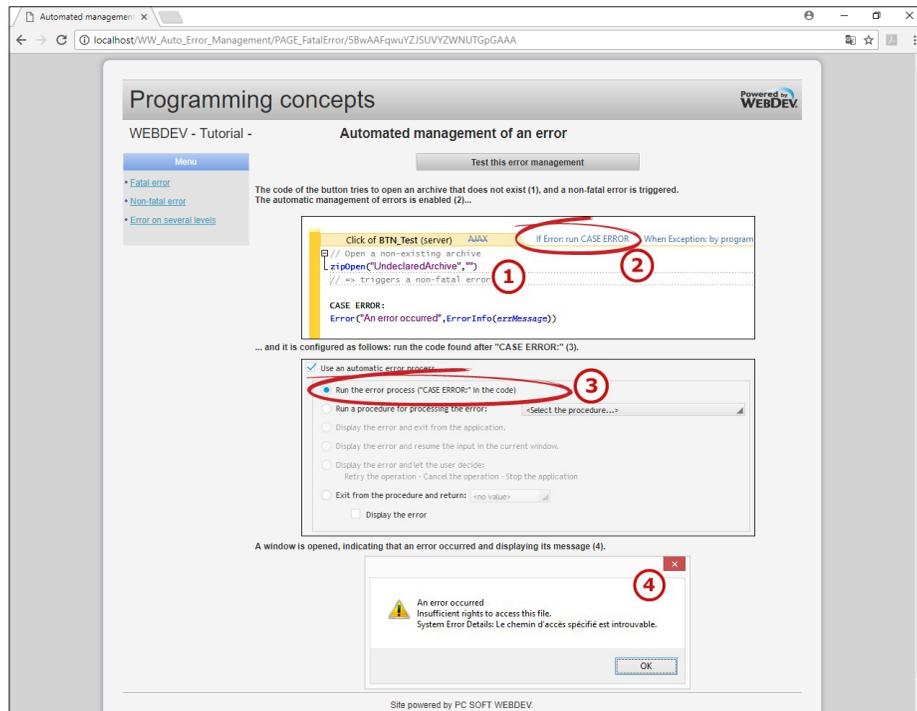
Automatic management of errors: a training example

- ▶ To understand the different error cases as well as their management, we will be using a training example supplied with WEBDEV.
 1. Close the current project if necessary. The home window is displayed.
 2. In the home window, click "Open an example". The list of complete examples and training examples supplied with WEBDEV is displayed.
 3. Type "Error" in the search area. Only the examples containing this word are listed.



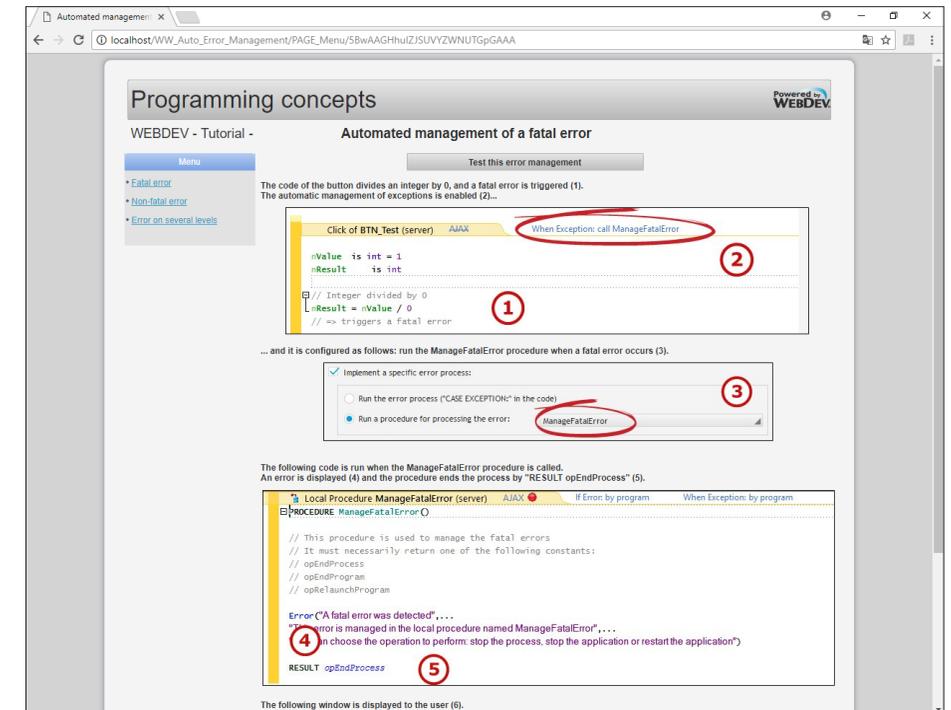
4. Select the "WW_Auto_Error_Management" project. The project is loaded.

- ▶ This project explains how to manage:
 - a non-fatal error (opening an archive that does not exist).
 - a fatal error (division by 0).
 - of an error on several levels.
- ▶ Run the project test by clicking  (among the quick access buttons).
 1. Click "Non-fatal error".
 2. The following page is displayed.



3. Click the "Test this error management" button.
4. When running the code line that triggers the error, an error message is displayed.
5. Validate the error message.

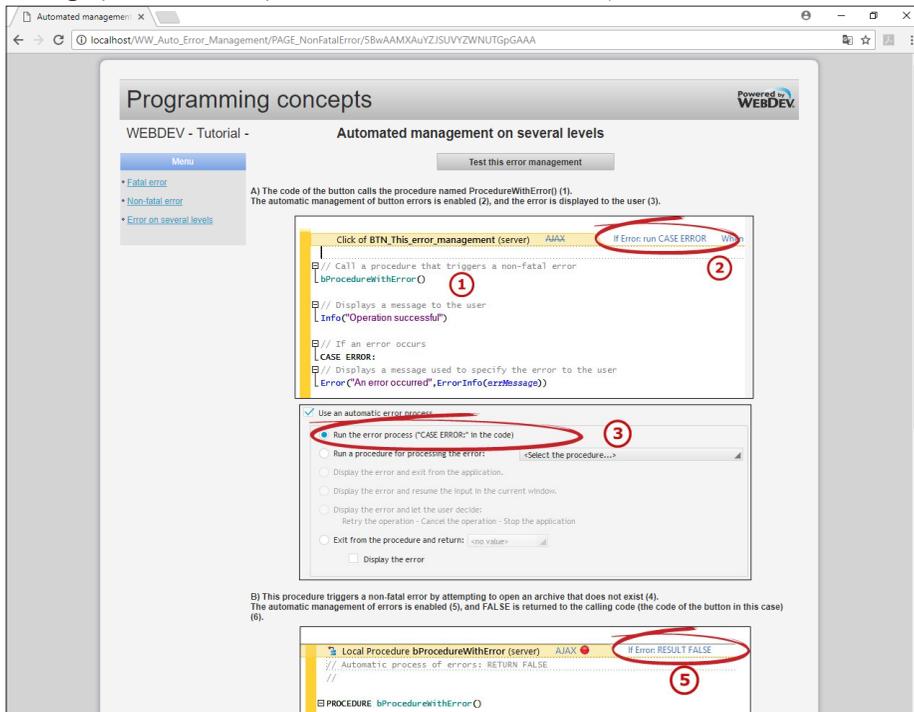
- ▶ Click "Fatal error".
 1. The following page is displayed. This page is used to test the error when an integer is divided by 0.



2. Click the "Test this error management" button.
3. When running the code line that triggers the error, a procedure is automatically called. This procedure is used to display the error message and to stop the current process.
4. Click "OK".

► Click "Error on several levels".

1. The following page is displayed. This page is used to test an error on several levels (process calling a procedure that opens an archive that does not exist).



2. Click the "Test this error management" button.

3. When running the code line that triggers the error:

- the procedure returns "False" to the calling process.
- the calling process displays an error message and stops the process.

4. Validate the error message.

► Close the browser to end the test.



Example

WEBDEV also proposes an automatic management of HFSQL errors. See the online help for more details (keyword: HFSQL, Managing errors).

LESSON 9.3. IMPORT/EXPORT

This lesson will teach you the following concepts...

- Importing elements from a project to another one.
- Exporting the elements of your project.
- Specific import operations (HTML pages)



Estimated time : 10 mn

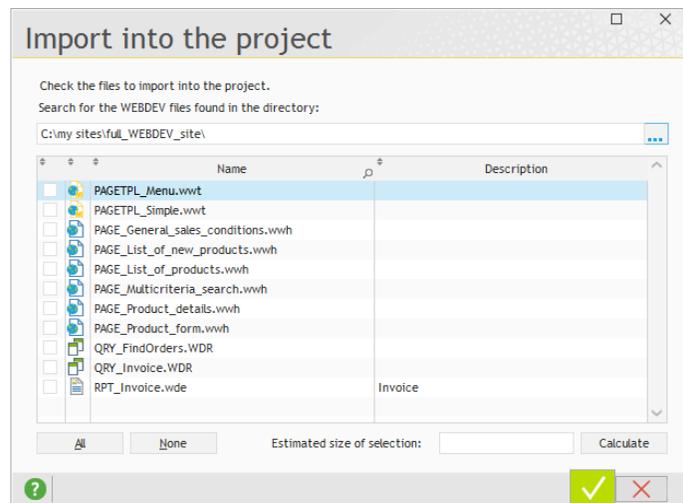
Importing elements

You have the ability to import existing WEBDEV elements into your project. All types of WEBDEV elements can be imported:

- pages, reports,
- classes, components,
- procedures, ...

► To import existing elements into the current project:

1. On the "Project" pane, in the "Project" group, expand "Import" and select "WEBDEV elements and their dependencies...".
2. Click "..." and select the directory containing the elements to import (the directory must include WEBDEV elements).
3. Validate. WEBDEV lists the directory elements that can be imported (the sub-directories are ignored).



4. Select the elements to import and validate. The elements (and all the files used by these elements : images, ...) are now included in the project.



Note

The "Calculate" button (found in the import window) is used to calculate the size of selected elements along with their dependencies.

Exporting elements

You also have the ability to export elements from your project to another directory for example. These elements can be re-used in other projects.

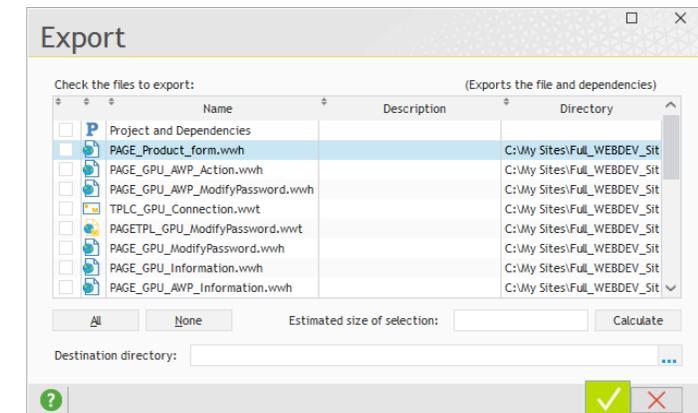


Caution!

Exporting elements is a lot more than a simple copy of elements. It is actually the only safe method for transferring a project element along with its dependencies (images, icons, ...). A practical solution for transmitting pages along with their dependencies by email for example.

► To export elements from your project:

1. On the "Home" pane, in the "General" group, expand "Save" and select "Export .. To a directory...".
2. In the window that is displayed, select the project elements that will be exported.



3. Specify the name of destination directory (or select it with the "..." button).
4. Validate. The elements are exported to the specified directory. These elements are still available in your project.



Note

The "Calculate" button (found in the import window) is used to calculate the size of selected elements along with their dependencies.

Specific import operations

Importing a WINDEV project

WEBDEV allows you to import a window or a full WINDEV project into a WEBDEV project.

During this import:

- The windows are changed into pages.
- The codes are changed into server codes.
- The elements with no equivalent in WEBDEV are imported as comments or as separated elements.

► To import a WINDEV project into a WEBDEV project:

1. On the "Project" pane, in the "Project" group, expand "Import" and select "A WINDEV or WINDEV Mobile project". The wizard for importing a WINDEV or WINDEV Mobile project starts.
2. Select the WINDEV project to import. This project will not be modified and a new WEBDEV project will be created.
3. Specify the name and location of the WEBDEV project to create.
4. If an analysis is linked to the project, specify its name, its location, and whether the analysis must be used by the WEBDEV project.
5. Specify the elements that will be shared between the WEBDEV project and the WINDEV project (common elements such as "Reports", "Classes", ...). If the elements are shared, they will not be copied into the WEBDEV project.
6. Validate. The WINDEV project is changed into WEBDEV project.

► To import WINDEV elements into a WEBDEV project :

1. Open the WEBDEV project into which the element must be imported.
2. On the "Project" pane, in the "Project" group, expand "Import" and select "WINDEV or WINDEV Mobile elements". The import wizard starts. Select the WINDEV project containing the elements to import or select the WINDEV elements to import. Go to the next step.
3. Specify the elements that will be shared between the WINDEV and WEBDEV projects. These elements will not be copied into the WEBDEV project.
4. Validate. The specified elements are automatically imported into the current WEBDEV project.

Importing an HTML page

A page found in a non-WEBDEV site is useful? You want to retrieve its interface? Nothing's easier. WEBDEV gives you the ability to import your existing HTML pages into your WEBDEV project. Each imported page becomes an element of your WEBDEV site. This page can be modified!

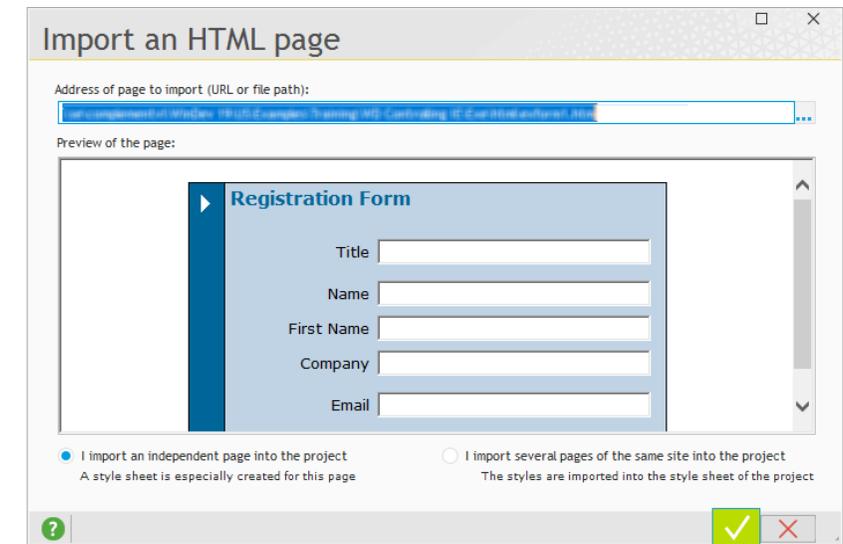


Note

The feature for importing HTML pages is a great help given to the developers of WEBDEV sites. This is not an Internet site "sucking" feature. However, some display differences may be noticed between the HTML page before the import and the HTML page after the import.

► To import an HTML page into a WEBDEV project:

1. Open your WEBDEV project (for example: an existing project or a new project).
2. On the "Project" pane, in the "Project" group, expand "Import" and select "An HTML page...". The wizard for importing HTML pages starts.



3. Specify the location of HTML page. This page can be a file found on your computer ("..." button) or an Internet address (for example: <http://www.mywebdevsite.com/mypage.html>). The HTML page is displayed in the "Page preview" frame.
4. Validate. A new WEBDEV page was just created.
5. Save the page. The page is automatically added to the list of project elements.

PART 10

Optimizing and debugging a project



LESSON 10.1. OVERVIEW

This lesson will teach you the following concepts ...

- Why optimize a site?
- Example project



Estimated time : 5 mn

Overview

Your site is created. It operates. You want to deploy it.

Did you think of using WEBDEV tools to optimize your site? WEBDEV proposes several tools and features allowing you to quickly optimize your site and to easily avoid predictable bugs.

This section presents these tools and their use.

A project to optimize was prepared, allowing you to handle these features!

Opening the project

- ▶ Start WEBDEV (if not already done). Close (if necessary) the current project to display the home window.
- ▶ Open the "WW_Optimization" project. if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Optimize a project".

LESSON 10.2. PROJECT AUDITS

This lesson will teach you the following concepts ...

- What is an audit and what is its purpose?
- Starting and studying the static audit
- Starting and studying the dynamic audit



Estimated time : 20 mn

What is an audit?

The audits provide a set of features used to automatically improve the project quality and performances, and to follow the conditions in which it is implemented.

Two types of audits are available:

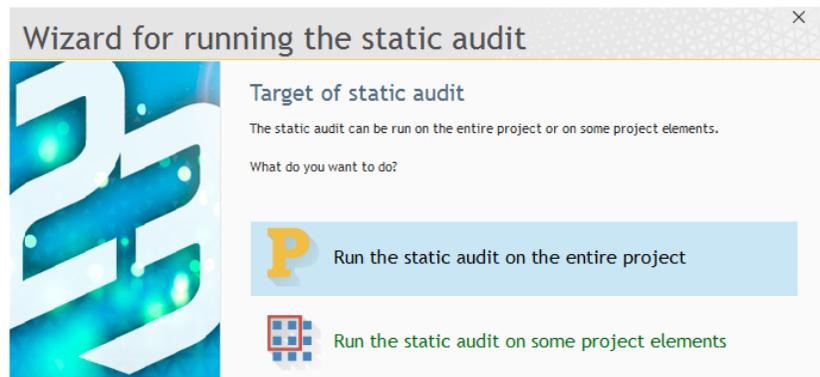
- **The static audit.** The static audit returns a full report regarding a project and its elements. This audit is performed from the project editor.
- **The dynamic audit.** The dynamic audit studies the behavior of a project during its execution. This audit can be performed in test mode or in the production environment.

We are going to test these audits on the "WW_Optimization" project.

Static audit

The static audit is an environment feature used to study the source code of a project in order to detect different problems and to propose improvements..

- ▶ To start the static audit on the "WW_Optimization" project:
 1. In the ribbon, on the "Project" pane, in the "Audit and performances" group, expand "Edition audit" and select "Trigger the edition audit".
 2. The wizard starts. We are going to define the target of static audit.



3. Select "Run the static audit on the entire project".
4. Validate the wizard.

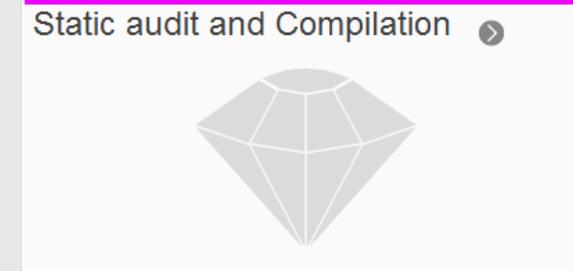
The static audit of project can also be started from the project dashboard, via the "Static audit and Compilation" widget.

You must:

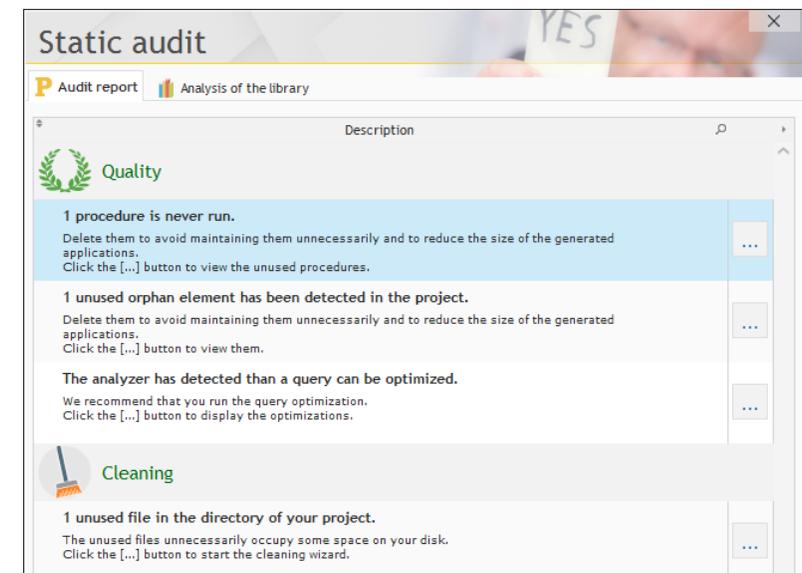
- display the Widget if necessary:
 - Switch the dashboard to modification mode (right mouse click, "Modification mode" option).
 - Add the Widget (right mouse click, "Add .. Static audit" option).
 - Close the modification mode (right mouse click, "Modification mode" option).
- click the arrow.



Notes



5. The audit report is displayed:



The static audit includes:

- The static audit of project.
- The audit of content of application library.
- ▶ Let's study the topics presented by this report.

Procedure not run

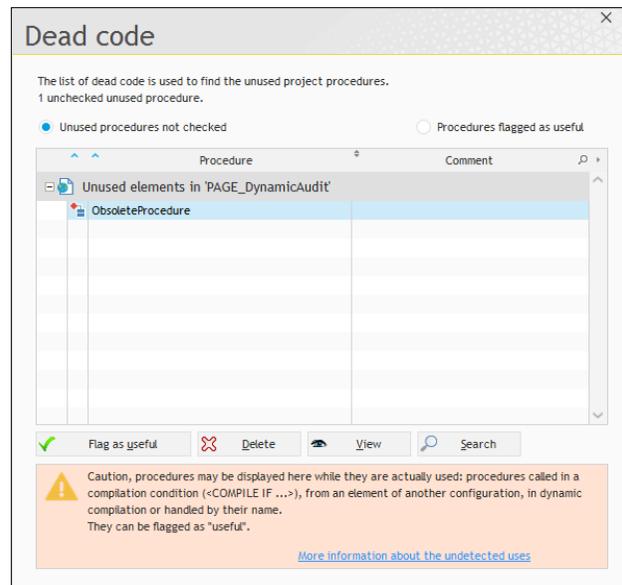
In our project, the audit indicates that a procedure is never run.

In huge projects, you may want to create one or more procedures to perform a process then, further to a code reorganization, the procedure is no longer used but it remains in the project.

The presence of unused procedures unnecessarily consumes the resources distributed to the end users.

► To fix this problem:

1. Click the [...] button to get more details. The window that lists the dead codes is displayed.



2. The "ObsoleteProcedure" procedure is never called. This window is used to:
 - specify that the procedure is still used ("Mark as useful" button). In this case, the procedure will not be taken into account anymore during the audit.
 - delete the procedure if it is actually unused ("Delete" button).
 - see the procedure ("View" button).
 - find the use cases in strings for example ("Search" button).
 3. In our case, this procedure is actually unused, click "Delete".
 4. A window is displayed, asking you to confirm the deletion. Click the "Delete" button to confirm the deletion.
 5. Close the window of dead code (click the cross in the top right corner).
- In the window of static audit, click the "Refresh" button to update the audit report.

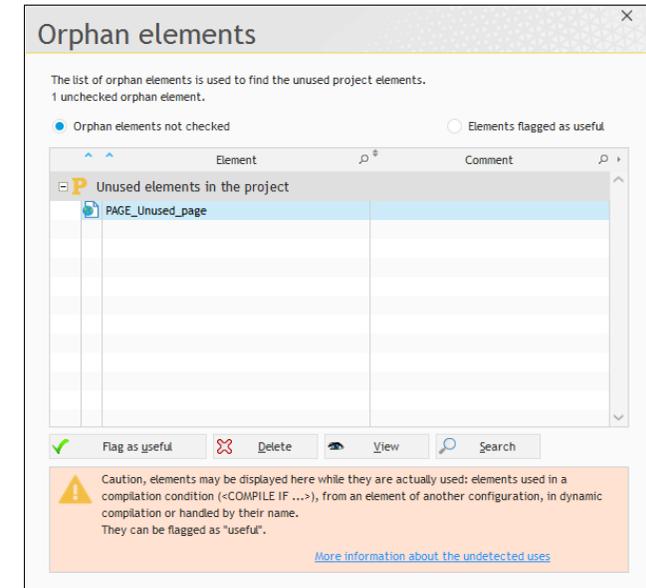
Orphan element

The audit tells us that our project contains an orphan element.

As for the procedures run, you may create windows or reports in order to run a quick test, save them and forget everything about them. The presence of orphan elements in the executable unnecessarily consumes the resources distributed to the end users.

► To fix this problem:

1. Click the [...] button to get more details. The window that lists the orphan elements is displayed.



2. The "PAGE_Unused_Page" page is never called. The window that lists the orphan elements is used to:
 - specify that the element is still used ("Mark as useful" button). In this case, the page will not be taken into account anymore during the audit. This option can be interesting when using a test page specific to the development for example.
 - delete the page if it is actually unused ("Delete" button).
 - see the page ("View" button).
 - find the use cases in strings for example ("Search" button).
 3. In our case, this "PAGE_Unused_Page" page is actually unused, click "Delete".
 4. Close the window of orphan elements (click the cross in the top right corner).
- In the window of static audit, click the "Refresh" button to update the audit report.

Cleaning the project

Our project contains several unused files. You have the ability to clean the project in order to keep the necessary elements only. The client setup is not weighted with images, external files, ... not used.

► To fix this problem:

1. Click the [...] button to get more details.
2. The wizard for cleaning a project starts. This wizard indicates the unused files that can be deleted.
3. Go to the next step.
4. Select the type of cleaning to perform. You can:
 - create a zip file with the useless files.
 - move the useless files into a specific directory.
5. Validate the option proposed by default and go to the next step.
6. End the wizard.

Our project was optimized according to the tips given by the static audit.

The static audit is used to get an overall status on the source code of your project. Our advice: run it on a regular basis!

Let's see what happens at run time by starting the dynamic audit.

Dynamic audit

The dynamic audit is used to study the application execution. The audit is used to detect problems such as:

- Excessive memory consumption,
- Slowness of algorithms used,
- Errors "hidden" at run time,
- ...

A dynamic audit can be performed in a test environment or on a live application.

The "WW_Optimization" project contains a specific page triggering errors that can be detected by the dynamic audit.

The dynamic audit and the project test will be started at the same time.

► To start the dynamic audit on the "WW_Optimization" project:

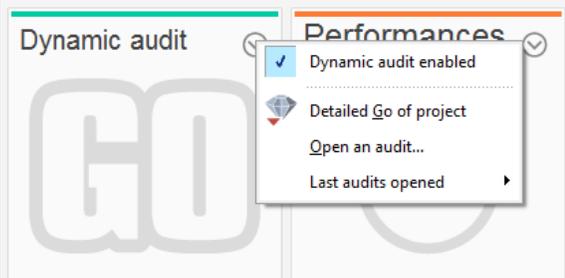
1. In the ribbon, on the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug the project while the audit enabled". The project test is run.



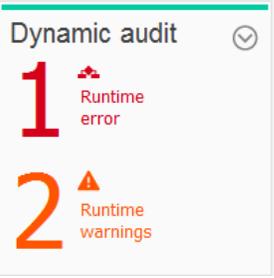
Notes

The dynamic audit of the project can also be started from the project dashboard, via the "Dynamic audit" widget. You must:

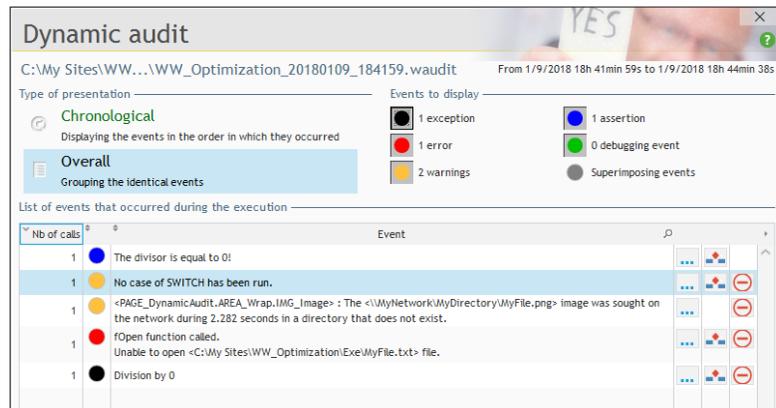
- enable the Widget if necessary (click the link "Click here to re-enable").
- expand the arrow and select "Detailed Go of project".



Note: The dynamic audit is automatically run whenever a project GO is performed. A status report is automatically displayed in the dashboard:



2. Click the "Test page of dynamic audit" button.
3. Click the different buttons found in the page.
 - At the end of each process, a toast message is displayed to specify that the process is over. Wait for the toast to be displayed before clicking another button.
 - For the "Assertion and Exception" option, an assertion is displayed: click "Continue" in order for the message to appear in toast format.
4. Stop the site test.
5. The report window of dynamic audit is displayed.



► Let's study this window:

- The top section of this window is used to choose the display mode of data. You can:
 - choose a chronological display (respecting the order in which the events occurred) or an overall display, used to group the different types of problems. During a chronological display, a chronological border allows you to see the position and importance of problems.
 - choose the type of problem to display (error, assertion, ...). This allows you to concentrate on the major errors, ...
- The bottom section displays the different events that occurred and that may cause problems in the application.

► In this example, the dynamic audit detects several problems:

- A SWITCH statement for which no CASE is run,
- A triggered assertion,
- A triggered exception,
- the assignment of an image file that does not exist to an Image control.

For each problem, a "... " button is used to access the details of the event. If the event is linked to a specific code line, the button  is used to open the code editor at the corresponding location in order to fix the problem.

► Close the window of the dynamic audit.



Notes

The dynamic audit of a project can also be performed when the site is deployed. All you have to do is call `dbgEnableAudit` to trigger the dynamic audit. The audit generates a ".waudit" file, this file must be loaded in the development environment to study the result. See the online help for more details (keyword: "Dynamic audit").

LESSON 10.3. PERFORMANCE PROFILER

This lesson will teach you the following concepts ...

- Overview
- Starting the performance profiler
- Studying the result



Estimated time : 15 mn

Overview

The performance profiler (also called Profiler) is a tool used to check and optimize the execution time of processes found in your site.

The principle is straightforward:

- You run the test of your site.
- During this test, the performance profiler keeps track of all the actions performed and saves the amount of time it took each process to run.

At the end of the test, the performance profiler displays:

- the 10 most time consuming operations,
- the duration and the number of calls of all the processes run.

The "WW_Optimization" project contains a specific page used to view the interesting results with the performance profiler.

Starting the performance profiler

The performance profiler can be started:

• from the WEBDEV editor:

In this case, the project is automatically run in test mode. You can use your application and start the processes of your choice.

To go back to the WINDEV editor, all you have to do is exit from your site.

Then, the performance profiler displays the result of study. This result is saved in a WPF file.

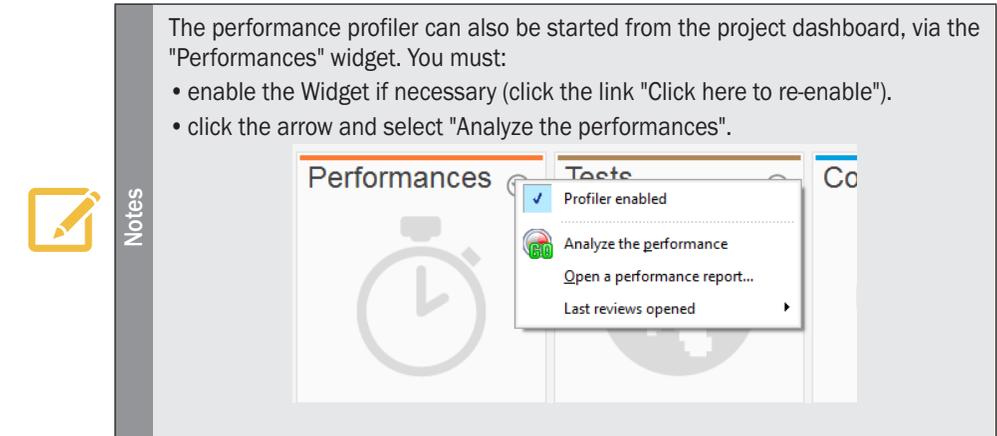
• from one of your processes in WLanguage, via the following functions:

ProfilerStart	Starts "collecting data" for the performance profiler.
ProfilerEnd	Stops "collecting data" for the performance profiler.

In this case, only the code found between **ProfilerStart** and **ProfilerEnd** is analyzed. The result is saved in a WPF file.

► The first method will be used in our example. To start the performance profiler on the "WW_Optimization" project:

1. In the ribbon, on the "Project" pane, in the "Audit and performances" group, expand "Analyze the performance" and select "Analyze the performance".



2. The project test is run.
3. Click the "Test page of performance profiler" button.
4. Click the "Process to analyze" button.
5. Validate the information window and stop the project test. The report window of performance profiler is displayed.

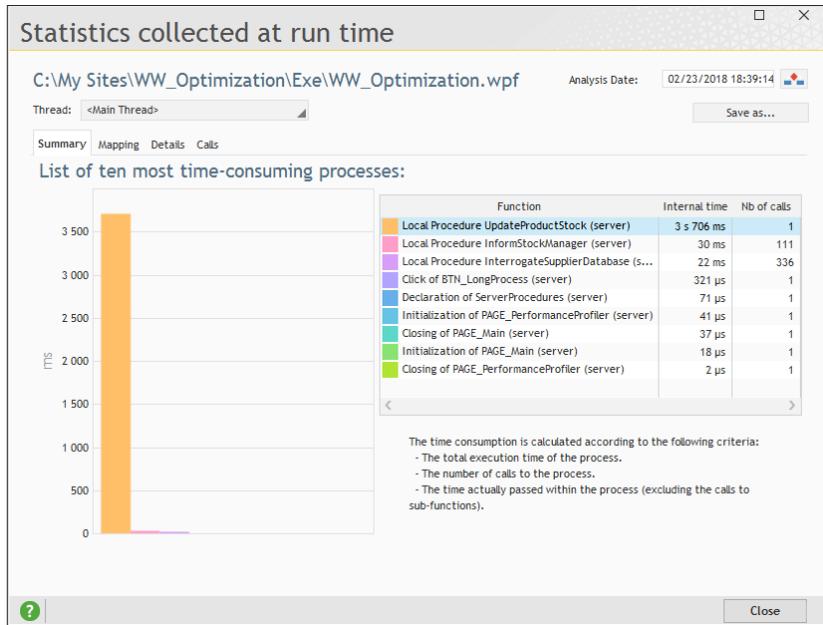
Studying the result

► Let's study the report window of performance profiler. The results are displayed in several tabs:

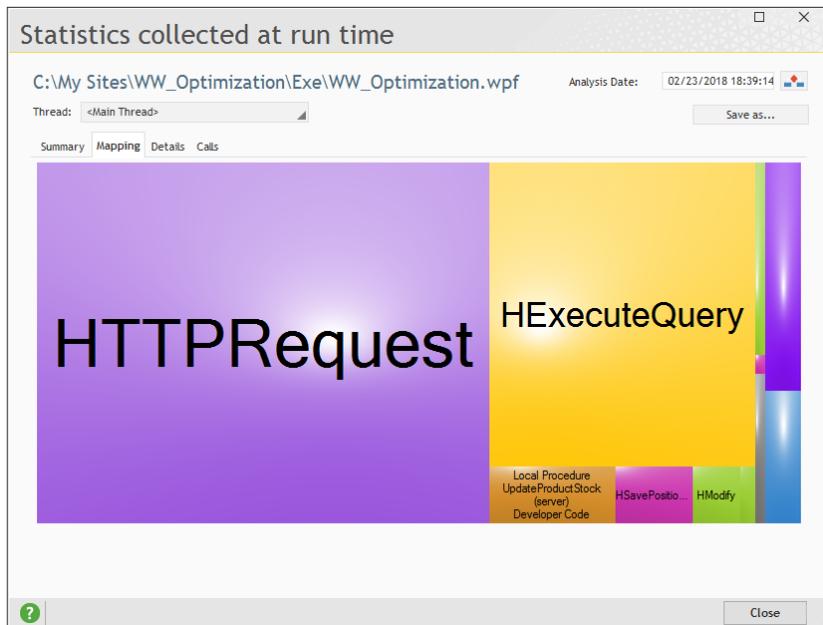
- the "Summary" tab presents the ten longest processes.
- the "Mapping" tab presents a graphical view of the main processes.
- the "Details" tab presents all the processes run during the test of the application (from the slowest one to the fastest one).
- the "Calls" tab is used to view the details of the operations performed in a process.

► Let's present these different tabs in our example.

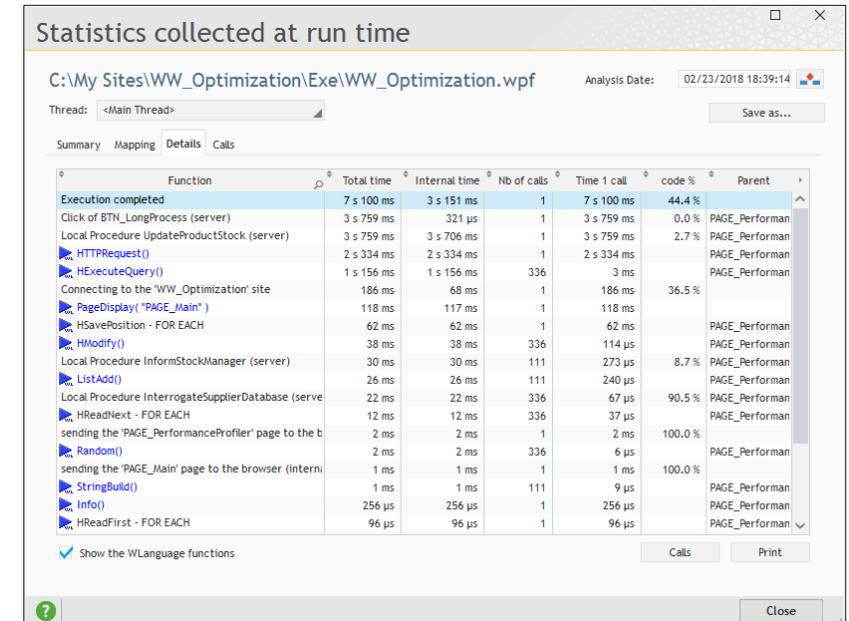
- The "Summary" tab presents the ten longest processes. In our example, you can see that the local procedure named "UpdateProductStock" takes more than 3 seconds to run (this time may change according to the power of your computer).



- The "Mapping" tab is used to visually identify what took the longest time. In our case, it is a call to **HTTPRequest**:



- The "Details" tab presents all the processes run, from the slowest one to the fastest one.



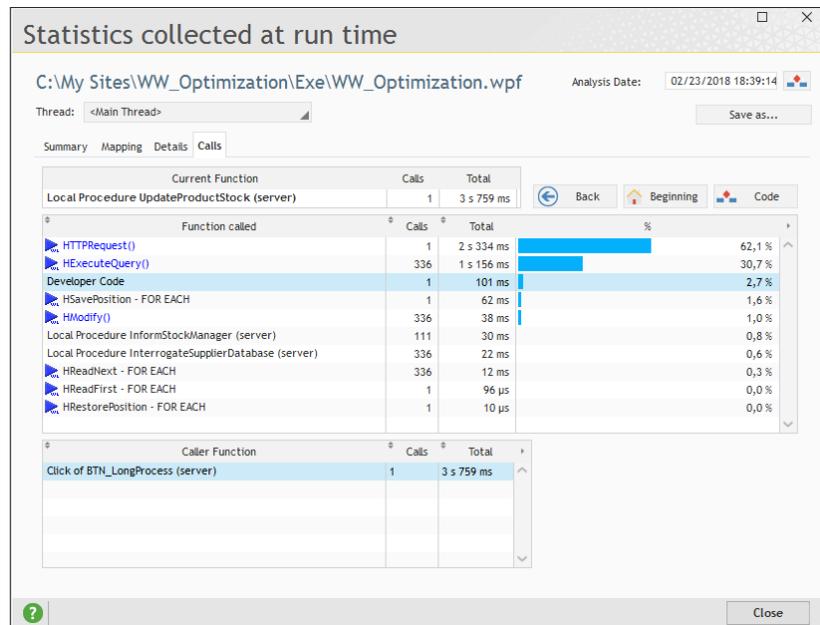
The following information is displayed for each process:

- Function: Function, process or procedure run.
- Total Time: Execution time of function.
- Internal time: Execution time due to the engine.
- Nb of calls: Number of calls made to the function (procedure or process).
- Time 1 call: Execution time of a call to the function (procedure or process).
- % Code: Percentage of time spent in the process of the function or procedure (developer code that can be optimized).
- Parent: Element that contains the process.

In our case, the "Details" tab indicates that the call to **HTTPRequest** is one of the elements taking the longest time.

Select this line. We are going to check in the code whether this slowdown is caused by a specific problem.

- Click the "Calls" button to display the details of the calls to the **UpdateProductStock** procedure. Select the "HTTPRequest" line and click the "Code" button: the corresponding code line is displayed in the code editor.



- Close the performance profiler.
- The following code line is run:

```
HttpRequest ("supplier-addr")
```

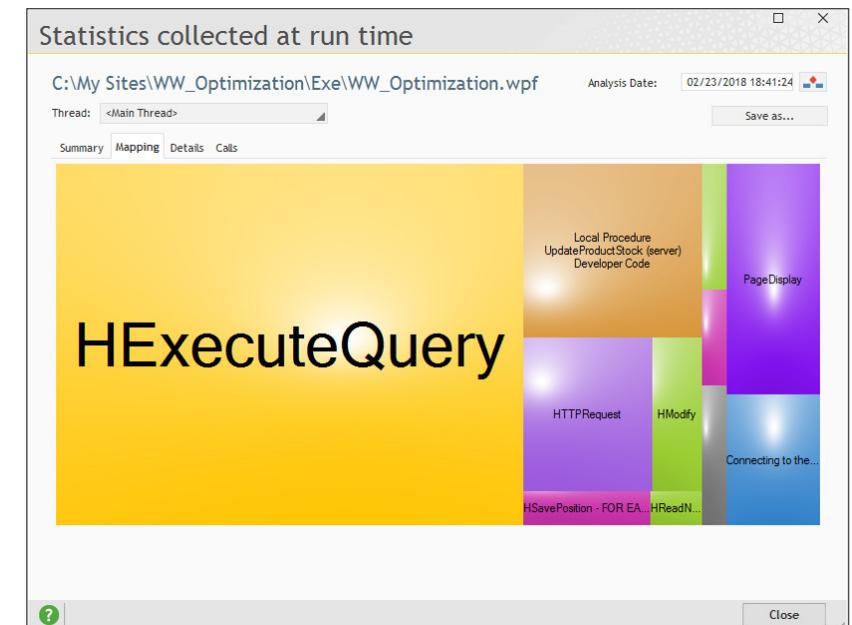
The slowdown is caused by the fact that the address specified for *HttpRequest* is not accessible.

- Let's check the operating mode of application by optimizing this code:
 1. Replace the code line containing *HttpRequest* by the following code line:

```
// Checks the accessibility of supplier server
HttpRequest ("www.google.com")
```

2. Save the code (CTRL S).

- We are now going to restart the performance profiler:
 1. On the "Project" pane, in the "Audit and performances" group, expand "Analyze the performance" and select "Analyze the performance".
 2. The project test is run.
 3. Click the "Test page of performance profiler" button.
 4. Click the "Process to analyze" button.
 5. Validate the information window and stop the project test. The report window of performance profiler is displayed.
 6. In the "Mapping" tab, *HttpRequest* does not appear with the same importance.



- Close the report window of the performance profiler.

LESSON 10.4. DEBUGGING A PROJECT

This lesson will teach you the following concepts ...

- Overview
- Using the debugger



Estimated time : 15 mn

Overview

Let's take a look at the debugger that is supplied with WEBDEV.

What is the debugger?

The debugger is a powerful tool used to follow the progress of a code or application, step by step. Enhancing a process or even an application is child's play.

We are going to use the debugger on the long process found in the PAGE_PerformanceProfiler page.

Using the debugger

► To debug the PAGE_PerformanceProfiler page:

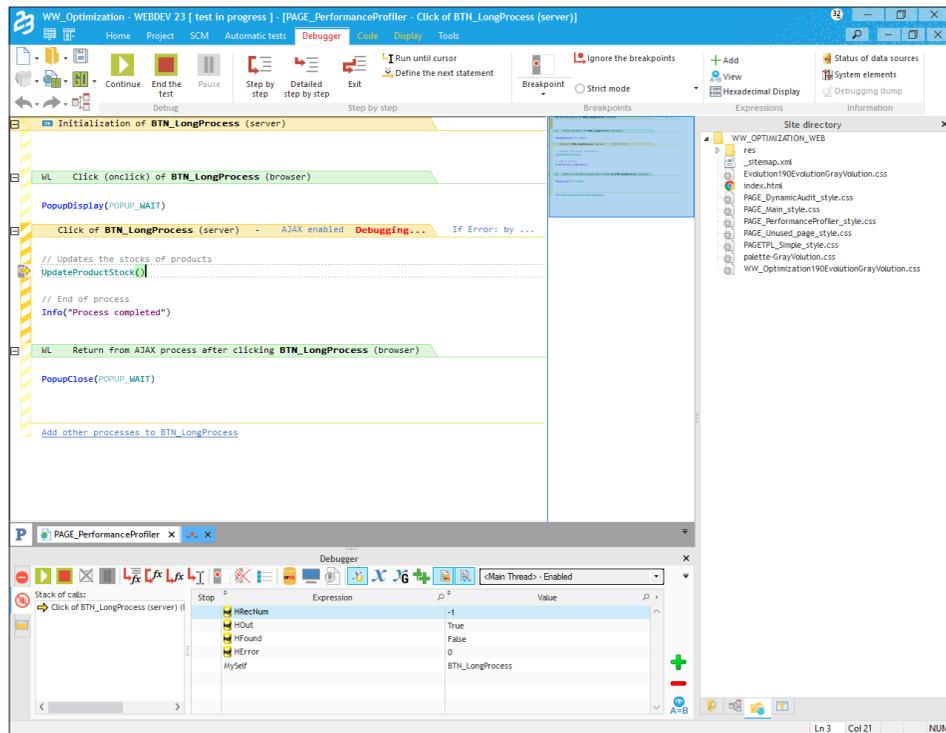
1. Open the "PAGE_PerformanceProfiler" page in the page editor (double-click its name in the project explorer).
2. In the WEBDEV editor, position a breakpoint in the code of "Process to analyze" button: click in front of the first code line of server click process or press CTRL B, a red dot appears. The debugger will be automatically started when the line preceded by the breakpoint is run.
3. Run the test of "PAGE_PerformanceProfiler" page ( among the quick access buttons).



Note

Several methods can be used to start the debugger. See the online help for more details (keyword: "Debugger, Run a test").

4. Click the "Process to analyze" button. The debugger starts: the code editor is displayed in "Debugger" mode. The current line is preceded by a little arrow.



The "Debugger" pane appears in the lower section of the screen. This pane displays two distinct areas:

- the call stack: this area is used to find out the hierarchy of processes displayed in the debugger. In our example, we are currently debugging the click process on the BTN_LongProcess button.
- the list of expressions to evaluate. By default, the main variables used in the code are displayed in this section. You have the ability to add variables to follow their evolution (this feature will be presented later).

We are going to perform some modifications in the debugger to check its capabilities.

- First of all, we are going to run the different statements step by step and see the content of variables:
 1. Press the [F7] key (or press the "Detailed step by step" button). Note: The values of variables are modified (if necessary) in the "Debugger" pane (displayed at the bottom of the screen).
 2. Keep pressing the [F7] key until you reach the line "Product.Stock = InterrogateSupplierDatabase ...", hover "Product.Stock" with the mouse cursor. A tooltip is displayed with the expression value:

```
// Browses the products
FOR EACH Product
    // Interrogates the supplier database
    Product.Stock = InterrogateSupplierDatabase(Product.Reference)
    HModify(Product.Stock = 20)
```

3. The value of "Product.Stock" is displayed in a tooltip. This value corresponds to 20 because the code line was not run.
4. Use the [F8] key to run the line.
5. Hover "Product.Stock" again. The value of "Product.Stock" displayed in the tooltip corresponds to the result of **InterrogateSupplierDatabase** procedure.

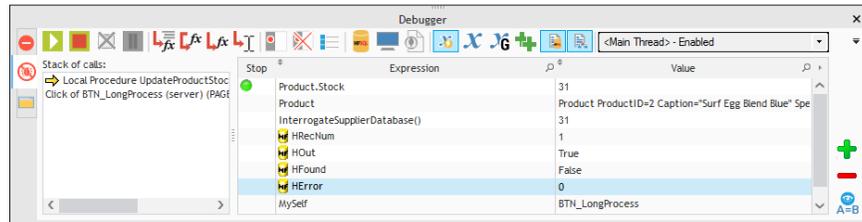
```
// Browses the products
FOR EACH Product
    // Interrogates the supplier database
    Product.Stock = InterrogateSupplierDatabase(Product.Reference)
    HModify(Product.Stock = 50)
```

- We are now going to add an expression to monitor the evolution of its value in the "Debugger" pane. This expression can have any type: variable, function, operation on variables, ... The result of the expression is calculated and displayed.

This expression is used to perform a custom debugging. For example, you can find out the content of a variable as it is being used in the application.

1. Select "Product.Stock" in the code and display the popup menu (right mouse click). Select "Add the expression into the debugger".

2. The expression is automatically added into the debugger pane at the bottom of the screen.



► The debugger can also be used to run a set of code lines:

1. Position the cursor on the following line:

```
IF HExecuteQuery(QRY_QuantityOrdered) THEN
```

2. Press the [F6] key (or click the "Run until cursor" button found in the ribbon).
3. The arrow indicating the line currently run moves until it reaches the code line where the cursor is positioned. The code lines found before the cursor are automatically run.

► We are now going to add a breakpoint and to run the code until it reaches the breakpoint:

1. Click in the hatched area with the mouse, in front of **HModify**. A breakpoint (red bullet) appears.

```
// Browses the products
FOR EACH Product
    // Interrogates the supplier database
    Product.Stock = InterrogateSupplierDatabase(Product.Reference)
    HModify(Product)

    // Checks whether the product is very popular
    QRY_QuantityOrdered.pRéférence = Product.Reference
    IF HExecuteQuery(QRY_QuantityOrdered) THEN

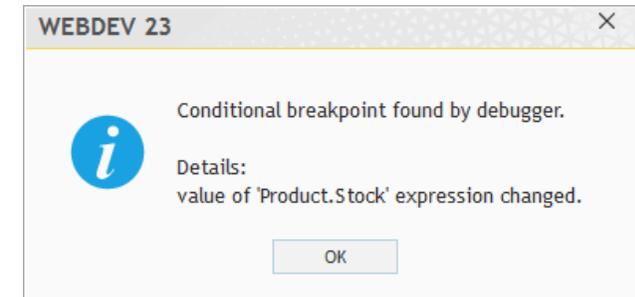
        // If the total ordered is greater than 92 AND if the stock is less than 50
        IF QRY_QuantityOrdered.TotalQuantity > 92 AND Product.Stock <= 50 THEN

            // Displays an information
```

2. Press the [F5] key (or click the "Continue" button found in the ribbon). The code is run until it reaches the breakpoint. The arrow used to identify the current line moves until it reaches the breakpoint.
3. Click the breakpoint to remove it.

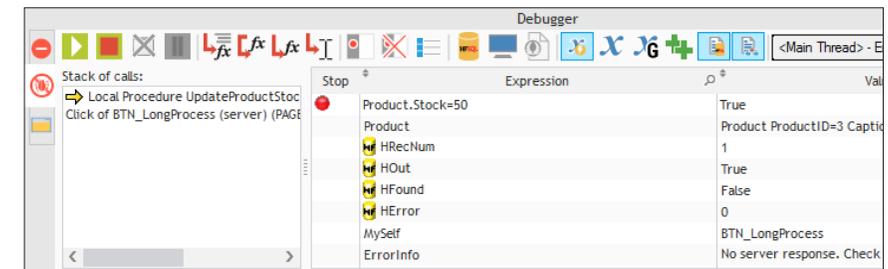
► To end this test, we will be using an "Auto-stop" expression. An "Auto-stop" expression is used to start the debugger as soon as a condition is checked or when the value of a variable is modified. In our example, the debugger will be started as soon as the stock value is equal to 50:

1. In the "Debugger" pane, select the "Product.Stock" expression that was added beforehand.
2. Click the green circle.
3. Press [F5] to continue the test.
4. A message is displayed, indicating that the value of "Product.Stock" expression changed.



5. Validate.

6. In the "Debugger" pane, select the "Product.Stock" expression. Click the expression again: the "Expression" column becomes editable. In the "Expression" area, add "=50". You will get "Product.Stock = 50".



7. Press the [F5] key. The program continues to run. The debugger is started again when the value of Product.Stock variable is equal to 50.

► That's it, the main features of debugger have been presented here. To stop the test in the debugger, click "End the test" found in the ribbon.

PART 11

**Managing
a HFSQL Client/
Server database**

LESSON 11.1. INTRODUCTION

This lesson will teach you the following concepts ...

- Principle of Client/Server
- Why switch an application to HFSQL Client/Server?



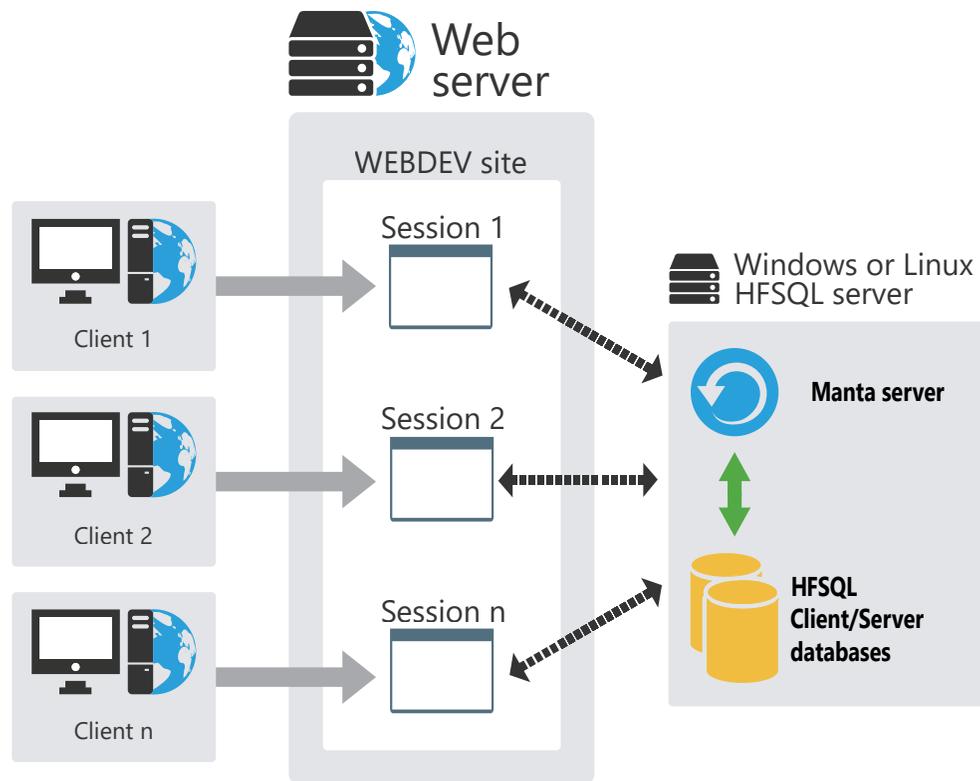
Estimated time : 5 mn

Overview

WEBDEV allows you to create applications/sites that access HFSQL Client/Server databases. A database in Client/Server mode is used to host the databases on a server (this server can differ from the one where the site is deployed).

This operating mode:

- increases your data's security,
- lets you easily manage the multi-user mode,
- simplifies the maintenance,
- allows the connections from the outside (headquarters by back office for example).



WEBDEV allows you to:

- create a site that is using a HFSQL Client/Server database.
- modify a site that is using an HFSQL Classic database so that it is using an HFSQL Client/Server database.

Why switch a site to HFSQL Client/Server mode?

The main benefits of a site in HFSQL Client/Server mode compared to a site in HFSQL Classic mode are as follows:

- The use of HFSQL Client/Server is more secured (use of a login and password and definition of rights granted to the users).
- No management of directories: all the database files are grouped at the same location.
- The databases in Client/Server mode can be used by an Internet connection.
- Management of native multi-user mode: the performances are optimized in multi-user mode.

LESSON 11.2. IMPLEMENTATION OF A CLIENT/SERVER DATABASE

This lesson will teach you the following concepts ...

- Installing a local HFSQL server
- Creating a site that is using a HFSQL Client/Server database
- Adapting a site to manage a HFSQL Client/Server database
- Features available in Client/Server mode



Estimated time : 10 mn

Overview

In this lesson, we are going to perform all the operations required to develop and to deploy a site that is using a HFSQL Client/Server database.

Installing a local HFSQL server

The first operation to perform before developing consists in installing a HFSQL server.

This server can be installed locally on the development computer (that's what we are going to do). In deployment, this server can be installed on a specific computer or on the server that is hosting your site.

The setup program of the HFSQL server is available on the WEBDEV DVD. If you do not own this DVD, the setup of HFSQL server is also available from the PC SOFT site (www.windev.com).

To install the HFSQL server locally:

1. Start the setup program of WEBDEV.
2. Choose "HFSQL Client/Server Setup".
3. Then, select "Install or update a HFSQL Client/Server server".
4. Accept the license agreement.
5. Choose the platform ("For Windows on this computer").
6. If HFSQL servers are already installed on the current computer, select "Install a new server".
7. Select the setup directory and specify the server name and port. The port 4900 will be used by default.



Notes

Remember to open this port on the firewall to connect to the HFSQL server from another computer.

8. Install the HFSQL Control Center if it is not already present or accessible from your computer.



Caution!

The HFSQL Control Center is required to manage the HFSQL Client/Server database.

9. The wizard proposes to configure the sending of notifications in order to identify the server dysfunctions in real time. In our example, go to the next step and indicate that the setting will be performed "Later".

10. The wizard proposes to configure the authentication via Active Directory. In our example, go to the next step directly.

11. Validate (or not) the participation in product improvement by allowing us to collect information regarding the use of product. This optional and anonymous collect allows PC SOFT to improve the product features.

12. The setup is performed. By default, to connect to the server in administrator mode, use the "admin" user without password.



Notes

For security reasons, don't forget to change the administrator password.

Creating a site that is using a HFSQL Client/Server database

Creating a WEBDEV site that is using a HFSQL Client/Server database is child's play. You must:

1. Create the project by requesting to create a new database.
2. Create the analysis by specifying that the databases used by the project will be "HFSQL Client/Server" databases.
3. Specify the characteristics of connection to the HFSQL Client/Server server that will be used.
4. When creating a file in the analysis, indicate that this file is in Client/Server mode and specify the connection used.



Notes

You can also describe the connection to the HFSQL server by programming. See the online help for more details: "HDescribeConnection".

Adapting a site to use a HFSQL Client/Server database

Overview

Switching a database from the HFSQL Classic mode to the Client/Server mode is the most common operation.

WEBDEV propose several solutions to perform this adaptation:

- perform the switch in the data model editor.
- perform the switch in the HFSQL Control Center.

To better understand the different steps, we are going to switch the site that was created in part 4 of this tutorial to Client/Server mode by using the first method, the data model editor.

Adapting the example



Answer

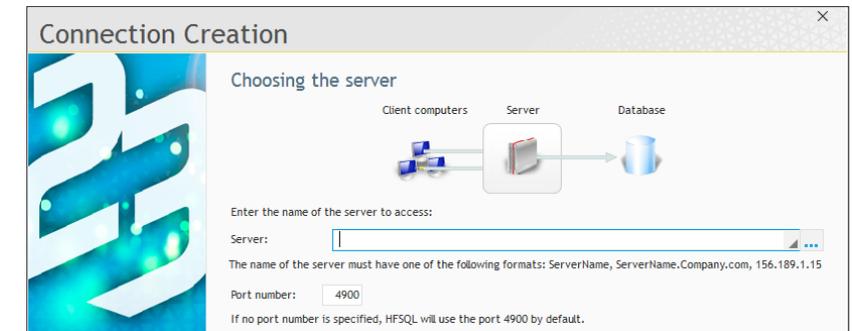
If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (Answer)"

► To adapt the project:

1. Open the "Full_WEBDEV_Site" project if necessary.
2. Load the analysis of your project in the data model editor: click  among the quick access buttons of the WEBDEV menu. The data model editor is displayed.
3. In the ribbon, on the "Analysis" pane, in the "Connection" group, click "New connection". A wizard is opened, allowing you to create a connection.
4. Select the type of connection to create: "HFSQL Client/Server". Go to the next step.

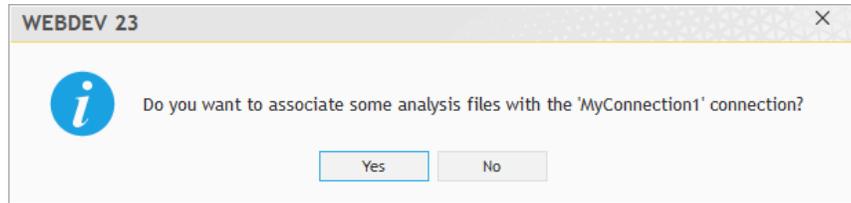


5. In the following steps, specify:

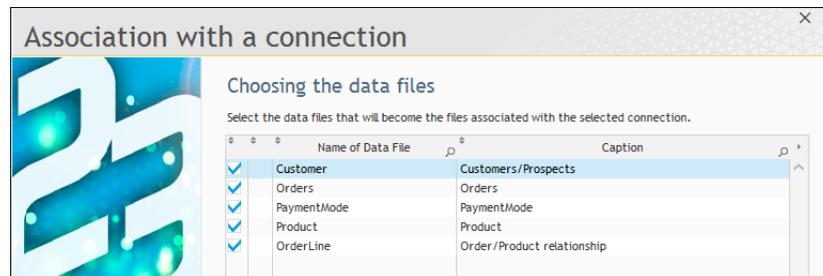


- the name of the server on which the HFSQL Client/Server server was installed (localhost if it was installed on your computer for example) and the port number. Go to the next screen.
 - the user name and password (leave this information empty to use the administrator). Go to the next screen.
 - the database name ("Full_WEBDEV_Site" in our example). Go to the next screen.
6. Type the connection name (keep the proposed name).

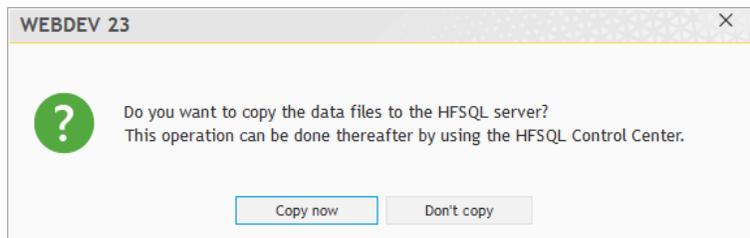
7. Go to the next step and validate. The connection to the database is automatically created. The wizard proposes to associate the different data files found in the analysis with the connection that was just created.



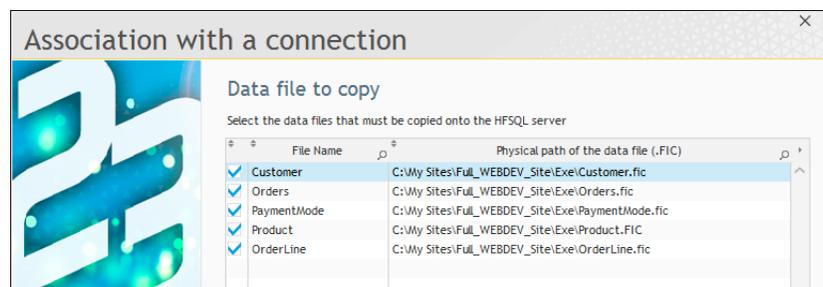
8. Click "Yes".



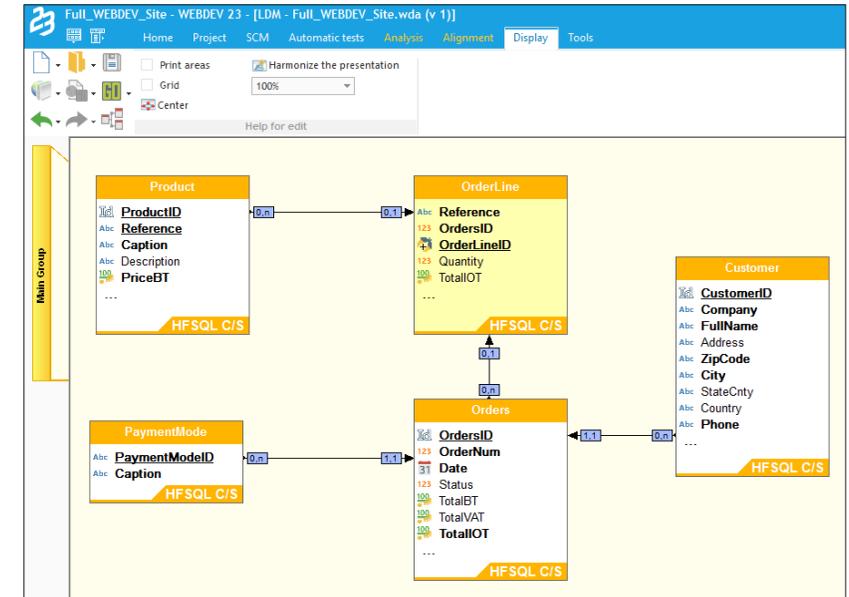
9. In the next screen, select all the proposed files. Go to the next step.
 10. Then, the wizard proposes to copy the data files onto the server.



11. Validate ("Copy now").
 12. Select the analysis files that will be copied onto the server: in our case, all the data files found in the EXE directory.



13. Go to the next step and validate.
 14. The data files of the analysis are automatically transformed into HFSQL Client/Server data files and associated with the selected connection.



15. Generate the analysis: on the "Analysis" pane, in the "Analysis" group, click "Generation". An automatic modification of data files is automatically performed. You have the ability to cancel the automatic modification of data files if all the data files are updated.



Notes

Switching to Client/Server mode: some tips

- Check the code of your project : in HFSQL Client/Server, functions such as *HSubstDir*, ... are useless.
- According to the parameters specified when creating the connection, the connection defined in the analysis can be modified by *HOpenConnection* and *HChangeConnection*.
- *HOpenConnection* can be used to go back to HFSQL Classic mode: all you have to do is specify the path of directory containing the HFSQL Classic data files.

16. The development project was successfully adapted. You may also have to adapt the deployed site (if the deployed site is using HFSQL Classic files for example). This operation is configured when creating the setup program of the site.

Features available in HFSQL Client/Server mode

HFSQL Client/Server proposes several features:

- Transactions,
- Logs,
- Stored procedures,
- Triggers,
- Hot automatic data modification,
- Hot reindexing,
- Scheduled backups,
- Incremental backups,
- Universal replication.

These features will not be described here (some of them have been presented in this tutorial in HFSQL Classic mode). See the online help for more details.

LESSON 11.3. MANAGING A CLIENT/ SERVER DATABASE

This lesson will teach you the following concepts ...

- The HFSQL Control Center
- Creating a user in the HFSQL Control Center
- Saving the database



Estimated time : 20 mn

Overview

Now that we know how to create and/or adapt a site so that it operates in HFSQL Client/Server mode, let's see how to manage the associated database.

Indeed, a Client/Server database requires:

- a specific configuration of computers (setup of HFSQL server, ...).
- a management performed via the HFSQL Control Center.

Configuring the computers

To use a HFSQL Client/Server database, a HFSQL server must be installed on the server. Several HFSQL servers that use different ports can be installed on the same computer. However, for performance reasons, this configuration is not recommended. One or more databases can be installed on each server.

For example, a test HFSQL server that includes a test database and a production HFSQL server that is using a different port can be installed on the same computer.

The HFSQL Control Center

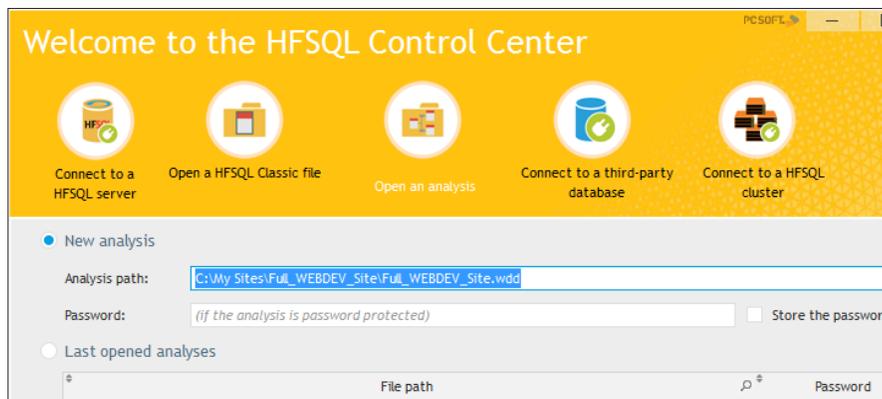
The HFSQL Control Center is used to perform all the management operations on the HFSQL Client/Server servers and databases.

We are going to present the most important features.

First, we are going to start the HFSQL Control Center from the WEBDEV project.

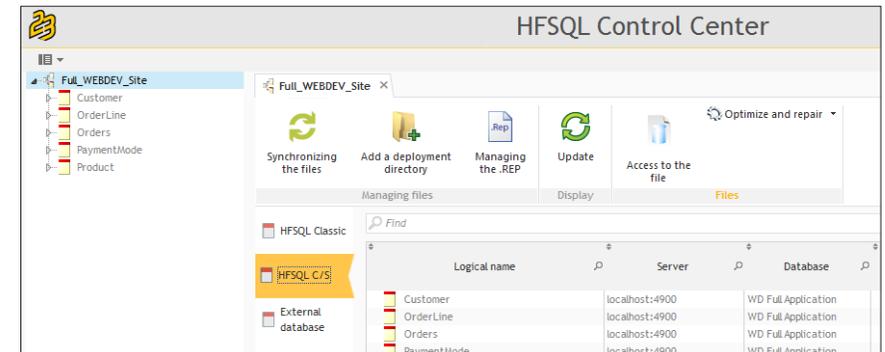
► To start the HFSQL Control Center from WEBDEV and to access the data:

1. In the WEBDEV menu, on the "Tools" pane, in the "Database" group, click "HFSQL". The HFSQL Control Center is displayed.
2. The home window is displayed in the HFSQL Control Center. The analysis of the current project is automatically selected.



3. Validate the screen. The HFSQL Control Center is displayed. This type operating mode lets you see the various files linked to the analysis of the current project.

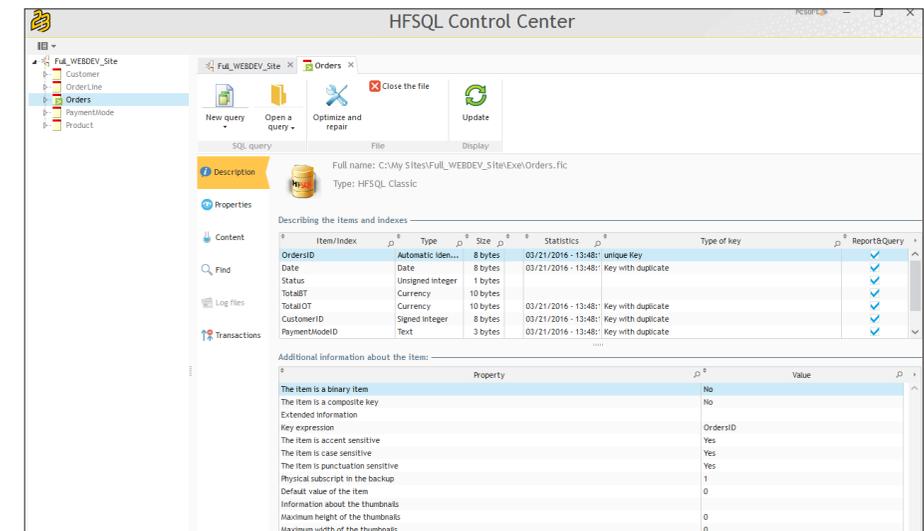
4. Click the vertical tab named "HFSQL C/S" if necessary. The list of data files in HFSQL Client/Server format is displayed.



The Control Center lists the Client/Server data files found in the analysis linked to the current project. No connection is established: the files are grayed.

5. To see the data found in the files, double-click one of the data files in the list on the left ("Orders" for example). If the HFSQL Control Center does not recognize all the connection parameters, a connection window is used to perform the effective connection to the HFSQL Client/Server server used. If this window is displayed, specify the password and validate.

6. The information about the selected data file that is using this connection is displayed in a new tab:



- The "Description" tab gives information about the data files (file items, ...).
- The "Content" tab displays the records found in the data files.

The entire HFSQL Client/Server database can be managed from the HFSQL Control Center.

Creating a user account in the HFSQL Control Center

A single user account is created when installing a HFSQL server and when creating a database: the administrator account ("Admin" login without password).

Using a user account allows you to secure the access to data. Indeed, all the users of the site are not administrators. Specific rights can be granted to each user (or group of users).

Caution!

The user rights specified in the HFSQL Control Center are granted for the database and not for accessing the site.

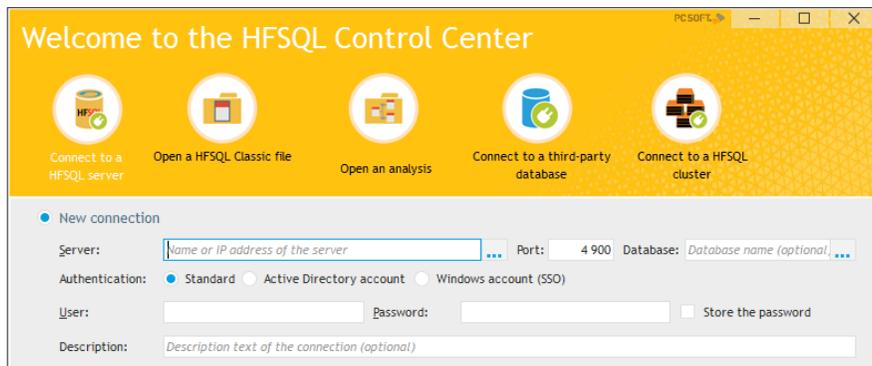
Do not confuse right management for the Client/Server databases with the user groupware, which we've seen in a previous lesson.

Some users may not have the rights to write into some data files for example.

To run a simple test, we are going to create a user and allow this user to see the records found in Customer file.

▶ To directly connect to the database found on the server:

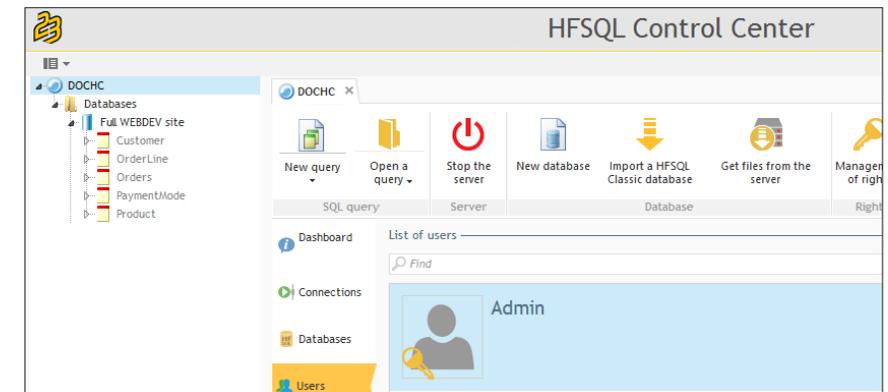
1. Expand the menu found at the top left of the HFSQL Control Center and select "Connect to a HFSQL server".
2. The home window of HFSQL Control Center is displayed.



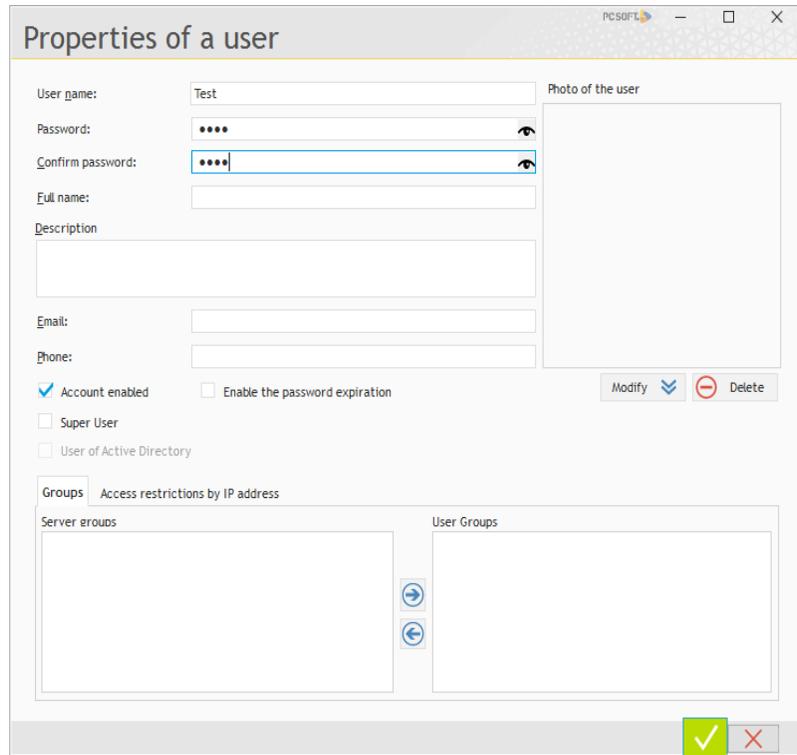
3. The option "Connect to a HFSQL server" is selected by default. Specify the characteristics of the server installed in the previous lesson.
4. The characteristics of the HFSQL server are displayed in the different panels:
 - the name of HFSQL server as well as the list of databases found on this server are displayed in the left panel.
 - in the right section of the screen, a new tab allows you to see the characteristics of HFSQL server.



5. In the right section of the screen, select the "Users" tab. This tab is used to manage the users of the server.
6. Only the "Admin" user exists at this time.



7. To create a new user, in the ribbon, in the "Users" group, click the "New" button. The screen used to define the characteristics of the user is displayed.



8. Type the following information: (use "Test" as password for example).



Notes

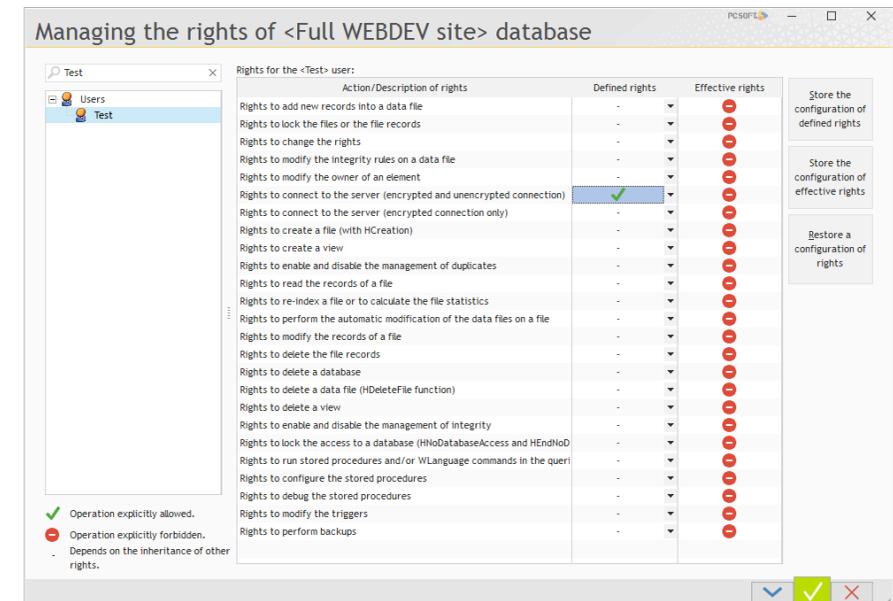
Several characteristics can be noticed:

- Super User: The users defined as "Superuser" are allowed to perform all types of actions on the server, the databases and the files.
- Account enabled: If this option is not checked, the user exists but he is not enabled (users on holiday for example).
- Password expiration: You have the ability to specify a password valid for a limited number of days (configurable).

9. Validate the user creation. By default, no rights are granted to this user.

We are now going to grant rights to the user: the "Test" user can connect to the database and he can read the Customer file.

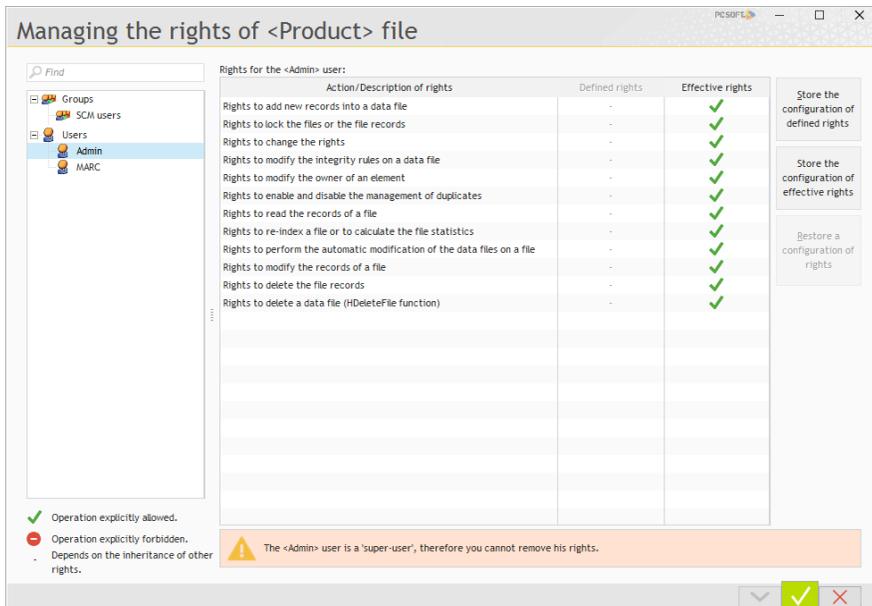
- To grant the rights to connect to the database:
 1. In the HFSQL Control Center, double-click the "Full_WEBDEV_Site" database. A new tab appears.
 2. On the "Full_WEBDEV_Site" pane, in the "Rights" group, click "Management of rights".
 3. Select the "Test" user in the list of users.
 4. In the list of rights, for the action named "Rights to connect to the server (encrypted and unencrypted connection)", click in the "Defined rights" column and select the green checkmark.



5. Click the "Apply" button found at the bottom of window (blue arrow). The rights become effective.

6. Close the window for managing rights.

- ▶ To grant rights to read the Customer file:
 1. In the HFSQL Control Center, expand the "Full_WEBDEV_Site" database and double-click the Customer file (on the left of the screen).
 2. On the "Client" pane, in the "Rights" group, click "Manage the rights".
 3. Select the "Test" user in the list of users.
 4. In the list of rights, for the action named "Rights to read the file records", click the "Defined rights" column and select the green checkmark.



5. Click the "Apply" button found at the bottom of window (blue arrow). The rights become effective.
6. Close the window for managing rights.

Similarly, the rights can be defined:

- on the HFSQL server,
- on the database,
- on the database files.

In our example, the "Test" user will have the ability to browse the records found in Customer file. If this user tries to perform another action, a message will be displayed: "The Test user has no sufficient rights to XXXX" (where XXXX corresponds to the action performed).

Once the account is created, it can be used when the application connects to the server (when **HOpenConnection** is used).



Notes

The users and their rights can also be managed by programming with the WLanguage functions. See the online help for more details.

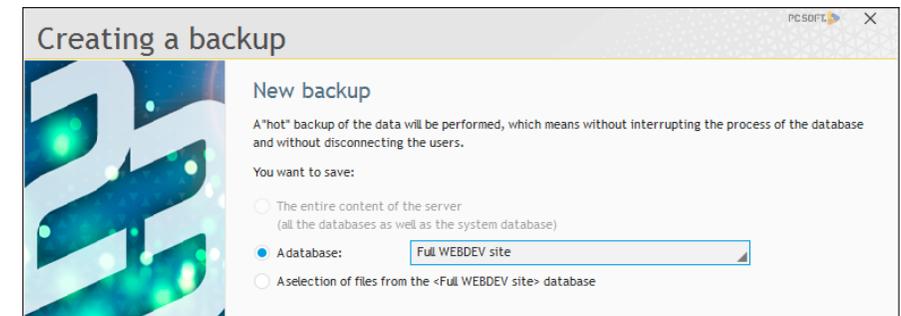
Saving the database

To save the database:

1. Position on the tab corresponding to the "Full_WEBDEV_Site" database.
2. Select the vertical tab named "Backups".
3. In the menu, in the "Backups" group, expand "New backup" and select "New hot backup".

Note: This button is accessible in the "Backups" group:

- on the tab corresponding to the HFSQL server,
- on the tab corresponding to the database.



Conclusion

The HFSQL Control Center is a tool for managing databases, allowing you to:

- stop or restart a server if a problem occurs,
- manage the users and their rights,
- reindex the data files if necessary,
- perform backups of the database.

The HFSQL Control Center is a redistributable tool that can be installed on the computers of the users who are working with HFSQL Client/Server databases. The HFSQL Control Center must be used by the database administrator.

PART 12

**Source Code
Manager**



LESSON 12.1. SCM

This lesson will teach you the following concepts ...

- Overview
- The Source Code Manager



Estimated time : 15 mn

Introduction

The development of a large IS system requires the participation of several developers. These developers must work on a single WEBDEV project while sharing the different resources (pages, classes, ...).

WEBDEV is supplied with a Source Code Manager named "SCM" used to share the source code of different projects between developers and to find out the full history of the modifications performed (in the code, in the interface, ...).

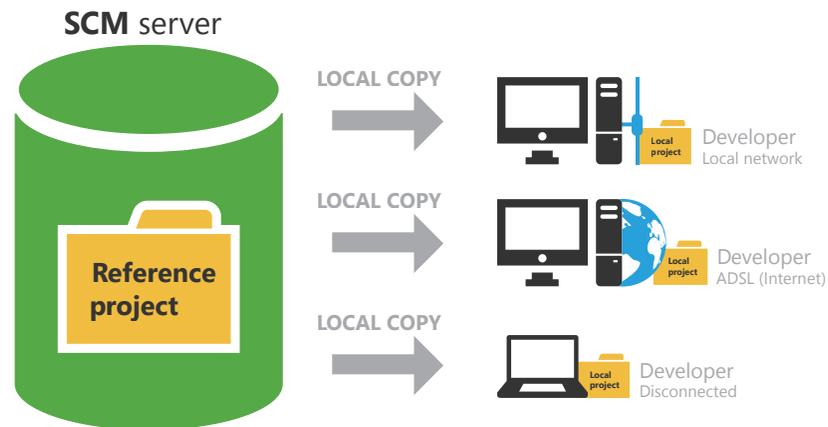
SCM (Source Code Manager)

Principle of SCM

The Source Code Manager is used to store and share the projects and their elements.

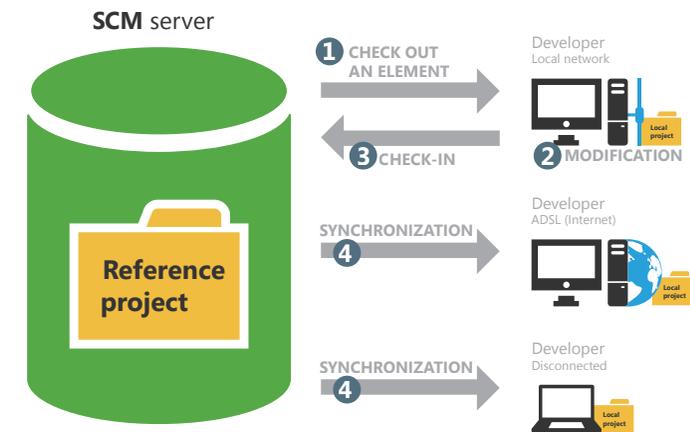
The principle is as follows:

- A reference version of each one of your projects is found on a server. All these versions are called "SCM database".
- Each developer has a local copy of the different projects on his computer.



- Whenever a developer wants to modify a project element (page, report, query, ...), he notifies the SCM that he is becoming the temporary owner of this element. To do so, this element will be checked out from the SCM database by the developer.
- This developer gets exclusive rights on this element: all the requested modifications can be performed on this element.
- The other developers are still working on the copy of the reference version of this element (found in the SCM database).
- Once the modifications have been made by the developer, the checked-out element is checked back into the SCM database.

- The other developers are automatically notified of this check-in operation. They can now update their local copy.



The SCM supports teamwork and it allows you to find out the history of all the modifications. The SCM can also be used to manage and control the elements shared between several projects.

Creating the SCM database

To share a project via the Source Code Manager, an SCM database must be created. This SCM database must be created once only on a server.

This SCM database can be created:

- when installing WEBDEV.
- when creating a project that is using SCM.
- when importing a project into SCM.
- whenever you want, from WEBDEV directly or from the SCM administrator.

The SCM database can be installed in the following modes:

- HFSQL Classic,
- HFSQL Client/Server,
- Cloud. The Cloud mode allows you to access the sources of projects from anywhere and at any time. Check the www.pcscloud.net site for more details

- ▶ In the next lesson, our SCM database will be created when importing a project into the SCM.



Notes

We advise you to perform backups of the SCM database on a regular basis. To do so, you must:

- connect as administrator to the management tool of SCM.
- on the "Management" pane, in the "Backups" group, click "Full backup of database".

LESSON 12.2. INCLUDING A PROJECT IN SCM

This lesson will teach you the following concepts ...

- Adding a project into SCM.
- Opening a project from SCM.
- Configuring SCM.



Estimated time : 15 mn

Including a project in SCM

Adding the project into SCM

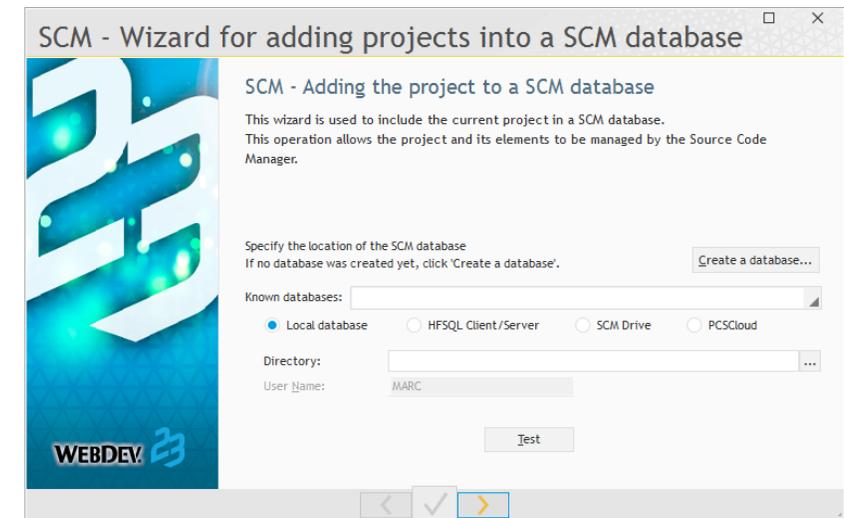
To use an existing project with the Source Code Manager, all you have to do is include this project in the SCM database.



Answer

If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WEBDEV site (Answer)".

- We are now going to include the "Full_WEBDEV_Site" project in the SCM database:
1. In the ribbon, on the "SCM" pane, in the "Project" group, click "Add the project". The wizard for adding projects into SCM starts:



The SCM database was not created yet.

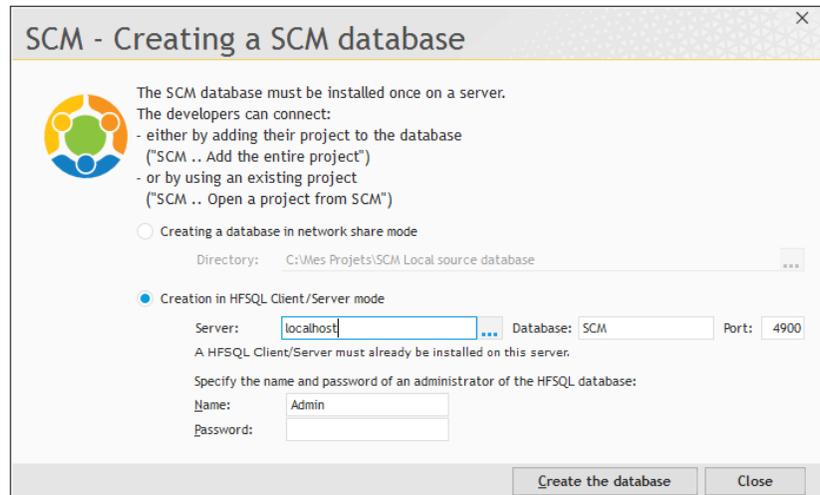


Notes

We are going to create a "Client/Server" SCM database on the server that was installed in part 10. If you did not follow this part, you must follow the part!

2. Click the "Create a database" button.

3. The screen used to create the database is displayed.



The SCM database can be in HFSQL Classic format (local or network) or in HFSQL Client/Server format. We are going to create a SCM database in HFSQL Client/Server format.

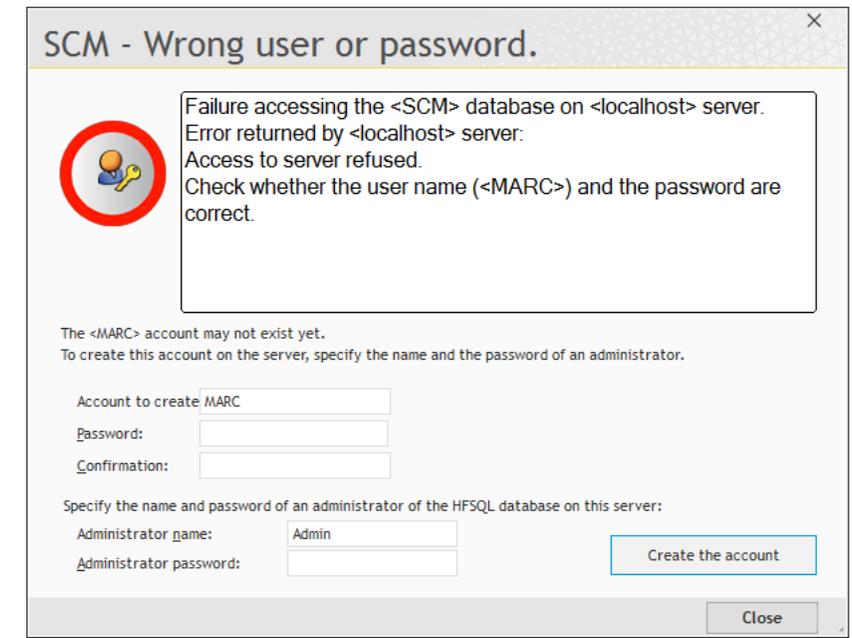


Notes

If the SCM database is in HFSQL Client/Server format, this SCM database can be used remotely.

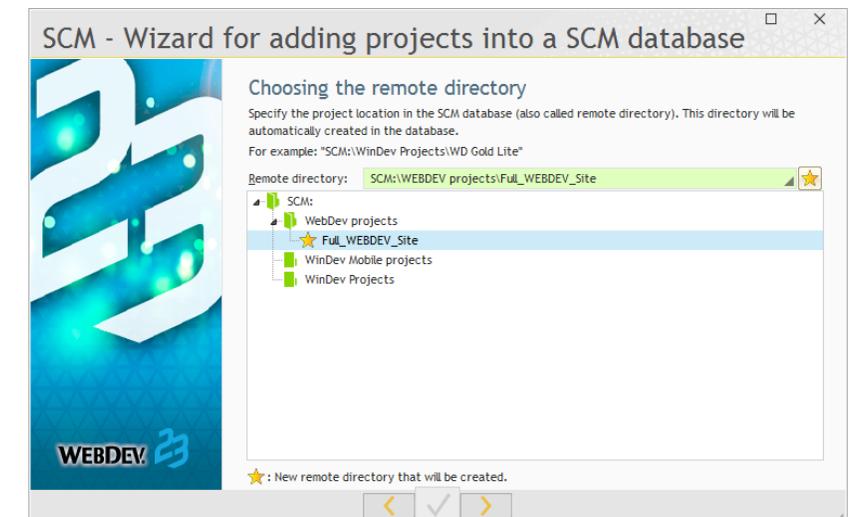
4. Select "Creation in HFSQL Client/Server mode".
5. Specify the parameters of the server that was installed beforehand:
 - The server name, its port.
 - The administrator name, his possible password.
6. Validate the creation of SCM database ("Create the database" button). This operation can take quite a long time in Client/Server mode.
7. The SCM database is now created. We are going to include our project in this SCM database.

8. Go to the next step. The current user is not saved in the database of users of HFSQL server.



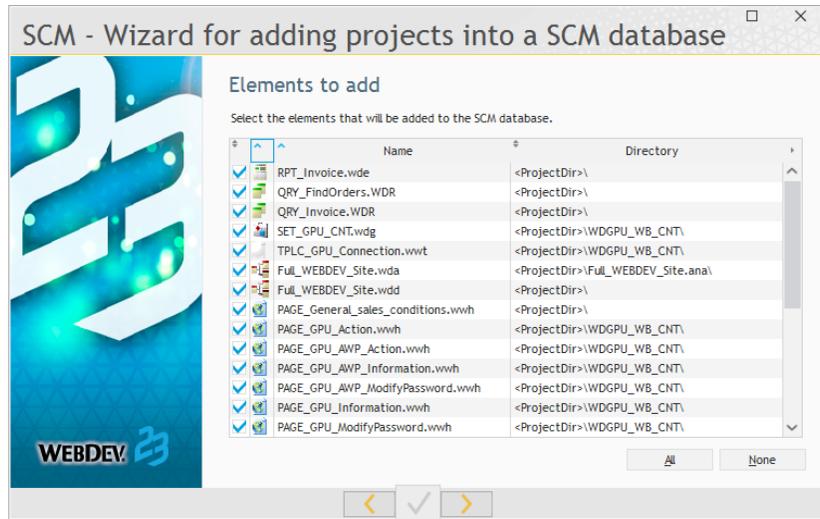
We are going to create this user: click the "Create the account" button.

9. The wizard proposes to place the project in the "WEBDEV projects" sub-directory of SCM database.



Accept this location. Go to the next step.

10. The wizard asks you to select the project elements that will be added into the SCM database.



We want to add all the project elements. Go to the next step.

11. The wizard asks you to select the project dependencies that will be added into the SCM database. We want to add all the project dependencies. Go to the next step.

12. Validate the inclusion of project in the SCM. The project and its elements have now been added into our SCM database.



Notes

Sharing project elements

When the projects that share the same resources (same analysis, same windows, ...) are included in the SCM, the relevant elements can be shared between the different projects. Therefore, the same element is checked in once only into the SCM and the modifications are automatically reported in the other projects.

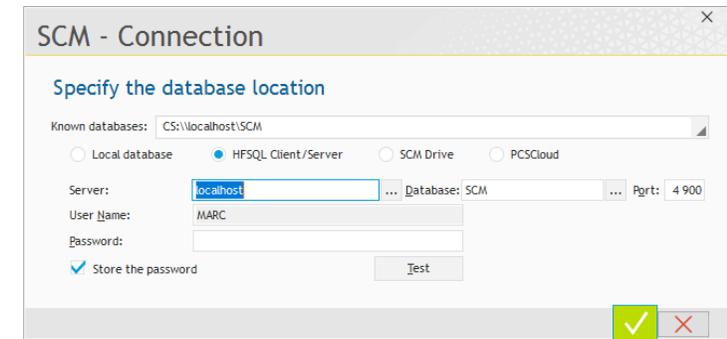
Opening a project from SCM

In our example, the project is included in the SCM and it can be used directly.

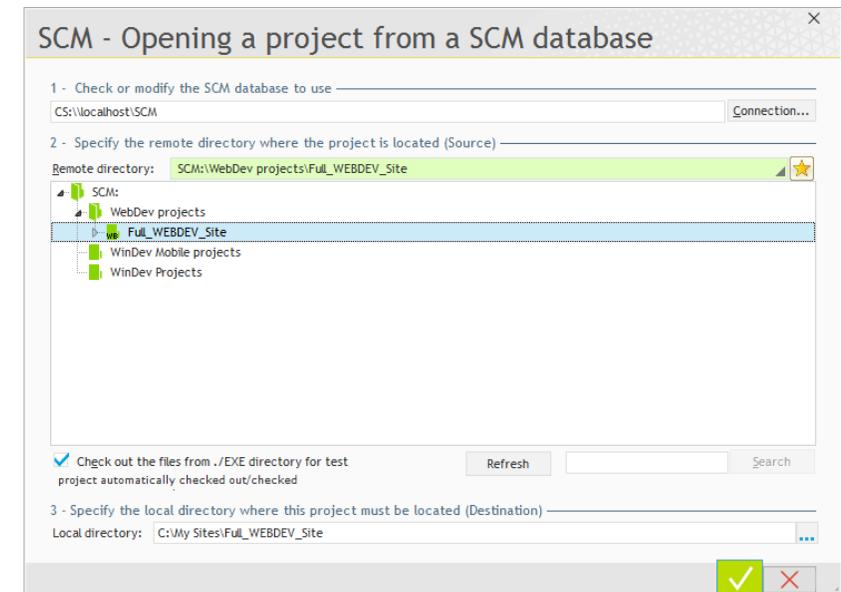
In a real case, in order for other developers to work on a project found in the Source Code Manager, they must retrieve a copy of this project locally.

To do so, the following operations must be performed:

1. Open the project from the Source Code Manager: on the "Home" pane, in the "General" group, expand "Open" and select "Open a project from SCM".
2. Specify the location parameters of SCM database and validate (this step is required only if the current project in the editor does not belong to the SCM):



3. In the screen that is displayed, specify (if necessary) the connection and the local directory:



Note: If the project was already opened from SCM, the SCM proposes to open the project as usual or to overwrite the content (to retrieve the entire project).

This operation must be performed once only by each developer who is using the project.
The developer who added the project into the Source Code Manager (you in this case!) has no operation to perform.



Notes

The next openings of a project managed by SCM are identical to the openings of a project not managed by SCM: all you have to do is open the project (".WWP" file) corresponding to the local copy.

Configuring SCM

Before you start working on the elements of the project found in SCM, you must configure the mode for checking out the project elements.

When working on the elements of a project found in SCM, the element must be checked out from the SCM database before it can be modified, then it must be checked back in once the modifications have been performed. Therefore, the modified element becomes available to all SCM users.

WEBDEV proposes two modes for checking out the project elements:

- **the standard mode:** if you display an SCM element that is not checked out, a dialog box indicates that this element must be checked out before it can be modified. The element can be checked out immediately (check-out button found in the dialog box).
- **the automatic mode:** if you try to modify an SCM element that is not checked out, the SCM automatically proposes to check out this element. Once the check-out is validated, the element can be modified.

Note: this mode is not recommended when using SCM with a slow Internet connection.

In this tutorial, we will be using the automatic check-out.

- ▶ To make sure that the automatic check-out is enabled, on the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV". In the "General" tab, check (if necessary) "Check out the elements during the first modification".

LESSON 12.3. USING A PROJECT IN THE SCM

This lesson will teach you the following concepts ...

- Modifying a project parameter.
- Modifying a project page.
- Synchronizing the project.
- Offline mode.
- SCM administrator.



Estimated time : 30 mn

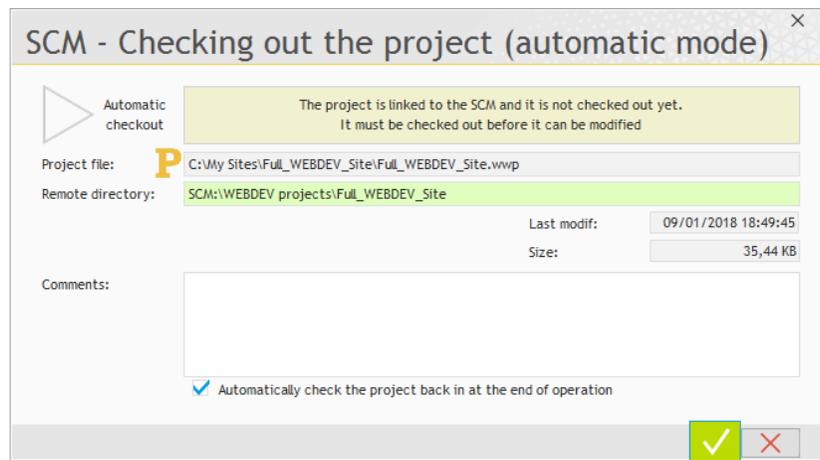
Overview

We are now going to work with SCM in real conditions by performing the following operations:

- Modify a project parameter.
- Modify a project page.

Modifying a project parameter

- ▶ We are going to modify the project by asking to display the skin on the dialog boxes:
 1. Display the project description: in the ribbon, on the "Project" pane, in the "Project" group, click "Description".
 2. Click the "Skin/Styles" tab.
 3. Check "Apply the skin to the dialog boxes".
 4. Validate the description window of project.
- ▶ Several SCM windows appear:
 1. First of all, the window for automatic project check-out is displayed. Indeed, we want to modify a project characteristic therefore the project must be checked out.



2. The option "Automatically check the project back in at the end of operation" is used to automatically check the project back in at the end of modification. Keep this option.
3. Validate this window.
4. WEBDEV proposes to add elements into the SCM database. Accept (green button).
5. The description window of project is closed and the project is automatically checked back into the SCM database.

Modifying a project page

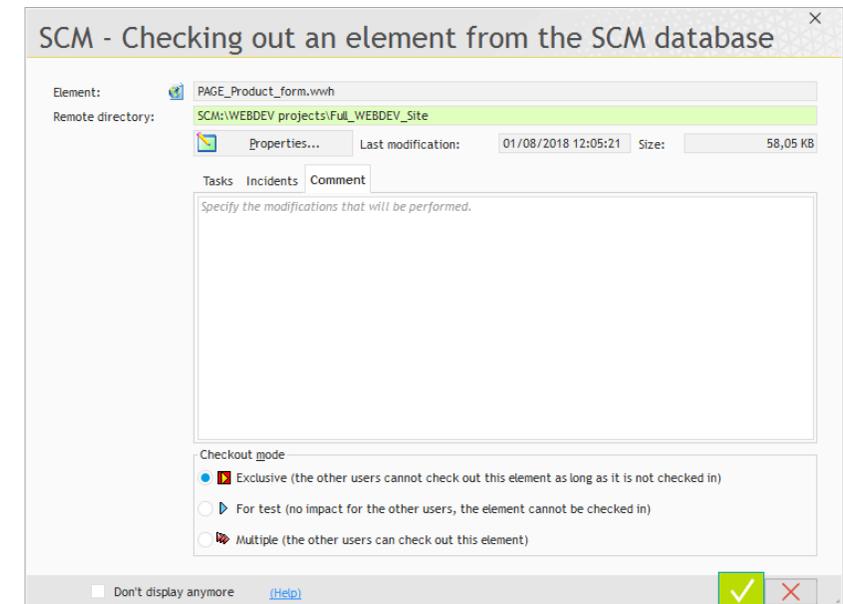
We are now going to modify the "PAGE_Product_form" page. We are going to move the "Cancel" and "Validate" buttons.

To modify a project element, this element must be checked out.

- ▶ To modify the "PAGE_Product_form" page:
 1. Select the "PAGE_Product_form" page in the project explorer and double-click the element in order to open it in the page editor.



2. The automatic check-out is enabled during the first modification: all you have to do is move a control to check the element out.
3. You also have the ability to click the "Check out" icon found in the ribbon of SCM pane ()
4. The check-out window is displayed:



5. Three check-out modes are available:
 - **Exclusive** (recommended mode): no one else will be able to check out this element until it is checked back in. The element can be checked out for test only.

- **For test:** the element can be modified but the modifications will not be checked back in.
 - **Multiple:** the element can be checked out by other users. In this case, the differences between the different element versions can be viewed when the element is checked back in. This mode is reserved to specific cases and to experienced developers.
6. The page will be checked out in exclusive mode. Keep "Exclusive" checked.
 7. Type a comment ("Modifying the buttons" for example). This comment will be useful for the other developers.
 8. Validate the check-out.
The page is checked out.

Modifying the checked-out element

The method for modifying a checked-out element (GUI, code, ...) is the same as the method for modifying an element in a project not managed by SCM.

However, the modifications performed on a checked-out element are not visible to the other developers.

If another developer runs the checked-out element, the element that is currently found in the SCM database will be used.

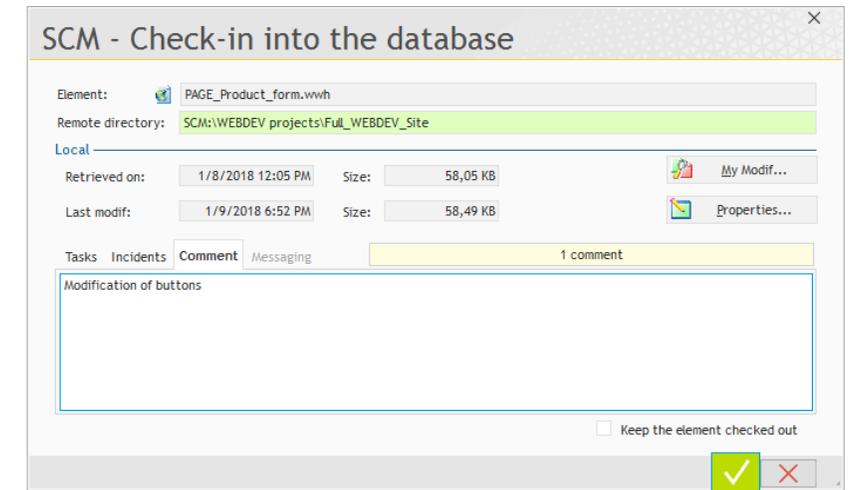
This allows you to make an application evolve by keeping a steady version in the SCM database.

- ▶ Modify the checked-out page:
 1. Select the "Cancel" and "Validate" buttons.
 2. Move the buttons to the right (below the download control).
 3. Save your page (CTRL S).
- ▶ Check your modifications.

Checking the checked-out element back in

Now that the modifications have been made, we are going to check the page back into SCM. Then, your modifications will be made accessible to the other developers.

- ▶ On the "SCM" pane, in the "Current element" group, click the "Check in" button. The following screen is displayed:



This screen is used to:

- find out the modifications performed by comparing the element found in the SCM database with the checked-out element ("My Modif" button).



Notes

Merging code

You have the ability to compare an element to one of its earlier versions. This allows you to compare the code in order to retrieve a section of code that was "lost" or accidentally deleted by another developer.

- access the history of the element found in the SCM database ("Properties" button).
- enter a comment about the modifications performed. By default, WEBDEV proposes the comment that was typed during the check-out.
- send a message to the other developers.
- check in the modifications made to the element while keeping the element checked out ("Keep the element checked out").



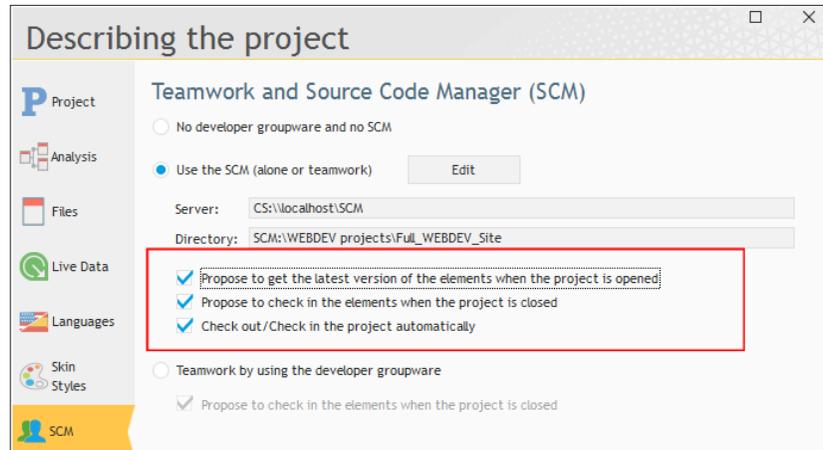
Notes

If you are using the Control Centers, the current task can be ended when the element is checked back into the Source Code Manager. This feature is useful to follow the monitoring of tasks, the corrections of bugs, ...

4. Validate the check-in.

Synchronizing the project

Several options can be used to configure a project handled by SCM. These options are grouped in the "SCM" tab of project description (to display it, click "Description" on the "Project" pane).



These options are as follows:

- *Propose to get the latest version of the elements when the project is opened.*
When opening a project found in the SCM database, this option proposes to get the latest version of project elements.
- *Propose to check the elements back in when the project is closed.*
When closing the project, this option is used to display the list of all elements currently checked out so that they can be checked back in.
By default, the checked-out elements are not checked back in when the project is closed.
- *Check out/Check in the project automatically.*
This option is used to automatically check out or check in the project when handling an element. This option is selected by default.

Off-line mode (or mobile mode)

The SCM allows you to work in off-line mode (or mobile mode).

This mode allows a developer who is using a laptop computer to continue to work on a project found in the SCM database while being disconnected from the SCM database.

The principle is straightforward:

- before the disconnection, on the "SCM" pane, in the "Other actions" group, expand "Remote work" and select "Disconnect for a mobile use".
- during the reconnection, on the "SCM" pane, in the "Other actions" group, expand "Remote work" and select "Reconnect and synchronize". Then, all you have to do is check back in the modified elements.

In mobile mode, two solutions are available for checking out elements:

- No element is checked out from the SCM. The other developers will be able to work on the same elements as you while you are working in off-line mode. When you reconnect to SCM, the modifications made by yourself to the element will have to be merged with the modifications made by the other developers.
- The elements that you want to modify are checked out in exclusive mode. No one else can use the element while you are working in off-line mode.

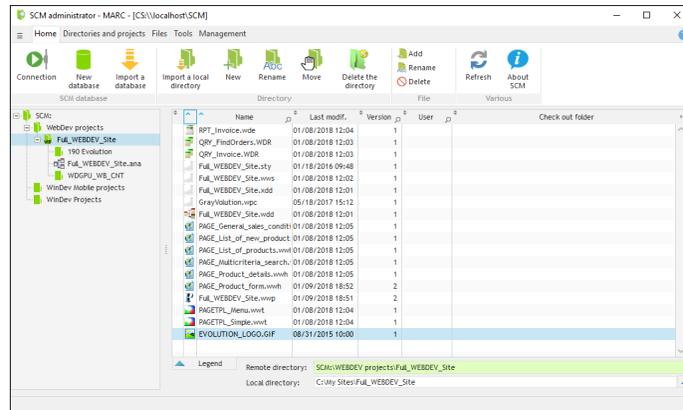
SCM administrator

The SCM administrator is used to directly handle the different projects included in the source code manager.

It allows you to:

- manage the SCM databases (creation, connection to a SCM database).
- manage the files and directories found in a project of SCM database (add, delete, rename, ... files and directories).
- manage the different files found in the SCM database (check-in, check-out, share, ...).
- start some tools (options, maintenance, ...).
- show the history of an element.
- show the status of the elements.
- perform backups.
- grant rights to the different SCM users.
- list the projects in which you are taking part in order to dissociate from them (if necessary).

- Start the SCM administrator: on the "SCM" pane, in the "SCM database" group, click the "Manage" button. All the project elements are listed in the administrator.



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

Disconnecting from SCM

To stop using SCM on a project:

1. Display the description window of project: on the "Project" pane, in the "Project" group, click "Description".
2. In the "SCM" tab, select "No developer groupware and no SCM".
3. Validate the description window of project.

DEVELOP 10 TIMES FASTER

APPENDICES

The WLanguage basics

APPENDICES 1. VOCABULARY

This lesson will teach you the following concepts ...

- The terms used by WINDEV, WEBDEV and WINDEV Mobile



Estimated time: 1 hour

Main terms used

AAF

Automatic Application Features.

Ajax

AJAX (Asynchronous Javascript and XML) is a technology used to refresh the data modified in an HTML page without having to redisplay the entire page.

Alignment

Method for organizing the controls in a window or page. For example, center a control in a page, define the same width for several controls, ...

Analysis

Description of structure of data files and relationships.

Anchoring

Mechanism that consists in defining positioning or resizing rules in order for the content of a window or page to adapt when resizing a window or a browser.

Application RAD

Fast development method of program from a program template.

Array

Type of variable that contains several values. The values can be accessed via a subscript. [] are used in the language to access the array elements.

Assignment

Operation that consists in assigning a value to a variable or control. For example:

```
// Assign the value MOORE to CustomerName variable
CustomerName = "MOORE"
```

The = sign is the assignment operator.

AWP

Page format directly addressable (fixed URL), referenceable, without automatic context.

Block (report)

Element constituting a report. For example, a Header block, a Footer block, a Body block.

Break (report)

Mechanism that consists in grouping and splitting data based on a value.

For example, create a break in a report that is used to list the customers by city. The customers who live in the same city are grouped together. Visually, we separate the customers for each new city.

Class (OOP)

Element defined in Object-Oriented Programming. A class gathers methods (actions) and members (data).

Classic (file)

Type of access to a HFSQL file. A HFSQL file is of Classic type when it is accessed directly from its directory.

Client/Server (file)

Type of access to a HFSQL file. A HFSQL file is of Client/Server type when it is accessed by connecting via a HFSQL engine to a server containing the file.

Context (Page)

Part stored in memory on a Web server that was used to build a page that is visible in a browser.

Control (window or page)

Graphic element used to build the GUI of a program or site.

Control template

Container of one or more controls (with processes) that can be indefinitely re-used in pages.

Main template characteristic: if the initial template is modified, the modifications are automatically applied to the different template uses.

CSS

Description language for the styles of different elements in an HTML Page.

Data binding

Method that is used to associate a graphic GUI element to a data (variable, item).

Database

Element containing the program data. The data is organized in files (also called tables).

Deployment

Consists in installing a program on the user computer.

Dynamic site

Project developed in WEBDEV containing static and dynamic pages. The dynamic pages are used to manage the access to the data stored in the database.

Editor

Program that is used to create a project element (window editor, page editor, ...).

Event-driven

Type of programming. A user action on a window or on a page induces a code to run. The code to run in response to the action is typed in the event representing the user action.

For example, the “Button click” event corresponds to the user clicking this button with a mouse.

External component

Software brick that is used to export one or more business rules in order to re-use them.

File (Table)

Element that constitutes a database. A file is used to store data entered in a program.

For example, a CUSTOMER file will contain the customer details that have been entered in a program.

Global (variable or procedure)

Corresponds to the scope in memory of a variable or procedure. A global element is accessible from any other project element. The opposite is local.

GUI

Graphical User Interface. Description of windows or pages that constitute a program.. This is what the user sees when using the program.

Homothetic

Method for resizing an image in order to maximize its display without distorting its content.

HTML

HyperText Markup Language.

Language used to describe the elements of a Web page.

Index (file)

Synonym: key

Integrity constraint

Rule to respect associated with a file item in order to insure the consistency of data in a database.

Internal component

Container grouping elements from a project (window, page, query, report, class, ...) in order to allow and simplify the share with another project.

Item

Element that belongs to the structure of a data file or table (found in an analysis). For example, a CUSTOMER file could be made of the FirstName and LastName items.

Javascript

Programming language that is used by the browsers to perform calculations and processes in an HTML page (on the computer of Web user).

Key (file)

Characteristics of a file item. A key item is used to optimize the searches and sorts in a data file.

Synonym: index

Link (analysis)

Used to describe the nature of relationship or the common point between 2 analysis files. From a link, description of integrity rules to respect when writing into the relevant files.

Live Data

Mechanism that consists in displaying real data coming from the database when creating the GUI. This mechanism is only used if the element is linked to the data file.

Local (variable or procedure)

Corresponds to the scope in memory of a variable or procedure. A local element can only be accessed in the process where it was defined. The opposite is global.

Member

Variable belonging to a class or structure.

Method

Procedure belonging to a class used to act on the class data (members).

Native access (connector)

Method for connecting to a database from a program.

n-tier

Programming method in layers. Each layer is independent and it can be changed without impacting the other layers.

Benefit: Simplified maintenance

Drawbacks: Difficulty and development time.

Object-Oriented Programming (OOP)

Advanced programming method, opposed to procedural programming.

In OOP, we handle objects, which means grouped sets of variables and methods associated with entities that include these variables and these methods.

In procedural programming, we define functions that call each others. Each function or procedure is associated with a specific process that can be divided into sub-processes until we get basic functions.

OOP

Abbreviation of Object-Oriented Programming.

Page

Element of a WEBDEV project where the graphical site interface is defined. In most cases, a site includes several pages linked together.

Page template

Container representing a standard page that can be applied to one or more project pages.

Main template characteristic: if the initial template is modified, the modifications are automatically applied to the different template uses.

Parameter (window, page, procedure, method, ...)

Element expected in a window, page, procedure, method, ... during the call to this one. Each value passed in parameter must be assigned to a variable.

Popup

Type of window (or page). A popup is a window (or a page) that is opened above another window (or page). You can still see the content of the window (or page) below while performing the input in the popup. Private

Variable or procedure that can only used in the current element.

Popup menu

Drop-down menu containing possible actions according to the location where the right mouse click occurred and to the type of element on which this click was performed.

Procedure

Project element containing the code of a process to run.

Project

Element that groups all the elements that constitute a program or a site. A project contains for example an analysis, pages, reports, queries, ...

Project code

Code run when starting a program or a site.

Project configuration

Description of output format of project: Windows executable, Linux, JAVA, ...

Property (control, window, ...)

Keyword representing a characteristic of an element. The properties are used to handle and modify the characteristics of project elements by programming.

Public

Variable or procedure hat can be used from all the elements.

Query

Element written in SQL language that is used to access (in read-only or in read/write) the content of a relational database.

RAD

Acronym for Rapid Application Development

Fast development method of program from an analysis (description of data files).

Report

Project element that defines a print to perform.

Report template

Container representing a standard report that can be applied to one or more project reports.

Main template characteristic: if the initial template is modified, the modifications are automatically applied to the copies.

SCM

Source code manager. Tool for organizing, sharing project resources, managing rights, ...

Skin

Style book of a WEBDEV site.

Element in which is defined the graphic layout of the pages found in a WEBDEV project.

Static site

Project developed in WEBDEV containing static pages only, which means pages that do not have access to a database.

Structure

Type of variable that includes several sub-variables.

Style

Element used to describe the graphic style of a control in a window or in a page. A style includes, for example, a font, the character size, the character color, etc.

Style sheet

Contains the list of styles used in a project.

Table (control)

Graphic element found in a window or page. A table control includes one or more columns and several rows.

Table (file)

Element that constitutes a database. A table is used to store the data entered in a program. For example, a CUSTOMER table will contain the customer names and addresses that have been entered in a program.

User groupware

Tool for describing and managing the access rights to GUI for the users of a program or site. For example, prevent a user from clicking a "Delete" button according to his login or to his group.

Variable

Element used to store a program value in memory. Several types of variables are available. Each type corresponds to the nature of the value that will be stored. For example, a string variable to store the name of a person, a currency variable to store an amount.

Webservice

Program installed on a Web server whose processes are accessible via the Web.

XML

Language for organizing data in order to normalize and simplify the exchange of data (mainly used in the exchanges with the Webservices).

Zoning

Method for defining the splitting of a page.

APPENDICES 2. THE VARIABLES

This lesson will teach you the following concepts ...

- What is a variable?
- The different types of variables
- The scope of variables
- The String type in details
- The Array type in details



Estimated time: 1 hour

What is a variable?

In a programming language, a variable is used to store data. These memory sections contain strings, numbers, etc.

The variables are used to perform calculations, to perform comparisons or to store information that will be used later.

Declaring a variable

```
Price is currency
```

↑ ↑
Variable Variable
name type

Initializing a variable

```
Price = 500.32
```

↑
Variable value

A variable is represented by:

- a name: Name given to the variable so that it can be used by the language.
- a type: Nature of data stored in the variable (see Types of variables).
- a value: Information stored in the variable.
- a scope: Limit for using the variable in the program (see Scope of variables). The scope is mainly defined by the location where the variable is declared.

Declaring a variable

The variable must be declared (which means created) before it can be used.

- Example of simple declaration:

```
Price is currency
```

- **Price** represents the variable name.
- **is** allows you to declare the variable. WLanguage is using the natural language.
- **currency** corresponds to the type of variable.
- Example of multiple declaration:

```
LastName, FirstName are strings
```

- **LastName, FirstName** represent the names of variables.
- **are** allows you to declare a set of variables.
- **strings** represents the type of variables.

Assignment and use

When the variable is declared, you have the ability to assign it (or to give it a value).

For example:

```
// Assign a currency variable
Price = 1256.67
// Assign a string variable
LastName = "Doe"
```

The = operator is used to perform this assignment.



Notes

In WLanguage, the " character (double quote) is the character used to delimit a character string. In the above example, the doubles quotes are used to assign the Doe value to the Name variable.

The variable content can be read and handled: all you have to do is use the name given to the variable in order to access it.

The following example is used to read and display the content of Price variable on the screen:

```
Info(Price)
```

The types of variables

The variable type is used to specify the kind of information that will be stored in the variable. The most common types are:

- boolean (True or False),
- string ("Doe"),
- integer (1234),
- currency (12.32),
- real (7.766666),
- etc.



Important

Use the type corresponding to the information that must be stored. Therefore, you will optimize the memory and you will avoid calculation or process errors when using variables in the WLanguage functions.

Most of these types of variables will be used in this tutorial. See the online help regarding the relevant type for more details.



Notes

Other types are available like the arrays, the structures, the dates, the times, ... Advanced variables are also available. These advanced types group all the characteristics of the element used in a single variable. Advanced types can be used to handle the XML documents, the emails, the XLS files, ... This type of variable will be used later in this tutorial.

The scope of variables

The variables can be declared anywhere in the code. However, according to the position of its declaration, the variable cannot be used to perform processes or calculations. We talk of variable scope.

Two types of scope are available:

- Global.
- Local.

Global scope

Global means that the variable has an extended visibility in the code. The variable is visible outside the location where it was declared. Several levels are available:

- Project and Set of procedures,
- Window, Mobile window, Page, Report.

A variable declared **at project level** has the highest visibility in the program. The variable is visible anywhere, in all the program processes. However, you should not declare too many variables with this scope ; indeed, the memory occupied by the variable is always reserved even if the variable is not used. Using a large number of global variables is not recommended in the program architecture. To pass variables to a process, we recommend that you use parameters (see "Parameters of a procedure", page 464 for more details).

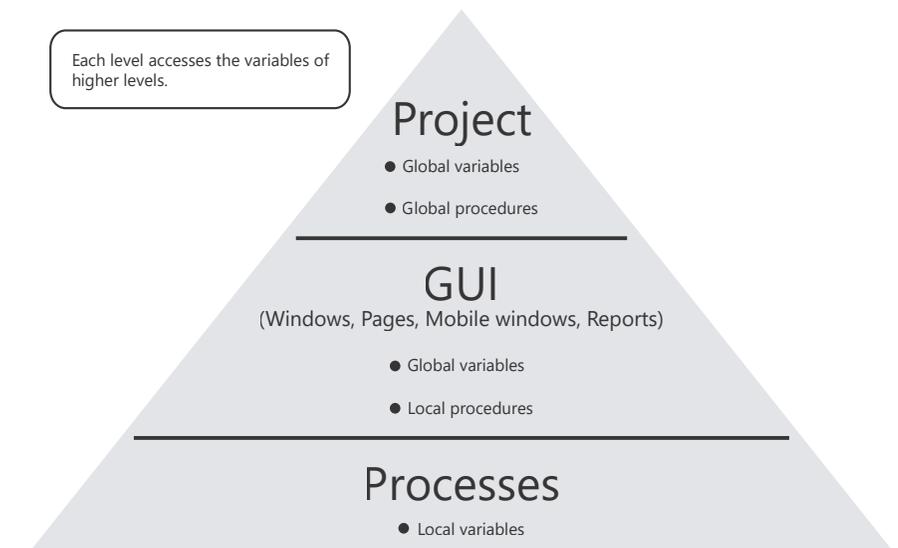
A variable declared **at Set of Procedures level** and a variable declared at project level have the same visibility. The benefit to declare a variable at Set level is to group (or classify) the variables by theme in order to make the initialization process of the project more readable.

A variable declared **at Window, Mobile Window, Page or Report level** limits the scope of the variable to all the processes of the element (Window, Mobile Window, Page or Report) and its controls. This is used to encapsulate and limit the uses.

Local scope

Local means that the variable has a limited visibility in the code. The variable is visible in the process where it was declared. This is used to restrict the use of the variable to the process.

Summary scope diagram



A variable is global when it is declared:

- in the initialization code of the project (or in the declaration code of the set of procedures). The variable is global to the project.
- In the declaration code of global variables of window, page or report. The variable is global to the element (window, page or report) where it was declared.

In all the other cases, a variable is local to the process where it is declared.

Simple operations on the variables

Several mathematical operators can be used to perform calculations on variables:

- + to perform an addition.
- - to perform a subtraction.
- * to perform a multiplication.
- / to perform a division.

Other operators can be used to perform calculations:

- ++ to increment from 1 (add 1 to the variable).
- -- to decrement from 1 (subtract 1 from the variable).
- += to assign by adding a value.
- -= to assign by subtracting a value.

Examples:

```
// Declaration of variables
Ctr is int
V1 is int
Res is numeric

// Assignment
Ctr = 10
V1 = 3

// Use of operators
Ctr = Ctr + 3 // Ctr is equal to 13
Ctr ++ // Ctr is equal to 14
Ctr -= 8 // Ctr is equal to 6
Ctr = Ctr * V1 // Ctr is equal to 18
Res = Ctr / 5 // Res is equal to 3.6
```

Comparison operators are also available:

- < less than
- > greater than
- <= less than or equal to
- >= greater than or equal to
- <> different from
- = equal to

Other operators are available. See the online help for more details (keyword: "Operators").

Tips

- It is very convenient to name the variables with long names (and to avoid short names such as i, j, k, ...). When reading the program again, you will be able to easily find out the purpose of variable.
- To define the name of variables, all the Unicode characters (including the accented characters) are accepted. Meaning improved readability! Caution: some characters are not allowed: space, =, dot, comma, ...
- It is very important to give the proper type to the variable according to its use. For example, to store several digits, you may have to:
 - use a numeric variable if this variable must be used for calculations.
 - use a string variable if this variable must be used to store digits without performing calculations (to store the social security number for example).

Details of variable type: the String variables

The String variables are the most often used types of variables.

Let's present in details some features available for this type of variable.

The String type

The String type is used to store and handle the characters and the character strings.

We have already seen how to initialize a string variable:

```
LastName is string
// Assign a string variable
LastName = "Doe"
```

There is no need to declare the string length: this length automatically adapts when using the variable.



TIP

To initialize a string variable with a text on several lines, use the following syntax:

```
<Variable Name> = [
    <Text of Line 1>
    <Text of Line 2>
]
```

For example:

```
MyString is string
MyString = [
    Example of
    multi-line string
]
```

You also have the ability to assign a string variable with the content of a control handling strings. The following code is used to assign a string variable with the content of an edit control:

```
LastName is string
// Assign a string variable
// with the content of the EDT_LASTNAME edit control
LastName = EDT_LastName
```

In addition to the main comparison operators, several powerful operators are used to manage the extractions and concatenations in advanced mode.

Some examples:

- "+" : To concatenate strings
- "~" : To check the flexible equality

Specific WLanguage functions are used to perform various operations: search, extraction, size, switch to uppercase characters, ...

Examples:

```
str is string
str = "WEBDEV is a great tool"
// Extract a sub-string from left
Info(Left(str,6)) // Displays "WEBDEV"
// Extract a sub-string from right
Info(Right(str,3)) // Displays "ool"
```



Notes

The different WLanguage functions can be nested. A WLanguage function can be used as parameter of another WLanguage function.

For example:

```
Info(Middle(Left(str,13),8,3)) // Displays "is"
```

See the help about the character strings and about the functions for handling character strings for more details.

Practical example

To handle the different concepts of this lesson, we are going to create different windows. These windows will be created in the "My_First_Pages" project.

- ▶ Open the "My_First_Pages" project if necessary. If the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My first pages (Exercise)".

To handle the Character String variables, we are going to create the following page:



This page is used to:

- find a string inside another one.
- compare two strings.
- ▶ Create a new blank page:
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Page" then "Page". The wizard for page creation is displayed.
 2. Select "Blank page" and "Simple layout".
 3. Validate. The page is automatically created in the editor. The backup window of created element is displayed.
 4. Specify the page title: "Variables". The page name ("PAGE_Variables") is automatically proposed.
 5. Click the green button to validate the information displayed in the backup window.
- ▶ To create the edit control containing the string:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click the location where the edit control will be created (at the top, in the middle of the page for example).
 3. Right-click the control and select "Description".
 4. In the "General" tab, specify:
 - The control name: "EDT_Text".
 - The control caption: "Text".

5. Select the "Content" tab. This tab is used to define the default value of edit control. Type "WEBDEV is a great tool".
6. Validate the description window. The text typed is directly displayed in the control.

► The control content is truncated in the editor. To display the control properly:

1. Select the control.
2. Enlarge the control in width with the sizing handles in order for the control content to be entirely displayed.

Finding a string

► To create the "Find a string" button:

1. On the "Creation" pane, in the "Usual controls" group, click .
2. Click the location where the button will be created (below the edit control for example).
3. Click the button that was just created. The text displayed in the button becomes editable. Enter the caption: "Find a string".
4. Press the [ENTER] key to validate the input.
5. Display the processes associated with the button ([F2] key).
6. Type the following code in the server "Click" process of the button:

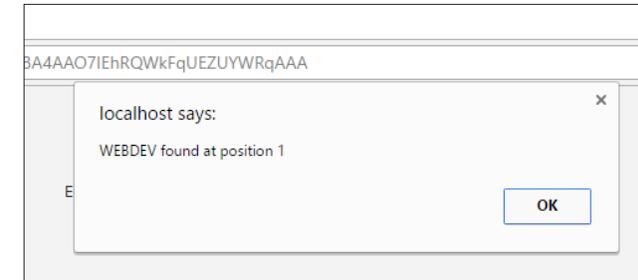
```
sStringToFind is string = "WEBDEV"
nPos is int
nPos = Position(EDT_Text, sStringToFind)
IF nPos = 0 THEN
    Info(sStringToFind + " not found in the text")
ELSE
    Info(sStringToFind + " found at position " + nPos)
END
```

Let's study this code:

- Two variables are declared: a String variable corresponding to the sought string and an Integer variable corresponding to the position of sought string.
- Position is used to find a string inside another one. In our example, we search for the content of sStringToFind variable in the content of EDT_Text control.
- The conditional IF statement is used to check the result of Position. This type of statement will be presented later in this section on WLanguage.
- If Position returns 0, it means that the sought string was not found. A message is displayed by Info.

► Let's now run the page test:

1. Click  among the quick access buttons (or press [F9]).
2. The window that was just modified is automatically saved then it is started in execution. Click the button. The WEBDEV word is found.



3. Modify the content of edit control (by replacing WEBDEV by WB for example) and click the button. The WEBDEV word is not found.
4. Close the browser.

Comparing two strings

► To create the "Compare two strings" button:

1. On the "Creation" pane, in the "Usual controls" group, click .
2. Click the location where the button will be created (on the right of existing button for example).
3. Click the button that was just created. The text displayed in the button becomes editable. Enter the caption: "Compare two strings".
4. Press the [ENTER] key to validate the input.
5. Adapt the control size.
6. Display the processes associated with the button ([F2] key).
7. Type the following code in the server click process:

```
sStringToCompare is string = "WEBDEV"

IF EDT_Text ~= sStringToCompare THEN
    Info("The text displayed corresponds to " + sStringToCompare)
ELSE
    Info("The text displayed does not correspond to "+...
        sStringToCompare)
END
```

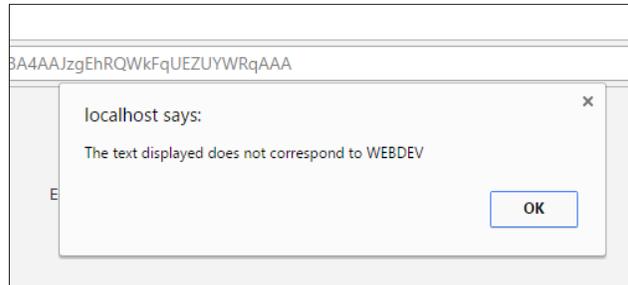
Let's study this code:

- A String variable is declared. This variable contains the string to compare.
- The ~= operator corresponds to a flexible equality. This operator is used to perform a comparison while ignoring the case, the spaces found before and after the string and the accented characters.

Note: to type the "~" sign, press [ALT GR]+[2].
- If the comparison is true, the edit control corresponds to the word found in the string to compare regardless of the case used.

► Let's now run the page test:

1. Click  among the quick access buttons (or press [F9]).
2. Click the "Compare two strings" button. The message indicates that the comparison is not correct.



3. In the edit control, type "WEBDEV" in uppercase characters and click the "Compare two strings" button. The WEBDEV word is found.
4. Modify the content of edit control (by replacing WEBDEV by WEBDEV for example) and click the button. The WEBDEV word is also found.
5. Close the browser.

Details of another variable type: the arrays

The arrays are a common type of variable.

An array is a structured type that is used to group a set of elements of the same type. Each array element can be directly accessed by its subscript.

Specific WLanguage functions are used to handle the arrays and their elements. These functions start with "Array".

Declaration

The declaration of an Array variable is performed as follows:

```
<Array Name> is array of <Type of Array Elements>
```

For example:

```
arrString is array of strings
arrInt is array of int
```

Filling an array and accessing the elements

During its declaration, the array is empty. **Add** is used to add elements via the following syntax:

```
Add(<Array Name>, <Element Value>)
```

To access the array elements, use the following syntax:

```
<Array Name> [<Element Subscript>]
```



Important

The subscripts of array elements start from 1.

Example:

```
// Create an array of strings
MyArray is array of strings
// Add elements
Add(MyArray, "WINDEV")
Add(MyArray, "WEBDEV")
Add(MyArray, "WINDEV Mobile")
// Display the content of third element
Trace(MyArray[3]) // Displays "WINDEV Mobile"
```

Fast array initialization

To initialize an array, you also have the ability to use the following syntax :



Tip

```
// Declares an array
arrDay is array of strings
// Initialization with
// the names of the days of week
arrDay = ["Monday", "Tuesday", "Wednesday", ...
          "Thursday", "Friday", "Saturday", "Sunday"]
```

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

Advanced arrays

Advanced arrays are also available: array with several dimensions, array of arrays, associative array, array of structures, array of objects, ... See the online help for more details (keyword: "Table").

APPENDICES 3. THE CONDITIONAL STATEMENTS

This lesson will teach you the following concepts ...

- The IF statement
- The SWITCH statement
- Practical example



Estimated time: 1 hour

Overview

WLanguage is a set of statements used to handle data.

The conditional statements are used to compare variables and/or values between themselves in order to perform different processes. Several conditional statements are available:

- IF... THEN... ELSE... END
- SWITCH ...

The IF and SWITCH statements

The IF statement

This statement is used to run an action or another one according to the result of an expression. If the 'expression is checked, a process is run ; if the 'expression is not checked, another process can be started.

The IF statement can be used as follows:

```
IF <Expression to check> THEN
    Process to run if the expression is checked
ELSE
    Process to run otherwise
END
```

Code sample: The following code selects a number at random and displays a message according to the value.

```
Tot is currency
// Selects a number at random between 100 and 4000
Tot = Random(100, 4000)
IF Tot>2000 THEN
    Info("The amount is greater than 2000")
ELSE
    Info("The amount is less than or equal to 2000")
END
```

In this case, the expression to check corresponds to "Tot>2000".

Note: Several code lines can be run during the process corresponding to a condition. In this case, the following syntax must be used:

```
IF <Expression to check> THEN
    Code line 1
    Code line N
ELSE
    Code line 1
    Code line N
END
```

The SWITCH statement

This statement is used to evaluate an expression and to run a process for each possible expression value.

The SWITCH statement can be used as follows:

```
SWITCH <Expression>
CASE Value 1:
  Process 1...
CASE Value 2:
  Process 2...
...
CASE Value N:
  Process N...

OTHER CASE
  Process ...
END
```

Example: The following code retrieves today's date and displays a different message according to its value. A specific message is displayed for the 1st and for the 15th of the month. In the other cases, today's date is displayed.

```
D is Date
D = Today()
SWITCH D..Day // Checks the day of the date
CASE 1: Info("Today is the first day of the month")
CASE 15: Info("Today is the 15th of the month.")
OTHER CASE:
Info("Today is: " + DateToString(D))
END
```

Notes:

- If the code line "CASE 1:..." is run, the other code lines corresponding to the possible values are not run.
- Several values can be grouped in the same case. The different values are separated by a comma. For example:

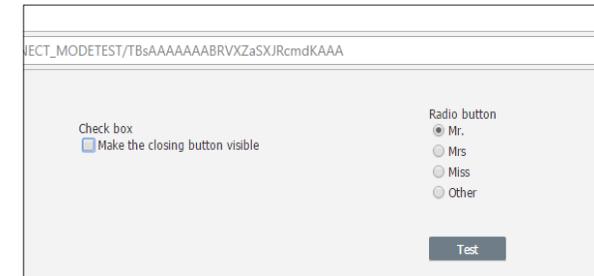
```
Sub is int = 2
SWITCH Sub
CASE 1,2: Info("Case 1 or 2")
CASE 3: Info("Case 3")
OTHER CASE: Info("Other case")
END
```

- Several code lines can be run during the process corresponding to an expression value. In this case, the following syntax must be used:

```
SWITCH <Expression>
CASE Value 1:
  Process 1 - Code line 1...
  Process 1 - Code line 2...
CASE Value N:
  Process N - Code line 1...
  Process N - Code line 2...
OTHER CASE
  Process ...
END
```

Practical example: Using the IF and SWITCH statements

To use the conditional statements, we are going to create the following window:



Two operations are available:

- If the user clicks the check box, the "Close" button is displayed.
- If the user clicks the "Test" button, the box checked in the check box is displayed.

Project used

To handle the different concepts of this lesson, we are going to create different windows. These windows will be created in the "My_First_Pages" project.

- ▶ Open the "My_First_Pages" project if necessary. If the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My first pages (Exercise)".

Creating the page

- ▶ Create a new blank page:
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Page" then "Page". The wizard for page creation is displayed.
 2. Select "Blank page" and "Simple layout".
 3. Validate. The page is automatically created in the editor. The backup window of created element is displayed.
 4. Specify the page title: "Conditional statements". The page name ("WIN_Conditional_statements") is automatically proposed.
 5. Click the green button to validate the information displayed in the backup window.

Creating the page controls for the conditional IF statement

Two controls must be created:

- a Check Box control used to display (or not) the "Close" button.
 - a "Close" button used to close the browser window.
- ▶ To create the Check Box control:
 1. On the "Creation" pane, in the "Usual controls" group, click "Check box" then click the position where the control will be created in the page (at the top of the window for example).
 2. Click the control that was just created: the "Option 1" caption becomes editable.
 3. Type the option caption: "Make the closing button visible".
 4. Press the [ENTER] key to validate the input.
 - ▶ The option caption is truncated in the editor. To display the control properly:
 1. Select the control.
 2. In the popup menu (right mouse click), select "Resize".
 3. The control is immediately enlarged and the caption is entirely displayed.
 - ▶ To create the "Close" button:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click the location where the button will be created (on the right of Check Box control for example).
 3. Click the button that was just created. The text displayed in the button becomes editable. Type the caption: "Close".
 4. Press the [ENTER] key to validate the input.
 - ▶ This button is used to close the browser window. We are going to type the corresponding WLanguage code:
 1. Select the "Close" button.
 2. Press the [F2] key: the code editor displays the processes associated with the button.
 3. Type the following code in the "Click BTN_Close" browser process:

```
BrowserClose ()
```

BrowserClose is used to close the current window of Internet browser.

- ▶ The "Close" button must be invisible when opening the page. This information corresponds to its "initial status". We are going to modify this status in the description window of control:
 1. Select the "Close" button.
 2. Display the popup menu of the control and select "Description".
 3. In the "GUI" tab, modify the initial status of control by clicking "Visible". This option is now unchecked.
 4. Validate the description window of control.
 5. The "Close" button is still visible in the editor. Its status was changed in execution only.
 6. Save the modifications by clicking  among the quick access buttons (on the left of ribbon) or with the [CTRL] + [S] key combination.

Conditional IF statement

In our example, a click on the check box must trigger the display of Close button.

We are going to type the WLanguage code associated with the check box.

- ▶ We are going to type the WLanguage code associated with the check box.:
 1. Select the Check Box control.
 2. Press the [F2] key.
 3. In the "Exit with modification from CBOX_NoName1 (browser code)" process, type the following code:

```
IF CBOX_NoName1=True THEN
  BTN_Close..Visible = True
ELSE
  BTN_Close..Visible = False
END
```

Let's study this code:

- CBOX_NoName1 and BTN_Close respectively correspond to the names of Check Box and Button controls.



Tip

To handle the current control in one of the processes associated with it, you have the ability to use the MySelf keyword.

In our example, you also have the ability to write:

```
IF MySelf = True THEN
```

- The Check Box control is a 2-state control: checked/unchecked. It corresponds in programming to a Boolean variable. If the control is checked, its value is set to True ; if the control is unchecked, its value is set to False.
- This code checks the value of check box.
 - If the check box is checked, the BTN_Close button becomes visible via the WLanguage **..Visible** property.
 - If the check box is unchecked, the BTN_Close button becomes invisible.



Notes

Several functions and properties are available in WLanguage. The functions can accept parameters and they return results. The properties are directly applied to the controls via the following syntax:

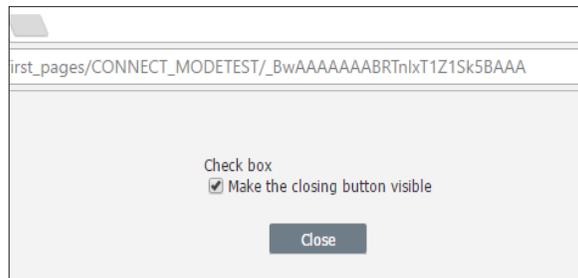
<Control name>.<Property name>.

In our example, the **Visible** property is used on the BTN_Close control.

The online help of a function or property can be displayed at any time by pressing the [F1] key. See "How to access the online help?", page 16 for more details.

► Let's now run the page test:

1. Click  among the quick access buttons (or press [F9]).
2. In the page in execution, click the check box. The "Close" button becomes visible.



3. Click the "Close" button: the browser tab is closed and the page editor is displayed.

Creating the page controls for the conditional SWITCH statement

Two controls must be created:

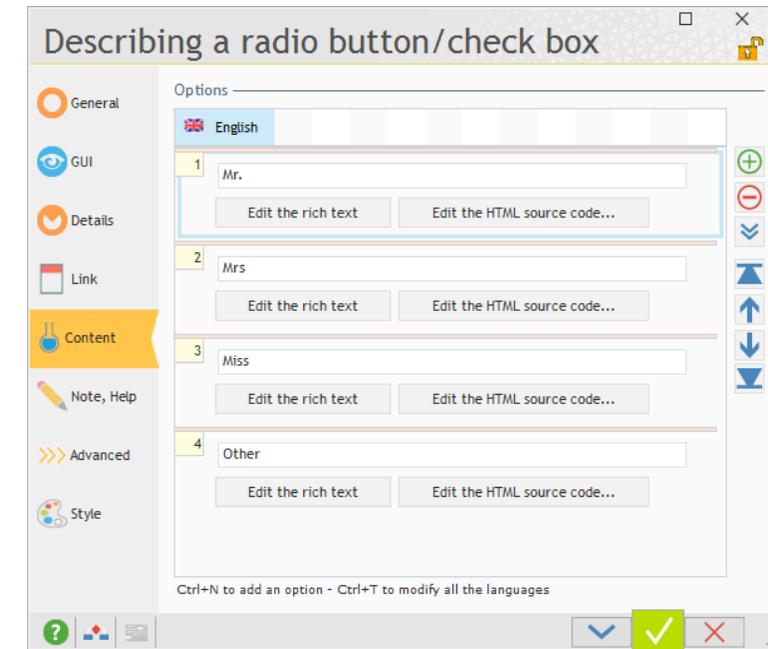
- a Radio Button control used to select the value to check.
- a Button control used to run the test.

► To create the Radio Button control:

1. On the "Creation" pane, in the "Usual controls" group, click "Radio button" (click the arrow).
2. Click in the window to create the control (below the Check Box control created beforehand for example).
3. Display the popup menu of the control (right mouse click) and select "Description".
4. In the "Content" tab, add two new options by clicking the button .

5. Specify the captions for the different options:

- "Option 1" becomes "Mr."
- "Option 2" becomes "Mrs."
- "Option 3" becomes "Miss"
- "Option 4" becomes "Other".



6. Validate the description window of Radio Button control (green button).

► To create the "Test" button:

1. On the "Creation" pane, in the "Usual controls" group, click .
2. Click the location where the button will be created (on the right of Radio Button control for example).
3. Click the button that was just created. The text displayed in the button becomes editable. Type the caption: "Test".
4. Press the [ENTER] key to validate the input.

SWITCH statement

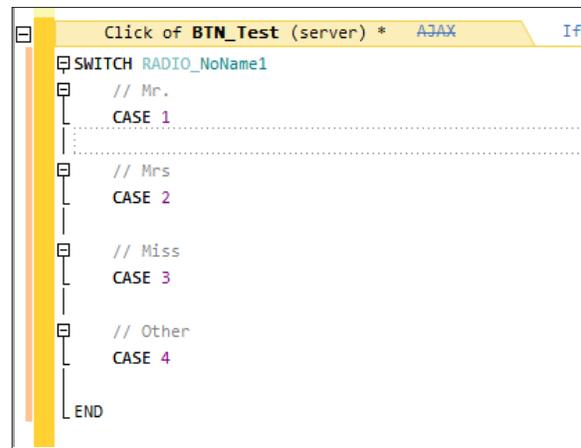
In our example, the Test button is used to check the selected value and to display a message.

► We are going to type the WLanguage code associated with the Test button:

1. Select Test button.
2. Press the [F2] key.
3. Type the following code in the "Click of BTN_Test (Server code)" process:

```
SWITCH RADIO_NoName1
```

4. When pressing the [ENTER] key to go to the next line, the code editor automatically displays the different SWITCH capabilities:



5. Type the following code:

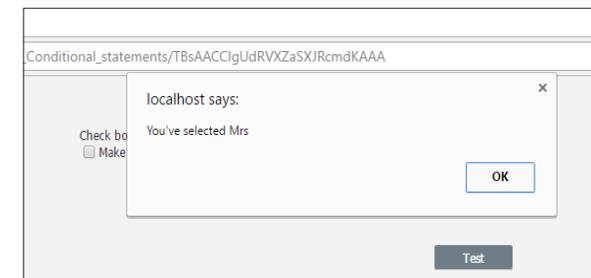
```
SWITCH RADIO_NoName1
// Mr
CASE 1
  Info("You've selected Mr")
// Mrs
CASE 2
  Info("You've selected Mrs")
// Miss
CASE 3
  Info("You've selected Miss")
// Other
CASE 4
  Info("You've selected Other")
END
```

Let's study this code:

- RADIO_NoName1 corresponds to the name of the Radio Button control.
- The Radio Button control is a control proposing several options (4 in our case). A single option can be checked at a time. The Radio Button control corresponds in programming to an Integer variable. Each menu option is associated with a value. If this option is checked, the Radio Button control takes for value the identifier of checked option.
- This code checks the value of radio button. A message is displayed according to the value of radio button.

► Let's now run the page test:

1. Click  among the quick access buttons (or press [F9]).
2. In the page whose test is run, select an option and click "Test": the message corresponding to the selected option is displayed.



3. Close the browser.

APPENDICES 4. THE LOOPS

This lesson will teach you the following concepts ...

- The FOR statement
- The LOOP statement
- The WHILE statement
- Practical example



Estimated time: 1 hour

Overview

The loop statements are used to run a process in a recurring way. A specific loop statement is used according to the number of occurrences. Several statements can be used to perform loops:

- FOR ...
- LOOP ...
- WHILE ...

The FOR statement

The FOR statement is used when the number of occurrences to process is known. This statement is used to manage the number of occurrences via a variable in which the passages performed in the loop will be counted.

The syntax of the FOR statement is as follows:

```
FOR Subscript = Start Value TO End Value
  Process to run
END
```

For example, the following code runs the process 2000 times:

```
FOR nCtr = 1 TO 2000
  // Process to run
END
```

Note: An increment step of subscript can be defined via the STEP keyword. For example, the following code runs the process 200 times and the nCtr variable decreases from 10 to 10:

```
FOR nCtr = 2000 TO 1 STEP -10
  // Process to run
END
```

The LOOP statement

The LOOP statement is used to perform loops when the number of occurrences to process is not known. In this case, a test must be used to exit from the loop.

The syntax of the LOOP statement is as follows:

```
LOOP
  Process to run
  IF <Expression> THEN BREAK
END
```

For example:

```
Counter is int
Counter = 10
LOOP
  // Process to run
  Counter = Counter - 1
  IF Counter = 0 THEN BREAK
END
```



Tip

The LOOP statement and the FOR statement can have the same behavior: all you have to do is use the syntax with exit according to the number of iterations:

```
LOOP (<Number of Iterations>)
```

```
...
END
```

Example:

```
LOOP(10)
  // Process to run
END
```

The WHILE statement

The WHILE statement and the LOOP statement operate according to the same principle. The difference is that the test of exit condition is performed BEFORE running the loop code. This test is used to compare a variable. This variable starts from a start value and it is modified in the loop until it reaches the value that triggers the exit from the loop.

The syntax of the WHILE statement is as follows:

```
<Initialize the variable to its start value>
WHILE <Compare the variable to its end value>
  Process to run
  <Modify the variable>
END
```

For example:

```
Counter is int
Counter = 0
WHILE Counter<10
  // Process to run
  Counter = Counter + 1
END
```

Practical example: Using loops

To check the use of loops, we are going to create a page into which an Image control will be moved.

Project used

To handle the different concepts of this lesson, we are going to create different pages. These pages will be created in the "My_First_Pages" project.

- ▶ Open the "My_First_Pages" project if necessary. If the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My first pages (Exercise)".

Creating the page

- ▶ Create a new blank page:
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Page" then "Page". The wizard for page creation is displayed.
 2. Select "Blank page" and "Simple layout".
 3. Validate. The page is automatically created in the editor. The backup window of created element is displayed.
 4. Specify the page title: "Loops". The page name ("PAGE_Loops") is automatically proposed.
 5. Click the green button to validate the information displayed in the backup window.

Creating controls

We are going to create the Image control then the buttons used to handle this image.

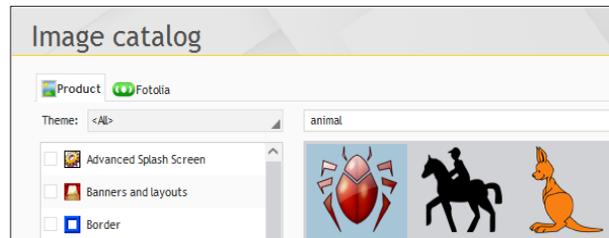
- ▶ To create an Image control:
 1. On the "Creation" pane, in the "Usual controls" group, click "Image". Position the control in the page.
 2. Right-click the control and select "Description".
 3. We are going to associate an image to the tab via the image catalog of WEBDEV.



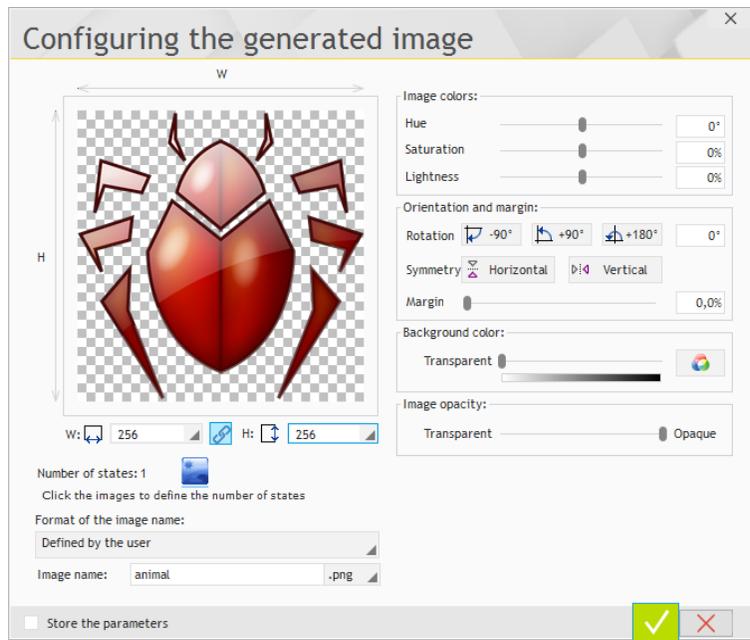
Notes

As soon as an image can be displayed in a control or page, WEBDEV proposes to use the image catalog. This image catalog is started via the "Catalog" option (available by clicking the button ) . This catalog contains hundreds of images, cliparts, ...

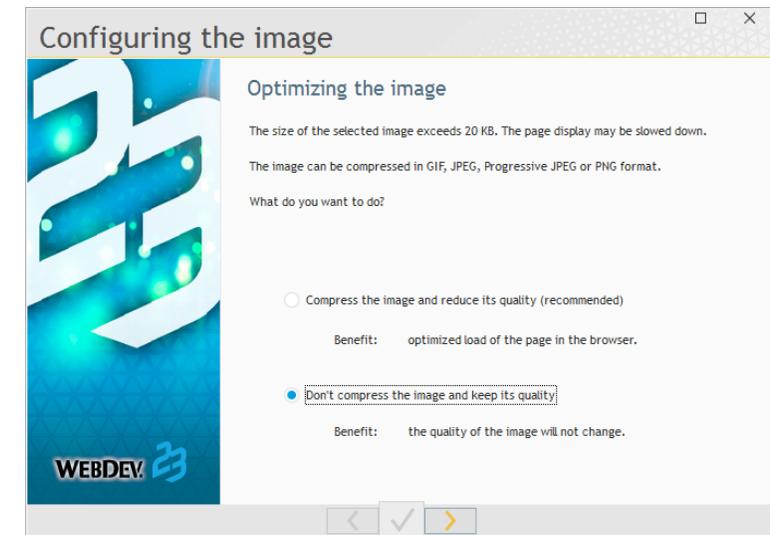
- Click the button  on the right of "Image" control. Select "Catalog" from the popup menu that is displayed. The window of image catalog is displayed.
- Type "Animal" in the search area. Click the magnifier to start the search.



- Select the image  and validate (green button).
- A window used to configure the image is displayed. Keep the default options and validate.



- A window for configuring the image is displayed. Select "Don't compress the image and keep its quality".



- Go to the next wizard step.
- Keep the default options and end the wizard.
- The path of the image is displayed in the description window of the Image control.
- In the "GUI" tab, check "The control can be stacked".
- Validate the description window of control.
- Position the Image control at the top left of the page.
- Save the page (CTRL S or click  among the quick access buttons).
- We are now going to create a button in order for the image to be moved by 300 pixels to the right in the page. to do so, we will be using the WLanguage FOR statement.
 - On the "Creation" pane, in the "Usual controls" group, click .
 - Click the location where the button will be created (below the Image control for example).
 - Click the button that was just created. The text displayed in the button becomes editable. Type the caption: "FOR statement".
 - Press the [ENTER] key to validate the input.
 - Display the description window of control (press [ALT] [ENTER]).
 - In the "General" tab, in the "Operations on controls" area, select "None".
 - Validate the description window of control.
 - Display the processes associated with the button ([F2] key).
 - Type the following browser click code:

```
// Moves the image horizontally by 300 pixels
FOR i=1 TO 300
  IMG_Animal..X++
END
```

This code is used to modify the X coordinate of the Image control (..X property) in a loop from 1 to 300. At each loop turn, the X coordinate is increased by one pixel.



Notes

When typing conditional statements in the code editor, the different possible syntaxes are proposed in a list by the code editor.

If you select one of these syntaxes, the structure of the statement is automatically inserted into the code editor. All you have to do is type the code corresponding to each statement section.

- ▶ Let's now create a button used to move the image until the image border is positioned at 1200 pixels. To do so, we will be using the WHILE statement of WLanguage.

 1. On the "Creation" pane, in the "Usual controls" group, click **Ok**.
 2. Click the location where the button will be created (on the right of FOR button for example).
 3. Click the button that was just created. The text displayed in the button becomes editable. Type the caption: "WHILE statement".
 4. Press the [ENTER] key to validate the input.
 5. Display the description window of control (press [ALT] [ENTER]).
 6. In the "General" tab, in the "Operations on controls" area, select "None".
 7. Display the processes associated with the button ([F2] key).
 8. Type the following browser click code:

```
WHILE IMG_Animal..X < 1200
  IMG_Animal..X++
END
```

This code is used to modify the X coordinate of the Image control (..X property) until a condition is true. In our case, this condition is as follows: the image position (its ..X property) must correspond to 1200.



Tip

When a code line is too long to be displayed in the window of code editor, you have the ability to cut it in 2 by using a carriage return.

- ▶ Let's now create a last button used to move the image until it reaches the left border of page. To do so, we will be using the LOOP statement of WLanguage.

 1. On the "Creation" pane, in the "Usual controls" group, click **Ok**.
 2. Click the location where the button will be created (on the right of WHILE button for example).
 3. Click the button that was just created. The text displayed in the button becomes editable. Type the caption: "LOOP statement".
 4. Press the [ENTER] key to validate the input.
 5. Display the description window of control (press [ALT] [ENTER]).
 6. In the "General" tab, in the "Operations on controls" area, select "None".
 7. Validate the description window of control.
 8. Display the processes associated with the button ([F2] key).

- 9. Type the following browser click code:

```
// Moves the image to the left
LOOP
  IMG_Animal..X--
  IF IMG_Animal..X <= 0 THEN BREAK
END
```

This code is used to modify the X coordinate of Image control (..X property). At each turn, a condition is checked. If this condition is true, the program exits from the loop. In our case, the condition is as follows:

- the image position is set to 0,
- the image position is less than 0 (used to manage the case where the user presses the button several times).



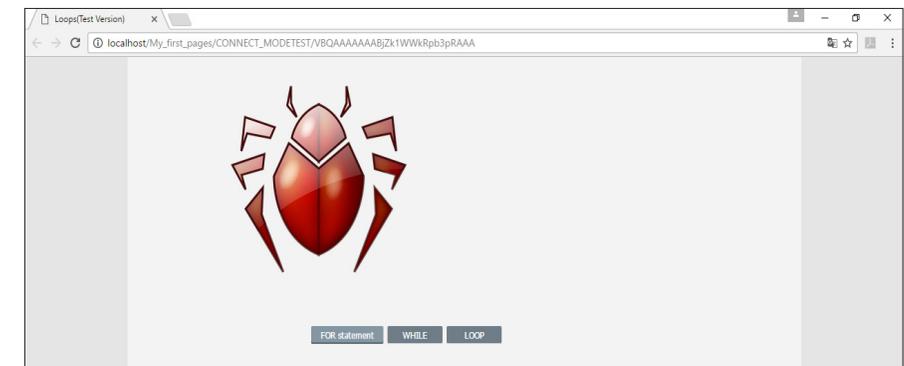
Notes

When typing this code, the LOOP statement may be underlined by a green line and a warning may appear in the pane of compilation errors.

This warning reminds you that an exit statement must be found in the code of loop to avoid an infinite loop.

Page test

- ▶ Let's now run the page test:
 1. Click  among the quick access buttons (or press [F9]).
 2. The created page is started in execution.
 3. Click the different buttons.



- 4. Close the test page.

APPENDICES 5. THE PROCEDURES

This lesson will teach you the following concepts ...

- Types of procedures
- Creating and calling a procedure
- Parameters of a procedure
- Result of a procedure
- Practical example



Estimated time: 1 hour

Overview

A procedure is used to associate an identifier with a code section in order to re-use it. In this lesson, we are going to present the different types of procedures available in WLanguage, their creation mode, how to call them, pass parameters and retrieve a result.

Types of procedures

Three types of procedures are available:

- **Global procedure:** can be used in all the project processes (declared in a set of procedures).
- **Procedure local** to a Window, Page or Mobile Window: can be used in all the processes that depend on the object in which this procedure was declared.
- **Procedure internal** to a process: can only be used in the process where it was declared.



Notes

Scope of procedures

The procedures comply with the scope rules presented for the variables (see page "The scope of variables", page 434).

Creating and calling a procedure

Global procedure

To create a **global procedure**, you must:

1. Create (if necessary) a set of procedures (via the "Project explorer" pane, "Procedures" folder). Give a name to the set of procedures.
2. Create a global procedure in the set of procedures (via the "Project explorer" pane, "Procedures, Set name" folder). Give a name to the procedure.
3. Type the code of global procedure. The procedure code has the following format:

```
PROCEDURE <Name of Global Procedure> ()
```

Local procedure

To create a **local procedure**, you must:

1. Select the element associated with the procedure (window, page, ...).
2. Create a local procedure (via the "Project explorer" pane, expand the element name, "Local procedures" folder).
3. Give a name to the procedure.
4. Type the code of local procedure. The procedure code has the following format:

```
PROCEDURE <Name of Local Procedure> ()
```

Internal procedure

To create an internal procedure, type the following code in the relevant process:

```
INTERNAL PROCEDURE <Procedure Name> ()
<Code of internal procedure>
END
```

Calling a procedure

To call a procedure, use the name of procedure (with the possible parameters that will be passed to it).

```
<Procedure Name>(<Parameter 1>, ..., <Parameter N>)
```

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

Parameters of a procedure

What is a parameter?

A parameter is a value sent to a procedure during the call to the procedure.

The following example is used to call the **Multiply10** procedure and to pass in parameter the value that will be handled in the procedure:

```
Multiply10(50)
```

You have the ability to pass from 0 to several values in parameter to a procedure. These values can have any type (like for the variables).

The parameter is specified in the procedure declaration in the format of a variable. For example, for the **Multiply10** procedure, the procedure code is:

```
PROCEDURE Multiply10(P)
P=P*10
```

P is the parameter expected by the procedure.

**Notes**

To specify the role of parameter in the procedure, you have the ability to typecast the parameter in the procedure declaration.

For example, to use numeric values only, you have the ability to declare:

```
PROCEDURE Multiply10(P is numeric)
```

In the following example, the **Multiplication** procedure expects two Integer parameters and returns the result of multiplication.

The procedure code is as follows:

```
PROCEDURE Multiplication(Nb1 is int, Nb2 is int)
Result is int
Result = Nb1 * Nb2
RESULT Result
```

The code used to call the procedure is as follows:

```
res is int
res = Multiplication(10, 50)
// Res is equal to 500
```

How to use the parameters?

By default, **passing parameters in WLanguage is performed by reference** (or by address). The parameter in the procedure represents (references) the variable passed during the call.

Therefore, when a statement of the procedure modifies the parameter value, the value of the variable corresponding to this parameter is modified.

Example:

- The procedure code is as follows:

```
PROCEDURE Test_address(P1)
P1 = P1 * 2
```

- The code used to call the procedure is as follows:

```
T is int
T = 12 // T is set to 12 before the call
Test_address(T)
// T is set to 24 after the call
```

To avoid modifying the value of the variable corresponding to the parameter, the **parameters must be passed by value**. Passing parameters by value allows you to handle a copy of the parameter value. If the procedure code modifies the value of the variable, the value of the variable corresponding to the parameter is not modified.

To force a parameter to be passed by value to a procedure, the **LOCAL** keyword must be used in front of the parameter name in the declaration of the procedure. This keyword indicates that the following parameter will not be modified by the procedure.

Example:

- The procedure code is as follows:

```
PROCEDURE Test_value(LOCAL P1)
// Local indicates that the parameter will be passed by value
P1 = P1 * 2
```

- The code used to call the procedure is as follows:

```
T is int
T = 12 // T is set to 12
Test_value(T)
// T does not change
```



Notes

In the same procedure, some parameters can be passed by address while other parameters can be passed by value. All you have to do is used the LOCAL keyword in front of each parameter passed by value.

Mandatory or optional parameters?

The parameters received in the procedure can be mandatory or optional parameters. A mandatory parameter must be filled during the call to the procedure while an optional parameter can be omitted: in this case, it will take the default value defined when declaring the procedure parameters.



Notes

When declaring a procedure, the optional parameters are the last parameters (they are always specified after all the mandatory parameters).

In the following example, the **Multiplication** procedure is using an optional parameter, Nb2. This optional parameter is indicated after the mandatory parameters, by specifying its default value. In this example, the default value of the optional parameter is set to 10.

```
PROCEDURE Multiplication(Nb1 is int, Nb2 is int=10)
Result is int
Result = Nb1 * Nb2
RESULT Result
```

The code used to call the procedure is as follows:

```
res is int
res = Multiplication(6)
// Res is equal to 60
```

In this example, the second parameter was not specified. Therefore, its default value will be used.

Result of a procedure

The procedures can return one or more results. The result can be typecasted. The RESULT keyword must be used to return a value.

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

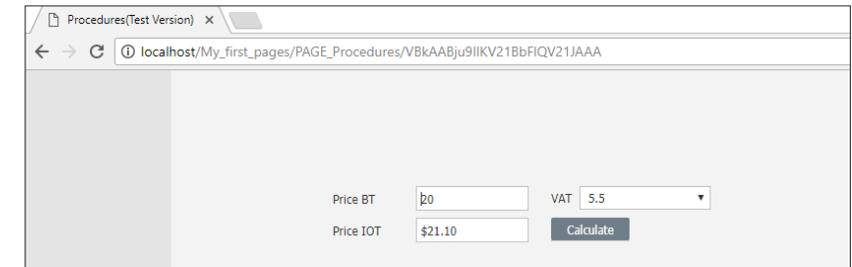
Practical example: Using a procedure

In a new page, we are now going to:

- Create two numeric edit controls containing the value BT and the value IOT.
- Create a Combo Box control used to choose the VAT rate.
- Create a button used to calculate and display the value IOT corresponding to the amount BT.

The result of the calculation will be displayed in the "Price IOT" control.

This page is as follows:



Project used

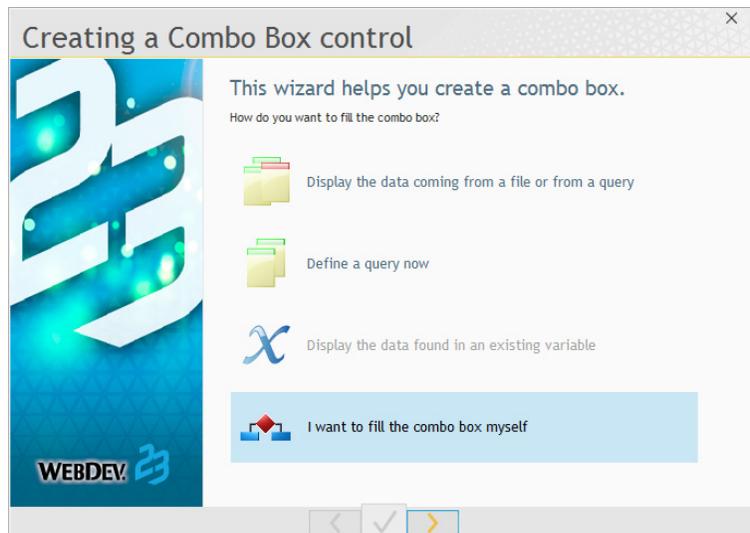
To handle the different concepts of this lesson, we are going to create different pages. These pages will be created in the "My_First_Pages" project.

- ▶ Open the "My_First_Pages" project if necessary. If the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My first pages (Exercise)".

Implementation

- ▶ Create a new blank page:
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Page" then "Page". The wizard for page creation is displayed.
 2. Select "Blank page" and "Simple layout".
 3. Validate. The page is automatically created in the editor. The backup window of created element is displayed.
 4. Specify the page title: "Procedures". The page name ("PAGE_Procedures") is automatically proposed.
 5. Validate the backup window (green button).
- ▶ To create the edit control corresponding to the price BT:
 1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow). The list of available edit controls is displayed. Select the "Currency" control and position the control in the page.
 2. Right-click the control and select "Description".

3. In the description window:
 - Type the control name: "EDT_PriceBT".
 - Type the caption: "Price BT".
 4. Validate.
- To create the control where the result will be displayed:
1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow). The list of available edit controls is displayed. Select the "Currency" control and position the control in the page (below the edit control displaying the price BT for example).
 2. Type the control information: right-click the control and select "Description".
 - Specify the control name: "EDT_PricelOT".
 - Modify the caption to "Price lOT".
 3. The result displayed in this control must not be modifiable. Click the "GUI" tab and choose "Read-only" for the initial status.
 4. Validate.
 5. Save the page.
- To create the Combo Box control for VAT selection:
1. On the "Creation" pane, in the "Usual controls" group, click "Combo box" then click the position where the control will be created in the page (beside the "Price BT" control).
 2. The wizard for creating a Combo Box control is displayed. This wizard is used to define the main control characteristics.
 3. Select "I want to fill the combo box myself".

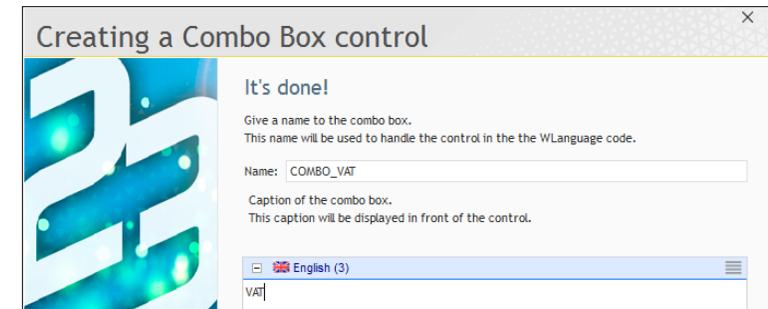


4. Display the next screen.
5. Keep the default options. Display the next screen.

6. In the screen named "Initial content", type the list of possible VAT values:
 - 5.5
 - Press the [ENTER] key.
 - 10
 - Press the [ENTER] key.
 - 20.



7. Display the next screen: give a name (COMBO_VAT) and a caption (VAT) to the control.



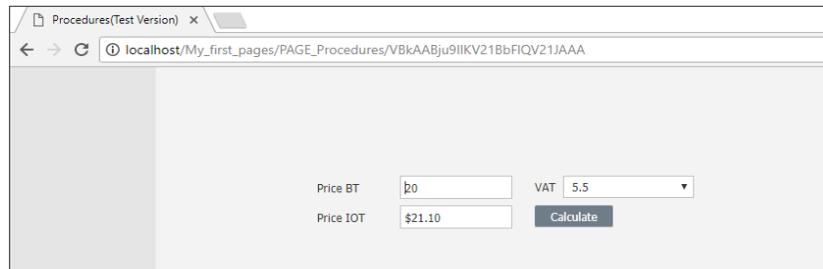
8. Validate.
- To create the "Calculate" button:
1. On the "Creation" pane, in the "Usual controls" group, click **Ok**.
 2. Click the location where the button will be created (below the Combo Box control for example).
 3. Click the button that was just created. The text displayed in the button becomes editable. Type the caption: "Calculate".
 4. Press the [ENTER] key to validate the input.

- ▶ The amount IOT will be calculated in the process of the "Calculate" button.
 1. Display the code of "Calculate" control ("Code" from the popup menu).
 2. Type the following code in the "Click BTN_Calculate (server code)" process:

```
SWITCH COMBO_VAT..DisplayedValue
// 5.5 %
CASE 5.5
EDT_Price_IOT = EDT_Price_BT * 1.055
// 10 %
CASE 10
EDT_Price_IOT = EDT_Price_BT * 1.1
// 20 %
CASE 20
EDT_Price_IOT = EDT_Price_BT * 1.2
END
```

This code calculates the amount IOT by using the value selected in the Combo Box control (returned by `..DisplayedValue`).

3. Run the test of your page (among the quick access buttons). Give a price BT. Select the different values in the Combo Box control and click the "Calculate" button.



Our page operates properly now. However, instead of using a formula 3 times to calculate the price, a procedure can be used to perform this calculation.

- ▶ Close the test page to go back to the code editor.

Creating and using a procedure

- ▶ To create the procedure for calculating the amount IOT:
 1. Click the page background.
 2. Display the processes associated with the page ([F2] key).
 3. In the code editor, on the "Code" pane, in the "Procedures" group, expand "New" and select "New local procedure (Server)".
 4. In the window that is displayed, type the name of local procedure ("Calc_IOT") and validate ("Add" button).
 5. The new procedure local to the page is automatically created and its code is displayed in the code editor.

6. Type the following code:

```
PROCEDURE Calc_IOT(PriceBT, VATRate)
cyIOT is currency
cyIOT = PriceBT * (1 + VATRate/100)
RESULT cyIOT
```

Let's study this code:

- The Calc_IOT procedure expects 2 parameters: the price before tax and the VAT rate.
 - This procedure declares a currency variable. This variable is used to store the calculated price IOT.
 - This procedure calculates the price IOT and returns the calculated value.
7. Close the code editor.



Notes

When creating a procedure, comments are automatically generated BEFORE the procedure code. These comments are mainly used to specify the content of parameters and return value.

It is important to fill these comments. Indeed, they will be automatically displayed in a tooltip when typing the call to the procedure in the code editor.

- ▶ Let's now call the procedure from the calculation button.
 1. Select the "Calculation" button.
 2. Display the code of the button ([F2] key for example).
 3. Replace the existing code by the following code:

```
SWITCH COMBO_VAT..DisplayedValue
// 5.5 %
CASE 5.5
EDT_Price_IOT = Calc_IOT(EDT_Price_BT,5.5)
// 10 %
CASE 10
EDT_Price_IOT = Calc_IOT(EDT_Price_BT,10)
// 20 %
CASE 20
EDT_Price_IOT = Calc_IOT(EDT_Price_BT,20)
END
```

This code calls the Calc_IOT procedure to calculate the amount IOT. Two parameters are passed in parameter: the price BT and the VAT rate. The result returned by the procedure is assigned to the EDT_Price_IOT control.

► Let's now run the page test:

1. Run the test of your page ( among the quick access buttons). Give a price BT. Select the different values in the Combo Box control and click the "Calculate" button.
2. The amount BT is displayed.
3. Close the test page.

Conclusion

This section allowed you to get familiar with the main concepts of WLanguage programming in WEBDEV. Several other features are available. Other ones will be presented later in this tutorial.

WLanguage is a very powerful language that allows you to develop applications that use:

- the Object-Oriented Programming (OOP),
- the MVP (Model View Presenter),
- a 3-tier programming,
- advanced types (XML, Email, ...), ...

See the online help regarding the relevant topic for more details.

APPENDICES 6. QUESTIONS/ANSWERS

This lesson will teach you the following concepts ...

- Questions/Answers about the code editor



Estimated time: 10 mn

Questions/Answers

Question How to view the element to which the current process belongs?

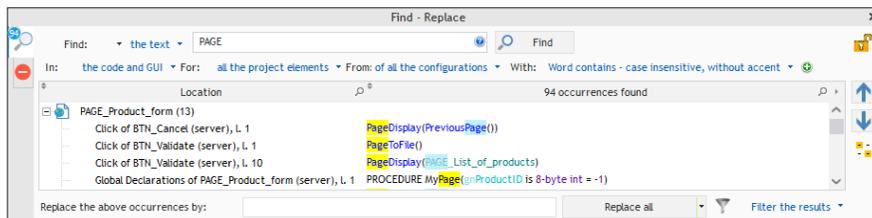
To view the element corresponding to the current process, on the "Code" pane, in the "Navigation" group, click "Go to object" (CTRL + ALT + F2). The page containing the requested element is displayed.

Question How to print the source code?

To print the current source code, click  in the "Home" pane or press [CTRL] + [P].

Question How do I perform a "find and/or replace"?

The functions for performing searches or replacements (in the code, in the interface, ...) can be accessed from the "Home" pane, in the "Find" group or in the "Find - Replace" pane (CTRL + F):



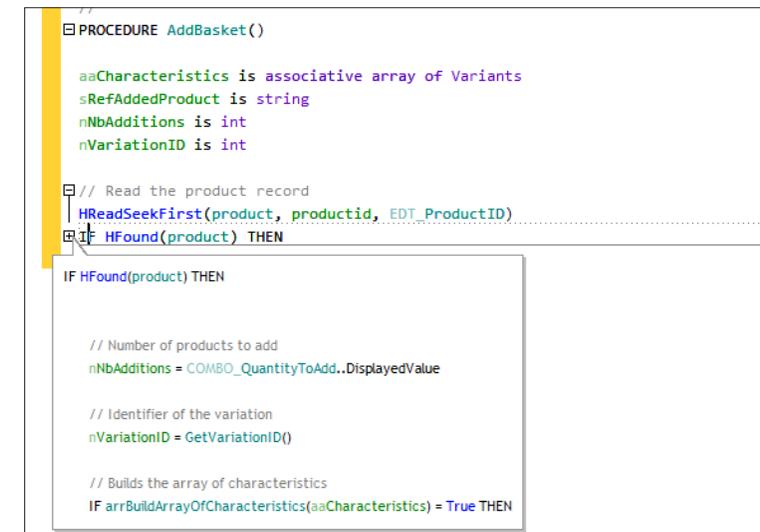
Then, you have the ability to define the different characteristics of the search to perform.

Question What is the meaning of the "+" and "-" signs in the code editor?

The code editor allows you to expand or collapse the WLanguage code. This feature is very useful if your processes use a lot of structured statements (loops, conditions, browses, ...).

To collapse a code section, on the "Display" pane, expand "Collapse" and select "Collapse all" (or press [CTRL] + [SHIFT] + * (on the numeric keypad)).

Only the comments remain visible. The associated code is displayed in a tooltip when the comment line is hovered by the mouse cursor:



Press [CTRL] + * (on the numeric keypad) to expand the entire code. A click performed on the "-" or "+" symbol allows you to collapse or expand the corresponding code section.

Question Is it possible to identify the person who wrote a code line?

Press [F6] to display information (name and creation/modification date) about each code line.

Question Is it possible to find out the number of a code line?

In the code editor, to enable (or not) the numbering of code lines, on the "Display" pane, in the "Help for edit" group, click "Display the line numbers" (shortcut: [CTRL]+[SHIFT]+[G]).

Question **Is there a method to easily display the syntax or the help about a function?**

When typing a function in the code editor:

- the name of each parameter is displayed in a tooltip for the code line currently in edit. For the parameter currently in edit, hovering the parameter name with the mouse is used to display the parameter details in a tooltip. If several syntaxes are available, press [CTRL] + [ALT] + Right Arrow or [CTRL] + [ALT] + Left Arrow to go from a syntax to another one.
- the function syntax is displayed in the status bar of editor.

In the help displayed, the parameters enclosed in [and] are optional parameters.

For the functions that require the names of data files, controls, pages or reports, the assisted input is used to display the list of project elements corresponding to the parameter of the function being typed.

Examples of assisted input for **HReadFirst**: This wizard asks you questions regarding the use of the function and it automatically generates the corresponding code.

A help page is associated with each WLanguage function and property. This help page can be directly accessed from the code editor: to do so, press [F1] on the name of the requested function or property.

Question **What are the useful shortcuts in the code editor?**

- [F2], when it is pressed on the name of a control, class, procedure or report block, displays the process of this object.
- [CTRL]+[F2] is used to go back to the initial process. To go back to the initial process, press [CTRL]+[F2] the same number of times.
- [CTRL]+[L] deletes the current line.
- [CTRL]+[D] duplicates the current line (or the selected lines) on the line below.
- [TAB] and [SHIFT]+[TAB] are used to manage the indent for the selected lines.
- [CTRL]+[/] converts the selected lines into comments, [CTRL]+[SHIFT]+[/] removes the comments (Caution: [/] key on the numeric keypad).
- [CTRL]+[R] is used to automatically indent the code displayed.

CONCLUSION

The tutorial is over now!

This tutorial has discussed a variety of subjects, but not all the features of WEBDEV, far from it! You are now familiar with the main concepts.

You can also explore the examples supplied with WEBDEV: some are simple and only address one topic, while others are more complex. These examples will show you the different aspects of WEBDEV. Reading the source code is also a good way to learn.

It would take too much room to discuss all the available topics (there are hundreds, even thousands !). WEBDEV proposes several other features not presented in this lesson:

- HTTP functions
- nested reports, queries with parameters ...
- dynamic compilation, calls to DLL, ...

See the online help for more details.

We wish you a great development experience with **WEBDEV 23!**